

CA SiteMinder®

Policy Design Guide

r6.0 SP6



Second Edition

This documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2011 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Arcot [™]
- CA Arcot RiskFort [™]
- CA Arcot WebFort [®]
- CA SiteMinder[®]
- CA SiteMinder[®] Federation Security Services
- CA SiteMinder[®] SAML Affiliate Agent

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: SiteMinder Overview 33

Overview	33
SiteMinder Components	33
SiteMinder Features	34
Policy Server Overview	35
Policy Server Management	37
Policy Server System Management Overview	37
Policy Server Object Management Overview	38

Chapter 2: Policy Server Object Overview 39

Policy Server Object Types	39
System Objects	39
Policy Domain Objects	41
Global Objects	42
Configuration Order	44
Administrative Roles	45

Chapter 3: Implementing Policy-based Security 47

Overview	47
Strategies for Managing Security and Users	47
About Access Control Lists	48
About SiteMinder Security Policies	49
How the End-user Experience Is Managed	51
How a Security Model Is Established	54
How a Security Model Requirements Table Is Created	55
How Organization and Resource Requirements Are Established	55
Define Task-Assessment Requirements	57
Define Implementation Requirements	58
How Role-Based Access Control Is Implemented	59
Access Roles in Identity Manager	59
Identity Manager Environment Objects	60

Chapter 4: Policy Server User Interface 63

Policy Server User Interface Overview	63
Policy Server Administrator Accounts	63

Default Super User Administrator Account	63
Start the Policy Server User Interface	64
Use the Policy Server User Interface	65
How to Work with the Object Pane	68
SiteMinder List Pane	68
SiteMinder Menu Bar	69
SiteMinder Tool Bar	72
Scroll Between Groups of Items	73
Search with the SiteMinder Administration Window	73
Policy Server Object Icons	74
Work with Policy Server Objects	75
Required Fields	75
Configure a Policy Server Object	75
View the Properties of a Policy Server Object	76
Edit an Existing Policy Server Object	78
Delete a Policy Server Object	81
How the Web Agent and Policy Server Calculate Time	83

Chapter 5: User Sessions 85

User Session Overview	85
Non-Persistent and Persistent Cookies	85
Non-Persistent and Persistent Sessions	85
Session Tickets	86
How SiteMinder Manages User Sessions	87
How a User Session Begins	89
How Sessions Across Realms Are Maintained	90
How Sessions Across Multiple Cookie Domains Are Maintained	90
How a User Session Is Validated	91
How Session Information Is Delegated	92
Session Timeouts	92
How Agent Key Management and Session Timeouts are Coordinated	93
How a User Session Ends	94
How a Windows User Security Context Is Obtained	94
How Persistent Sessions for User Security Contexts Are Maintained	95
How SiteMinder Is Configured to Provide a Windows User Security Context	96

Chapter 6: Agents and Agent Groups 99

Agents Overview	99
Web Agents	100
SAML Affiliate Agents	101
RADIUS Agents	103

Application Server Agents	103
TransactionMinder XML Agents	104
The SiteMinder Web Agent Model	104
Web Agent Configuration Options	104
Advantages of Centrally Configuring Web Agents	105
Agent Components	106
Policy Server Components	107
Trusted Hosts	107
Register a Trusted Host with the Policy Server	107
Trusted Host Configuration Settings	108
Trusted Host Configuration Parameters	108
View Trusted Host Objects	110
Delete Trusted Host Objects	111
How a Host Configuration Object is Configured for a Trusted Host	111
Create a Host Configuration Object	112
Add Multiple Policy Servers to the Host Configuration Object	114
Edit a Host Configuration Object	115
Configure Policy Server Clusters for a Host Configuration Object	117
Dynamic Host Configuration Object (HCO) Updates	118
Delete a Host Configuration Object	119
Configure a Web Agent	120
Web Agent Parameter Descriptions	121
Configure Web Agents Centrally	121
Configure Web Agents Locally	121
Combined Central and Local Configuration	122
Create an Agent Object to Establish a Web Agent Identity	123
Set the Configuration Parameters in the Agent Configuration Object	126
Configure an Agent Configuration Object for a Web Agent	126
Required Agent Configuration Object Parameters	127
Create an Agent Configuration Object	128
Modify an Agent Configuration Object	130
Delete an Agent Configuration Object	132
Enable a Web Agent	133
Configure Agent Identities for Non-Web Agents	133
Configure a RADIUS Agent	134
Create Custom Agents	136
Configure a Custom Agent Type	136
Create a Custom Agent Object for the Agent Identity	137
Resource Protection with a SiteMinder Agent	138
Agent Group Usage	138
Configure an Agent Group	139
Add Agents to an Agent Group	140

Chapter 7: User Directories 141

User Directory Connections Overview	141
LDAP Overview	142
ODBC Database Overview	153
Windows Directory Overview	154
Active Directory Overview	155
Custom Directory Overview	156
Configure User Directory Connections	156
Define the Same User Directory Connection in Multiple Policy Stores	157
Specify Directory Attributes	158
Configure CA, Oracle, and IBM Directory Server LDAP Directory Connections	160
Domino Directory Connections Prerequisites	163
Configure Domino Directory Connections	163
Configure Oracle Internet Directory Connections	164
Configuration of Novell eDirectory LDAP Server Connections	166
Configure Active Directory Connections	171
Connect the Policy Server to an Active Directory Global Catalog	176
Configure an ADAM Directory Server	177
Load Balancing/Failover Configuration for LDAP Directory Connections	180
Configure ODBC Directory Connections	183
How to Configure an LDAP User Directory Connection over SSL	188
Configure SQL Query Schemes	197
Configure a New SQL Query Scheme	197
Edit an Existing SQL Query Scheme	199
Select an SQL Query Scheme for an ODBC User Directory Connection	199
Configure SQL Query Schemes for Authentication via Stored Procedures	200
Asynchronous Call Support During Failover and Connection Pooling	202
Windows Directory Connections	204
WinNT Domain Connection Requirements	204
Configure a Windows Directory Connection	205
Configure Custom Directory Connections	209
Access the SiteMinder SQL Query Scheme Dialog	209
View User Directory Contents	210
Search Directories	211
Universal IDs	212
How SiteMinder Uses UIDs	212

Chapter 8: Directory Mapping 213

Directory Mapping Overview	213
Configure a Directory Mapping	215
Assign an Authorization Directory to a Realm	216

Delete a Directory Mapping	217
Configure Authentication and Validation Directory Mapping	217
Directory Mapping Examples	219
Employee Accesses an Engineering Realm Resource	220
Temporary Employee Accesses a Quality Assurance Realm Resource	220
Directory Mapping by Universal ID	221
Directory Mapping Case Sensitivity	221
Directory Mapping and Responses	222

Chapter 9: Policy Server Administrators 223

Administrator Concepts	223
Administrator Scope and Tasks	225
Configure an Administrator	226
Configure Administrator Identity and Authentication Details	226
Administrator Scope and Tasks	227
Precautions on Deleting or Editing an Administrator	229

Chapter 10: Authentication Schemes 231

Authentication Schemes Overview	231
Overview of Authentication Scheme Processing	232
Limit Policy Server Search to One User Store during Authentication	233
Types of Authentication Schemes	233
Protection Levels	237
Supported Authentication Schemes	238
Supported Authentication Schemes and Password Policies	239
Set Up an Authentication Scheme Object in the Policy Server User Interface	240
Multiple Instances of a Single Authentication Scheme Configuration	241
Basic Authentication Schemes	241
Basic Scheme Prerequisites	242
Configure a Basic Authentication Scheme	242
Basic Over SSL Authentication Schemes	243
Basic over SSL Scheme Prerequisites	243
Configure a Basic Over SSL Authentication Scheme	244
HTML Forms Authentication Schemes	244
HTML Forms Scheme Prerequisites	247
HTML Forms Authentication Templates	248
Configure an HTML Form Authentication Scheme	257
Windows Authentication Schemes	258
Windows Authentication Scheme Prerequisites	259
Windows Authentication Scheme Considerations	260
Configure a Windows Authentication Scheme	260

MS Passport Authentication Schemes	261
Passport Authentication Support in the Policy Server	262
Passport Authentication Prerequisites	263
Configure an MS Passport Authentication Scheme	264
RADIUS CHAP/PAP Authentication Schemes	265
PAP Overview	265
CHAP Overview	265
RADIUS CHAP/PAP Scheme Overview	266
RADIUS CHAP/PAP Scheme Prerequisites	266
Configure a RADIUS CHAP/PAP Authentication Scheme	266
RADIUS Server Authentication Schemes	267
RADIUS Server Scheme Prerequisites	267
Configure a RADIUS Server Authentication Scheme	267
SafeWord Server Authentication Schemes	268
SafeWord Server Scheme Prerequisites	268
Configure a SafeWord Server Authentication Scheme	268
SafeWord Server and HTML Forms Authentication Schemes	269
SafeWord Server and HTML Forms Scheme Prerequisites	269
Configure a SafeWord Server and HTML Forms Authentication Scheme	270
SecurID Authentication Schemes	270
SecurID Scheme Prerequisites	274
Configure a SecurID Authentication Scheme	274
SecurID with HTML Forms Support Scheme Prerequisites	275
Configure a SecurID and HTML Forms Authentication Scheme	275
Forms Support for Re-activating and Verifying SecurID Users	276
Forms Support for Activating New User Accounts	276
X.509 Client Certificate Authentication Schemes	277
How SiteMinder Establishes an SSL Connection for Certificates	278
How SiteMinder Identifies Users Based on Certificates	278
X.509 Client Certificate Scheme Prerequisites	278
Configure an X.509 Certificate Authentication Scheme	279
X.509 Client Certificate and Basic Authentication Schemes	279
X.509 Client Certificate and Basic Scheme Prerequisites	280
Configure an X.509 Certificate and Basic Authentication Scheme	280
X.509 Certificate or Basic Authentication Schemes	281
X.509 Client Certificate or Basic Scheme Prerequisites	282
Configure an X.509 Certificate or Basic Authentication Scheme	283
X.509 Client Certificate and HTML Forms Authentication Schemes	283
X.509 Client Certificate and HTML Forms Scheme Prerequisites	284
Configure an X.509 Certificate and HTML Forms Authentication Scheme	285
Agent API Support	285
X.509 Client Certificate or HTML Forms Authentication Schemes	286

X.509 Client Certificate or HTML Forms Scheme Prerequisites	287
Configure an X.509 Certificate or HTML Forms Authentication Scheme	288
Agent API Support	288
Anonymous Authentication Schemes	288
Anonymous Scheme Prerequisites	289
Configure an Anonymous Authentication Scheme	289
Custom Authentication Schemes	290
Custom Scheme Prerequisites	290
Configure a Custom Authentication Scheme	290
Federation Security Services Authentication Schemes	291
TransactionMinder Authentication Schemes	291
Impersonation Authentication Schemes	292
Impersonation Scheme Prerequisites	292
Configure an Impersonation Authentication Scheme	293
Certificate Mapping	293
Configure a Certificate Mapping	294
Test a Certificate Mapping	295
How SiteMinder Confirms the Validity of Certificates	296
Custom Mapping Expressions	300
Custom Certificate Mapping for Multiple Attributes of the Same Type	304

Chapter 11: Domains 307

Policy Domain Overview	307
Domains and User Membership	308
Configure a Policy Domain	309
Disable Global Policy Processing for a Domain	310
Assign User Directories	310
Assign an Administrator	311
Create a Realm	312
Add Identity Manager Environments to a Policy Domain	312
Affiliate Domains	313
Domain Editing	313
Domain Deletion	313

Chapter 12: Grouping Resources in Realms 315

Overview	315
Realms	315
How Resources Are Identified by Agent, Realm, and Rule	316
Nested Realms	318
Realms in Request Processing	320
Configure a Realm	320

Configure a Realm Protected by a SiteMinder Web Agent	321
Configure a Realm Protected by a RADIUS Agent	322
Configure a Realm for a 4.x Affiliate Agent	323
Configure a Nested Realm	324
Realm Editing and Deletion	325
Realm Editing	325
Realm Deletion	325
Flush a Realm from the Resource Cache	326

Chapter 13: Rules **327**

Rules Overview	327
How Rules Work as Part of a Policy	328
How the Policy Server Processes Rules	328
Rules and Nested Realms	329
Rule Actions	330
Advanced Rule Options	334
Configure a Rule for Web Agent Actions	334
Configure a Rule for Authentication Event Actions	336
Configure a Rule for Authorization Event Actions	337
Policy Considerations When Configuring OnAccessReject Rules	338
Configure a Rule for Impersonation Event Actions	339
Configure a Rule for an Affiliate Agent	340
Resource Matching and Regular Expressions	341
Standard Resource Matching	341
Regular Expressions for Resource Matching	342
Enable and Disable Rules	343
Advanced Rule Options	344
Add Time Restrictions for Rules	344
Configure an Active Rule	346
Rule Deletion	347

Chapter 14: Rule Groups **349**

Rule Group Overview	349
Create a Rule Group	350
Add a Rule To or Remove a Rule From a Rule Group	350
Rule Group Property Editing	351
Rule Group Deletion	351

Chapter 15: Responses and Response Groups **353**

Responses	353

How SiteMinder Processes Responses	354
Types of Responses	355
Response Attributes	356
Responses and Directory Mappings.....	357
Configure a Response	357
Response Attributes	359
Configure Response Attribute Caching	363
Use Variable Objects in Responses	364
Response Property Editing	366
SiteMinder Generated User Attributes	366
Active Responses Created for Custom Agents	372
Response Deletion	372
Response Groups	372
Configure a Response Group	373
Response Group Property Editing	374
Response Group Deletion	374

Chapter 16: Policies 375

Policy Overview	375
Policies Explanation	377
Policy Bindings	377
Confidence Levels in Policies	378
Configure a Policy	378
Add Users to a Policy	379
Add Rules to a Policy	380
Associate a Rule with a Response or Response Group	380
Associate a Rule with a Global Response	381
Exclude a User or Group from a Policy	381
Allow Nested Groups in Policies	382
AND Users/Groups Check Box	382
Specify AND/OR Relationships between Users/Groups	384
Add Users by Manual Entry	385
Enhance Policy Server's LDAP Authorization Performance	386
Add LDAP Expressions to Policies.....	387
Add Roles to a Policy	387
Exclude Roles in a Policy	388
Add a Confidence Level to a Policy	389
Enable and Disable Policies	390
Advanced Policy Options	391
Specify Allowable IP Addresses for Policies	391
Add and Remove Time Restrictions for Policies	394

Configure an Active Policy	395
Policy Binding Establishment	396
Policy Bindings for LDAP Directories	396
Policy Bindings for WinNT User Directories	400
Policy Bindings for Microsoft SQL Server and Oracle User Directories	402
Bind Policies to SQL Queries	404
Policy Deletion	405
Policy Processing	405
Sample SiteMinder Configuration with Nested Realms	405
Authentication Processing for Hierarchical Policies	410
Authorization Processing for Hierarchical Policies	412
Directory Mapping for Hierarchical Policies	413
Expressions in Policies	413
Add an Expression to a Policy	414

Chapter 17: Certificate Authorities and Web Services Variables 415

Overview	415
Keys and Certificates in the Key Database	415
The smkeydatabase.properties File	416
Create and Manage the Key Database Using Smkeytool	417

Chapter 18: Variables 421

eTelligent Rules	421
SiteMinder eTelligent Rules Benefits	421
CA Directory Variable Requirements	422
eTelligent Rules Configuration	422
Variables Overview	425
Variable Types	426
Variable Use in Policies	427
Variable Use in Responses	428
How the Policy Server Processes Variables	428
Web Service Variables	430
Security Requirements When Resolving Web Services Variables	432
Configure the Web Service Variable Resolver	432
Sample WebServiceConfig.properties Configuration File	433
Create a Variable	433

Chapter 19: Global Policies, Rules, and Responses 435

Global Policies	435
Global Policy Object Characteristics	436

SiteMinder Global Policy Concept	438
Global Policy Processing	440
Configure Global Policies	440
Global Rule Objects	440
Global Response Objects	445
Response Attributes for Global Responses	446
Configure Global Policy Objects	449
Configure a Global Active Policy	454

Chapter 20: Impersonation **457**

Impersonation Overview	457
Impersonation Process	458
Security Considerations for Impersonation	459
Effects of Authentication Scheme Protection Levels	460
Session Idle Timeouts	460
Restricting Impersonation	460
Impersonation Realms and Events	461
Policy Server Objects for Impersonation	461
How an Impersonation Session is Initiated	463
Configure an Agent to Use startimp.fcc	466
Configure an Impersonation Authentication Scheme	466
Enable Impersonation through an .fcc File	466
Impersonation Event Configuration	471
Configure Policies for Impersonation	471
Multiple Cookie Domain Support	473
Sample Implementation of Impersonation	473
Sample Impersonation Implementation Assessment	474
Sample Forms for Impersonation	475

Chapter 21: Password Policies **479**

Password Services Overview	479
How Password Services Work	480
Apache Web Server Prerequisites	482
Important Considerations	482
Change the Default Redirect URL	483
Create Password Policies	483
Configure Password Expiration	485
Configure Password Composition	487
Password Regular Expressions	488
Regular Expressions Syntax	488
Configure Regular Expression Matching	489

Configure Password Restrictions	490
Configure Advanced Password Options	491
User-initiated Password Changes	492
How a Change Password Link is Added	492
Password Self-Changes	496
Remove the Login ID When Redirecting for Password Services	496
CGI-based Password Services HTML Form Customization	497
CGI-based Password Services HTML Form Templates	497
Modify CGI-based Password Services HTML Templates	499
Password CGI-based Services Properties Files	500
Servlet-based Password Services JSP Form Customization	502
Overview of Servlet-based Password Services JSP Forms	502
Modify Servlet-based Password Services JSP Forms	504
Localize Servlet-based Password Services	505
Authentication Schemes Requiring Additional Attribute Information	507
Enhanced Active Directory Integration with Password Services	510
Enhanced Active Directory Integration Considerations	510
User Account Management with Active Directory Integration	511
Password Services with Active Directory Integration	512
Password Policy Troubleshooting	512
New User Passwords are Rejected	513
User Accounts are Mistakenly Disabled	513
User Accounts are Prematurely Disabled	513
Password Changes are Forced	514
LDAP Users Do Not Disable	514
Active Directory Users Cannot Change Passwords	514
Incorrect Password Message Does Not Appear	515

Chapter 22: SiteMinder Test Tool **517**

Test Tool Overview	517
Configure Your Test Environment Agent	518
Policy Server Identification	519
Specify Resource Information	521
Specify User Credentials	522
Set the Encoding Spec	522
Save and Load Configurations	523
Run a Functionality Test	523
Functionality Test Results	525
Calculate an Average Elapsed Time	527
Perform a Regression Test	527
Stress Tests	528

Thread Control Files	528
Report Viewing	530
Certificate-based Authentication Tests	531
Certificate Attributes that Require Custom Mappings	531
Custom Attribute Mappings for Testing	532

Chapter 23: Agent and Related Dialogs Reference 535

Agent Dialog	535
Agent Object Prerequisites	535
Navigate to the Agent Dialog	536
Agent Dialog Fields and Controls	536
Tasks Related to the Agent Dialog	537
Host Configuration Object Dialog	538
Host Configuration Object Prerequisites	538
Navigate to the Host Configuration Object Dialog	538
Host Configuration Object Dialog Fields and Controls	538
Agent Configuration Object Dialog	540
Agent Configuration Object Prerequisites	540
Navigate to the Agent Configuration Object Dialog	540
Agent Configuration Dialog Fields and Controls	541
Tasks Related to the Agent Configuration Object Dialog	541
Agent Group Dialog	541
Agent Group Object Prerequisites	541
Navigate to the Agent Group Dialog	542
Agent Group Dialog Fields and Controls	542
Tasks Related to the Agent Group Dialog	543
Edit Parameter Dialog	543
Edit Parameter Dialog—Plain Radio Button Selected	543
Edit Parameter Dialog—Encrypted Radio Button Selected	543
Edit Parameter Dialog—Multi-value Radio Button Selected	543
Cluster Setup Dialog	544
Agent Group Items Dialog	544

Chapter 24: User Directory and Search Dialog Reference 545

User Directory Dialog	545
User Directory Object Prerequisites	546
Navigate to the User Directory Dialog	546
Navigate to the User Directory Dialog for an Existing User Directory Connection	547
Tasks Related to the User Directory Dialog	547
User Directory Dialog—LDAP Namespace	547
User Directory Dialog—ODBC Namespace	553

User Directory Dialog—WinNT Namespace	557
User Directory Dialog—Custom Namespace	561
User Directory Dialog—AD Namespace	566
LDAP Setup Dialog	571
LDAP Setup Dialog Prerequisites	571
Navigate to the LDAP Setup Dialog	572
LDAP Setup Dialog Fields and Controls	572
Tasks Related to the LDAP Setup Dialog	574
ODBC Setup Dialog	574
ODBC Setup Dialog Prerequisites	574
Navigate to the ODBC Setup Dialog	575
ODBC Setup Dialog Fields and Controls	575
Tasks Related to the ODBC Setup Dialog	576
SQL Query Scheme Dialog	576
SQL Query Scheme Dialog Prerequisites	576
Navigate to the SQL Query Scheme Dialog	576
Navigate to the SQL Query Dialog for an Existing Query Scheme	577
SQL Query Scheme Dialog Fields and Controls	577
Tasks Related to the SQL Query Scheme Dialog	582
Directory List Dialog	582
Directory List Dialog Prerequisites	583
Navigate to the Directory List Dialog	583
Directory List Dialog Fields and Controls	583
Tasks Related to the Directory List Dialog	583
Search LDAP/AD Dialog	584
Search LDAP/AD Dialog Prerequisites	584
Navigate to the Search LDAP/AD Dialog	584
Search LDAP/AD Dialog Fields and Controls	584
Tasks Related to the Search LDAP/AD Dialog	585
Search Database Dialog	586
Search Database Dialog Prerequisites	586
Navigate to the Search Database Dialog	586
Search Database Dialog Fields and Controls	587
Tasks Related to the Search Database Dialog	587
Search WinNT Directory Dialog	587
Search WinNT Dialog Prerequisites	587
Navigate to the Search WinNT Dialog	588
Search WinNT Dialog Fields and Controls	588
Tasks Related to the Search WinNT Dialog	588
User Lookup Dialog	589
User Lookup Dialog Prerequisites	589
Navigate to the User Lookup Dialog	589

User Lookup Dialog Fields and Controls	590
Tasks Related to the User Lookup Dialog	590
Directory Mappings Dialog	590
Directory Mappings Dialog Prerequisites	591
Navigate to the Directory Mappings Dialog	591
Directory Mappings Dialog Fields and Controls	591
Tasks Related to the Directory Mappings Dialog	592
Add Directory Mapping Dialog	592
Navigate to the Add Directory Mapping Dialog	592
Add Directory Mapping Fields and Controls	592
Tasks Related to the Add Directory Mapping Dialog	593
Edit Directory Mapping Dialog	593
Navigate to the Edit Directory Mapping Dialog	593
Edit Directory Mapping Fields and Controls	594
Tasks Related to the Edit Directory Mapping Dialog	594
Delete Dialog	594
Tasks Related to the Delete Dialog	595
AuthValidate Directory Mappings Dialog	595
AuthValidate Directory Mappings Dialog Prerequisites	595
Navigate to the AuthValidate Directory Mappings Dialog	595
AuthValidate Directory Mappings Dialog Fields and Controls	596
Tasks Related to the AuthValidate Directory Mappings Dialog	596
Add AuthValidate Directory Mapping Dialog	596
Navigate to the Add AuthValidate Directory Mapping Dialog	596
Add AuthValidate Directory Mapping Dialog Fields and Controls	596
Edit AuthValidate Directory Mapping Dialog	597
Navigate to the Edit AuthValidate Directory Mapping Dialog	597
Edit AuthValidate Mapping Directory Dialog Fields and Controls	597

Chapter 25: Administrator and Related Dialogs Reference 599

Administrator Properties Dialog	599
SiteMinder Administrator Properties Prerequisites	599
Navigate to the Administrator Properties Dialog	599
Administrator Properties Dialog General Group Box	600
Tasks Related to the Administrator Properties Dialog	603
Domain Manager For These Domains Dialog	604
Domain Manager For These Domains Dialog Prerequisites	604
Navigate to the Domain Manager For These Domains Dialog	604
Domain Manager For These Domains Dialog Fields and Controls	604
Tasks Related to the Domain Manager For These Domains Dialog	605

Chapter 26: Authentication Scheme and Related Dialogs Reference 607

Authentication Scheme Dialog	607
Authentication Scheme Object Prerequisites	608
Navigate to the Authentication Scheme Dialog	608
Navigate to the Authentication Scheme Dialog for an Existing Authentication Scheme	608
Authentication Scheme Dialog Fields and Controls	609
Authentication Scheme Dialog—Basic Template	610
Authentication Scheme Dialog—Basic over SSLTemplate	611
Authentication Scheme Dialog—HTML Form Template	612
Authentication Scheme Dialog—Windows Template	615
Authentication Scheme Dialog—MS Passport Template	619
Authentication Scheme Dialog—RADIUS CHAP/PAP Template	622
Authentication Scheme Dialog—RADIUS Server Template	623
Authentication Scheme Dialog—SafeWord Template	624
Authentication Scheme Dialog—SafeWord HTML Form Template	625
Authentication Scheme Dialog—SecurID Template	627
Authentication Scheme Dialog—SecurID HTML Form Template	628
Authentication Scheme Dialog—X509 Client Cert Template	631
Authentication Scheme Dialog—X509 Client Cert and Basic Template	632
Authentication Scheme Dialog—X509 Client Cert or Basic Template	634
Authentication Scheme Dialog—X509 Client Cert and Form Template	637
Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template	639
Authentication Scheme Dialog—Anonymous Template	642
Authentication Scheme Dialog—Custom Template	644
Authentication Scheme Dialog--SAML Artifact Template	645
Authentication Scheme Dialog--SAML POST Template	652
Authentication Scheme Dialog -- SAML 2.0 Template	658
Authentication Scheme Dialog—Custom Template for Federation	660
Authentication Scheme Dialog—Impersonation Template	662
Tasks Related to the Authentication Scheme Dialog	664
Certificate Mappings Dialog	665
Certificate Mappings Dialog Prerequisites	665
Navigate to the Certificate Mappings Dialog	666
Certificate Mappings Dialog Fields and Controls	666
Tasks Related to the Certificate Mappings Dialog	666
Certificate Mapping Properties Dialog	667
Certificate Mapping Properties Dialog Prerequisites	667
Navigate to the Certificate Mapping Properties Dialog	667
Certificate Mapping Properties Dialog Fields and Controls	667
Tasks Related to the Certificate Mapping Properties Dialog	672
Certificate Map Test Dialog	672

Certificate Map Test Dialog Prerequisites	672
Navigate to the Certificate Map Test Dialog	672
Certificate Map Test Dialog Fields and Controls	673
Tasks Related to the Certificate Map Test Dialog	673

Chapter 27: Domain Dialog Reference 675

Domain Dialog	675
Domain Dialog Prerequisites	675
Navigate to the Domain Dialog	675
Navigate to the Domain Dialog for an Existing Policy or Object	675
Domain Dialog Fields and Controls	676
Tasks Related With the Domain Dialog	679

Chapter 28: Realm Dialog Reference 681

Realm Dialog	681
Realm Object Prerequisites	681
Navigate to the Realm Dialog to Create a New Realm	681
Navigate to the Realms Dialog to Edit an Existing Realm	682
Realm Dialog Fields and Controls	682
Tasks Related to the Realm Dialog	689

Chapter 29: Rule and Rule Group Dialogs Reference 691

Rule Dialog	691
Rule Dialog Prerequisites	691
Navigate to the Rule Dialog to Create a Rule	691
Navigate to the Rule Dialog to Edit an Existing Rule	692
Rule Dialog Fields and Controls	692
Tasks Related to the Rule Dialog	698
Rule Group Properties Dialog	698
Rule Group Properties Dialog Prerequisites	699
Navigate to the Rule Group Properties Dialog to Create a Rule Group	699
Navigate to the Rule Group Properties Dialog to Edit an Existing Rule	699
Rule Group Properties Dialog Fields and Controls	700
Tasks Related to the Rule Group Properties Dialog	701
Rule Group Items Dialog	702
Rule Group Items Dialog Prerequisites	702
Navigate to the Rule Group Items Dialog	702
Rule Group Items Dialog Fields and Controls	702
Tasks Related to the Rule Group Items Dialog	703

Chapter 30: Response and Related Dialogs Reference 705

Response Dialog	705
Response Object Prerequisites	705
Navigate to the Response Dialog	705
Response Dialog Fields and Controls	705
Tasks Related to the Response Dialog	707
Response Attribute Editor Dialog	707
Response Attribute Editor Dialog Prerequisites	708
Navigate to the Response Attribute Editor Dialog to Create an Attribute	708
Navigate to the Response Attribute Editor Dialog to Edit an Attribute	708
SiteMinder Response Attribute Editor Dialog Fields and Controls	708
Tasks Related to the Response Attribute Editor Dialog	712
Response Group Properties Dialog	712
Response Group Properties Dialog Prerequisites	713
Navigate to the Response Group Properties Dialog	713
Response Group Properties Dialog Fields and Controls	713
Tasks Related to the Response Group Properties Dialog	714
Response Group Items Dialog	715
Response Group Items Dialog Prerequisites	715
Navigate to the Response Group Items Dialog	715
Response Group Items Dialog Fields and Controls	715

Chapter 31: Variable-related Dialogs Reference 717

Static Variable Editor Dialog	717
Static Variable Editor Dialog Prerequisites	717
Navigate to the Static Variable Editor Dialog	718
Static Variable Editor Dialog Fields and Controls	718
Tasks Related to the Static Variable Editor Dialog	719
Request Context Variable Editor Dialog	719
Request Context Variable Editor Dialog Prerequisites	720
Navigate to the Request Context Variable Editor Dialog	720
Request Context Variable Editor Dialog Fields and Controls	721
Tasks Related to the Request Context Variable Editor Dialog	721
User Context Variable Editor Dialog	721
User Context Variable Editor Dialog Prerequisites	722
Navigate to the User Context Variable Editor Dialog	722
User Context Variable Editor Dialog Fields and Controls	723
Tasks Related to the User Context Variable Editor Dialog	725
POST Variable Editor Dialog	725
Post Variable Editor Dialog Prerequisites	725
Navigate to the Post Variable Editor Dialog	726

Post Variable Editor Dialog Fields and Controls	726
Tasks Related to the Post Variable Editor Dialog	726
Web Services Variable Editor Dialog	727
Web Services Variable Editor Dialog Prerequisites	727
Navigate to the Web Services Variable Editor Dialog	727
Web Services Variable Editor Dialog Fields and Controls—General Tab	728
Web Services Variable Editor Dialog Fields and Controls—SOAP Document Tab	730
Web Services Variable Editor Dialog Fields and Control - HTTP Headers Tab	730
Tasks Related to the Web Services Variable Editor Dialog	730
HTTP Header Dialog	730
HTTP Header Dialog Prerequisites	731
Navigate to the HTTP Header Dialog	731
HTTP Header Dialog Fields and Controls	731
Variable Lookup Dialog	731

Chapter 32: Policy and Related Dialogs Reference 733

Policy Dialog	733
Policy Dialog Prerequisites	733
Navigate to the Policy Dialog	734
Policy Dialog Fields and Controls	734
Tasks Related to the Policy Dialog	738
Expression Editor	738
Expression Editor Prerequisites	738
Navigate to the Expression Editor	739
Expression Editor Fields and Controls	739
Tasks Related to the Expression Editor	742
Rule Items Dialog	742
Rule Items Dialog Prerequisites	742
Navigate to the Rule Items Dialog	742
Rule Items Dialog Fields and Controls	743
Tasks Related to the Rule Items Dialog	743
Set Response Dialog	744
Set Response Dialog Prerequisites	744
Navigate to the Set Response Dialog	744
Set Response Dialog Fields and Controls	745
Tasks Related to the Set Response Dialog	745
Policy Expression Dialog	745
Policy Expression Dialog Prerequisites	746
Navigate to the Policy Expression Dialog	746
Policy Expression Dialog Fields and Controls	746
Tasks Related to the Policy Expression Dialog	749

Chapter 33: Global Policies, Rules, and Responses Dialogs Reference 751

Global Rule Dialog	751
Global Rule Object Prerequisites	751
Navigate to the Global Rule Dialog	751
Global Rule Dialog Fields and Controls	751
Tasks Related to the Global Rule Dialog	754
Global Response Dialog	755
Global Response Object Prerequisites	755
Navigate to the Global Responses Dialog	755
Global Response Dialog Fields and Controls	755
Tasks Related to the Global Response Dialog	756
Global Response Attribute Editor Dialog	757
Global Response Attribute Editor Dialog Prerequisites	757
Navigate to the Global Response Attribute Editor Dialog	757
SiteMinder Global Response Attribute Editor Dialog Fields and Controls	757
Tasks Related to the Global Response Attribute Editor Dialog	761
Global Policy Dialog	761
Global Policy Object Prerequisites	761
Navigate to the Global Policy Dialog	761
Global Policy Dialog Fields and Controls	761
Tasks Related to the Global Policy Dialog	764
Rule Items Dialog	765
Rule Items Dialog Prerequisites	765
Navigate to the Rule Items Dialog	765
Rule Items Dialog Fields and Controls	766
Tasks Related to the Rule Items Dialog	766
Set Global Response Dialog	767
Set Global Response Dialog Prerequisites	767
Navigate to the Set Global Response Dialog	767
Set Global Response Dialog Fields and Controls	768
Tasks Related to the Set Global Response Dialog	768

Chapter 34: Password Services Dialog Reference 769

Password Policy Dialog	769
Password Policy Dialog Prerequisites	769
Navigate to the Password Policy Dialog to Create a New Policy	769
Navigate to the Password Policy Dialog to Modigy an Existing Policy	770
Password Policy Dialog Fields and Controls	770
Tasks Related to the Password Policy Dialog	786

Chapter 35: SAML 1.x Affiliate Dialog Reference **787**

Affiliate Dialog	787
Navigate to the Affiliates Dialog	787
Affiliate Dialog Fields and Controls	788
Affiliate Dialog Prerequisites	790
Affiliate Dialog--Users Tab	791
Affiliate Dialog--Assertions Tab	791
Affiliate Dialog--Session Tab	793
Affiliate Dialog--Attributes Tab	793
Affiliate Attribute Editor Dialog	794
Affiliate Attribute Editor Prerequisites	794
Navigate to the Affiliate Attribute Editor Dialog	794
Affiliate Attribute Editor Dialog Fields and Controls	795
Affiliate Attribute Editor Dialog--Attribute Setup Tab	795
Affiliate Attribute Editor Dialog--Advanced Tab	796
Affiliate Dialog--IP Addresses Tab	797
Affiliate Dialog--Time Tab	797
Affiliate Dialog--Advanced Tab	798
Tasks Related to the Affiliate Dialog	798

Chapter 36: SAML 1.x Authentication Scheme Properties **799**

Authentication Scheme Dialog Prerequisites	799
Navigate to the Authentication Scheme Dialog	799
Authentication Scheme Dialog--SAML Artifact Template	800
Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab	801
Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab	808
Authentication Scheme Dialog--SAML POST Template	808
Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab	809
Authentication Scheme Dialog--SAML POST Template--Advanced Tab	814
Tasks Related to the SAML 1.x Authentication Scheme Dialog	814

Chapter 37: SAML 2.0 Service Provider Reference **815**

SAML Service Provider Dialog	815
Navigate to the SAML Service Provider Dialog	815
SAML Service Provider Dialog Prerequisites	816
SAML Service Provider Properties Dialog Fields and Controls	816
SAML Service Provider Dialog--Users Tab	819
SAML Service Provider Dialog--Name IDs Tab	819
SAML Service Provider Dialog--General Tab	821
SAML Service Provider Dialog--Backchannel Properties	823

SAML Service Provider Dialog--SSO Tab	824
Assertion Consumer Services Dialog	827
Restrictions Dialog for an SP	827
Navigate to the Restrictions Dialog	827
Restrictions Dialog Prerequisites	828
Restrictions Dialog Fields and Controls	828
Tasks Related to the Restrictions Dialog	828
SAML Service Provider Dialog--Attributes Tab	829
Navigating to the SAML Service Provider Attribute Editor Dialog	829
SAML Service Provider Attribute Editor Dialog Prerequisites	829
SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab	830
SAML Service Provider Attribute Editor Dialog--Advanced Tab	832
SAML Service Provider Dialog--SLO Tab	832
SAML Service Provider Dialog--IPD Tab	833
SAML Service Provider Dialog--Encryption Tab	834
SAML Service Provider Dialog--Advanced Tab	835
SAML 2.0--Additional URL Configuration Dialog	836
Customize Session Validity Duration	838
Tasks Related to the SAML Service Provider Attribute Editor Dialog	839

Chapter 38: SAML 2.0 Authentication Scheme Properties Reference 841

Authentication Scheme Dialog -- SAML 2.0 Template	841
SAML 2.0 Authentication Properties--Scheme Setup Tab	842
SAML 2.0 Authentication Properties--Advanced Tab	844
Navigate to the SAML 2.0 Auth Scheme Properties Dialog	844
SAML 2.0 Auth Scheme Properties Dialog	844
SAML 2.0 Auth Scheme Properties Dialog Prerequisites	844
SAML 2.0 Auth Scheme Properties Dialog Fields and Controls	845
SAML 2.0 Auth Scheme Properties Dialog--Users Tab	846
SAML 2.0 Auth Scheme Properties Dialog--SSO Tab	847
SAML 2.0 Auth Scheme Properties Dialog--SLO Tab	852
SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab	853
SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab	854
SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab	854
SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab	854
SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab	856
SAML 2.0--Additional URL Configuration	858
Tasks Related to the SAML 2.0 Auth Scheme Properties Dialog	859

Chapter 39: WS-Federation Resource Partner Properties Reference 861

Resource Partner Properties Dialog	861

Resource Partner Properties Dialog Prerequisites	861
Navigate to the Resource Partner Properties Dialog	862
Resource Partner Properties Dialog Fields and Controls	863
Resource Partner Properties Dialog--Users Tab	865
Resource Partner Properties Dialog--Name IDs Tab	866
Resource Partner Properties Dialog--General Tab	868
Resource Partner Properties Dialog--SSO Tab	869
Restrictions Dialog for a Resource Partner	870
Restrictions Dialog Prerequisites	870
Restrictions Dialog Field and Controls	870
Resource Partner Properties Dialog--Attributes Tab	871
Resource Partner Attribute Dialog Prerequisites	871
Resource Partner Attribute Dialog--Attribute Setup Tab	872
Resource Partner Attribute Dialog--Advanced Tab	873
Resource Partner Properties Dialog--Signout Tab	873
Resource Partner Properties Dialog--Advanced Tab	874
Tasks Related to the Resource Partner Properties Dialog	875

Chapter 40: WS-Federation Authentication Scheme Reference 877

Authentication Scheme Properties Dialog--WS-Federation Template	877
WS-Federation Auth. Scheme Properties Dialog Prerequisites	878
Navigate to the Auth. Scheme Properties Dialog--WS Federation Template	878
Federation Authentication Scheme Dialog--Scheme Setup Tab	879
WS-Federation Authentication Scheme Dialog--Advanced Tab	880
WS-Federation Auth. Scheme Fields and Controls	881
Auth. Scheme Properties--WS-Federation--Users Tab	882
Auth. Scheme Properties--WS-Federation--SSO Tab	883
Auth. Scheme Properties--WS-Federation--Signout Tab	885
Auth. Scheme Properties--WS-Federation--Advanced Tab	886
Tasks Related to the WS-Federation Authentication Scheme Properties Dialog	887

Chapter 41: Attribute Authority and Query Reference 889

SAML 2.0 Attribute Authority Reference	889
Attribute Svc Dialog	889
Attribute Svc Dialog Prerequisites	889
Navigate to the Attribute Svc Dialog	889
Attribute Svc Dialog Fields and Controls	890
SAML 2.0 Attribute Query Reference	891
Attributes Dialog Prerequisites	891
Navigate to the Attributes Dialog	892
Attributes Dialog for Attribute Queries	892

Name IDs Dialog Fields for Attribute Queries	894
Federation Attribute Variable Editor Reference	896
Tasks Related to Attributes and Attribute Queries	896

Chapter 42: SAML Affiliations Reference 897

SAML Affiliation Properties Dialog	897
SAML Affiliation Dialog Prerequisites	897
Navigate to the SAML Affiliation Dialog	897
SAML Affiliation Dialog Fields and Controls	898
SAML Affiliation Dialog--Name IDs Tab	899
SAML Affiliation Dialog--Users Tab	900
SAML Service Providers Tab	902
SAML Auth Schemes Tab	902

Chapter 43: Registration Dialog Reference 903

Chapter 44: Shared Dialog Reference 905

Shared Dialog Reference Overview	905
Time Dialog	905
Time Dialog Prerequisites	906
Navigate to the Time Dialog	906
Time Dialog Fields and Controls	906
Tasks Related to the Time Dialog	909
Active Rule Editor Dialog	909
Active Rule Editor Dialog Prerequisites	909
Navigate to the Active Rule Editor Dialog	909
Active Rule Editor Dialog Fields and Controls	910
Tasks Related to the Active Rule Editor Dialog	910
Active Policy Editor Dialog	910
Active Policy Editor Prerequisites	910
Navigate to the Active Policy Editor	911
Active Policy Editor Fields and Controls	911
Tasks Related to the Active Policy Editor	912
Users/Groups Dialog	912
Users/Groups Dialog Prerequisites	912
Navigate to the Users/Groups Dialog	913
Users/Groups Dialog Fields and Controls	913
Tasks Related to the Users/Groups Dialog	916
Add an IP Address Dialog	916
Add an IP Address Dialog Prerequisites	916

Navigate to the Add an IP Address Dialog	916
Add an IP Address Dialog Fields and Controls	917
Tasks Related to the Add an IP Address Dialog	918
Chapter 45: TransactionMinder-related User Interface Elements	919
Overview	919
Chapter 46: Troubleshooting	921
Policy Server Hangs When Web Agent Connection Is Idle	921
Policy Server Does Not Connect to LDAP Stores Using SSL with AKI	922
Troubleshooting SSL Authentication Schemes	923
Determine SSL Connection Ability	923
SSL Configuration	924
SSL Troubleshooting	928
Chapter 47: LanMan User Directories	933
About LanMan User Directories	933
LanMan Directory Connection Prerequisites	933
Configure a LanMan Directory Connection	934
Configure Registry Keys for a LanMan Directory Connection	934
Configure a LanMan User Directory Connection	935
Failover for Windows User Directories	935
LanMan User Directory Search Criteria	936
Chapter 48: Customizing Registration Services	937
Customizing Registration Services Overview	937
The Cascading Style Sheet	937
Cascading Style Sheet Customization	940
Links Between The Cascading Style Sheet to Registration Services Pages	941
Properties Files	941
Locale Properties Files	942
Presentation Properties Files	943
Object Properties Files	944
Map Properties Files	947
JavaServer Pages (JSPs)	949
Search Customization	950
Modify the Search Fields	950
Change the Search Attributes	953
Specify Search Limits	954

Customize the Search Results	954
Task Selector Page Customization	958
Modify the Links	958
Collect Additional Attributes	963
Extend the Object Class	963
Add Attributes to Properties Files	964
Add Fields to Collector Pages	966
Localize Registration Services	970
Configure Registration Services for Different Languages	971

Chapter 49: CA SSO/WAC Integration 977

Overview	977
SiteMinder and CA SSO Integration Architectural Examples	978
Example 1: User Accesses SiteMinder-Protected Resource Before CA SSO	979
Example 2: Authenticated CA SSO Client User Accesses SiteMinder Resource	980
Example 3: User Accesses WAC-Protected Resource Before SiteMinder	982
Prerequisites Before Integrating SiteMinder and CA SSO	983
Configure Single Sign-On from SiteMinder to CA SSO	984
Configure Single Sign-On from CA SSO Client to SiteMinder	987
Configure Single Sign-On from CA SSO to SiteMinder	988
Configure an smetssocookie Web Agent Active Response Attribute	990
Configure an smauthetso Custom Authentication Scheme	992

Appendix A: SiteMinder Kerberos Authentication 995

Kerberos Overview	995
How To Configure SiteMinder Kerberos Authentication	996
Kerberos KDC Configuration at the Domain Controller	996
Kerberos Authentication Configuration at the Policy Server	997
Kerberos Authentication Configuration at the Web Server	998
Kerberos Authentication Configuration at the Windows Workstation	999
Configure a Kerberos Authentication Scheme	999
Configure Kerberos External Realm on Windows Host	1001
Kerberos Configuration Examples	1002
KDC Configuration on Windows 2003 Example	1002
KDC Configuration on UNIX Example	1005
Kerberos Configuration at the Policy Server on Windows Example	1006
Kerberos Configuration at the Policy Server on UNIX Example	1008
Verify that a Resource is Protected	1010
Troubleshooting SiteMinder Kerberos Authentication	1011

Appendix B: CA Arcot Integration **1015**

CA Arcot WebFort and RiskFort	1015
Authentication in an On–Premise Arcot Integration	1016
Confidence Levels and SiteMinder Authorization	1017
Risk Scores and Confidence Levels Compared	1018
CA Arcot Integration Use Cases	1019
User Store Consideration	1023
CA Arcot A-OK	1023
Authentication in a Hosted CA Arcot Integration	1024
Confidence Levels and SiteMinder Authorization	1025
Risk Scores and Confidence Levels Compared	1025
CA Arcot A-OK Integration Use Cases	1026
User Store Consideration	1029

Index **1031**

Chapter 1: SiteMinder Overview

This section contains the following topics:

[Overview](#) (see page 33)

[SiteMinder Components](#) (see page 33)

[SiteMinder Features](#) (see page 34)

[Policy Server Overview](#) (see page 35)

[Policy Server Management](#) (see page 37)

Overview

SiteMinder provides all of the critical capabilities to meet the stringent needs of large, e-business and corporate sites. It provides an infrastructure for centralized and secure policy management that scales to meet the needs of even the largest enterprise Web sites. It provides a way to uniquely identify users before they are authenticated as named users, and to track and manage each user's privileges so that they get access only to those applications or operations for which they have been authorized. In addition, SiteMinder provides for easy personalization of content, as well as single sign-on.

SiteMinder Components

SiteMinder consists of two core components:

Policy Server

Provides policy management, authentication, authorization, and accounting.

SiteMinder Agents

Integrated with a standard Web server or application server, SiteMinder Agents enable SiteMinder to manage access to Web applications and content according to predefined security policies. Other types of SiteMinder Agents allow SiteMinder to control access to non-Web entities. For example, a SiteMinder RADIUS Agent manages access to RADIUS devices, while a SiteMinder Affiliate Agent manages information passed to an affiliate's Web site from a portal site.

SiteMinder Features

SiteMinder is a platform for secure portal, extranet, and intranet management. It provides key authentication, authorization, and personalization features for building and managing secure Web sites. SiteMinder can provide significant benefits for efficient application development and deployment, easier security management, and greater flexibility for centralized access control. It also provides tools to customize content to help retain user loyalty.

SiteMinder is a directory-enabled, standards-based system that can work with heterogeneous Web servers, operating systems, and application development platforms.

SiteMinder can do the following:

- Operate across multiple Web server platforms:
 - Microsoft IIS (Windows)
 - iPlanet Enterprise Web Server (Windows, Solaris, HP-UX)
 - Apache (Solaris)

- Centralize control of user access privileges

- Leverage existing user stores

SiteMinder provides native integration with industry-standard LDAP directory servers, NT domains, and Microsoft SQL Server or Oracle databases for authentication and access control.

In addition, SiteMinder can authenticate users against one directory and authorize users against another directory. This is particularly useful in environments that use legacy directories.

- Deliver an improved user experience

Using personalization, user registration services, anonymous user support, and single sign-on, SiteMinder improves Web site usability for each user.

- Provide solutions for registering and managing users

Registration Services include the following:

- User Self Registration—Allows users to register themselves at your Web site using a set of customizable forms.
- User Update of Profiles—Allows registered users to make changes to their own profile information.
- Management of Registered Users—Allows administration of self-registered users through a set of customizable HTML forms.

- Scale for large or small sites

SiteMinder can support very large sites, with millions of users and applications as well as extranets and intranets with a smaller user base.

- Integrate applications and improve workflow

SiteMinder can integrate directories and external databases in its policies. This means that when a SiteMinder event occurs, for example, an authentication or authorization, SiteMinder can call external applications or libraries, and extract the necessary information from these sources. As a result, legacy applications and systems, such as mainframe applications, can still be used.

- Allow easier Web application development

Web site developers can use SiteMinder to deliver secure personalized Web applications on time and within a budget by managing all of the complex security and entitlement requirements for those applications.

More information:

[Overview](#) (see page 33)

[SiteMinder Components](#) (see page 33)

[Policy Server Overview](#) (see page 35)

[Policy Server Management](#) (see page 37)

Policy Server Overview

The Policy Server typically runs on a separate Windows or Solaris system, and performs SiteMinder's key security operations. In particular, it provides the following:

Authentication

The Policy Server supports a range of authentication methods. It can authenticate users through user names and passwords, forms-based authentication, and public-key certificates.

Authorization

The Policy Server is responsible for managing and enforcing access control rules established by the Policy Server administrator. These rules define the operations that are allowed for each protected resource.

Administration

The Policy Server can be configured using the Policy Server User Interface (UI). The Administration service of the Policy Server is what allows the UI to record configuration information in the Policy Store.

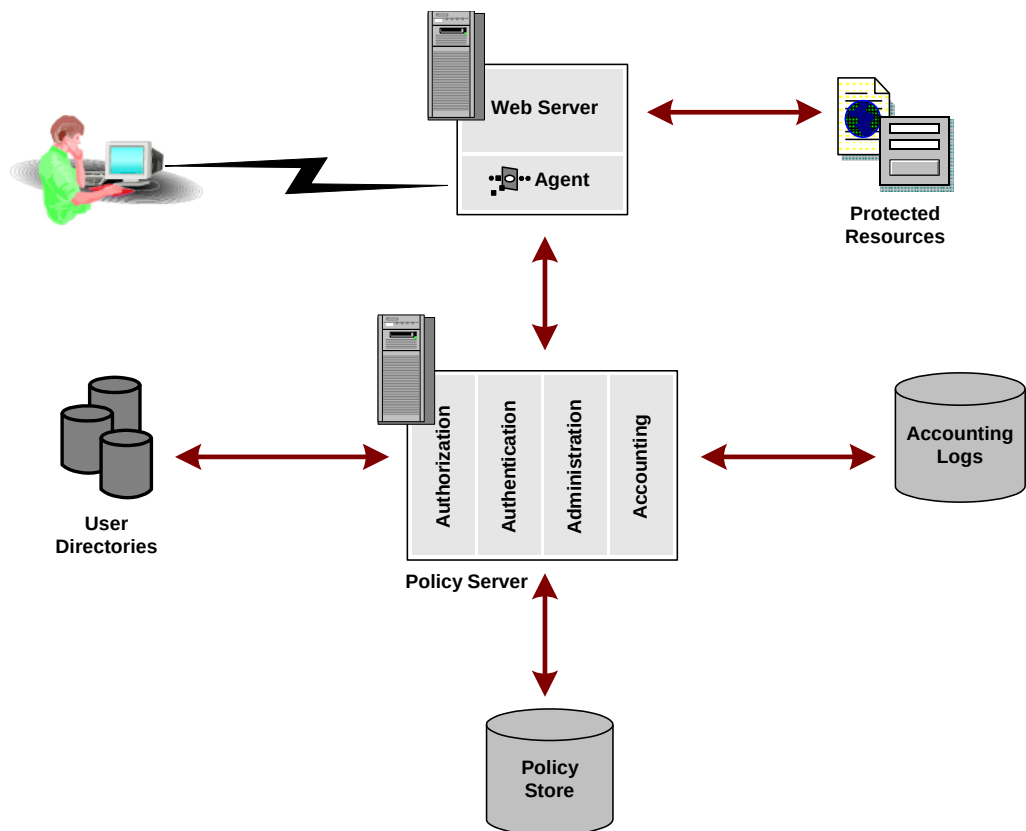
Accounting

The Policy Server generates log files that contain auditing information about the events that occur within the system. These logs can be printed in the form of predefined reports, so that security events or anomalies can be analyzed.

Health Monitoring

The Policy Server provides features for monitoring activity throughout a SiteMinder deployment.

The following figure illustrates a simple SiteMinder environment.



In a Web implementation, a user requests a resource through a browser. That request is received by the Web Server and intercepted by the SiteMinder Web Agent. The Web Agent determines whether the resource is protected, and if so, gathers the user's credentials and passes them to the Policy Server. The Policy Server authenticates the user against native user directories, then verifies if the authenticated user is authorized for the requested resource based on rules and policies contained in the Policy Store. Once a user is authenticated and authorized, the Policy Server grants access to protected resources and delivers privilege and entitlement information.

Note: Other types of Agents can be created using the Agent API.

More information:

[Create Custom Agents](#) (see page 136)

Policy Server Management

As a Policy Server system administrator, you may be responsible for one or more of the following areas of administrative responsibility:

System management

System-level configuration, management of Policy Server processes, and general administration of the SiteMinder environment, including:

- Management of system-level, domain-level, and global policy objects
- Management of users
- Management of keys and password policies
- Registering of trusted hosts

Domain management

Creation and maintenance of the objects that establish connections to network resources and determine access to those resources, including:

- Management of domain-level objects
- Management of users
- Management of password policies within specific domains

Policy Server System Management Overview

You perform most fundamental system configuration and management tasks using the Policy Server Management Console. Others tasks are performed using the Policy Server User Interface.

Policy Server system management tasks include the following:

- Starting and stopping Policy Server processes
- Configuring Policy Server Executives
- Cache Management
- Key Management

- Global Settings
- User Management

For more information, see the *Policy Server Management* guide.

Policy Server Object Management Overview

As you work with Policy Server you will configure many Policy Server objects. Some of these objects represent connections to existing network entities, while other objects represent groupings of resources and policies that determine entitlements to those resources.

More information:

[Policy Server Object Overview](#) (see page 39)

Chapter 2: Policy Server Object Overview

This section contains the following topics:

[Policy Server Object Types](#) (see page 39)

[Configuration Order](#) (see page 44)

[Administrative Roles](#) (see page 45)

Policy Server Object Types

There are three main categories of objects that the policy server uses:

- [System Objects](#) (see page 39)
- [Policy Domain Objects](#) (see page 41)
- [Global Objects](#) (see page 42)

System Objects

System objects can be used throughout a SiteMinder deployment. They include connections to existing user directories, administrators, Agents, authentication schemes, registration schemes, and password policies.

The following tables lists all of the system objects.

Object	Description
Agents	An Agent is installed on Web servers, application servers, or other network entities to secure access to resources. Once an Agent is installed on a server, a SiteMinder object must be configured for the Agent in the Policy Server User Interface.
Agent Groups	An Agent group is a Policy Server object that points to a group of Agents. The Agents in the group can be installed on different servers, but all of the Agents protect the same resources. Typically Agent groups are configured in SiteMinder for groups of servers that distribute the workload for access to a popular set of resources.
Agent Configuration Objects	An Agent Configuration Object holds configuration parameters for one or more Web Agents.

Object	Description
Host Configuration Objects	A Host Configuration Object holds configuration parameters for the Trusted host.
User Directories	A user directory in SiteMinder is an object that contains details for connecting to an existing user directory that resides outside of SiteMinder. This allows an administrator to configure a simple connection to an existing user directory, instead of replicating user information within SiteMinder.
Policy Domains	A policy domain is a logical grouping of one or more user directories, administrators, and realms. This Policy Server object is the basis for entitlement data. By creating policy domains, an administrator creates a container for entitlements that surround a particular groups of resources (realm), as well as the users who may access the resources, and the administrator who sets up entitlements.
Affiliate Domains	An affiliate domain is a logical grouping of SAML affiliates associated with one or more user directories and administrators. For more information, see the <i>Federation Security Services Guide</i>. This system object is only available if the SiteMinder Option Pack v6.0 SP 3/4 for the Policy Server is installed on the Policy Server.
Administrators	An administrator is an object that contains profile information for a SiteMinder administrator account. Everyone who logs into SiteMinder is considered an administrator. The privileges and activities of an administrator account vary by administrative role.
Authentication Schemes	An authentication scheme is a Policy Server object that determines the credentials a user will need to access a protected resource. Authentication schemes are assigned to realms. When a user tries to access a resource in a realm, the authentication scheme of the realm determines the credentials that a user must supply in order to access the resource.
Registration Schemes	A registration scheme is a Policy Server object that allows users to register themselves for access to a group of resources on a network and administrators to manage registered users. Registration schemes simplify the task of managing a large user database.
Agent Types	An Agent Type is a Policy Server object that defines the actions and response attributes supported by a type of Agent, such as Web, Affiliate, RADIUS, or custom.

Object	Description
SQL Query Schemes	A SQL Query Scheme is an object that stores SiteMinder SQL queries. These queries are used to retrieve information, such as a list of user groups, from relational databases (for example, Oracle) used as SiteMinder user directories.
Password Policies	Password policies are Policy Server objects that contain rules for passwords, including expiration dates, constraints, and composition requirements.
SAML Affiliations	A SAML affiliation is a group of SAML 2.0 entities that share a name identifier for a single principal. For more information, see the <i>Federation Security Services Guide</i> . Note: This system object is only available if the SiteMinder Option Pack v6.0 SP 3/4 for the Policy Server is installed on the Policy Server.
Trusted Hosts	A Trusted Host object represents the client component that connects to the Policy Server.
IdentityMinder Environment	An IdentityMinder Environment is a Policy Server object produced by IdentityMinder that must be associated with a directory and a policy domain. For more information, see <i>CA Identity Manager Operations Guide</i> . Note: This system object is only available if the IdentityMinder Extensions for Policy Server are installed.

Policy Domain Objects

A policy domain is a grouping of objects that deal with a specific domain of resources. For example, a company may divide its network resources by business unit, creating a policy domain for marketing, a separate domain for engineering, etc. Policy domain objects are those objects that pertain to a specific policy domain. These objects include rules and policies for controlling access to resources.

The following are policy domain objects:

Object	Description
Realms	A realm is a Policy Server object that identifies a group of resources. Realms typically define a directory or folder and possibly its subdirectories.

Object	Description
Rules	A rule is a Policy Server object that identifies a resource and the actions that will be allowed or denied for the resource. Rules can also include actions associated with specific events, such as what to do if a user fails to authenticate correctly when asked for their credentials.
Rule Groups	A rule group is a Policy Server object that contains multiple rules. Rule groups are used to tie together different rules that will be used in a single policy.
Responses	A response is a Policy Server object that determines a reaction to a rule. Responses are included in policies, and take place when a rule is triggered.
Response Groups	A response group is a Policy Server object that contains a logical grouping of responses. Response groups are most often used when many responses will be included in a policy.
Policies	A policy is a Policy Server object that binds users, rules, responses, and optionally, time restrictions and IP address restrictions together. Policies establish entitlements for a SiteMinder protected entity. When a user attempts to access a resource, the policy is what SiteMinder ultimately uses to resolve the request.
Variables	A variable is an object that can be resolved to a value which you can incorporate into the authorization phase of a request. The value of a variable object is the result of dynamic data and is evaluated at runtime.
Affiliates	An affiliate object binds users, and optionally, time restrictions and IP address restrictions together. It also contains configuration data and a list of user entitlement attributes to be passed to an affiliate after a user is authenticated. For more information, the <i>Federation Security Services Guide</i>. This system object is only available if the SiteMinder Option Pack v6.0 SP 3/4 for the Policy Server is installed on the Policy Server.

Global Objects

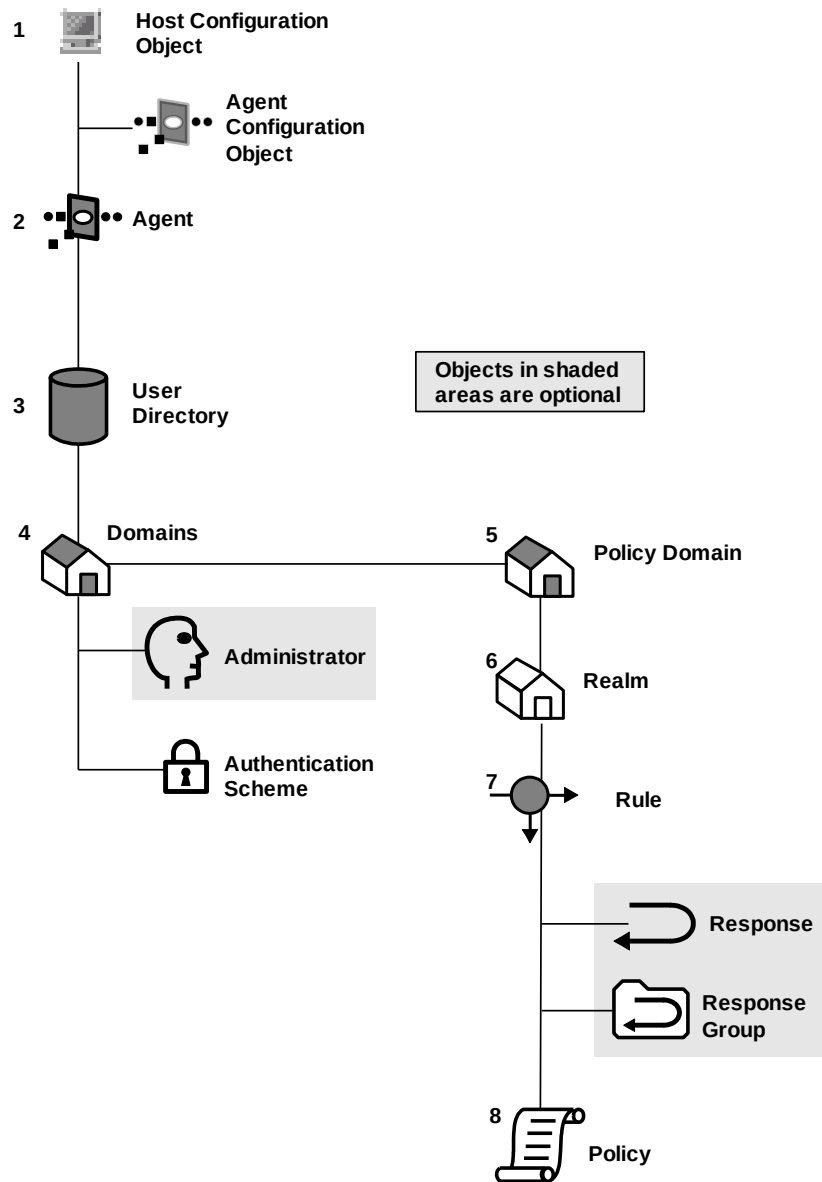
In addition to configuring policies for specific resources in a domain, you can also configure global policy objects that apply to all resources.

The following are global objects:

Object	Description
Global Rules	A global rule is a Policy Server object that specifies a filter used to apply a global policy to a large group of resources.
Global Responses	A global response is a Policy Server object that determines a reaction to a global rule. Global responses are included in global policies, and take place when a global rule is triggered.
Global Policies	A global policy is a Policy Server object that binds users, global rules, global responses, and optionally, time restrictions and IP address restrictions together. When a user attempts to access a resource, the global policy is what SiteMinder ultimately uses to resolve the request.

Configuration Order

When you begin working with the Policy Server User Interface to configure objects for your SiteMinder deployment, you will discover that certain objects are prerequisites of other objects. The following figure illustrates the configuration order for the core SiteMinder objects.



The numbers in the diagram indicate the sequence in which the objects must be configured.

Administrative Roles

The Policy Server User Interface allows you to configure administrator accounts that grant access to all Policy Server objects, specific subsets of resources called domains, or subsets of features, such as security or user management. You can assign accounts on a user name and password basis, or specify administrator privileges for specific administrators or groups of administrators who are stored in an existing user directory.

An administrator's privileges are set up using the SiteMinder Administrator dialog. For detailed information about delegating administrative privileges to individual administrator accounts, see [Policy Server Administrators](#) (see page 223).

Chapter 3: Implementing Policy-based Security

This section contains the following topics:

[Overview](#) (see page 47)

[Strategies for Managing Security and Users](#) (see page 47)

[How a Security Model Is Established](#) (see page 54)

[How Role-Based Access Control Is Implemented](#) (see page 59)

Overview

Managing users and securing resources successfully are critical aspects of administering an application, a Web site, or a portal. To manage users and secure resources effectively, you must:

- Provide users with easy and fast access to appropriate information.
- Provide security users can trust.
- Protect resources from users who should not have access.

Failing to manage users and secure resources effectively can have negative affects on you and your clients, such as:

- **Jeopardizing resources** if security is inadequate.
- **Losing sales** if potential customers lack confidence in the security of your site and avoid the site.
- **Frustrating customers** if the process of accessing resources is cumbersome.
- **Losing consumer data** if you can't track users adequately.

Developing and implementing a strategy that secures your resources and also manages your user base is critical when you build a Web site or Web application.

Strategies for Managing Security and Users

A complete security solution should answer the following questions:

- Where is security needed and how much security is needed?
- What kind of tasks will be performed? Security-related tasks?
- What resources need to be protected and what level of protection is required?

- Who has access to the resource and what are they authorized to do?
- How is control over user access to protected resources enforced?

Two of the most common methods for managing users and securing resources are *access control lists* (ACLs), and *security policies*.

Security policies provide the most complete security solution by defining not only the type of access a user or user group has to a resource but also what happens when a user or user group accesses the resource. Security policies go beyond the capabilities of ACLs by enabling you to manage the user experience. The SiteMinder authorization model is based on security policies.

About Access Control Lists

An access control list is an object associated with a resource that defines access privileges for individual users or groups of users. ACLs are associated with resources to establish:

- Who can access a resource
- What type of access the user has

ACLs provide a straightforward way of granting or denying a specified user or groups of users access to a resource. For example:

`{Manager:ALL, Clerk:READ, Others:NONE}`

The access control list above assigns users in the manager group complete access, users in the clerk group read-only access, and users in all other groups no access to the resource.

Several drawbacks are associated with ACLs:

- Controlling a user's interaction with a resource once the user is authorized and authenticated is difficult. For example, you cannot use ACLs to define session timeouts.
- Managing ACLs for large numbers of resources can be prohibitively time consuming and costly. ACLs create a significant administrative burden.
- Profiling user information is difficult. Users are added to ACLs, which are then associated with resources, making it difficult to determine all of the resources a user has access to at any given moment.
- Qualifying access rights is difficult. It's virtually impossible to restrict access based on a time or a location.

- Personalizing content and defining responses is difficult. If a user has access to a resource, it's difficult to personalize the content or define what happens when the user is allowed or denied access. For instance, you can't use an ACL to display or hide a set of buttons when the user accesses the resource.

ACLs are an effective way to protect a resource but an ineffective way to manage the user experience.

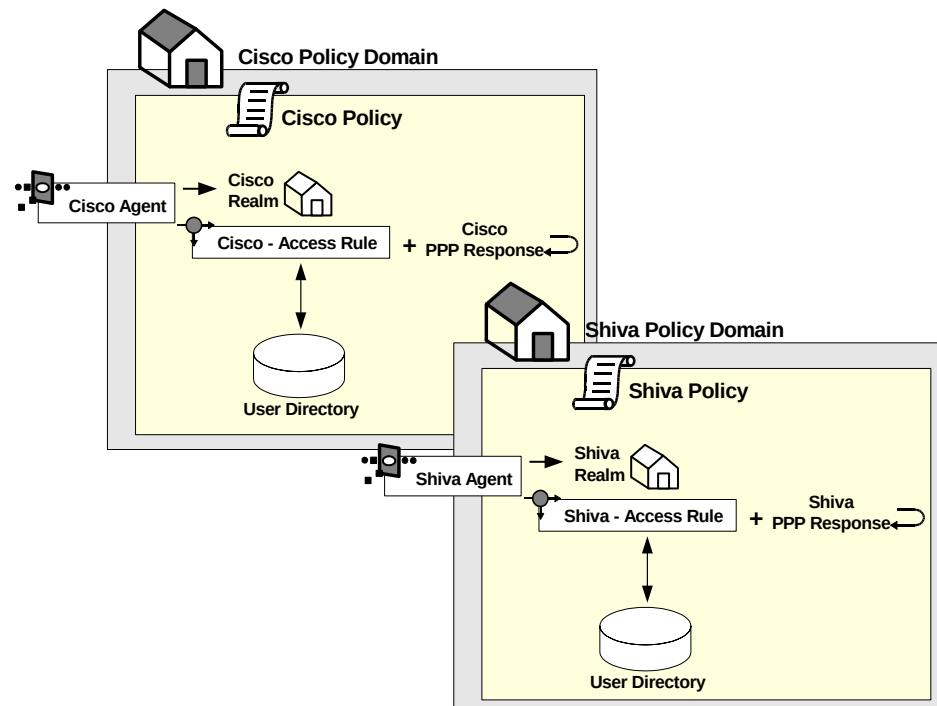
About SiteMinder Security Policies

Unlike ACLs, policies serve a dual purpose: policies provide security and manage the user experience. Policies are user-centric: policies are constructed around the user group rather than the resource.

Policies define access permissions using rules, responses, and time/location constraints. Policies are then associated with users or user groups to establish:

- Where the resource is located
- Who can access a resource
- What type of access the user has
- When they can access a resource
- What happens when they access the resource
- What happens if they can't access the resource

The following graphic provides a definition of a SiteMinder security policy.



Policies provide an effective means of managing users and securing resources for the following reasons:

- Policies provide more granularity and the ability to personalize content. Responses enable you to define what happens when a user is allowed or denied access, such as which graphics are shown if a user is allowed access or where to redirect the user if the user is denied access.
- Policies are easy to maintain. When a user is modified, added to the group, or deleted from a group, all policy definitions that include the user are automatically updated.
- Policies provide fine-grained security. Policy definitions can include time restrictions and location (I.P.) restrictions.

Because of the power and flexibility of policies, authorization models based on security policies are more efficient and effective than models based on ACLs.

How the End-user Experience Is Managed

In addition to securing resources, policies in SiteMinder can be used to manage the end-user experience by:

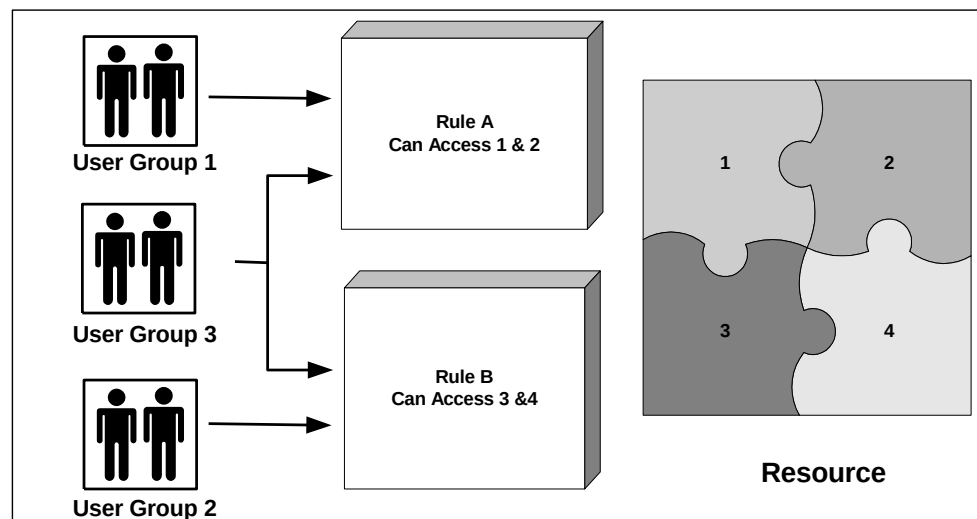
- [How Privileges Are Established](#) (see page 51)
- [How Content Is Personalized](#) (see page 51)
- [How Sessions Are Managed](#) (see page 52)

How Privileges Are Established

In a policy, the privilege to access a resource is established by adding a rule to a policy. Rules identify specific resources and either allow or deny the user access to the resources. A single policy can establish privileges for many users: policies can be associated with an individual user, a user group, or the members of an entire user directory.

For example, in the following graphic:

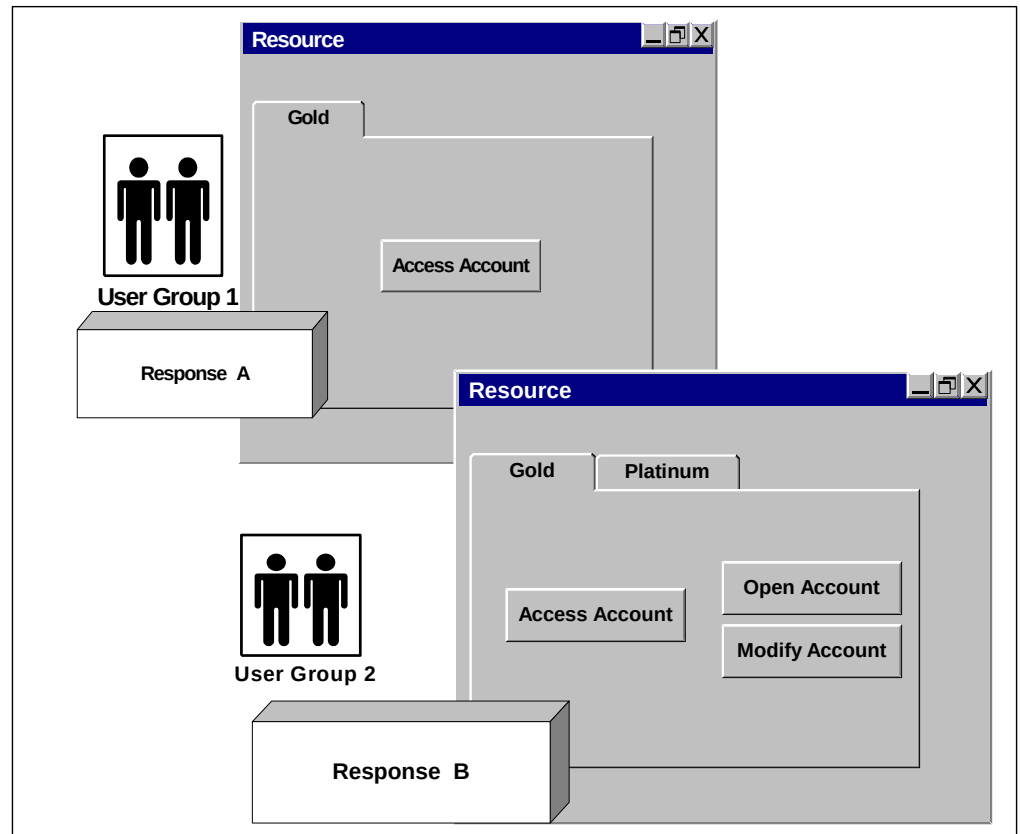
- Group 1 has been assigned Rule A and can access resources 1 & 2,
- Group 2 has been assigned Rule B and can access resources 3 & 4, and
- Group 3 has been assigned both rules and can access all resources.



How Content Is Personalized

Once a user has been granted access to a resource, the policy can then personalize the user's view of the resource. Policies can customize the view of a resource through the use of responses. Responses are paired with rules and are triggered when a rule fires.

For example, in the following graphic, both Group 1 and Group 2 can access the Resource dialog. However, the view each group has of the dialog differs. The policy for Group 1 uses Response A, which does not provide two buttons (Open Account and Modify Account) or the Platinum tab that Response B makes available.



How Sessions Are Managed

By managing sessions, you control how long an authenticated and authorized user can access the resource. You can control sessions by:

- Specifying the amount of time a user can remain idle, without interacting with the resource.

Idle timeouts protect against unauthorized use of the resource by limiting the amount of time the session remains active if it is not being used. The idle timeout is particularly useful in cases where users leave their computer without logging out of their session. When the idle timeout limit is reached, the session automatically ends.

- Specifying the maximum amount of time a user can access a resource before they must re-authenticate.

Maximum timeouts protect against unauthorized use of a resource by forcing an authenticated user to re-authenticate after a specified time. This safeguard ensures that if an authenticated user leaves the computer without logging out and someone else uses the open session, the session will end after a specified amount of time, and the user must re-authenticate to continue using the resource.

- Revoking a user's access to a resource at any time.

In addition to managing how long a session can remain active, you can also end a session immediately if you suspect the integrity of a resource has been compromised. Once a user session has been revoked, the user is disabled in the user directory until you have re-enabled the user with the Policy Server User Interface.

For more information about managing sessions, see the *SiteMinder Policy Server Management Guide*.

- Enabling persistent sessions.

You can also implement persistent sessions to provide Windows security context functionality and support for Federated Web Services.

More information:

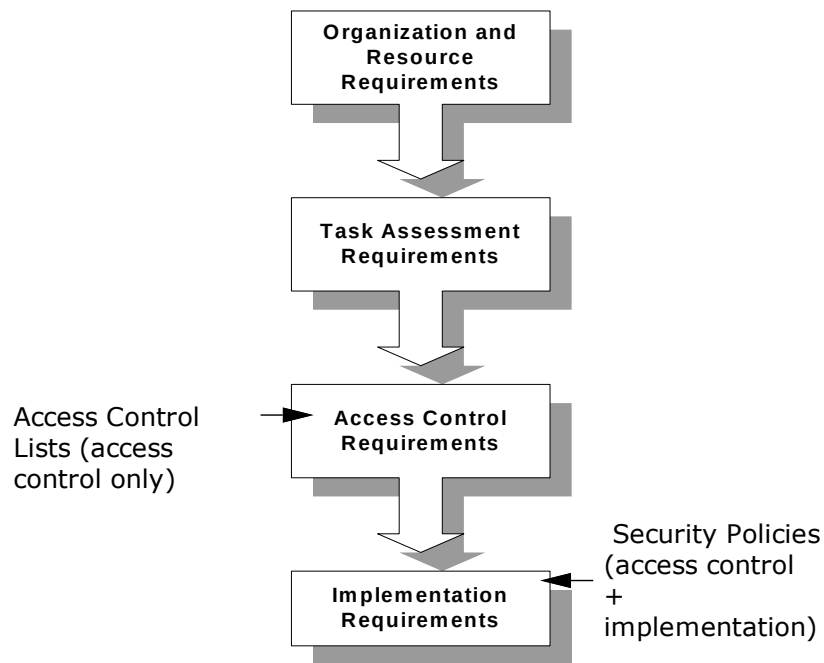
[How Privileges Are Established](#) (see page 51)

[How Content Is Personalized](#) (see page 51)

[How Persistent Sessions for User Security Contexts Are Maintained](#) (see page 95)

How a Security Model Is Established

To implement a security model that best meets the needs of your organization, you may create security policies using information gathered in the design phases shown and described below.



1. Organization and resource requirements—set the basic objective of the security model and identify the resources.
2. Task assessment requirements—identify users and roles, and link the roles to tasks.
3. Access control requirements—establish access requirements for users based on their role requirements.

Authorization models based only on access control lists (ACLs) end at this point.

4. Implementation requirements—define how the access is implemented (in terms of how users are tracked and how content is personalized for users) and how user sessions are managed.

Authorization models based on SiteMinder security policies incorporate both access control and implementation models.

More information:

[How Organization and Resource Requirements Are Established](#) (see page 55)

[Define Task-Assessment Requirements](#) (see page 57)

How a Security Model Requirements Table Is Created

When completing each phase of the security model design methodology, use a table similar to the one shown below to organize your information. Once you have completed the columns in this table, you can use the information to build SiteMinder security policies.

Resource	Role	Task	Access	Implementation

The following sections describe how to gather information to complete this table.

How Organization and Resource Requirements Are Established

Before you can setup a security model, you must establish the organization and resource needs of your organization. Some general issues to consider include:

- Are there security guidelines, regulations, or laws your organization is required to meet?
- If security needs conflict with business requirements, which requirements take precedence?
- Has a lack of security affected the reputation of your organization?
- Have security incidents in the past caused financial loss or taken down the site?
- Once you have established the overall security requirements and concerns, you are ready to define the more specific security requirements outlined below.

How Organization Security Requirements Are Defined

To determine the more specific security needs, consider the following issues:

These organization requirements:	Affect these tasks:
■ Who requires access to the resources? ■ How much access do they require? ■ Can you categorize users with similar access requirements into groups?	Configuring user directory connections
■ Which resources require protection? ■ Do different resources require different levels of protection?	Creating policy domains and realms
■ How sensitive and valuable is the information? ■ How much do you trust your users? ■ Are your users local or remote? ■ What type of security do your users expect? ■ Will you lose customers if security does not match their expectations?	Creating authentication schemes using from authentication templates
■ Are there security guidelines, regulations, or laws your organization is required to meet? ■ Do different objects require fine-grained protection or personalization? ■ What type of actions do you want to control?	Defining rules
■ What type of security and controls do your users and customers expect? ■ Do different groups of users require different views of the resource? ■ What events should take place when a user is authenticated or authorized?	Defining responses
■ How will you implement your requirements?	Defining policies

Identify Resources and Roles

The second part of establishing organization requirements is to identify resources and map resources to roles.

The purpose of this step is to link resources with roles. By linking these two components, you will have a better understanding of what needs protection and what type of protection is required.

When identifying resources:

- Identify all known resources, including resources that are planned but do not yet exist. Planning security for all known resources at once, whether they currently exist or not, will save you time.
- Create a site map for Web sites to better understand the structure of the resources.

How this applies to policies:

Resources are defined in realms and rules. Roles of users are implied based on the user group to which they belong or based on their user attributes. In an airline application, for example, a user belonging to the Pilot user group would perform tasks associated with the Pilot role.

To identify resources and roles

1. Using a table or chart similar to the security model table described in [How a Security Model Requirements Table Is Created](#) (see page 55), list the resources in the Resources column.
2. Identify all subdivisions of a single resource. For example, if a directory called /bidding had two subdirectories below, such as /bidding/flights and /bidding/standby, both subdirectories would be listed as resources. By treating each subdirectory as a separate resource, it will be easier to determine if each resource requires separate security.
3. Next to each resource, list the roles that will need access to the resource.

Define Task-Assessment Requirements

Task assessment requirements bridge the gap between roles and access-control requirements. When you identify task-assessment requirements, you define the tasks each role performs using the resource.

How this applies to policies:

Although tasks are not specifically defined in policies, you will use this information when assessing access rights for each resource-role pairing in the next section.

To define task-assessment requirements

1. For each role, identify the tasks the role must perform using the resource.
2. Enter the task in the Task column, next to the associated resource and role.

More information:

[Define Access Control Requirements](#) (see page 58)

[How a Security Model Requirements Table Is Created](#) (see page 55)

[How Organization and Resource Requirements Are Established](#) (see page 55)

[Define Implementation Requirements](#) (see page 58)

Define Access Control Requirements

Access control requirements establish whether or not a role requires access to a resource, and if so, the type of access they require. Two roles that access the same resource may not require the same access to the resource.

For example, two groups of users might perform tasks in the same directory. Group A might use this resource to view and modify the resource, while Group B members would view the resource. Because each user group uses the resource in a different way, access rights would differ:

- Group A: Get, Post
- Group A: Get

How this applies to policies:

Access rights are defined in rules.

To define access control requirements

1. For each task, identify the type of access that is required to perform the associated task.
2. Enter the access requirement in the Access column.

Define Implementation Requirements

Implementation requirements define how the resource is used. How a resource is used can be configured by many variables, including:

- Personalization
- Time limitations for using the resource
- Redirections

How this applies to policies:

Implementation requirements are defined in responses.

To define implementation requirements

1. For each task, identify how access should be implemented.
2. Enter the implementation requirement in the Implementation column.

With the security model information complete, you can begin constructing policies that are appropriate for your site.

More information:

[How a Security Model Requirements Table Is Created](#) (see page 55)

[How Organization and Resource Requirements Are Established](#) (see page 55)

[Define Task-Assessment Requirements](#) (see page 57)

How Role-Based Access Control Is Implemented

In addition to creating access control policies based on users or groups, you can implement access control based on roles. If you have purchased CA Identity Manager (formerly IdentityMinder), you can create roles and assign roles to users and groups. For more information, see the *CA Identity Manager Operations Guide*.

More information:

[Access Roles in Identity Manager](#) (see page 59)

[Identity Manager Environment Objects](#) (see page 60)

[Overview](#) (see page 47)

[Strategies for Managing Security and Users](#) (see page 47)

[How a Security Model Is Established](#) (see page 54)

Access Roles in Identity Manager

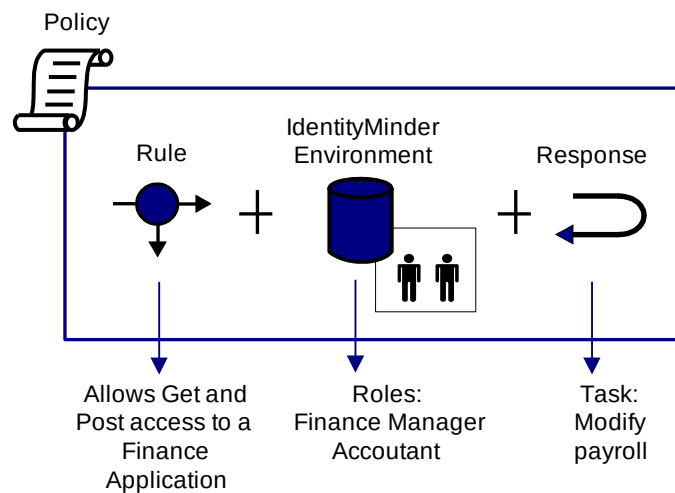
Access roles enable centralized management of user privileges in external applications secured by SiteMinder. Identity Manager administrators can create and assign roles in Identity Manager that determine users' access to applications outside of Identity Manager. For example, a Role Administrator may create roles in Identity Manager that control access to a finance application, and grant the ability to assign the roles to the Help Desk administrator. The Help Desk administrator can assign or revoke that role through the Identity Manager interface.

Access roles are enabled through integration with SiteMinder. SiteMinder associates roles with policies to determine which users can access a protected resource and to deliver user-specific role and task information to protected resources.

To configure roles-based access control to protected resources, a SiteMinder administrator associates an Identity Manager Environment with a Policy Domain in the Policy Server User Interface. The administrator creates a policy to protect an application and associates a role or roles with that policy. Users who have an associated role can access the protected application.

SiteMinder can also provide details about entitlements that a user has in protected applications. To provide entitlement information to an application, a SiteMinder administrator associates a response with an access rule in the application's policy (see the following figure).

The response contains a response attribute that specifies a SiteMinder-generated user attribute. The SiteMinder-generated user attribute retrieves task information from Identity Manager. The Policy Server passes this information to the Web Agent as an HTTP header variable or a cookie. The Web Agent makes the header variable or cookie available to the protected application for fine-grained access control.



Identity Manager Environment Objects

An Identity Manager Environment is a SiteMinder object produced by Identity Manager that must be associated with a directory and a policy domain. It provides a view of a management namespace that allows an administrator to manage objects such as users, groups, and organizations with a set of associated roles and tasks.

For information about adding Identity Manager Environments to policy domains, see [Add Identity Manager Environments to Policy Domains](#) (see page 312).

For information about configuring a policy to include roles, see [Add Roles to a Policy](#) (see page 387).

Chapter 4: Policy Server User Interface

This section contains the following topics:

[Policy Server User Interface Overview](#) (see page 63)

[Policy Server Administrator Accounts](#) (see page 63)

[Start the Policy Server User Interface](#) (see page 64)

[Use the Policy Server User Interface](#) (see page 65)

[Work with Policy Server Objects](#) (see page 75)

[How the Web Agent and Policy Server Calculate Time](#) (see page 83)

Policy Server User Interface Overview

The SiteMinder Policy Server is managed through a graphical user interface. The user interface has been designed to make managing SiteMinder deployments as intuitive as possible. This chapter discusses how to login to the Policy Server User Interface, the basic parts of the interface, and the common procedures you will use while configuring and managing Policy Server objects.

Policy Server Administrator Accounts

In order to use the Policy Server User Interface, you must log into Policy Server User Interface using a valid administrator account. Each administrator account has a set of privileges.

More information:

[Policy Server Administrators](#) (see page 223)

Default Super User Administrator Account

When you install the Policy Server, the installation process creates a single “Super User” administrator account automatically. The account has the maximum system privileges, and can be accessed using the following credentials:

User Name

SiteMinder

Password

The password that you entered during the installation

You can use this account to configure any type of Policy Server object, including other administrator accounts.

More information:

[Administrator Concepts](#) (see page 223)

Start the Policy Server User Interface

In order to use the Policy Server User Interface, you must login as a valid administrator.

To start the Policy Server User Interface

1. Open your browser.
2. Enter `http://hostname.domain/siteminder` in the URL field and press Enter.

The *hostname* is the name of the machine on which the Policy Server is installed. The *domain* is the fully qualified domain name which must use two periods. For example, a host name of `siteminder` and a domain of `.security.com` requires the following to log into Policy Server User Interface:

`http://siteminder.security.com/siteminder`

Note: You must use a fully qualified host name in the URL. If the Web server on the host machine is not running on the standard HTTP port (80), you must append a colon and a port number to the end of the domain. For example:

`http://siteminder.security.com:81/siteminder`

Your browser displays the main Policy Server page.

3. Click Administer Policy Server.

A status bar appears while the Policy Server User Interface loads. There may be a short delay. When the application has loaded, the SiteMinder Administration Login window opens.

4. Enter a valid user name and password in the appropriate fields.

If you are accessing Policy Server User Interface for the first time, use the default super user administrator account, which you created during Policy Server installation.

5. Click the Login button.

The system displays the SiteMinder Administration window. This is the main window of the Policy Server User Interface. The contents of this window differ based on the privileges of the administrator account you use to login to the user interface. The differences are based on whether your administrator account has privileges to manage all Policy Server objects or only a subset of policy domain objects.

More information:

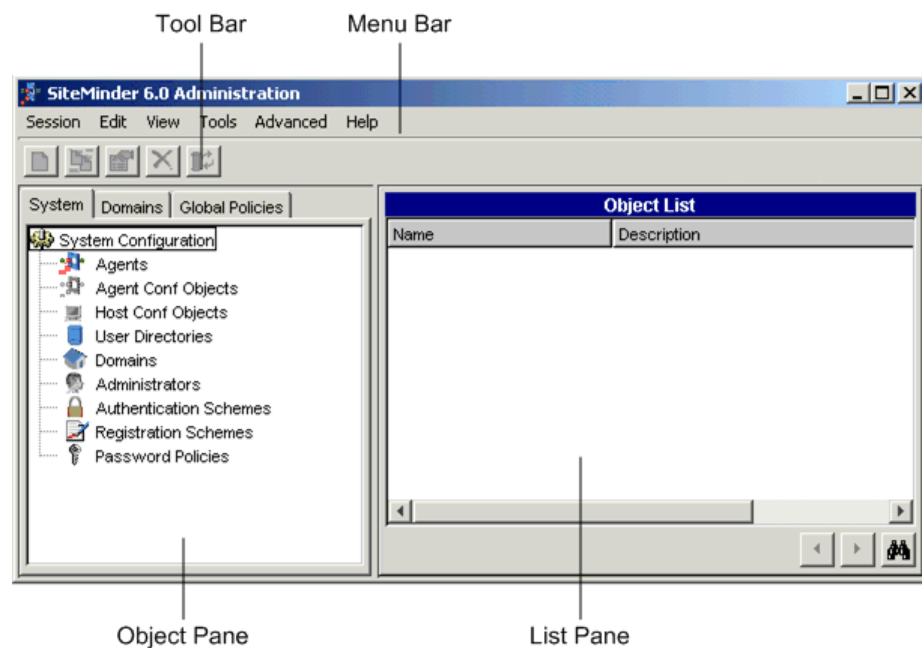
[Default Super User Administrator Account](#) (see page 63)

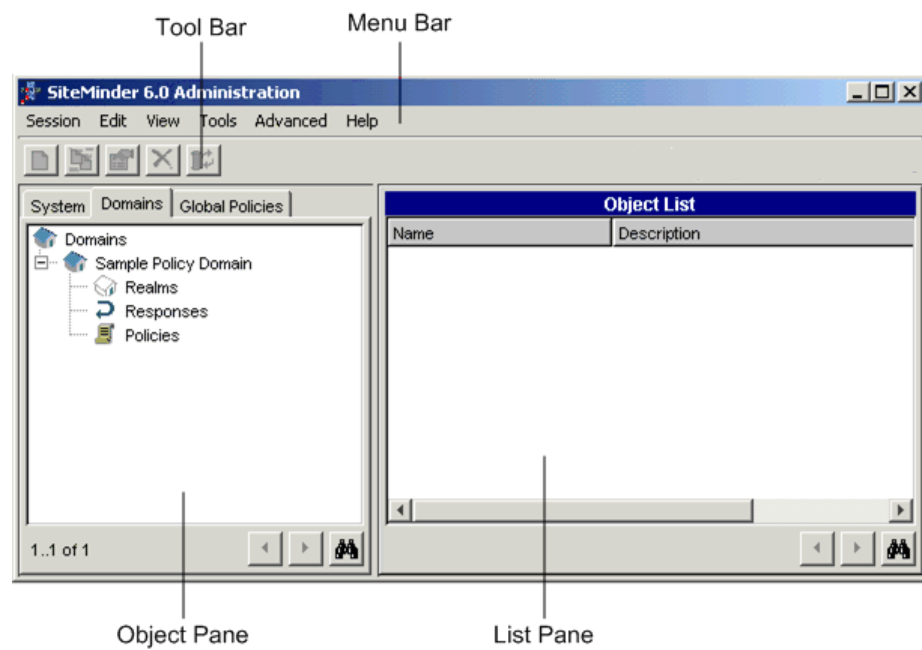
[Administrator Concepts](#) (see page 223)

Use the Policy Server User Interface

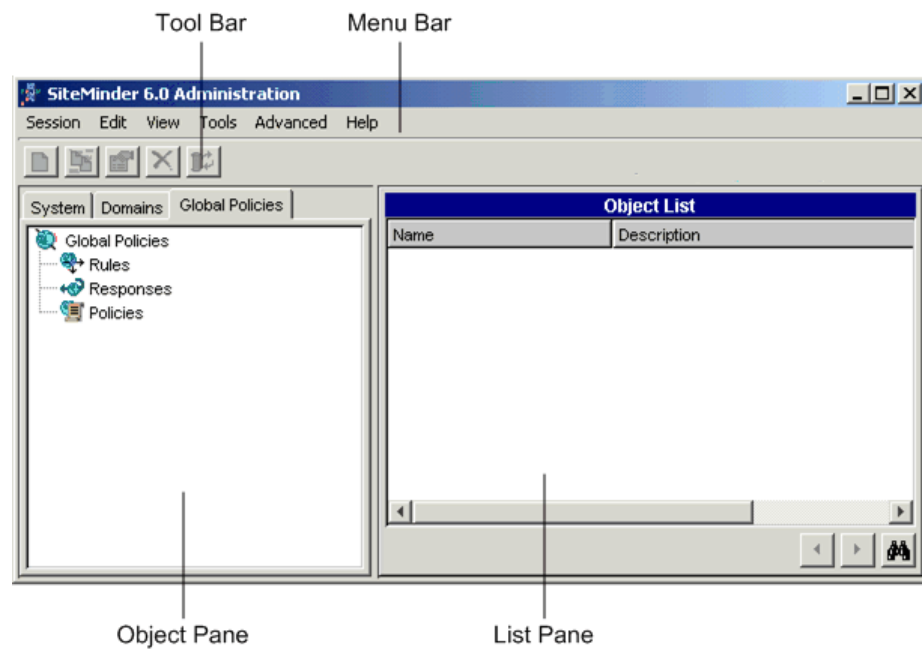
The Policy Server User Interface enables you to manage all of the objects you can configure with the Policy Server User Interface. This section discusses the basic techniques you can use while creating and administering your Policy Server objects.

The following figures show the Policy Server User Interface as it appears for an administrator who can manage all system and domain objects, and for an administrator who can manage only policy domain objects.





Note: If no policy domains exist, the Object pane is blank.



The Policy Server User Interface contains the following controls:

Menu Bar

The menu bar contains menus that allow you to log off of SiteMinder, create, edit, and delete SiteMinder objects, access reports, administration tools and advanced features, and open SiteMinder's on-line documentation.

Tool Bar

The tool bar contains icons that allow you to create, edit, copy, and delete SiteMinder objects.

Object Pane

The object pane can contain three tabs: System, Domain, and Global Policies. These tabs allow you to access a hierarchy of SiteMinder objects.

Note: For an administrator with access to only specific policy domains, the Object pane contains neither a System, Domains, and Global Policies tab. Instead it contains only the policy domains to which the administrator has access.

System Tab

This tab contains a list of objects you can add to a SiteMinder installation.

The objects include Agents, Agent Types, Agent groups, user directories, policy domains, administrators, authentication schemes, registration schemes, password policies, and SQL query schemes. The following chapters describe what these objects can do, and how you can configure them with the Policy Server User Interface.

Note: To view icons for Agent Types, Agent Groups, Registration Schemes, or SQL Query Schemes, select the appropriate menu item from the View menu.

Domains Tab

For an administrator with access to all SiteMinder objects, this tab contains a list of all existing policy domains and allows the administrator to manage SiteMinder objects associated with a policy domain.

Policy domain objects include: realms, rules, rule groups, responses, response groups, and policies.

Note: To have Rule Groups or Response Groups appear in the Object Pane of the SiteMinder Administration window, select the appropriate menu item from the View menu.

Global Policies Tab

For an administrator with the Manage System and Domain Objects privilege, the tab pane contains a tab for Global Policies. The objects in this tab allow you to create policies that apply to all protected resources.

List Pane

The columns of information in the List pane change based on your selection in the Object pane. When you select an object in the Object pane, the List pane displays all of the objects of the selected type that have been configured for your SiteMinder installation.

More information:

[Global Policies, Rules, and Responses](#) (see page 435)

How to Work with the Object Pane

The following sections describe methods for using the controls in the Policy Server User Interface. For specific information about the objects you can access from the SiteMinder Administration window, see the chapters in this guide that deal with specific SiteMinder objects.

Change Tabs

If you are logged into SiteMinder as an administrator with access to all SiteMinder objects, the Object pane contains three tabs. These tabs contain system objects and policy domain objects.

To switch between tabs, click on a tab name. The selected tab becomes the current tab in the Object pane.

More information:

[Start the Policy Server User Interface](#) (see page 64)

Expand and Collapse Objects

Policy Domains listed in the Domains tab of the Object pane can be expanded and collapsed to show or hide nested objects. For example, if you have created multiple policy domains, each one of the policy domains contains some number of nested realms, rules, and policies.

You can expand and collapse the nested objects by clicking on the + and - symbols to the left of the policy domain names in the Domains tab.

SiteMinder List Pane

The List pane displays information associated with the object you select in the Object pane. When you select an object in the Object pane, the List pane changes accordingly.

SiteMinder Menu Bar

The SiteMinder menu bar contains menus that allow you to access all of SiteMinder's administrative features. The menus include the following options:

- Session

- Exit and Logout**

- Closes the SiteMinder Administration window and logs you out of the system.

- Edit

- Create <selected object type>**

- Opens an empty dialog for an object of the type that is currently selected in the Object pane or the List pane.

- Duplicate <selected object>**

- Creates a duplicate of the selected object and opens the corresponding dialog.

- Properties of <selected object>**

- Opens a dialog and displays properties for the object selected in the List pane.

- Delete <selected object>**

- Deletes the object selected in the List pane. A dialog asks you to confirm the deletion.

- System Configuration**

- Expands a submenu that contains selections for creating each type of SiteMinder system object.

- <domain name>**

- This menu item's name changes to the name of a selected policy domain. It expands a submenu that contains selections for creating each type of SiteMinder policy domain object.

- <realm name>**

- This menu item's name changes to the name of the selected realm. It expands a submenu that contains selections for creating a nested realm, creating a rule, or flushing resources contained in the selected realm from the cache.

- View

Agent Groups

Adds/removes Agent groups from the System tab of the Object pane. A check mark to the left of *Agent Groups* in the View menu indicates that Agent groups are available in the System tab.

Agent Types

Adds/removes Agent types from the System tab of the Object pane. A check mark to the left of *Agent Types* in the View menu indicates that Agent types are available in the System tab.

Password Policies

Adds/removes password policies from the System tab of the Object pane. A check mark to the left of *Password Policies* in the View menu indicates that password policies are available in the System tab.

Registration Schemes

Adds/removes registration schemes from the System tab of the Object pane. A check mark to the left of *Registration Schemes* in the View menu indicates that registration schemes are available in the System tab.

Response Groups

Adds/removes response groups from the Domains tab of the Object pane. A check mark to the left of *Response Groups* in the View menu indicates that response groups are available in the Domains tab.

Rule Groups

Adds/removes rule groups from the Domains tab of the Object pane. A check mark to the left of *Rule Groups* in the View menu indicates that rule groups are available in the Domains tab.

SAML Affiliations

Adds/removes SAML affiliations from the System tab of the Object pane. A check mark to the left of *SAML Affiliations* in the View menu indicates that SAML affiliations are available in the System tab. For more information, see the *SiteMinder Federation Security Services Guide*.

SQL Query Schemes

Adds/removes SQL query schemes from the System tab of the Object pane. A check mark to the left of *SQL Query Schemes* in the View menu indicates that SQL query schemes are available in the System tab.

Trusted Hosts

Adds/removes trusted hosts from the System tab of the Object pane. A check mark to the left of *Trusted Hosts* in the View menu indicates that trusted hosts are available in the System tab.

Variables

Adds/removes variables from the Domains tab of the Object pane. A check mark to the left of *Variables* in the View menu indicates that variables are available in the Domains tab.

- Tools

Global Settings

Opens the SiteMinder Global Settings dialog. For more information, see the *SiteMinder Policy Server Management Guide*.

Manage Cache

Opens the SiteMinder Cache Management dialog. For more information, see the *SiteMinder Policy Server Management Guide*.

Manage Keys

Opens the SiteMinder Key Management dialog. For more information, see the *SiteMinder Policy Server Management Guide*.

Manage User Accounts

Opens the User Management dialog. For more information, see the *SiteMinder Policy Server Management Guide*.

- Advanced

Directory Mapping

Opens the Directory Mappings dialog.

Certificate Mapping

Opens the Certificate Mappings dialog.

- Help

SiteMinder Help

Opens SiteMinder online help.

Online Manuals

Opens your default browser and displays a PDF file with links to PDF versions of all of books in the SiteMinder documentation set.

Note: Your browser must be equipped with the Adobe Acrobat Reader plug-in, which you can download from <http://www.adobe.com>.

About SiteMinder

Opens the About SiteMinder window, which contains copyright, version, and policy store schema information about your installation of SiteMinder.

More information:

[Configure SQL Query Schemes](#) (see page 197)

[Directory Mappings Dialog](#) (see page 590)

[Certificate Mapping](#) (see page 293)

SiteMinder Tool Bar

The SiteMinder Administration window contains a tool bar with the following icons:

	Create <object>—Opens an empty dialog for an object of the type that is currently selected in the Object pane or the List pane.
	Properties of <object>—Opens a dialog and displays properties for the object(s) selected in the List pane.
	Delete <object>—Deletes the selected object(s) from the List pane. You must confirm any deletions in a confirmation dialog that opens when you select Delete <object>.
	Copy <object>—Copies the selected rule. Use this function to create new rules that are similar to an existing rule. When you highlight a rule in the List pane and click the Copy button, the SiteMinder Rule dialog appears. This dialog displays the properties of the rule you selected. Change the Name of the rule and any other parameters, then click OK to save the new rule.
	Refresh—Updates the contents of the object tree when an object in the Domains tab is selected.

More information:

[Rules](#) (see page 327)

Scroll Between Groups of Items

At the bottom of each pane of the SiteMinder Administration window, the number of items of the selected object type appears. For example, if the Object pane contains a list of 15 rules, the bottom of the List pane displays 1..15 of 15. If a policy domain contains 250 rules, the Object pane displays a scrolling list of 100 rules, and the number of items appears as 1..100 of 250.

The Object pane and the List pane also contain Previous and Next (Arrow) buttons.

These buttons retrieve records from the Policy Store database. The Policy Store is where SiteMinder stores all SiteMinder objects. If you have more than 100 records of a particular type that are currently displayed in either the Object pane or the List pane, you can click the Previous (Left Arrow) button to retrieve the previous 100 records from the Policy Store, or you can click the Next (Right Arrow) button to retrieve the next 100 records from the Policy Store.

In the example above, if you click the Next button, the next group of 100 rules appears in the List pane, and the number of items changes to 101..200 of 250.

Search with the SiteMinder Administration Window

The main SiteMinder Administration window contains an icon in the lower right corner of the window for the Search tool. The search icon looks like a pair of binoculars. In the main window, the Search tool allows you to locate a specific object quickly.

To search for an object

1. Click on the Search icon in the lower right corner of the SiteMinder Administration window.

The area along the bottom of the List pane displays the Search tool.

2. Enter a text string in the Filter field.

The text string can contain the * or & wildcard characters.

3. Click the Go button.

The Search tool closes and the List pane displays only the objects with names that match the text string you entered in the Filter field. The area to the right of the scroll buttons displays the text string you entered in the Filter field. To display all objects, click the Search icon again, then click the All button.

Note: If you select a different object type in the Object pane, the List pane displays only objects whose names match the text string you entered in the Filter field.

Policy Server Object Icons

All of the objects you create and configure using the Policy Server User Interface are associated with an icon. These icons can help you identify the type of Policy Server object with which you are working. Following is a list of icons used to represent objects:

System Objects	Domain Objects	Global Policies Objects
 Agents	 Realms	 Global Policies
 Agent Groups	 Rules	 Rules
 Agent Conf Object	 Rule Groups	 Responses
 Host Conf Objects	 Responses	 Policies
 User Directories	 Response Groups	
 Policy Domains	 Policies	
 Affiliate Domains	 Affiliates	
 Administrators		
 Authentication Schemes		
 Agent Types		
 Registration Schemes		
 Password Policies		
 SQL Query Schemes		
 Trusted Hosts		
 IdentityMinder Environment		

Work with Policy Server Objects

As you work with the Policy Server User Interface, you must create, view, edit, and delete Policy Server objects. Although the particular details for these tasks differ by object, the general methods are similar. For example, the process for deleting an Agent is similar to the process for deleting a response, etc.

The following sections describe the general processes for creating, viewing, editing, and deleting Policy Server objects. Other chapters in this guide provide detailed descriptions for configuring each type of object.

Required Fields

As you configure and edit Policy Server objects, you work in an array of dialogues. If you configure an object and one or more of the fields in the dialog is preceded by an asterisk (*), the field is required. You cannot save a Policy Server object unless all required fields contain values. The Policy Server User Interface displays an error message if you attempt to save an object without providing entries in all required fields.

Configure a Policy Server Object

There are many ways to configure the various policy server objects. Later topics will discuss in detail how the objects can be configured to best serve your needs.

To configure a policy server object

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the Policy Server object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to chose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

3. In the selected tab, do one of the following:
 - Double click on the object type.
 - Click on an object type, then click the Create button in the tool bar.

 - From the menu bar, select Edit , System Configuration , <object type>.

- Click on an object type, then from menu bar, select Edit , Create <object type>.
- Right click on an object type and select Create <object type> from the menu.

The dialog associated with the selected object type opens.

4. Enter appropriate information in the dialog.

See the chapter associated with the object type for detailed information about the fields and controls contained in the dialog or click the Help button in the dialog.

Note: The asterisk (*) and ampersand (&) characters may not be used in the Name field of any Policy Server object. These characters are invalid in object names and should be avoided.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

5. When you have finished configuring a new object, do one of the following:
 - Click the OK button to save the object and close the dialog.
 - Click the Apply button to save the object without closing the dialog.
 - Click the Cancel button to close the dialog without saving the object.

More information:

[Start the Policy Server User Interface](#) (see page 64)
[Administrator Concepts](#) (see page 223)

View the Properties of a Policy Server Object

Policy server objects contain numerous properties that can be viewed individually or as a group. The next section will describe how to view multiple objects. If you wish to view the properties of only a single object, however, use the following procedure.

To view the properties of a policy server object

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to chose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

3. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

4. In the List pane, do one of the following:

- Double click on the object you want to view.
- Click on the object you want to view and click the View Properties button in the tool bar.



- Click on the object you want to view and from the menu bar, select Edit , Properties of <object type>.
- Right click on the object you want to view and from the menu, select Properties of <object type>.

The system displays the appropriate dialog with the current properties of the selected object.

5. When you have finished viewing the properties, click the OK button to save the object and close the dialog.

More information:

[Administrator Concepts](#) (see page 223)

View Multiple Objects

The Policy Server User Interface allows you to select multiple objects from the List pane so that you can view the properties of multiple objects from one of the object dialogs. You can use the CTRL and SHIFT keys to select multiple objects in the List pane.

To view multiple Policy Server objects

1. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to chose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

2. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

3. Select multiple objects in the List pane by doing one of the following:
 - To select a range of objects, click on the first object, then click on the last object while holding down the SHIFT key.
 - To select objects that are not grouped together in the list pane, hold down the CTRL key and click on each of the objects in the List pane that you want to edit.
4. Open the dialog associated with the selected objects by doing one of the following:
 - Click the View Properties button in the tool bar.



- From the menu bar, select Edit , Properties of <object type>.

The properties dialog for the selected objects opens and displays the properties for the first object in the List pane that you selected. The bottom right corner of the dialog contains Previous and Next buttons. Use these buttons to scroll through the selected objects.

5. When you finish viewing object properties click OK to close the dialog.

More information:

[Administrator Concepts](#) (see page 223)

Edit an Existing Policy Server Object

The Policy Server User Interface allows you to edit the Policy Server objects that currently exist within the system.

To edit an existing Policy Server object

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to chose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

3. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

4. In the List pane, do one of the following:
 - Double click on the object you want to edit.
 - Click on the object you want to edit and click the View Properties button in the tool bar.



- Click on the object and from the menu bar, select Edit , Properties of <object type>.
- Right click on the object you want to edit and from the menu, select Properties of <object type>.

The system displays the appropriate dialog with the current properties of the selected object.

5. Modify the information in the dialog.

See the chapter associated with the object type for detailed information about the fields and controls contained in the dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

6. When you have finished editing a object, do one of the following:
 - Click the OK button to save the object and close the dialog.
 - Click the Apply button to save the object without closing the dialog.
 - Click the Cancel button to close the dialog without saving the object.

More information:

[Start the Policy Server User Interface](#) (see page 64)
[Administrator Concepts](#) (see page 223)

Edit Multiple Objects

The Policy Server User Interface allows you to select multiple objects from the list pane so that you can edit multiple objects from one of the object dialogs. You can use the CTRL and SHIFT keys to select multiple objects in the List pane.

To edit multiple Policy Server objects

1. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to choose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

2. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

3. Select multiple objects in the List pane by doing one of the following:

- To select a range of objects, click on the first object, then click on the last object while holding down the SHIFT key.
- To select objects that are not grouped together in the list pane, hold down the CTRL key and click on each of the objects in the List pane that you want to edit.

4. Open the dialog associated with the selected objects by doing one of the following:

- Click the View Properties button in the tool bar.



- Right click on the selected objects and from the popup menu, select Properties of <object type>.
- From the menu bar, select Edit , Properties of <object type>.

The properties dialog for the selected objects opens and displays the properties for the first object you selected. The bottom right corner of the dialog contains Previous and Next buttons. Use these buttons to scroll through the selected objects.

5. Modify the information in the dialog for each object.

See the chapter associated with the object type for detailed information about the fields and controls contained in the dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

6. When you finish editing an object, click the Apply button to save the object without closing the dialog.
7. When you finish editing the last object click the OK button to save the object and close the dialog.

More information:

[Administrator Concepts](#) (see page 223)

Delete a Policy Server Object

If a policy server object becomes unnecessary or unused, the policy server allows you to delete it.

To delete a policy server object

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to choose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

3. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

4. In the List pane, do one of the following:

- Click on the object you want to delete, then click the Delete button in the tool bar.



- Click on the object you want to delete, then from the menu bar, select Edit , Delete <object type>.
- Right click on the object you want to delete, then from the menu, select Delete <object type>.
- Click on the object you want to delete and press the DELETE key.

The system displays a Delete dialog and asks you to confirm the deletion of the object.

5. Ensure that the object you want to delete is listed in the Delete dialog and click Confirm.

The system deletes the selected object.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Administrator Concepts](#) (see page 223)

Delete Multiple Objects

The Policy Server User Interface allows you to select multiple objects from the List pane so that you can delete multiple objects.

To delete multiple Policy Server objects:

1. In the Object pane, click on the System Configuration or the Policy Domain Configuration tab.

The tab you select depends on whether the object must be configured by an administrator with the Manage All System and Domain Objects privilege or the Manage Domain Objects privilege.

Note: If you login as an administrator for specific policy domains, there are no tabs to choose from in the Object pane. Instead, the available objects are simply listed in the Object pane.

2. In the Object pane, click on an object type.

The List pane displays all existing objects of the selected type.

3. Select multiple objects in the List pane by doing one of the following:

- To select a range of objects, click on the first object, then click on the last object while holding down the SHIFT key.
- To select objects that are not grouped together in the list pane, hold down the CTRL key and click on each of the objects in the List pane that you want to edit.

4. Delete the selected objects by doing one of the following:

- Click the Delete button in the tool bar.



- From the menu bar, select Edit , Delete <object type>.
- Select the objects you want to delete and press the DELETE key.

The Policy Server User Interface displays a Delete dialog that lists the selected objects and asks you to confirm the deletion of the objects.

5. Ensure that the objects you want to delete are listed in the Delete dialog and click Confirm.

The system deletes the selected objects.

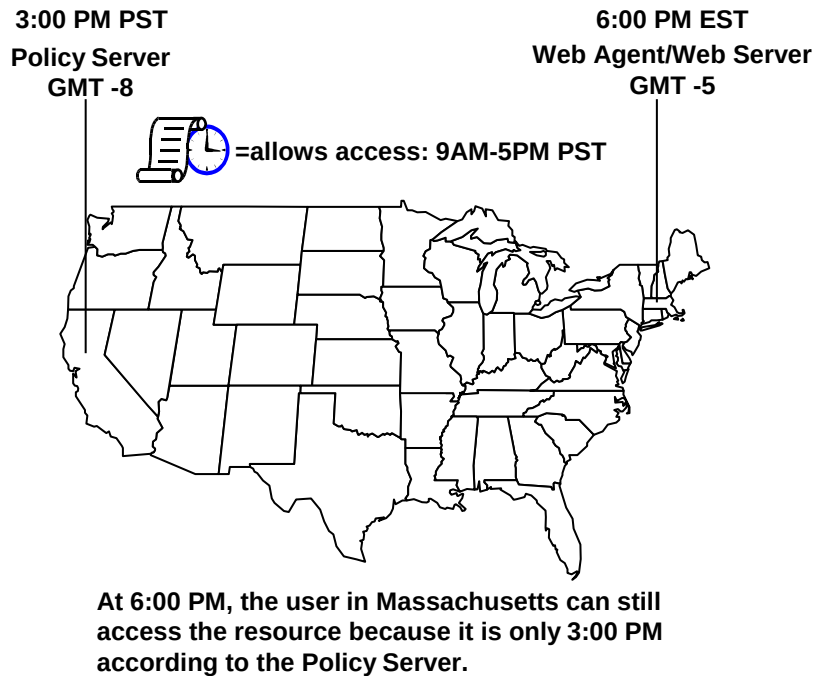
More information:

[Administrator Concepts](#) (see page 223)

How the Web Agent and Policy Server Calculate Time

For each system that has a Policy Server or Web Agent installed, you must set the system clock for the time zone appropriate to that system's geographical location. Policy Servers and Web Agents use the time zones to calculate time relative to Greenwich Mean Time (GMT).

The following figure shows how the Policy Server executes a policy relative to time. A resource is stored on a Web Server in Massachusetts and is protected by a Policy Server in California. The policy allows access to the resource between 9:00 a.m. and 5:00 p.m. However, the user in Massachusetts can still access the resource at 6:00 p.m. because the policy is based on the Policy Server's time zone, Pacific Standard Time (PST), which is 3 hours behind the Web Agent's time zone, Eastern Standard Time (EST).



Note: For Windows systems, the time zone and the time of day that you set in the Date/Time control panel must agree. For example, to reset a system in the USA from Eastern Standard Time to Pacific Time, you must set the system's clock back three hours and change the time zone to Pacific Standard Time. If these two settings do not match, single sign-on across multiple domains and agent key management will not work properly.

Chapter 5: User Sessions

This section contains the following topics:

[User Session Overview](#) (see page 85)

[How a User Session Begins](#) (see page 89)

[How Sessions Across Realms Are Maintained](#) (see page 90)

[How Sessions Across Multiple Cookie Domains Are Maintained](#) (see page 90)

[How a User Session Is Validated](#) (see page 91)

[How Session Information Is Delegated](#) (see page 92)

[Session Timeouts](#) (see page 92)

[How Agent Key Management and Session Timeouts are Coordinated](#) (see page 93)

[How a User Session Ends](#) (see page 94)

[How a Windows User Security Context Is Obtained](#) (see page 94)

User Session Overview

SiteMinder maintains and administers consistent user session across multi-tiered applications. Maintaining user session information is critical as applications and Internet businesses become location-independent and the environments and applications technology and security needs vary.

Non-Persistent and Persistent Cookies

Standard SiteMinder session cookies are non-persistent. A non-persistent cookie is one that is maintained only in the memory of the web browser. When users close their browser, the session cookie is destroyed, effectively logging them out.

In addition to maintaining the cookie in the web browser memory, you can configure SiteMinder to have the cookie written to the hard disk. Maintaining a SiteMinder cookie in the web browser and on hard disk is known as a persistent cookie. When using persistent cookies, users that close and reopen their browser remain logged in.

Note: For more information about configuring a persistent cookie, see the *Web Agent Configuration Guide*.

Non-Persistent and Persistent Sessions

SiteMinder also provides the ability to configure a persistent session to provide Windows security context functionality and support for Federated Web Services. A persistent session is one in which a SiteMinder cookie is maintained in a SiteMinder session store, in the memory of the web browser, and optionally the hard disk.

Before you implement persistent sessions, consider the following:

- Persistent sessions are configured on a per realm basis.
- Persistent sessions should only be used where necessary. Using session services to maintain sessions has an impact on system performance.

Note: For more information about enabling and configuring session services, see the Policy Server Administration Guide.

Session Tickets

SiteMinder implements session management using session tickets. A session ticket contains basic information about a user and that user's authentication information; it is used to identify the user's session across all sites in a single sign-on SiteMinder environment. Session tickets are encrypted and can only be read/validated by the Policy Server. SiteMinder Web Agents use session tickets to identify users and provide session information to the Policy Server.

The session ticket is handled differently depending upon whether the session is persistent or non-persistent.

Note: Non-persistent and persistent cookies are unrelated to the user's SiteMinder session being non-persistent or persistent.

Non-persistent session

The Web Agent places the session ticket in a cookie. The cookie contains the user session data; no user-specific data is kept in the cookie itself. The Web Agent is responsible for validating the cookie and enforcing session timeouts.

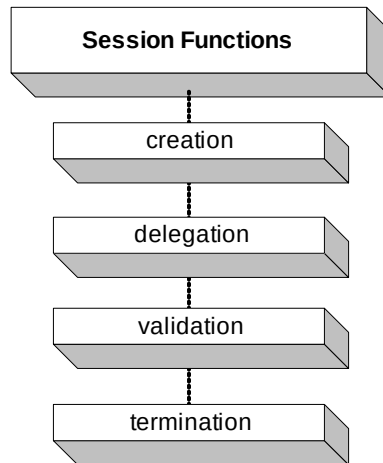
Persistent Session

The Web Agent places the session ticket in a session server database and, if possible, in an optional cookie on the client.

The session ticket data is used as an index into the Web Agent's cache, which contains the user session data. If a cookie is written, no user-specific data is kept in the cookie itself. The Web Agent is responsible for validating the session and enforcing the session timeouts.

How SiteMinder Manages User Sessions

For the most part, SiteMinder manages user sessions automatically, performing a number of session management functions during the life cycle of a user session, as illustrated below.



Session creation

Establishing a session when a user successfully logs into an application. If a user fails to authenticate, no session is established.

Session delegation

Passing session information across an application environment. Delegating session information is necessary when an application's logic crosses several application tiers.

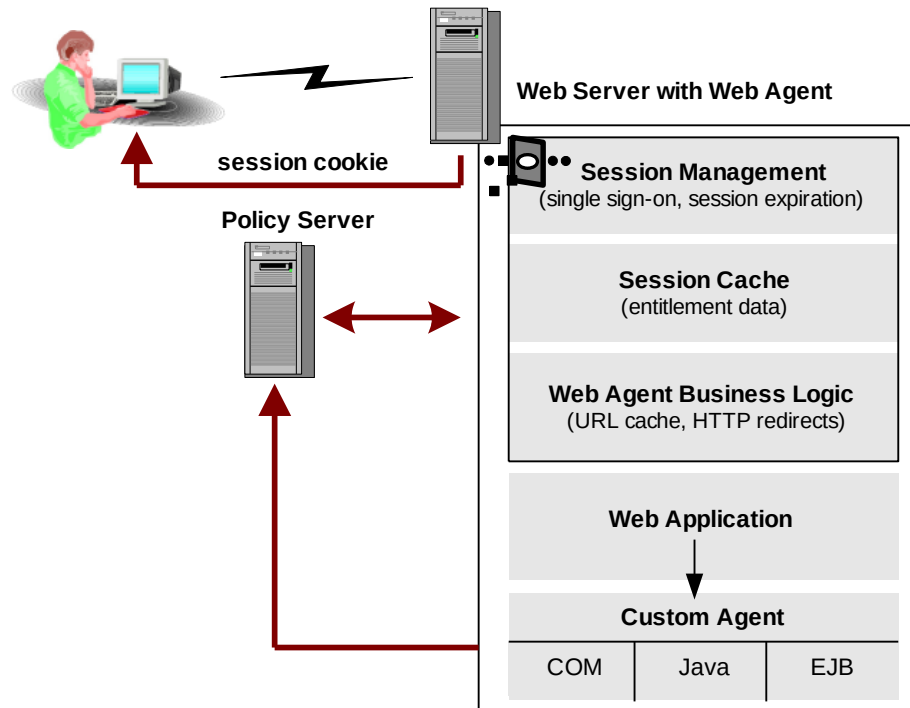
Session validation

Verifying the session ticket to make sure the user session is still active, that is, it has not expired or been terminated.

Session termination

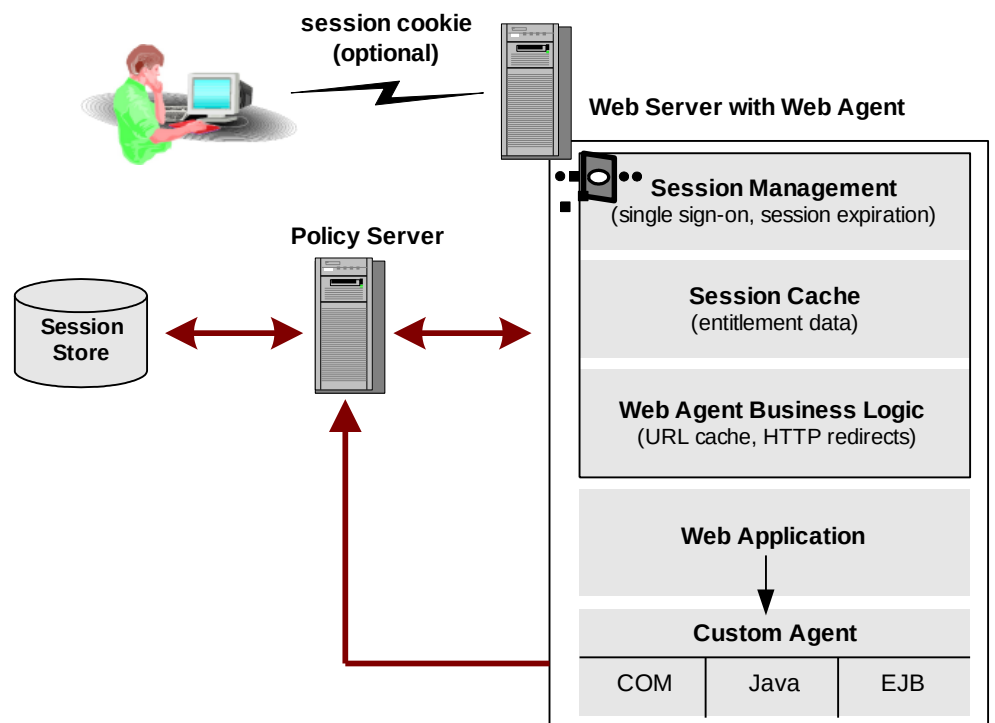
Ending a user session when a user logs out, when the configured session timeouts expire, or when a user is manually disabled by the SiteMinder System Manager. When a user logs out or the user session expires, they must log in again to create a new session. In the case of manual user disablement, the user can not re-initiate a session.

The following diagram illustrates how SiteMinder manages a non-persistent session.



The Web Agent passes the session ticket information across each application tier

The following diagram illustrates how SiteMinder manages a persistent session.



The Web Agent passes the session ticket information across each application tier

How a User Session Begins

A user session begins when a user logs in and is authenticated by SiteMinder. The Policy Server issues the session ticket, which is then required for all subsequent Agent requests for that user's session. If a user is forced to authenticate again during a session because of a higher protection level, the existing session is maintained.

A session is active until it is terminated when the user logs off, when the session expires, or when an administrator disables a user, thereby terminating the session.

More information:

[Protection Levels](#) (see page 237)

How Sessions Across Realms Are Maintained

When a user requests access to a resource, his or her session is created within the context of the realm that contains that resource. An authentication scheme is also associated with a realm, and it determines the type of credentials that the user must present to gain access to the resource.

This authentication context is made available to all Web servers in the SiteMinder installation through SiteMinder's default HTTP headers that define components, such as the authentication scheme being used, the namespace the user is authenticating against, and other relevant information. In addition to the default headers, you can configure response attributes in a policy to communicate information for a user, such as a social security number or a phone number, that helps to further identify a user.

If the SiteMinder installation is configured for single sign-on, the authentication scheme may have a protection level assigned to it by an administrator. The level can be a number from 1 through 1000, with 1 being the least secure and 1000 being the most secure. These protection levels enable administrators to implement single sign-on with a higher level of security and flexibility.

An authenticated user of one realm can be validated for a session in another realm if the second realm is protected by an authentication scheme of an equal or lower protection level as the first. As long as the protection level is the same or lower, that user does not need to re-authenticate, which means that the user session remains valid. If a user tries to access a resource protected by an authentication scheme with a higher protection level, SiteMinder prompts the user to re-enter his or her credentials, thereby ending one user session and creating a new session.

To configure protection levels for your single sign-on environment, see [Authentication Schemes](#) (see page 231).

How Sessions Across Multiple Cookie Domains Are Maintained

SiteMinder supports single sign-on across multiple cookie domains in environments with heterogeneous Web server platforms. If a user visits *companyA.com* and then goes to *companyB.com*, his or her session information stays with them. To maintain session information across multiple cookie domains, the Web Agents must be configured for single sign-on. With single sign-on configured, the cookie that contains session information can be made available to all Agents and servers in the single sign-on environment.

The ability to pass session and identification information across multiple cookie domains enables a user to authenticate at a site in one cookie domain and then navigate to a site in another cookie domain without being re-challenged for information.

Single sign-on is accomplished using a cookie provider. The cookie provider is an extension of the Web Agents in the single sign-on environment.

To achieve cross-domain logout for resources in separate cookie domains, you can enable persistent sessions for the realms in separate cookie domains. When a user logs out in one domain, the Policy Server sends a logout event terminating the user session.

Note: Single sign-on across multiple cookie domains does not require that the same user directory be used across the single sign-on environment. However, user directory connections configured in the Policy Server User Interface must share the same directory object name in each cookie domain.

More information:

[User Session Overview](#) (see page 85)

[How a User Session Begins](#) (see page 89)

[How Sessions Across Realms Are Maintained](#) (see page 90)

[How a User Session Is Validated](#) (see page 91)

[How Session Information Is Delegated](#) (see page 92)

[Session Timeouts](#) (see page 92)

[How Agent Key Management and Session Timeouts are Coordinated](#) (see page 93)

[How a User Session Ends](#) (see page 94)

[How a Windows User Security Context Is Obtained](#) (see page 94)

How a User Session Is Validated

When a user requests access to a resource, the Web Agent validates the session by checking whether or not it is still active. The Web Agent first checks its session cache for session information. If the Web Agent reads the cookie and has the information in cache, it can validate the session. If it does not, then it contacts the Policy Server to verify the user's identity and collect any other session and authorization information.

When a user accesses a resource that has a higher protection level than the one used when establishing the session, the session information is maintained, even though another authentication takes place.

More information:

[Protection Levels](#) (see page 237)

How Session Information Is Delegated

User sessions may be delegated between application tiers in a SiteMinder installation using the session ticket. The session ticket is the mechanism by which the user's identity is passed from one application tier to another. After getting the session ticket, each application can make authorization calls to the Policy Server themselves.

If your SiteMinder installation uses custom Agents, the custom Agent must have access to the information in the session ticket to maintain session information.

In addition to using the session ticket for delegation, the Web Agent makes a set of default HTTP headers available for session management that can be passed across different business application tiers such as Enterprise Java Beans (EJB) and Component Object Model (COM) based tiers. Included in these headers is a unique session ID and optionally, a universal ID. The session ID identifies an active user session.

The universal ID identifies the user to an application in a SiteMinder environment. This ID is typically not the same as the user's login ID, but is some other type of unique identifier like a customer account number. The universal ID helps facilitate identification between old and new applications. It delivers the user's identification automatically, regardless of the application. In addition, the ID is built into applications so that the applications have a user identification method that is separate from the user directory, which undergoes constant changes.

Both the session ID and the universal ID are shared among all the applications in a SiteMinder environment to maintain consistent user sessions.

Session Timeouts

Each user session includes session timeout information. The timeout values let you determine the length of an active session and the amount of session inactivity that can pass before a session is invalid. You configure session timeouts on a per-realm basis using the following timeout options.

Name	Purpose
Maximum Timeout (All sessions)	Specifies the maximum amount of time a user session can be active before the Web Agent challenges the user to re-authenticate. You can override this setting using the WebAgent-OnAuthAccept-Session-MaxTimeout response attribute.

Name	Purpose
Idle Timeout (All sessions)	Specifies the amount of time that a user session can be idle before the Web Agent terminates the session. If the session expires, a user must re-authenticate. Note: For persistent sessions, this value must be greater than that specified by Session Validation Period. You can override this setting using the WebAgent-OnAuthAccept Session-Idle-Timeout response attribute.
Session Validation Period (Persistent Sessions)	For persistent sessions only, specifies the maximum period between agent calls to the Policy Server to validate a session. Session validation calls perform two functions: informing the Policy Server that a user is still active <i>and</i> checking that the user's session is still valid.

More information:

[Grouping Resources in Realms](#) (see page 315)

How Agent Key Management and Session Timeouts are Coordinated

SiteMinder Web Agents use a key to encrypt and decrypt any cookies that pass between Web Agents in a SiteMinder environment. All keys must be set to the same value for all Web Agents communicating with a Policy Server.

A SiteMinder installation can be configured to use dynamic Agent keys that change on a periodic basis. Dynamic key rollover lets you update dynamic keys at regular intervals to ensure the security of encrypted cookies. You specify when key rollovers occur in the Set Rollover Frequency dialog box of the Policy Server User Interface. Key updates across a SiteMinder installation can take up to three minutes.

You must coordinate the updating of keys together with session timeouts or you may invalidate cookies that contain session information. This coordination is critical because the person designing policies in your organization may be different than the person configuring dynamic key rollover.

Session timeouts must be less than or equal to two times the interval configured between Agent key rollovers. If an administrator configures an agent key rollover to occur two times before a session expires, cookies written by the Web Agent before the first key rollover will no longer be valid. If a session timeout is greater than the specified rollover interval, a user may be re-challenged for their identification before their session terminates.

For example, if you configure key rollover to occur every three hours, you might want to set the Maximum Session timeout for 6 hours to ensure that multiple key rollovers do not invalidate the session cookie.

How a User Session Ends

A user session can end in one of three ways:

A user logs out

A user can terminate his or her own session by logging out.

Session timeouts expire

A user's session can expire because of configured session timeouts or policy-based response attributes, requiring the user to re-enter his or her credentials to begin a new session.

A user is disabled

A System Manager can disable a user account, flushing the session account and preventing that user from re-authenticating. For more information, see the *Policy Server Management* guide.

How a Windows User Security Context Is Obtained

In a Windows network, a security context defines a user's identity and authentication information. Web applications such as Microsoft Exchange Server or SQL Server need a user's security context to provide native security in the form of Microsoft's access control lists (ACLs) or other access control tools.

Note: In a Windows security context, the user store must be Active Directory (AD).

The SiteMinder Web Agent can provide a Windows user security context for accessing Web resources on IIS Web servers (Windows 2000 platforms). By establishing a user's security context, the server can use this identity to enforce access control mechanisms.

By providing a Windows user security context, SiteMinder offers the following benefits:

- Two levels of security

You can protect Web resources using SiteMinder's many capabilities with the additional benefit of working with Microsoft's own security tools to protect applications.

- The Agent can collect user credentials over SSL connections, then allow subsequent requests over non-SSL connections. SiteMinder combines this mechanism for credential collection with the Windows security context when permitting access to Web resources.

- SiteMinder can implement single sign-on using forms-based authentication, and pass on a user's identity to back-office applications.

Microsoft does not provide general-purpose forms credential collection.

How Persistent Sessions for User Security Contexts Are Maintained

For the Web Agent to provide a security context for specific resources, you need to enable persistent sessions for all realms that include those resources. Persistent sessions are maintained by the session server.

For conceptual information about persistent sessions, see [Persistent and Non-persistent Sessions](#). For instructions on enabling persistent sessions for a realms protected by Web Agents, see [Grouping Resources in Realms](#) (see page 315).

The session server, which resides on the same system as the Policy Server, stores a user's encrypted credentials and associates the user with a session ID. When a SiteMinder session is established between a client and a Web Agent, the Windows user account is established and linked to the session. If the Web Agent's user session cache becomes full and entries are purged, the Agent can retrieve the user's credentials from the session server and re-establish the session. The session server also stores the security context because this context must be propagated across a single sign-on environment.

How Sessions Are Revalidated

You can explicitly configure the Web Agent, working through the Policy Server, to contact the session server to revalidate a session. A session cookie stored in the user's browser contains the session ID, which the cookie uses to reacquire the user's credentials from the session server.

Note: If the session cookie becomes invalid, the credentials associated with the ID also become invalid and the user must reauthenticate.

The frequency at which the Agent revalidates a session is determined by a configurable value called the *validation period*. The validation period defines how long the Agent can keep a session active using the information in its cache before contacting the session server for updates.

If the validation period is too small, the Agent goes back to the session server frequently, slowing down SiteMinder performance when processing requests. Therefore, you want to set the validation value to a high number. If the number of active sessions is lower than the Agent's maximum user session cache value, the Agent uses the cached information instead of contacting the session server.

Note: If the session server is not operating, the validation period is infinite and the Agent will not contact the session server. For instructions on configuring the validation period, see [Grouping Resources in Realms](#) (see page 315).

How SiteMinder Is Configured to Provide a Windows User Security Context

Your IIS Web server environment must meet the following requirements to use the Windows security context feature:

- All IIS servers have to be trusted in the domain in which the user is authenticating. You can establish trust relationships among servers that provide distributed services for groups of users. To set up trust relationships, see Microsoft server documentation.
- Users must have the privileges to log on locally to the Web server. If there are multiple servers, users need the right to log on locally to all servers.

To enable a user to log on locally, configure this feature through the IIS Administrative Tools. See Microsoft documentation for instructions.
- If you are using a specific domain account and not the system account when starting the World Wide Web Publishing Service, you must set the user's account privileges to act as part of the operating system for the domain account running the service. This feature is configured through the Administrative Tools. See Microsoft documentation for detailed instructions.

Configuration Overview

Step	SiteMinder component	Supporting documentation
Designate a relational database as the session store.	Data tab of SiteMinder Policy Server Management Console	<i>Policy Server Management</i> guide

Step	SiteMinder component	Supporting documentation
Enable the user directory containing the Windows users to run the security context.	Run the Authenticated User's Security Context check box on the Credentials and Connection tab of the SiteMinder User Directory dialog	User Directories (see page 141) in the <i>Policy Design</i> guide
Associate one of the Supported Authentication Schemes (see page 97) with each realm	Resources tab of the Realms Properties dialog	Grouping Resources in Realms (see page 315) in the <i>Policy Design</i> guide
Enable persistent sessions for each realm and set a high Validation Enabled Period	Session tab of the Realms Properties dialog	Grouping Resources in Realms (see page 315) in the <i>Policy Design</i> guide
If your IIS Web server is not in a physically secure location, set insecureserver to YES	Agent Configuration Object or WebAgent.conf	<i>Web Agent Guide</i>

Supported Authentication Schemes

The following authentication schemes are supported:

- Basic Authentication Schemes
- Basic Over SSL Authentication Schemes
- HTML Forms Authentication Schemes
- X.509 Client Certificate and Basic Authentication Schemes
- X.509 Client Certificate and HTML Forms Authentication Schemes

The Web Agent cannot use certificates alone because the user's credentials are not provided. Certificate authentication must be combined with basic or forms to collect the user credentials.

You may want to use the NTLM authentication scheme to provide access to resources in a particular realm, such as those on an intranet. CA does not support selecting the NTLM authentication scheme as part of the Windows User Security Context configuration described in this section. However, you can use NTLM for some resources on a Web server and the Windows User Security Context mechanism (with a supported authentication scheme) for different resources on the same Web server. In that case, the resources accessed using NTLM need to be in a different realm than the resources accessed through the Windows User Security Context mechanism described in this section.

More information:

[Authentication Schemes](#) (see page 231)

Active Directory and NetBIOS Names

Although Active Directory domains are named according to DNS naming standards, a NetBIOS name must still be defined when creating Active Directory domains.

For SiteMinder to support seamless Microsoft security context, NetBIOS names should be named the same as the DNS domain name as recommended by Microsoft.

When the Active Directory domain has a DNS name that is different from its NetBIOS name, the user domain cannot be established for the web server to provide the security context when SiteMinder is configured to use an LDAP user directory.

Single Sign-on in Security Context

SiteMinder will provide single-sign-on across all supported web servers while preserving the security context required for seamless Microsoft security context.

However, this requires that any realm that may cause the user to be authenticated (and establish the SiteMinder session) be configured as “Persistent”. If the user authenticates in a realm that is not persistent, the user will be challenged again when SiteMinder needs to persist the security context.

Chapter 6: Agents and Agent Groups

This section contains the following topics:

[Agents Overview](#) (see page 99)

[The SiteMinder Web Agent Model](#) (see page 104)

[Trusted Hosts](#) (see page 107)

[How a Host Configuration Object is Configured for a Trusted Host](#) (see page 111)

[Configure a Web Agent](#) (see page 120)

[Configure an Agent Configuration Object for a Web Agent](#) (see page 126)

[Enable a Web Agent](#) (see page 133)

[Configure Agent Identities for Non-Web Agents](#) (see page 133)

[Configure a RADIUS Agent](#) (see page 134)

[Create Custom Agents](#) (see page 136)

[Resource Protection with a SiteMinder Agent](#) (see page 138)

[Agent Group Usage](#) (see page 138)

Agents Overview

An Agent in a SiteMinder environment is a network entity that acts as a filter to enforce network access control or Web access control. An Agent monitors requests for resources. When a user requests a protected resource, the Agent prompts the user for credentials based on an authentication scheme, and sends the credentials to a Policy Server.

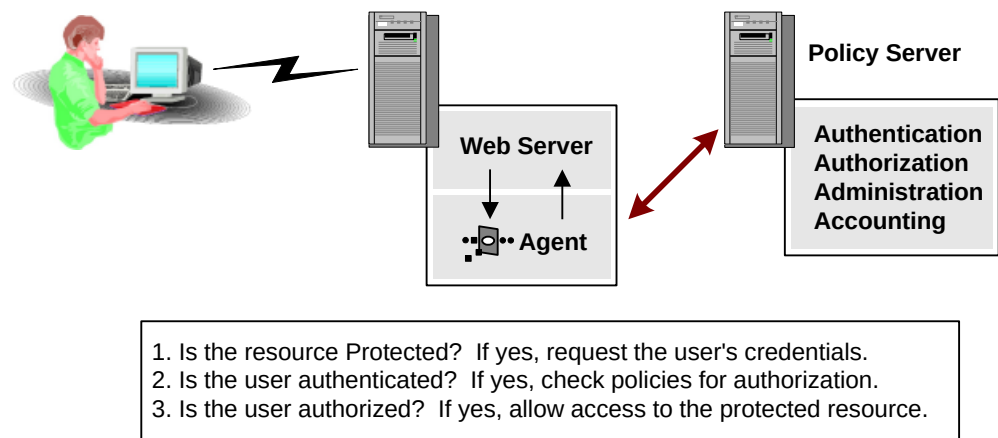
The Policy Server determines whether or not a user can be authenticated based on the credentials, and whether or not the user is authorized for the requested resource. The Policy Server then communicates with the Agent, which allows or denies access to the requested resource.

Web Agents, Affiliate Agents, EJB Agents, Servlet Agents, and RADIUS Agents, are available by default. All other Agents are considered Custom Agents that must be created using the Agent APIs. Once created, you can configure Custom Agents in the Policy Server User Interface.

Web Agents

Web Agents are SiteMinder Agents that operate with Web servers. When a user requests a page from the Web server, the Web Agent communicates with the Policy Server and processes authentication and authorization requests before the user can access the resource from a Web browser. In addition, the Policy Server can provide information that the Web Agent uses to provide personalized content based on a user's identity.

The following diagram illustrates the three most basic transactions that a Web Agent and Policy Server handle in order to provide access to a protected resource. These transactions can contain more detailed information to enable customized content and support other SiteMinder features, but the process is similar whenever a user attempts to access a resource through a Web server managed by a Web Agent.



The previous figure assumes that a user requests a protected resource for which the user is authorized. The Web Agent checks with the Policy Server to determine if the resource is protected, and the Policy Server indicates that it is protected. The Web Agent gathers credentials from the user and communicates them to the Policy Server.

The Policy Server authenticates the user and informs the Web Agent that the user has been properly identified. Finally, the Web Agent checks with the Policy Server to determine if the user is authorized for the resource. The Policy Server verifies that the user is authorized for the resource, communicates this to the Web agent, and the Web Agent allows the Web server to display the protected resource requested by the user.

Agents that control the same resources and are of the same Agent type (all Web Agents, or all RADIUS Agents) can be grouped.

Note: If you plan to configure support for virtual Web servers, see the *SiteMinder Web Agent Guide*.

More information:

[Agent Group Usage](#) (see page 138)

SAML Affiliate Agents

The SAML Affiliate Agent is part of the Federation Security Services solution. Federation Security Services enables business to share security information across multiple domains.

Note: Federation Security Services and the SAML Affiliate Agent are packaged as a separately-licensed items.

The SAML Affiliate Agent provides seamless single sign-on from a producer site, such as a portal, to a SAML consumer acting as an affiliate in a federated network. The affiliate provides resources and services related to the portal. For example, affiliateA.com and affiliateB.com have an agreement that visitors to affiliateA.com receive a 10% discount for purchases at affiliateB.com. These two sites are affiliates. Using the SAML Affiliate Agent eliminates the need for the a user to re-authenticate or provide additional information about themselves.

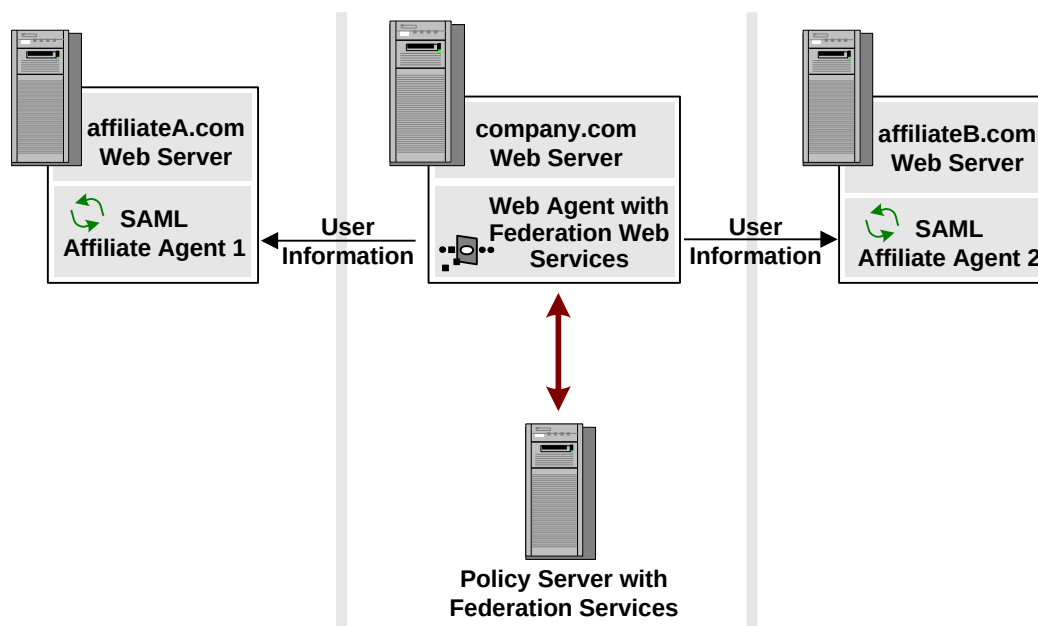
The communication exchange between a SAML Affiliate Agent and SiteMinder at the producer site results in the generation of a SAML assertion. A SAML assertion is an XML document containing authorization and authentication about a user. The producer sends this document to the SAML Affiliate Agent at the consumer site. The assertion confirms that a user has been authenticated at the main portal, and the information it contains enables the affiliate to provide user information to a Web server for use with its Web applications.

If you install a SAML Affiliate Agent at an affiliate site, it is the only SiteMinder component installed at that site. The affiliate site does not require a full installation, because a SAML Affiliate Agent does not protect resources; it only enables personalization.

The following components are required to support Federation Security Services at the producer site:

- Policy Server (provides assertion generator component)
- Web Agent
- Web Agent Option Pack-provides the Federation Web Services application

The following figure shows a federated network with SAML Affiliate Agents.



In the previous figure, the Policy Server is connected to a Web Agent on the company.com Web server. This Agent can pass user information about company.com users to the affiliateA.com and affiliateB.com Web servers using the SAML Affiliate Agents.

The producer site authenticates the user and passes information about the user to the affiliate site. This site has a full SiteMinder installation. The affiliate site has only a SAML Affiliate Agent and Web server; there is no Policy Server. The affiliate site does not require a full installation because a SAML Affiliate Agent does not protect resources in the same way as a Web Agent. The SAML Affiliate Agent simply provides user information to a Web server for use with its Web applications. The applications can use the information to personalize Web content and create a unique experience for each user.

SAML Affiliate Agent and Federation Security Services Configuration

Configuration of the SAML Affiliate Agent happens at the consumer site—you do not configure a SAML Affiliate Agent at the producer-site Policy Server. However, in the Policy Server User Interface, you set up Federation Security Services components, create affiliate domains, and configure affiliates for the affiliate domains.

Note: In the Policy Server User Interface, there is an option to create an Agent and choose Affiliate Agent as the Agent type. This option is not for the SAML Affiliate Agent—it is only applicable for 4.x Affiliate Agents. Do not select it.

For more information, see the *Federation Security Services Guide*.

RADIUS Agents

Remote Authentication Dial-In User Service (RADIUS) is a protocol that enables you to exchange session authentication and configuration information between a Network Access Server (NAS) device and a RADIUS authentication server. The RADIUS protocol is often used by NAS devices that serve as proxy services, firewalls, or dial-up security devices.

The Policy Server can be used as a RADIUS authentication server. RADIUS Agents allow the Policy Server to communicate with the NAS client devices.

Note: For complete information about configuring SiteMinder as a RADIUS authentication server, see the *SiteMinder Policy Server Management Guide*.

Application Server Agents

The Application Server Agent is a collection of Java components that provide a full-featured SiteMinder Agent for securing WebLogic and WebSphere application server resources. The Application Server Agent integrates SiteMinder with the J2EE platform.

The Application Server Agent can protect the following components:

- Web Applications (including servlets, HTML pages, JSP, image files)
- JNDI lookups
- EJB components
- JMS connection factories, topics, and queues
- JDBC connection pools

The Application Server Agent is a single Agent, but from the perspective of the SiteMinder Policy Server, there are different Agent types that protect application server resources. The Agent types give the Application Server Agent the flexibility to protect servlets, and EJB components in two ways: using the Servlet, or EJB Agent respectively, or using the Web Agent.

For complete information about how configure SiteMinder to work with application server Agents, see the *SiteMinder Application Server Agent Guide* that applies to your platform.

TransactionMinder XML Agents

The TransactionMinder XML Agent is an XML-enabled version of the SiteMinder Web Agent. It integrates with a Web server to authenticate and authorize requests for access to Web services bound to URLs served by that Web server. You can also use the core Web Agent functionality within XML Agents to protect other resources on a Web server.

For more information about XML Agents, see the *TransactionMinder Operations Guide*.

The SiteMinder Web Agent Model

The SiteMinder Web Agent model gives you the option of configuring and storing a set of Agent configuration parameters in the Policy Server's policy store to be used by a group of Web Agents. This is referred to as central agent configuration. Parameters are configured using the Policy Server User Interface.

Note: The Agent model described in this section does not apply to RADIUS, EJB, Servlet, or Custom Agents—those Agents can only perform local configuration.

Web Agent Configuration Options

There are two options for configuring a Web Agent:

Central configuration

The Web Agent is configured from the Policy Server. The policy store holds the set of parameters for a group of Web Agents.

The Agent configuration is specified in an Agent Configuration Object.

Local configuration

The Web Agent is configured from a local configuration file on each web server where the Agent is installed.

You can store some parameters centrally and others locally.

Note: You can only enable and disable the Web Agent from the local Agent configuration file, not from the Policy Server. This is true whether you are configuring centrally or locally.

More information:

[Agent Components](#) (see page 106)

[Combined Central and Local Configuration](#) (see page 122)

Advantages of Centrally Configuring Web Agents

When you centrally configure Web Agents, the settings are stored in the policy store, instead of locally, in a configuration file, on a Web Server.

Compared with local configuration, central configuration provides:

Improved Usability When Using Central Agent Configuration

- Updating the configuration settings of multiple Web Agents is easier and faster. You can change the setting in one place and have it propagate to multiple Web Agents.
- Installing the Web Agent no longer requires the administrator to specify a shared secret. The Policy Server generates the shared secret.

Added Security with Central Agent Configuration

- It is easier to control access to the Web Agent configuration. For example, you can prevent the Web Server administrator from changing the configuration settings.
- You can store the configuration settings behind an internal firewall, instead of on the Web Server.
- You can use a hardware-bound key to generate the shared secret, which is used by the client side to establish a secure connection to the Policy Server. The trusted host handles the connection to the Policy Server.

Agent Components

On the Agent-side of a SiteMinder network, there are several main components involved in Web Agent operation:

SiteMinder Web Agent

Virtual interface to a Web Server; triggers rules and enforces policies

Trusted Host

A client computer where one or more Web Agents is installed. It handles the connection to the Policy Server. You can have more than one trusted host on a physical server, but each must be identified by a unique name.

The trusted host is “trusted,” because it is registered with the Policy Server. You must register a trusted host so the Web Agents installed on that host can communicate with the Policy Server.

A trusted host is identified by the following data:

- Host name
- Host IP address
- Host description
- Shared secret
- Host Object Identifier (OID)

Web Agent Configuration File (WebAgent.conf or LocalConfig.conf)

Stored on the web server where the Agent resides, this file is used for local configuration. It holds the Agent configuration parameters for each Web Agent. All Web Agents use the WebAgent.conf file; however, Framework Web Agents use the WebAgent.conf file only for core settings needed for the Agent to start and connect to a Policy Server.

Host Configuration File (SmHost.conf)

Stored on the web Server where the Web Agent resides, this file holds initialization parameters for the trusted host. Once the trusted host connects to a Policy Server, the trusted host uses the settings in the Host Configuration Object stored at the Policy Server. The Host Configuration Object is named in the hostconfigobject parameter of this file.

More information:

[Web Agent Configuration Options](#) (see page 104)

Policy Server Components

On the Policy Server-side there are three policy objects related to Web Agent configuration:

Agent object

Names the Agent, establishing an Agent identity that can be mapped to a specific web server.

Agent Configuration Object

Contains the Web Agent configuration parameters. Use an Agent Configuration Object to centrally manage a group of Web Agents. Though this object is primarily for central Agent configuration, it also contains the parameter that tells the Policy Server to use local configuration. This object applies only to Web Agent.

Host Configuration Object

Contains the trusted host configuration parameters. Except for initialization parameters, trusted host parameters are always maintained in a Host Configuration Object.

Configuration objects are stored in the policy store. Use the Policy Server User Interface to create, modify, and view configuration objects.

More information:

[Web Agent Configuration Options](#) (see page 104)

Trusted Hosts

A *trusted host* is a client computer where one or more SiteMinder Web Agents can be installed. The term trusted host refers to the physical system.

Register a Trusted Host with the Policy Server

You register a trusted host from the system where you install a Web Agent; the host registration process is part of the Agent installation and configuration process. After registration is complete, the registration tool creates the SmHost.conf file. After this file is created successfully, the client computer becomes a trusted host. A trusted host must be registered to communicate with the Policy Server.

You cannot create a trusted host using the Policy Server User Interface; you can only view a trusted host once it is registered or delete a trusted host.

Note: You only register the host once, *not* each time you install and configure a Web Agent on a system.

Trusted Host Configuration Settings

Upon initialization, the Web Agent uses the trusted host's configuration settings in the Host Configuration File (SmHost.conf). The Web Agent attempts to connect to the first Policy Server listed under the PolicyServer parameter of the Host Configuration File. If the trusted host fails to connect to the first Policy Server, the trusted host attempts to connect to the next Policy Server listed, if any.

Once the Web Agent connects to its bootstrap Policy Server, the trusted host looks for the Host Configuration Object, named in the hostconfigobject parameter of the Smhost.conf file. You specify this value when you register the trusted host. The trusted host then retrieves its configuration.

Trusted Host Configuration Parameters

With the exception of the PolicyServer and RequestTimeout parameters, all the parameters discussed here can be set only in a Host Configuration Object, not in the Host Configuration File, SmHost.conf. Settings in the Host Configuration File apply only when the trusted host initializes. Once the trusted host initializes, the parameter settings in the Host Configuration Object take effect.

Request Timeout

Use the RequestTimeout parameter to specify the number of seconds that the trusted host should wait before deciding that a Policy Server is unavailable. This setting allows you to optimize the response time of the Web server.

The default value is 60 seconds.

Note: If the Policy Server is busy due to heavy traffic or a slow network connection, you may want to increase the RequestTimeout value.

Operation Mode

The operation mode determines how the trusted host works with multiple Policy Servers. There are two operation modes: failover and round robin.

Failover

Failover is a redundancy mode. If the primary Policy Server fails, there is a backup Policy Server to take over policy operations. Failover is the default operation mode. When the trusted host initializes, it operates in Failover mode.

In this mode, *every* trusted host request is delivered to the first Policy Server in the list. If that Policy Server does not respond, the trusted host marks it *unavailable* and redirects the request to the next Policy Server in the list. If a previously failed Policy Server recovers, it is returned to its original place in the list.

Round Robin

Round robin mode lets the trusted host distribute requests across multiple Policy Servers, which provides faster access to Policy Servers and therefore, more efficient user authentication and authorization. It also prevents a single Policy Server from becoming overloaded with requests.

In this mode, the trusted host delivers a request to the first Policy Server in the list. The next request is delivered to the second Policy Server in the list, and so on, until the trusted host has sent requests to all the available Policy Servers. After sending requests to all of the Policy Servers, the next request returns to the first Policy Server in the list and the cycle begins again.

If a Policy Server fails, the request is redirected to the next Server in the list. The trusted host marks the failed Server as *unavailable* and redirects all of the requests to other servers. After the failed server recovers, it is automatically restored to its original place in the list.

Implement an Operation Mode

The operation mode determines how the trusted host works with multiple Policy Servers. There are two operation modes: failover and round robin.

To implement an operation mode

1. Configure more than one Policy Server.
2. Configure all Policy Servers to use a common policy store.
3. Set the EnableFailover parameter.
 - To enable failover, set the EnableFailover parameter to yes.
 - To enable load balancing, set the EnableFailover parameter to NO.

The value for the EnableFailover parameter applies to all Policy Servers specified in the Host Configuration Object.

TCP/IP Connections

The trusted host and Policy Server communicate across TCP/IP connections. The number of available TCP/IP connections between the trusted host and Policy Server is determined by the available sockets for the Policy Server's authorization, authentication, and accounting ports.

The number of sockets per port controls the number of simultaneous threads accessing the Policy Server from the Web server. Each user access request is handled by a separate Web server thread, which requires its own socket. The Web server maintains a pool of threads for requests and only creates a new one when there are no more available threads. As traffic increases, the number of sockets per port needs to increase.

There are several settings that affect the TCP/IP connections between the trusted host and the Policy Server.

MaxSocketsPerPort

Defines the maximum number of TCP/IP connections used by the trusted host to communicate with the Policy Server. By default, this value is set to 20, which is generally sufficient for low- and medium-traffic Web sites. If you are managing a high-traffic Web site or if you have defined agent identities for virtual servers, you may want to increase this number.

MinSocketsPerPort

Determines the number of TCP/IP connections open for the Policy Server at start up. The default value is 2. If you are managing a high-traffic Web site, you may want to increase this number.

NewSocketStep

Specifies the number of TCP/IP connections that the Agent opens when new connections are required. The default value is 2. Modify the number of sockets that should be added at each required increment if you require more sockets.

Note: For more information about these values and how you might need to adjust them as your SiteMinder environment grows, see the information about scaling your SiteMinder environment in the *SiteMinder Policy Server Management Guide*.

View Trusted Host Objects

You register a trusted host when you install a Web Agent. After a host is registered, you can view it in the Policy Server User Interface. By default, the Policy Server User Interface does not display trusted host objects. To view trusted host objects, select Trusted Hosts from the View menu in the Policy Server User Interface.

Note: You cannot use the Policy Server User Interface to create, duplicate, or modify a trusted host. You can only delete a trusted host object. For instructions on registering a trusted host, see the *Web Agent Installation Guide*.

More information:

[Delete Trusted Host Objects](#) (see page 111)

Delete Trusted Host Objects

You can delete a trusted host and re-register it. You re-register a host using the Registration Tool, smreghost. This tool is installed with the Web Agent.

You can run the Web Agent Configuration Wizard to re-register a trusted host. If you do, delete or rename the SmHost.conf file first, or the Wizard does not prompt you to register the trusted host. We recommend using the smreghost tool.

For details on registering and re-registering trusted hosts, see the *Web Agent Installation Guide*.

To delete a trusted host object

1. Open the Policy Server User Interface.
2. In the System tab, select Trusted Hosts to display the Trusted Host List.
3. In the List pane, select the trusted host that you want to delete and right-click.
4. Select Delete Trusted Host.
The Deleting items dialog opens.
5. Confirm the correct trusted host name is going to be deleted, then click OK.

How a Host Configuration Object is Configured for a Trusted Host

Host Configuration Objects hold parameters for trusted hosts. After the trusted host connects to a Policy Server, it uses the settings in the Host Configuration Object.

On the Web Agent side, the Host Configuration Object being used by the trusted host is identified in the hostconfigobject parameter of SmHost.conf file. The settings in the SmHost.conf file are used by the Web Agent during the initialization phase of each web server child process. This means that the SmHost.conf file is not only used at start-up, but may also be used at run time by web servers that start new child processes to add capacity or to replace processes that stop unexpectedly. After initialization, the settings in the Host Configuration Object are used.

You need to create a Host Configuration Object before you create trusted host objects.

More information:

[Delete Trusted Host Objects](#) (see page 111)

Create a Host Configuration Object

The Policy Server User Interface displays Host Configuration Objects in the System tab. The Policy Server installation program creates a sample Host Configuration Object called DefaultHostSettings.

Do one of the following:

- To add a new Host Configuration Object, we recommend duplicating an existing object and modify the parameter values. You can duplicate the DefaultHostSettings object and use it as a template.
- Create an object without using an existing object.

Important! Do not directly modify and use the DefaultHostSettings object. Always copy this object and then modify it.

More information:

[Duplicate a Host Configuration Object](#) (see page 112)

[Create a Host Configuration Object Without Using an Existing Object](#) (see page 113)

Duplicate a Host Configuration Object

We recommend duplicating an existing object and modify the parameter values. You can duplicate the DefaultHostSettings object and use it as a template.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To duplicate a host configuration object

1. Open the Policy Server User Interface.
2. In the System tab, select Host Conf Objects to display the Host Conf Object List in the right pane.
3. In the List pane, select the object you want to duplicate and right-click.
4. Select Duplicate Configuration Object.

The Host Configuration Object Properties dialog opens.

5. In the Host Configuration Object Properties dialog, enter a Name for the object.

Note: This name cannot be the same as the name of an existing Agent object. If you use a name assigned to another Web Agent, you will see a message stating that the trusted host already exists.

6. Click OK to save the new object.

More information:

[Edit a Host Configuration Object](#) (see page 115)
[Host Configuration Object Dialog](#) (see page 538)

Create a Host Configuration Object Without Using an Existing Object

You can create a new Host Configuration Object without duplicating an existing object.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a host configuration object

1. Open the Policy Server User Interface.
2. From the System tab, right-click Host Conf Objects.
3. Select Create Host Conf Object.
The Host Configuration Object Properties dialog opens.
4. Select the General tab.
5. In the Name field, enter a name for the Host Configuration Object.
6. Optionally, in the Description field, enter a description for the object.
7. To specify parameters, click Add.
The Edit Parameter Dialog opens.
8. Use the Edit Parameter Dialog to specify parameters and values.
Although you can specify multiple values for some parameters, each parameter name must be unique.
9. Click Apply to write each parameter to the policy store.
10. When you are finished adding parameters, click OK.

More information:

[Host Configuration Object Dialog—General Tab](#) (see page 539)
[How a Host Configuration Object is Configured for a Trusted Host](#) (see page 111)

Add Multiple Policy Servers to the Host Configuration Object

You may want to configure multiple Policy Servers in the Host Configuration Object for:

- trusted host connections
- failover or round robin operation

To specify which Policy Server(s) can be accessed by a trusted host, add them to the Host Configuration Object.

Note: You must create a Host Configuration Object before deleting trusted host objects, and specify the destination Policy Server(s).

To add a Policy Server to the Host Configuration Object

1. Access the Policy Server User Interface.
2. Double-click a Host Configuration Object from the list.

The Host Configuration Object Properties opens.

3. Highlight the #PolicyServer parameter and click Edit.

The Edit Parameter Dialog opens.

4. Remove the pound sign (#) from the parameter name.
5. In the Value field, enter the address, including port numbers of the Policy Server.
The syntax is:

`<policy_server_IP_address>,<Accounting_port>,<Authentication_port>,<Authorization_port>`

Note: The entry must contain all these pieces of information.

6. Click OK.

When adding multiple Policy Servers to the Host Configuration Object, they must be added as a multi-valued parameter. For an example of entering a multi-valued parameter for Policy Servers, read the procedure in [Configure a Multi-Value Parameter](#) (see page 116).

Note: For load balancing or failover operation, you need to have more than one Policy Server defined in the Host Configuration Object.

More information:

[Delete Trusted Host Objects](#) (see page 111)

[Operation Mode](#) (see page 108)

[How a Host Configuration Object is Configured for a Trusted Host](#) (see page 111)

Edit a Host Configuration Object

To modify parameter settings in the Host Configuration Object, edit the name and/or value.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To edit a single, unencrypted parameter in the Host Configuration Object

1. Open the Policy Server User Interface.
2. In the System tab, select the Host Conf Objects to display the Host Conf Object List in the right pane.
3. Right-click the Host Configuration Object that you want to modify.
4. Select Properties of Configuration Object.

The Host Configuration Object Dialog box opens.

5. Do one of the following:
 - To edit a parameter that is listed, highlight it and click Edit.
 - To add a parameter that is not listed, click Add.

The Edit Parameter Dialog box opens with the Plain radio button selected by default.

6. In the Parameter Name field, specify a name. If you are editing only the value, go to the next step.

If applicable, remove the pound sign (#) to make the parameter active.

7. In the Value field, enter the modified value and click OK.

You return to the Host Configuration Object Properties dialog.

Note: New parameters are added to the end of the list in the Host Configuration Object Properties dialog. To move a parameter in the list, select it and use the up or down arrows.

8. Click OK.

Values can also be encrypted or Multi-Value. Details in:

- [Configure an Encrypted Parameter](#) (see page 116)
- [Configure a Multi-Value Parameter](#) (see page 116)

Configure an Encrypted Parameter

In order to configure an encrypted parameter, follow the instructions for [Edit a Host Configuration Object](#) (see page 115), with the following exceptions.

To configure an encrypted parameter

1. In the Edit Parameter Dialog box, select the Encrypted radio button.
2. In the Parameter Name field, enter a new or modified name for the parameter.
3. In the Encrypted Value field, enter the value that you want encrypted. It will be encrypted automatically. Re-enter this value in the Confirm encrypted value field.
4. Click OK to finish.

More information:

[Edit a Host Configuration Object](#) (see page 115)

Configure a Multi-Value Parameter

Multi-value entries enable you to specify more than one value for a single parameter. Only a subset of parameters can be multi-value. For example, if you want to use failover mode for the Policy Server, you are required to specify more than one Policy Server.

Note: Although you can specify multiple *values* for a parameter, each parameter name in the Host Configuration Object must be unique.

In order to configure a multi-value parameter, follow the instructions for [Edit a Host Configuration Object](#) (see page 115). However, make the following changes in the Edit Parameter Dialog box.

To configure a multi-value parameter

1. In the Edit Parameter Dialog box, select the Multi-value radio button.
2. In the Parameter Name field, specify the parameter name.
3. In the Enter Single Value field, enter the value, then click Add.
The entry is added to the Multiline Display box.
4. Repeat the previous step for each entry that you want to add to a single parameter.
5. You can also modify the entries in the following ways:
 - To remove a value, select it and click Remove
 - To modify a value, select it and click Edit.
 - To change the order of a value in the list, use the up and down arrows.
6. Click OK when you are finished modifying parameters.

Configure Policy Server Clusters for a Host Configuration Object

You can configure Policy Servers in a cluster for failover operation. Clustering servers enables failover from one group of servers to another group.

Policy Server clusters are defined as part of a Host Configuration Object. When a SiteMinder Web Agent initializes, the settings from the Host Configuration Object are used to setup communication with Policy Servers.

To configure a cluster

1. Open the Policy Server User Interface.
2. In the System tab of the SiteMinder Administration window, select Host Conf Objects.
3. In the List pane, select the host configuration object to which you want to add cluster configuration information.

Once you configure a cluster for a host configuration object, any Agent that uses the object to establish its Policy Server connections will use the specified cluster configuration.
4. From the Edit menu, select Properties of Configuration Object.

The Host Configuration Object Dialog box opens.
5. Select the Clusters tab.
6. In the Clusters tab, click the Add button.

The Cluster Setup Dialog box opens.
7. In the Add Server group box, do one of the following:
 - Select the IP Address radio button and enter the IP address of a Policy Server in the cluster in the provided fields.

Note: If you do not know the IP address of the Policy Server, but you know the host name, click the DNS Lookup button to search for the IP address of the Policy Server.
 - Select the Domain Name radio button and enter the domain name of the system where the Policy Server is installed. For example, server.company.com.
8. In the Server Port field, enter the port for Policy Server processing.
9. Click the Add to Cluster button.

The Policy Server appears in the servers list in the Current Setup group box.
10. Repeat steps 6 through 9 for any other Policy Servers you want to add to the cluster.
11. When you finish adding Policy Servers to the cluster, click the OK button to save your changes and return to the Host Configuration Object Properties dialog.

12. In the Failover Threshold Percent field, enter a percentage of Policy Servers that must be active and click Apply.

If the percentage of active servers in the cluster falls below the percentage you specify, the cluster fails over to the next available cluster in the list of clusters.

The Policy Server User Interface automatically calculates the Failover Threshold value displayed in the column to the right of the lists of servers in each cluster from the Failover Threshold Percent you specify by taking the specified percentage of servers in the cluster and rounding up to the nearest integer. For example, if the number of servers is 5 and the Failover Threshold Percent is in the range between [0 – 20], the Failover Threshold is 1.

When you set the Failover Threshold Percentage, it applies to all clusters that use the Host Configuration Object.

13. Click on the General tab.

14. Note the Policy Server parameter in the Parameter Name column.

The value of the Policy Server parameter is overridden by the contents of the Clusters tab. This parameter should be commented for clarity (preceded by a # symbol), since its value does not apply as long as Policy Servers are specified in the Clusters tab.

Note: In order for the value of the Policy Server parameter in the General tab to apply, no Policy Servers should be specified in the Clusters tab. If clusters are configured and you decide to remove the clusters in favor of a simple failover configuration using the Policy Server parameter in the General tab, be sure to delete all Policy Server information in the Clusters tab.

15. Click OK to save your changes.

Dynamic Host Configuration Object (HCO) Updates

You can add Policy Servers to and remove them from an existing Policy Server cluster or the default cluster when no cluster is configured. Without dynamic HCO updates, the Web Server must be rebooted and the Host Configuration Object initialized by the Web agent for Policy Server changes to take effect.

With dynamic HCO updates, you can add and remove Policy Servers without needing to reboot the Web Server for the changes to take effect. The Web agent picks up the Policy Server changes dynamically and the Host Configuration Object is updated after a delay that depends on the internal polling interval.

Note: You cannot add or remove Policy Server clusters or modify other HCO parameters without rebooting the Web Server.

Enable Dynamic Host Configuration Object (HCO) Updates

With dynamic HCO updates, you can add Policy Servers to and remove them from an existing Policy Server cluster without needing to reboot the Web Server for the changes to take effect.

To enable dynamic Host Configuration Object (HCO) updates

1. Open the following file with a text editor:
`web_agent_home\config\SmHost.conf`
web_agent_home
Specifies the directory where the Web Agent is installed.
2. Add or update the following line in the file:
`enableDynamicHCO="YES"`
3. Save and close the SmHost.conf file.
4. Restart the Web Server.

Dynamic HCO updates are enabled.

Note: For more information, see the *Web Agent Installation Guide*.

Delete a Host Configuration Object

You can delete unused or unnecessary host configuration objects using the Policy Server.

To delete a host configuration object

1. Open the Policy Server User Interface.
2. In the System tab of the SiteMinder Administration window, select Host Conf Objects.
3. In the List pane, select the Host Configuration Object that you want to delete and right-click.
4. Select Delete Configuration Object.
The Deleting items dialog displays.
5. Confirm that the correct Host Configuration Object is being deleted then click OK to remove it.

Configure a Web Agent

There are a number of tasks that must be completed in order to fully configure a Web Agent. These tasks apply to local and central configuration of a Web Agent.

Note: You must set up the Policy Server for Web Agent communication before you install a Web Agent and register a trusted host.

To configure a web agent

1. Install a Policy Server.
2. Create a Host Configuration Object.
3. Grant the Register Trusted Hosts privilege to a Policy Server Administrator. An administrator must have the privilege to register trusted hosts.

If you create an administrator with *only* the Register Trusted Hosts privilege, that administrator will not be able to use the Policy Server User Interface.
4. Create an Agent object to name the Agent. Do not confuse this object with an Agent Configuration Object.
5. Create an Agent Configuration Object.

If you plan to configure an Agent locally, you still need this object to enable the local configuration parameter, AllowLocalConfig.

6. At the client site, install the Web Agent.
7. Register the trusted host. Part of this process is to provide the name of the Host Configuration Object that you already created at the Policy Server.
8. When the Agent-related policy objects are configured, enable the Web Agent. This setting is in the local Agent configuration file.

Note: For details, see the *Policy Server Installation Guide* and *Web Agent Installation Guide*

More information:

[How a Host Configuration Object is Configured for a Trusted Host](#) (see page 111)

[Administrator Scope and Tasks](#) (see page 225)

[Create an Agent Object to Establish a Web Agent Identity](#) (see page 123)

[Configure an Agent Configuration Object for a Web Agent](#) (see page 126)

[Enable a Web Agent](#) (see page 133)

Web Agent Parameter Descriptions

The *SiteMinder Web Agent Guide* contains all the parameter descriptions, the default values, and instructions on setting the parameters. Whether you are configuring a Web Agent centrally or locally, see this guide for parameter descriptions.

Configure Web Agents Centrally

To centrally configure Web Agents, perform the steps outlined in [There are a number of tasks that must be completed in order to fully configure a Web Agent](#). These tasks apply to local and central configuration of a Web Agent.

If you specify any configuration parameters locally, the parameter values in the local Agent configuration file override the values in the corresponding Agent Configuration Object, merging the input from both configuration sources.

To use a local configuration exclusively, without combining input from an Agent Configuration Object and an Agent configuration file, configure the Agent Configuration Object with only the `AllowLocalConfig` parameter and set it to yes. This ensures that the Web Agent will only have configuration data from the local configuration file.

To better understand how central and local configuration work together, read [Combined Central and Local Configuration](#) (see page 122).

There are a number of tasks that must be completed in order to fully configure a Web Agent. These tasks apply to local and central configuration of a Web Agent.

Configure Web Agents Locally

The Web Agent reads both the Agent Configuration Object and the local Agent configuration file, overriding values in the Agent Configuration Object with the values in the local Agent configuration file. The Web Agent merges them together into one configuration source. This enables you to modify only a small subset of Agent parameters locally, then rely on the central Agent Configuration Object for the rest of an Agent's configuration.

To better understand how central and local configuration work together, read [Combined Central and Local Configuration](#) (see page 122).

To configure parameters locally

1. Obtain permission to perform local configuration from a Policy Server administrator.
2. Complete the steps in There are a number of tasks that must be completed in order to fully configure a Web Agent. These tasks apply to local and central configuration of a Web Agent.
3. In the Agent Configuration Object, set the AllowLocalConfig parameter to yes.
4. Edit the Web Agent configuration file (WebAgent.conf and/or LocalConfig.conf).

Be sure to modify a copy of the Web Agent configuration file and maintain a backup copy.

For all Web Agents except IIS 6.0, there is a WebAgent.conf.sample file in the <web_agent_home>\config directory. You should modify this file, then save it under the name WebAgent.conf to the appropriate web server location.

For IIS 6.0 Web Agents, this Agent uses the LocalConfig.conf file in <web_agent_home>\bin\IIS directory as its active configuration file. Modify this file if you want to make changes. The copy of the LocalConfig.conf file in <web_agent_home>\config is the original that you should not change.

Note: If you are using an IIS 6.0 Web Agent, the main configuration file is called LocalConfig.conf. The WebAgent.conf file is still used, but only for core Agent settings that enable the Agent to start and connect to the Policy Server. For additional information about local configuration and parameter descriptions, see the *SiteMinder Web Agent Guide*.

There are a number of tasks that must be completed in order to fully configure a Web Agent. These tasks apply to local and central configuration of a Web Agent.

Combined Central and Local Configuration

When a Web Agent is enabled, it searches the Agent Configuration Object for configuration information, and notes the value of the AllowLocalConfig parameter. If this parameter is set to yes, the Web Agent searches the corresponding Agent's local configuration file for modified or additional parameters, overriding any Agent Configuration Object parameters with the value from its configuration file.

Using the central and local configuration sources, the Agent creates a unified local copy of an Agent Configuration Object that it uses for configuration. The local copy does not alter the Agent Configuration Object that resides at the Policy Server.

Example of Using Central and Local Configuration

Scenario:

To configure multiple cookie domain single sign-on across your SiteMinder network without having to configure each Agent individually.

The CookieDomain parameter in the Agent Configuration Object is set to acmecorp.com. However, for one Web Agent in your network, you want to set the CookieDomain parameter to test.com while still using all the other parameter values set in the Agent Configuration Object.

Solution:

To implement the example configuration

1. Configure an Agent Configuration Object with all the parameters applicable for your environment.
2. In the Agent Configuration Object, set the AllowLocalConfig parameter to yes.
3. At one Web Agent, change only the CookieDomain parameter to test.com. Do not modify any other parameters.

The value for the CookieDomain parameter in the Agent configuration file overrides the value in the Agent Configuration Object, while the Agent Configuration Object determines the settings for all the other parameters.

Create an Agent Object to Establish a Web Agent Identity

You need to create an Agent object in the Policy Server User Interface. The Agent object creates an Agent identity by assigning the Agent a name. The identity establishes a mapping between the name and the IP address of the web server instance hosting a Web Agent. Additionally, the Policy Server uses the Agent identity to associate policies with the correct Web Agent.

An Agent identity is associated with its configuration by specifying the Agent name in a local Agent configuration file or Agent Configuration Object.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create an agent object to establish a web agent identity

1. Log into the Policy Server User Interface.

2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Agent.

The SiteMinder Agent dialog opens.

3. In the Name field, enter the name of the new Agent. This name is case-insensitive, and must be 7-bit ASCII characters, in the range 32-127.

You cannot create one agent named MyAgent and another named myagent.

Note: The value in the Name field must match the name entered in the DefaultAgentName or AgentName parameter in the Agent Configuration Object or Agent configuration file.

4. Optionally, in the Description field, enter a brief description of the Agent.
5. Since you are creating a 5.5 Web Agent identity, leave the Support 4.x agents check box unchecked.
6. In the Agent Type group box, ensure that the,
 - SiteMinder radio button is selected.
 - Web Agent is selected in the drop-down list.
7. Click OK to save the changes.

When you create realms, you will associate each realm with an Agent defined here.

More information:

[Grouping Resources in Realms](#) (see page 315)

[Start the Policy Server User Interface](#) (see page 64)

[Agent Dialog](#) (see page 535)

Agent Object Association with an Agent Configuration Object or File

The Agent name you specified for the Agent object must be specified in the DefaultAgentName or AgentName parameter of the Agent Configuration Object or configuration file.

Configure an Agent Object for a 4.x Web Agent Identity

Important! You will receive correspondence from CA regarding the end date for 4.x Web Agent support.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure an agent object

1. Log into the Policy Server User Interface.

2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Agent.

The Agent Properties dialog opens.

3. In the Name field, enter the name of the new Agent. This name is case-insensitive. You can't create an agent named MyAgent and a different agent named myagent.

Agent names must be 7-bit ASCII characters, in the range 32-127.

Note: For Web Agents, the Agent identity name must correspond to the Agent name that you specified when you installed the Agent. If you change the Agent name in the Web Agent Management Console or the WebAgent.conf file (see the *SiteMinder Web Agent Guide*), you must also edit the Agent name here.

4. Optionally, in the Description field, enter a brief description of the Agent.
5. Select the Support 4.x agents check box.

A further set of controls specific to 4.x agent types appears in the lower part of the dialog.

6. In the Agent Type group box, select the SiteMinder radio button.
7. In the drop-down list in the Agent Type group box, select Web Agent.
8. In the IP Address or Host Name field, enter the IP address or the host name of the server on which the Agent resides.

Note: For virtual servers, define virtual server names and IP addresses as you would define a single server. Each Agent on a virtual server must have a unique Agent Name.

9. In the Shared Secret Field, enter an alphanumeric shared secret.

Enter the shared secret assigned to the Agent when the Agent was installed on the Web server. The shared secret is used for mutual authentication when a 4.x Agent attempts to connect to a Policy Server. The secret must be between 1 and 255 characters in length and contain no embedded spaces.

Note: You must use the same shared secret for each virtual server on a single Web server.

10. In the Confirm Secret field, re-enter the shared secret.
11. Click Apply to save the changes, or click OK to save the changes and return to the SiteMinder Administration window.

When you create realms, you will associate each realm with a Web Agent identity.

You can only configure a SiteMinder 4.x Web Agent using a local Web Agent configuration file. For configuration parameter descriptions, see the *SiteMinder Web Agent Guide*.

There are a number of tasks that must be completed in order to fully configure a Web Agent. These tasks apply to local and central configuration of a Web Agent.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Agent Dialog](#) (see page 535)

[Grouping Resources in Realms](#) (see page 315)

Set the Configuration Parameters in the Agent Configuration Object

The following procedure contains the two general subprocedures required to set the configuration parameters of an agent configuration object.

To define the Web Agent's configuration

1. Create an Agent Configuration Object
2. Modify the configuration parameters in this object.

The *SiteMinder Web Agent Guide* contains all the parameter descriptions, the default values, and instructions on setting the parameters. Whether you are configuring a Web Agent centrally or locally, see this guide for parameter descriptions.

More information:

[Configure an Agent Configuration Object for a Web Agent](#) (see page 126)

Configure an Agent Configuration Object for a Web Agent

An Agent Configuration Object holds the parameters that define the Web Agent configuration. They are the Policy Server's counterpart to Web Agent configuration file. The Policy Server User Interface displays Agent Configuration Objects in the System tab.

When you configure a Web Agent you are prompted for the Agent Configuration Object. This is how you associate an object with a particular Web Agent. For more information, see the *Web Agent Installation Guide*.

There are a set of required Web Agent parameters for each Agent Configuration Object.

More information:

[Required Agent Configuration Object Parameters](#) (see page 127)

Required Agent Configuration Object Parameters

When you configure an Agent Configuration Object, the following parameters must be configured:

Web Agent	Required Parameters
All Agents	DefaultAgentName
IIS Agent	DefaultUserName DefaultPassword
Domino Agent	DominoDefaultUser DominoSuperUser

These parameters are also defined in the *Web Agent Installation Guide*.

Specify the IIS (Proxy) User

If you are configuring an IIS Web Agent, configure the following settings in the Agent Configuration Object:

Parameter Name	Parameter Value
DefaultUserName	Username of the IIS Proxy User
DefaultPassword	Password of the IIS Proxy User

The DefaultUserName and DefaultPassword identify an existing NT user account that has sufficient privileges to access resources on an IIS Web server protected by SiteMinder. When users want to access resources on an IIS Web server protected by SiteMinder, they may not have the necessary server access privileges. The Web Agent must use this NT user account, which is assigned by an NT administrator, to act as a proxy user account for users granted access by SiteMinder.

Note: If you plan to use the NTLM authentication scheme, or enable the Windows User Security Context feature, do not specify values for these IIS parameters.

Specify the Domino Users

If you are configuring a Web Agent on a Domino server, you need to edit the following settings to match your system:

DominoDefaultUser

If the user is not in the Domino Directory, and they have been authenticated by SiteMinder against another user directory, this is the name by which the Domino Web Agent identifies that user to the Domino server. This value can be encrypted.

DominoSuperUser

Ensures that all users successfully logged into SiteMinder will be logged into Domino as the Domino SuperUser. This value can be encrypted.

Specify the Agent Name

For all Web Agents, specify a value for the DefaultAgentName. This parameter identifies the Agent identity that the Web Agent uses when it detects an IP address on its Web server that does not have an Agent identity assigned to it. By default, the default Agent name is the name of the installed Web Agent.

Create an Agent Configuration Object

The installation program for the Policy Server creates sample Agent Configuration Objects for each supported Web Agent, such as IISDefaultSettings or iPlanetDefaultSettings. You can duplicate these sample objects and use them as template objects for your organization.

Do one of the following:

- To create a new Agent Configuration Object, we recommend duplicating an existing object and modify the parameter values. You can duplicate the DefaultHostSettings object and use it as a template.
- To create a new object without using an existing object, read [Create a Host Configuration Object Without Using an Existing Object](#) (see page 113).

Important! We recommend against modifying and using the default objects.

More information:

[Duplicate a Host Configuration Object](#) (see page 112)

Duplicate an Agent Configuration Object

If you need to make an Agent Configuration Object, we recommend duplicating an existing Agent Configuration Object and modify parameter values, than to create a new object.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To duplicate an agent configuration object

1. Open the Policy Server User Interface.
2. From the System tab, select Agent Conf Objects to display the Agent Conf Object List in the right pane.
3. In the List pane, select the Agent Configuration Object that you want to duplicate then right-click.
4. Select Duplicate Configuration Object.
The Agent Configuration Object Properties dialog opens.
5. Edit the Name field, if desired.
6. Click OK to save the new Agent Configuration Object.

To modify parameters for the newly created Agent Configuration Object, read [Modify an Agent Configuration Object](#) (see page 130).

More information:

[Agent Configuration Object Dialog](#) (see page 540)

[Start the Policy Server User Interface](#) (see page 64)

Create an Agent Configuration Object without Using an Existing One

To create an Agent Configuration Object, you need to add all of the desired parameters, including the parameter names.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create an agent configuration object

1. In the Policy Server User Interface, ensure that the System tab is active.
2. Right-click the node labeled Agent Conf Objects.
3. Select Create Agent Conf Object.

4. In the Name field of the Agent Configuration Object Properties dialog, enter a name for the Agent Configuration Object.
5. If you wish, in the Description field, enter a name for the Agent Configuration Object.
6. To specify parameters, click Add.
7. Use the Edit Parameter Dialog to specify parameters and values for your system.
Although you can specify multiple values for some parameters, each parameter name must be unique.
8. Click Apply to write each parameter to the policy store.
9. When you are finished adding parameters, click OK to close the Edit Parameter Dialog.

Modify an Agent Configuration Object

To modify an Agent Configuration Object, edit the parameter name and/or value. Values can be plain, encrypted, or multi-value.

For instructions on configuring encrypted or multi-value parameters, read:

- [Edit a Plain Parameter](#) (see page 130)
- [Configure an Encrypted Parameter](#) (see page 131)
- [Configure a Multi-Value Parameter](#) (see page 132)

For parameter descriptions and default settings, see the *SiteMinder Web Agent Guide*.

Edit a Plain Parameter

You can edit a single value in a plain parameter.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To edit a plain parameter

1. Open the Policy Server User Interface.
2. From the System tab, select Agent Conf Objects to display the Agent Conf Object List in the right pane.
3. From the List pane, select the Agent Configuration Object that you want to modify and right-click.

4. Select Properties of Configuration Object.

The Agent Configuration Object Properties dialog opens.

5. Do one of the following:

- To edit a parameter that is listed, highlight it and click Edit.
- To add a parameter that is not listed, click Add.

The Edit Parameter Dialog opens with the Plain radio button selected by default.

6. In the Parameter Name field, specify a name. If you are only editing the value, go to the next step.

If applicable, remove the pound sign (#) to make the parameter active.

7. In the Value field, enter the modified value.

8. Click OK.

You return to the Agent Configuration Object Properties dialog.

Note: New parameters are added to the end of the list displayed in the Agent Configuration Object Properties dialog. To reorder the list, select it then use the up or down arrows.

9. Click OK to exit the Agent Configuration Object Properties dialog.

Configure an Encrypted Parameter

Configuring an encrypted value is similar to editing a plain parameter. The encrypted parameter, however, requires additional security steps.

To configure an encrypted parameter

1. Follow the instructions for [Edit a Plain Parameter](#) (see page 130), with these exceptions:
2. In the Edit Parameter Dialog, select the Encrypted radio button.
3. In the Parameter Name field, enter a new or modified name for the parameter.
4. In the Encrypted Value field, enter the value that you want encrypted. It will be encrypted automatically. Re-enter this value in the Confirm encrypted value field.
5. Click OK.

Configure a Multi-Value Parameter

Multi-value entries enable you to specify more than one value for a single parameter. The following Web Agent parameters can be multi-value:

- AgentName
- IgnoreUrl
- OverrideIgnoreExtFilter

Note: Although you can specify multiple *values* for a parameter, each parameter name in the Agent Configuration Object must be unique.

To configure an multi-value parameter

1. Follow the instructions for [Edit a Plain Parameter](#) (see page 130), with the following changes:
2. In the Edit Parameter Dialog, select the Multi-value radio button.
3. In the Parameter Name field, specify the parameter name.
4. In the Enter Single Value field, enter the value, then click Add.
The entry is added to the Multiline Display box.
5. Repeat the previous step for each entry that you want to add to a single parameter.
6. You can also modify the entries in the following ways:
 - To remove a value, select it and click Remove
 - To modify a value, select it and click Edit.
 - To change the order of a value in the list, use the up and down arrows.
7. Click OK when you are finished modifying parameters.

Delete an Agent Configuration Object

You can use the Policy Server User Interface to delete unwanted agent configuration objects.

To delete an agent configuration object

1. Open the Policy Server User Interface.
2. From the System tab, select Agent Conf Objects to display the Agent Conf Object List in the right pane.
3. In the List pane, select the Agent Configuration Object that you want to delete then right-click.

4. Select Delete Configuration Object.
The Deleting items dialog displays.
5. Confirm that the correct Agent Configuration Object is being removed, then click OK.

Enable a Web Agent

Regardless of whether you configure a Web Agent centrally or locally, you can only enable and disable a Web Agent from the WebAgent.conf file.

The EnableWebAgent parameter value determines whether an Agent is enabled or disabled. Set it to yes to enable the Agent. It is set to no by default.

After you have set up all the necessary objects for the Policy Server and Agent to communicate, and you have installed and configured the Web Agent, then you can enable it.

For more information, see the *SiteMinder Web Agent Guide*.

Configure Agent Identities for Non-Web Agents

All SiteMinder Agents require an Agent identity. The Agent identity establishes a mapping between the name and the IP address of each Web, RADIUS, or application server instance hosting an Agent. The Policy Server uses the Agent identity to associate policies with the correct Agent.

You configure an Agent identity at the Policy Server by creating an Agent object in the Policy Server User Interface. In the Agent object, you assign a name to the Agent and define its agent type—Web Agent, EJB Agent, Servlet Agent, RADIUS Agent, Custom Agent.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Note: An Affiliate Agent is one of the Agent types for an Agent object—this Agent type applies only to 4.x Affiliate Agents. Do not use this option to configure SAML Affiliate Agents.

To configure an Agent object

1. Log into the Policy Server User Interface.

2. From the SiteMinder Administration menu bar, select Edit, System Configuration, Create Agent.

The Agent Properties dialog opens.

3. In the Name field, enter the name of the new Agent. This name is case-insensitive. You cannot create an Agent named MyAgent and a different agent named myagent.

Agent names must be 7-bit ASCII characters, in the range 32-127.

4. Optionally, in the Description field, enter a brief description of the Agent.

5. If applicable, select the Support 4.x agents check box.

A further set of controls specific to 4.x agent types appears in the lower part of the dialog.

6. In the Agent Type group box, select the SiteMinder radio button, unless you are configuring a RADIUS Agent.

The contents of the SiteMinder Agent Dialog change to support the agent type you have chosen.

7. Complete any remaining fields based on the appropriate instructions:

- For Web Agents, read [Create an Agent Object to Establish a Web Agent Identity](#) (see page 123)
- For RADIUS Agents, read [Configure a RADIUS Agent](#) (see page 134)
- For Custom Agents, read [Create Custom Agents](#) (see page 136)
- For EJB and Servlet Agents, see the appropriate SiteMinder Application Server Agent guide for WebLogic or WebSphere application servers.

More information:

[Start the Policy Server User Interface](#) (see page 64)

Configure a RADIUS Agent

A RADIUS Agent secures an entire application that communicates using the RADIUS protocol. For information about RADIUS and SiteMinder, see the *SiteMinder Policy Server Management Guide*.

After you configure a RADIUS Agent, install and configure a SiteMinder Agent on a RADIUS client or application server. You can only configure RADIUS Agents locally. For instructions, see the *SiteMinder Web Agent Installation Guide*.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a RADIUS Agent

1. Log into the Policy Server User Interface.
2. From the SiteMinder Administration menu bar, select Edit, System Configuration, Create Agent.

The Agent Properties dialog appears.

3. In the Name field, enter the name of the Agent. This name is case-insensitive, and must be 7-bit ASCII characters, in the range 32-127. You cannot create one Agent named MyAgent and another named myagent.

Note: The Agent name you specify must match the Agent name that you enter when you install the SiteMinder Agent. If you change the Agent name in at the Agent, you must also edit the Agent name at the Policy Server.

4. Optionally, in the Description field, enter a brief description of the Agent.
5. Select the Support 4.x agents check box.

Additional controls specific to 4.x agent types appear in the dialog.

6. In the Agent Type group box, select the RADIUS radio button.

The contents of the dialog change to support RADIUS Agents.

7. From the drop-down list in the Agent Type group box, select the vendor type of the RADIUS device the Agent will protect.

Generic RADIUS can be used to protect any type of RADIUS device, however, Generic RADIUS Agent types do not provide access to vendor-specific response attributes.

8. In the IP Address or Host Name field, enter the IP address or the host name of the RADIUS client (NAS device).

9. In the Shared Secret field, enter an alphanumeric shared secret.

Enter the shared secret used by the NAS device. The shared secret is used for mutual authentication between the NAS device and the Policy Server. The secret must be between 1 and 255 characters in length and contain no embedded spaces.

10. In the Confirm Secret field, re-enter the shared secret.

11. If you selected the RADIUS agent type, you can also fill in the Realm Hint Attribute.

Realm hints are needed if a RADIUS Agent protects a NAS device that must authenticate users in different domains, such as mydomain.com and yourdomain.com. A realm hint is a RADIUS attribute that provides SiteMinder with information about the domain in which to authenticate the user.

Note: For more information about realm hints, see the *SiteMinder Policy Server Management Guide*.

12. Click Apply to save the changes, or click OK to save the changes and return to the SiteMinder Administration window.

If you want to protect resources with the RADIUS Agent, you can now associate this Agent with a realm.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Agent Dialog](#) (see page 535)

[Configure a Realm Protected by a RADIUS Agent](#) (see page 322)

Create Custom Agents

If you have developed your own Agent, you must configure a new Agent type in the Policy Server User Interface. The Agent type defines the behavior of an agent. After you define a Custom Agent type, you can use the custom Agent to protect resources. For example, if you developed a custom FTP Agent, you would then need to define an FTP Agent type.

To create a custom Agent, you use the SiteMinder Agent API, either the C or Java version. The SiteMinder Agent API and associated documentation are provided by the Software Development Kit, which is available separately.

For more information about using an API to create a custom Agent, see the *Developer's Guide for C* or the *Developer's Guide for Java*.

Note: Custom Agents do not support central agent configuration.

Configure a Custom Agent Type

After you develop and Agent, you must configure a custom Agent type for that Agent. For instructions, see the *Developer's Guide for C* or the *Developer's Guide for Java*.

Create a Custom Agent Object for the Agent Identity

After you have created an Agent type using the Agent API, you must configure an Agent object in the Policy Server User Interface, which includes the Agent type you defined. The Agent object enables the Agent to establish an Agent identity.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create an Agent object

1. Log into the Policy Server User Interface.
2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Agent.

The Agent Properties dialog opens.

3. In the Name field, enter the name of the new Agent. This name is case-insensitive, and must be 7-bit ASCII characters, in the range 32-127.

You cannot create one agent named MyAgent and another named myagent.

4. Optionally, in the Description field, enter a brief description of the Agent.
5. Select the Support 4.x agents check box.

A further set of controls specific to 4.x agent types appears in the lower part of the dialog.

Note: For a custom agent, you must select the Support 4.x agents option. If you do not, the Policy Server will not communicate with the custom agent.

6. In the Agent Type group box, select the SiteMinder radio button.
7. From the drop-down list in the Agent Type group box, select the entry for your Custom Agent type.

The contents of the SiteMinder Agent Dialog change to support Custom Agents.

8. In the Secret field, enter a shared secret that the Custom Agent uses to decrypt data sent by the Policy Server.

This key must match the shared secret that was defined for the Affiliate Agent when it was installed. The encryption key must be between 1 and 255 characters in length and contain no embedded spaces.

9. In the Confirm Secret field, re-enter the shared secret.
10. Click OK to save the changes.

The *SiteMinder Web Agent Guide* contains all the parameter descriptions, the default values, and instructions on setting the parameters. Whether you are configuring a Web Agent centrally or locally, see this guide for parameter descriptions.

More information:

[Configure Agent Identities for Non-Web Agents](#) (see page 133)

[Configure a Custom Agent Type](#) (see page 136)

[Start the Policy Server User Interface](#) (see page 64)

[Agent Dialog](#) (see page 535)

Resource Protection with a SiteMinder Agent

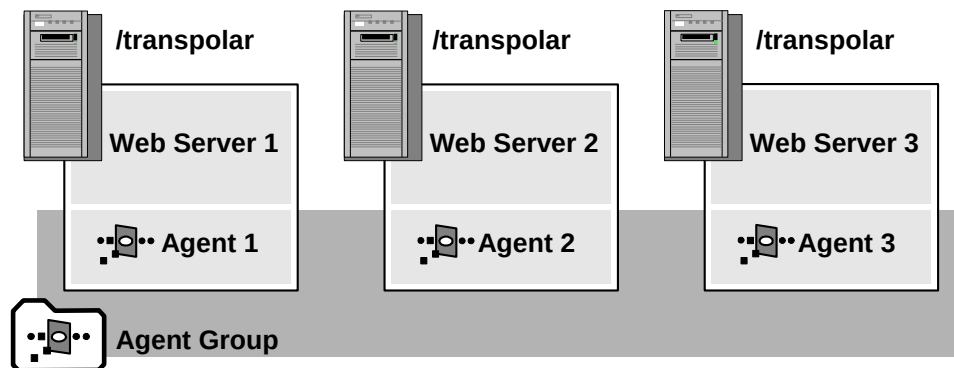
After you have configured an Agent, you associate it with a realm, which is a collection of resources that you want to protect. Realms are protected by rules, which get included in an access control policy.

To associate an Agent with a realm, read [Grouping Resources in Realms](#) (see page 315).

Agent Group Usage

An Agent group is a collection of Agents of the same type grouped together for common resource protection. The advantage of Agent Groups is that you can provide access to the same resource to a larger user base because the resource is duplicated on many web servers/Web Agents. It also saves time because you define only one policy for all of the Web Agents. For example, you can use an Agent group to protect a group of Web Servers that use round robin processing to supply access to the same resources (Web farm).

In the following figure, you can protect the Web farm with one set of policies, and create an Agent group that is bound to the set of policies.



You can create Agent groups even if the Agents you want to include in a group have not yet been created. Once you add a new Agent in SiteMinder, you can edit an Agent group and add the new Agent to the group.

Note: If you configure Agent groups, you cannot directly assign a set of parameter values to an Agent group. You can assign the same Agent Configuration Object to multiple agents and edit the object.

More information:

[Policies](#) (see page 375)

Configure an Agent Group

Before adding agents to an agent group, you should configure the agent group.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure an agent group

1. Log into the Policy Server User Interface.
2. From the menu bar of the SiteMinder Administration window, select View, Agent Groups.

The Policy Server places a check mark in the View menu to the left of the menu selection and Agent Groups appear in the System tab of the SiteMinder Administration window.

Note: If there is already a check next to Agent Groups in the View menu, selecting the option again will remove Agent Groups from the System tab.

3. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Agent Group.

The Agent Group Properties dialog opens.

4. In the Name field enter the name of the new Agent group.
5. In the Description field, enter a brief description of the Agent group.
6. In the Agent Type group box, select the type of Agents to include in the Agent group: SiteMinder or RADIUS.
7. Do one of the following:
 - If you selected the SiteMinder agent type, select Web Agent from the drop-down list.
 - If you selected the RADIUS Agent Type, you must also select a vendor from the drop-down list.
8. Go to [Add Agents to an Agent Group](#) (see page 140).

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Agent Group Dialog](#) (see page 541)

Add Agents to an Agent Group

You must add existing Agents to an Agent group. All of the Agents in an Agent Group must be of the same type, for example, all Web Agents, all Affiliate Agents, or all RADIUS Agents.

To add Agents to an Agent group

1. In the Agent Group Properties dialog, click Add/Remove.

The Agent Group Items dialog opens.

Note: For RADIUS Agent groups, the only items listed in the Available Members region are Agents supported by the selected RADIUS vendor. For example, if the Agent Group type is 3-Com, only 3-Com Agents are listed. If you selected Generic RADIUS, the Available Members list contains all defined RADIUS Agents.

2. Select the name of the Agent you want to add to the Agent group from the Available Members list.
3. Click the Left Arrow button to move the selected Agent to the Current Members list.
4. Repeat this procedure for all Agents you want to add to the Agent group.

If the list of Agents is long and the Agent you want to add to the Agent group is not displayed in the Available Members list, you can look for Agents using the search feature. The search feature is similar to the one that is available in the SiteMinder Administration window.

5. Click OK to save the member additions.

The Group Members group box in the SiteMinder Agent Group dialog displays the Agents you selected for the Agent group.

6. Click OK in the SiteMinder Agent Group dialog to save the Agent group and return to the SiteMinder Administration window.

More information:

[Agent Group Items Dialog](#) (see page 544)

[Search with the SiteMinder Administration Window](#) (see page 73)

Chapter 7: User Directories

This section contains the following topics:

[User Directory Connections Overview](#) (see page 141)

[Configure User Directory Connections](#) (see page 156)

[Configure SQL Query Schemes](#) (see page 197)

[Windows Directory Connections](#) (see page 204)

[View User Directory Contents](#) (see page 210)

[Search Directories](#) (see page 211)

[Universal IDs](#) (see page 212)

User Directory Connections Overview

User directories and user databases store user data, including organizational information, user and group attributes, and credentials such as passwords. The Policy Server User Interface allows you to configure connections to existing user directories and databases (user stores). The goal of configuring directory connections is to resolve how the Policy Server establishes a context for user identities. The Policy Server uses these connections to verify user identities and retrieve user attributes contained in the user stores.

The Policy Server supports the following as user directories:

LDAP

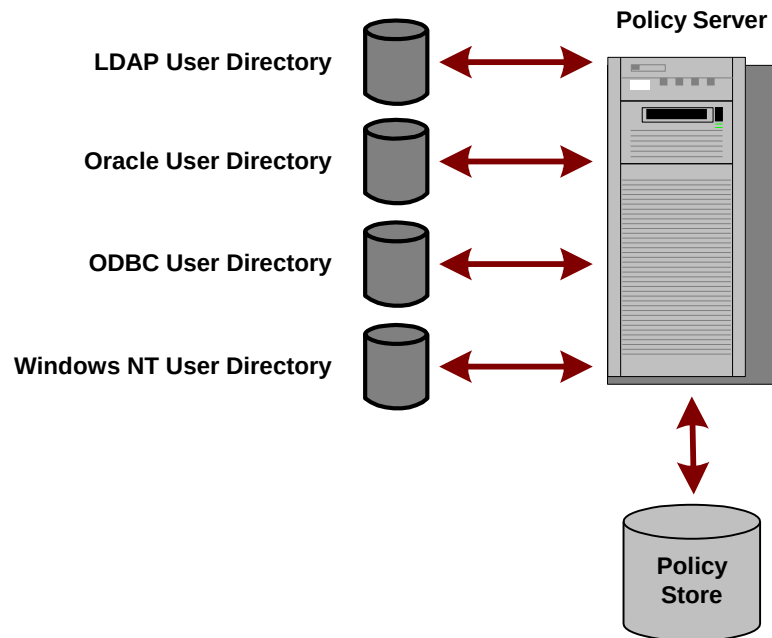
CA Directory, Oracle Directory Enterprise Edition (formerly Sun Java System Directory Server Enterprise Edition) LDAP Directory, Active Directory, Active Directory Application Mode, Active Directory Global Catalog, Novell eDirectory, Oracle Internet Directory, Domino LDAP, and IBM Directory Server

Databases

Microsoft SQL Server, Oracle, and IBM DB2

Custom user directories

As indicated in the following diagram, you can configure the SiteMinder Policy Server to connect to any number of user directories of the specified types.



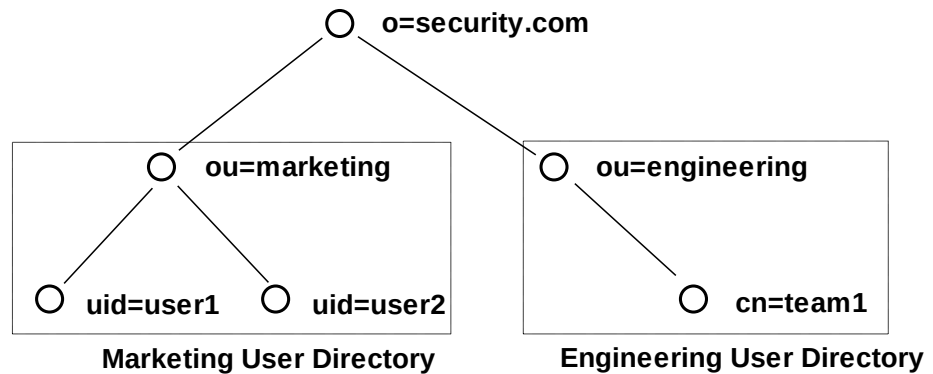
For a list of supported directories, see the Platform Support matrix for SiteMinder r6.0 SP6 on the Technical Support web site at <http://ca.com/support>.

LDAP Overview

The Policy Server can communicate with user directories that use the Lightweight Data Access Protocol (LDAP). LDAP user directories provide a powerful and flexible environment for providing user information to SiteMinder.

General Information About LDAP

LDAP user directories are created with an inverted tree structure. Due to this hierarchical structure, LDAP-enabled directories can contain multiple user namespaces. A namespace is a grouping of entities under a node in the LDAP Directory Information Tree (DIT). Any branch of an LDAP DIT can be defined in a user directory connection as a separate namespace. Typically, user directory connections are configured for DIT branches that represent an organization (o) or an organizational unit (ou). Users and user groups are located under an o= or ou= node in the directory structure.



Any node in an LDAP tree is identified by its distinguished name (DN), which is made up of a comma separated list of its own name and the names of the nodes above it in the directory tree. This method of naming allows each point in the user directory to have a unique DN.

For example, in the diagram above, one of the users in the Marketing department is identified by the following DN:

`uid=user1,ou=marketing,o=security.com`

The user group Engineering is identified as the following DN:

`ou=engineering,o=security.com.`

User Disambiguation in an LDAP Directory

User disambiguation is the process of locating a unique user in a user directory. There are two methods of locating users in a user directory: by DN or by search expression. In both of these methods, the Policy Server uses information entered in the Directory Setup tab in the User Directory Dialog, plus a user-supplied value, such as login name, to locate a user.

User Lookup via a DN

In this lookup option, the DN is constructed by concatenating values specified in the Start field in the LDAP User DN Lookup group box, the username as specified by the user during login, and the value specified in the End field.

The screenshot shows a configuration window titled "Directory Setup" with three tabs: "Directory Setup", "Credentials and Connection", and "User Attributes". The "Directory Setup" tab is active. It contains two main sections: "LDAP Search" and "LDAP User DN Lookup".

LDAP Search Section:

- Root: o=myorg.org
- Scope: Subtree (dropdown menu)
- Max time: 10 seconds
- Max results: 0

LDAP User DN Lookup Section:

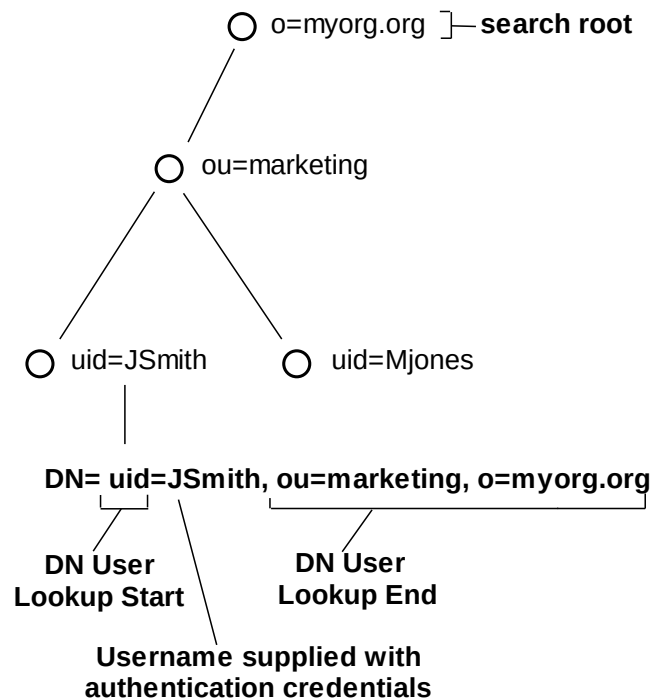
- Start: uid=
- End: ,ou=marketing, o=myorg.org
- Example button

At the top of the "Directory Setup" section, there are fields for *NameSpace: LDAP: (dropdown) and *Server: 123.123.12.12, with a "Configure..." button to the right.

The resulting DN has the following format:

<value in the Start field>, <username>, <value in End field>

See the following graphic for an LDAP Directory Information Tree (DIT) example.



In the previous diagram, the LDAP DIT design requires a DN to be of the form `uid=JSmith,ou=marketing,o=myorg.org`.

- The User DN Lookup Start property is `uid=`
- The User DN Lookup End property is `ou=marketing,o=myorg.org`

Only the unique part, *JSmith*, must be specified in the credentials when the user logs in.

User Lookup via a Search Expression

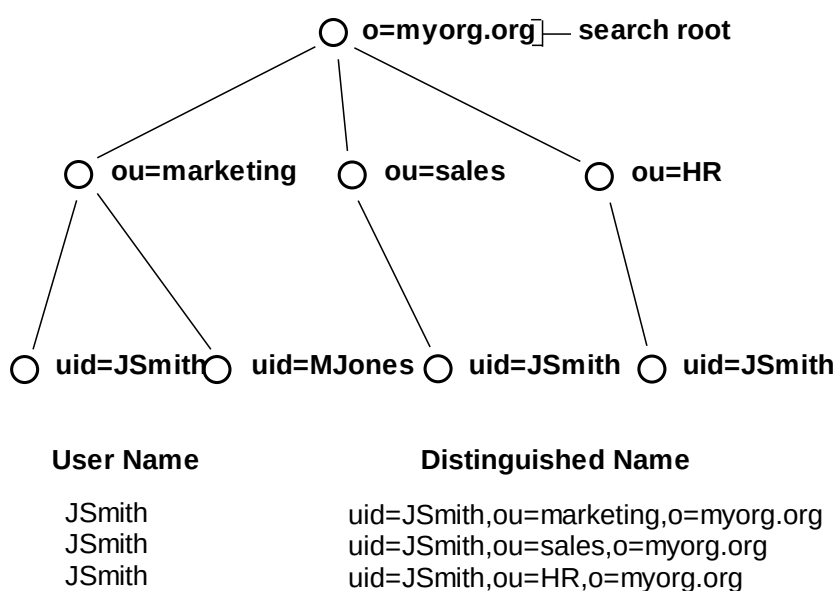
In situations where an LDAP directory server contains numerous users in complicated DITs, it may not be practical to create a large number of user directory connections. Some companies, for instance, have hundreds of organizational units and want to avoid having their end users log in with detailed string representations. Instead, one user directory connection pointing to a common root can be created with the User DN Lookup Start and User DN Lookup End properties defining an LDAP search expression. The result of the search expression is a list of user DNs for the Policy Server to try during authentication.

Example: Search expressions for user DN lookups

To locate a user across many organizational units, define the Start property as `(&(objectclass=inetOrgPerson)(uid=` and define the End property as `)`. Only the unique part, the *uid* value, must be specified in the credentials. In the section of the User Directory Dialog shown above, these values replace the values contained in the LDAP User DN Lookup group box.

Note: An InetOrgPerson is a common object class used in LDAP directory deployments.

See the following figure for the type of LDAP DIT where invoking a search expression is useful.



In this case, if JSmith from ou=sales wants to access a resource, JSmith can authenticate using only his or her name for credentials (as opposed to an entire DN string). By placing the uid= attribute between the User DN Lookup Start and User DN Lookup End fields with the search expressions in the corresponding fields, the Policy Server will find all DNs that match the LDAP query `(&(objectclass=inetOrgPerson)(uid=JSmith))`.

The Policy Server then has a list of DNs to choose from in giving access to the protected resource. Assuming the resource can only be accessed by the JSmith of ou=sales, the username/password for the DN `uid=JSmith,ou=sales,o=myorg.org` will be the one that is authenticated.

LDAP Search Filters

As you work with LDAP directory connections in the Policy Server, you may need to specify filters for LDAP search expressions. The following table provides a brief description of some common LDAP search filters.

Search Filter	Format	Description
Equality	attribute=value For example, to find a user whose user ID is jsmith, the search filter is uid=jsmith.	This filter finds a specific value for an attribute in an LDAP directory.
String Matching	attribute=*value, OR attribute=value*, OR attribute=val*ue, OR attribute=*value* For example, uid=*smith matches all values that end in smith, such as jsmith, msmith, etc. A value of uid=*smith* matches jsmith, msmith, and bsmith.	LDAP search filters support wild cards, which allow you to search for an attribute value based on a partial string. To find all of the values that match a partial string, use the wildcard character (*).
Greater than or equal to	attribute>=value For example, to find all of the users in a directory who are age 21 or over, part of the search filter would be age>=21.	This filter finds values that are greater than or equal to the specified value.
Less than or equal to	attribute<=value For example, to find all of the users in a directory who are age 21 or younger, part of the search filter would be age<=21.	This filter finds values that are less than or equal to the specified value.
Greater than	(!(attribute<=value)) For example, to find all of the users in a directory who are older than 21, part of the search filter would be (!(age<=21)).	LDAP does not support <i>greater than</i> expressions. In order to filter LDAP attribute values by <i>greater than</i> , you must use the Negation operator (!) in conjunction with a <i>greater than or equal to</i> expression.
Less than	(!(attribute>=value)) For example, to find all of the users in a directory whose age is less than 21, part of the search filter would be !(age>=21)).	LDAP does not support <i>less than</i> expressions. In order to filter LDAP attribute values by <i>less than</i> , you must use the Negation operator (!) in conjunction with <i>less than or equal to</i> expression.
Approximate	attribute~=value For example, the filter uid~=smith may return the values smithe and smitt.	This filter returns values that are similar to the value specified in the filter.

Search Filter	Format	Description
Presence	attribute=*	This filter determines if an attribute is present.
	For example, email=* returns all users who have an email address.	
Complex filters:	Intersection of Filter1 and Filter2: (&(filter1)(filter2))	Creates complex search filters.
And (&)	Union of Filter1 and Filter2: ((filter1)(filter2))	
Or ()	Satisfies Filter1, but not Filter2: (&(filter1)!(filter2))	
Not (!)	Note You must use parentheses to enclose the complex filter and each filter in the complex filter. For example, if you want to find all users whose User ID begins with the letter s and who are over 21 years old, you could use a filter of (&(uid=s*)(!(age<=21))).	

Objectclass Searches

Each entry in an LDAP table has at least one objectclass attribute. You can use a presence filter in conjunction with the objectclass attribute to build filters for searching your LDAP user directories. In SiteMinder environments, the objectclass attribute is most useful in the following cases:

- List all entries directly below an entry
To retrieve all entries one level below a directory entry, specify a search scope of One Level, and use a search filter of (objectclass=*). Since all LDAP directory entries have at least one objectclass attribute, the search filter returns a complete list of the entries below the root.
- List all entries in a subtree
To retrieve all entries in the branches below a directory entry, specify a search scope of Subtree, and use a search filter of (objectclass=*). The search filter returns a complete list of the entries in the entire subtree.

Filtered Characters in User IDs

SiteMinder provides LDAP search filter checking functionality that parses LDAP search filters to ensure that they comply with the LDAP standard (RFC).

All user login IDs are filtered for "(", ")", "\" characters by default before being checked against an LDAP user store. To disable this check, set the following EnableSearchFilterCheck registry value to 0:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\Ds\LDAPProvider\EnableSearchFilterCheck
```

Important! By disabling this check, you may expose your system to attack, and should not allow user IDs using these characters.

More information:

[User Directory Dialog—LDAP Namespace—Directory Setup Tab](#) (see page 548)

LDAP Referrals

The Policy Server supports LDAP referrals for LDAP user directory connections. An LDAP referral in one directory points to a location in another LDAP directory. There are two types of LDAP referrals: write referrals and read referrals. In addition, the Policy Server supports enhanced referral processing.

Note: In order for LDAP referrals to work correctly in a multiple policy store environment, the SiteMinder LDAP policy store schema must be applied to all replicas. For more information see the section describing LDAP policy store installation in the *Policy Server Installation Guide*.

Write Referrals

In a directory deployment that includes master and slave LDAP directories, LDAP write referrals allow updates to a master directory that can then be replicated to slave directories. In a SiteMinder deployment, you can specify a connection to a slave LDAP directory. If you use any of SiteMinder's features that require data to be written to the LDAP directory, SiteMinder automatically detects referrals that point to a master LDAP directory. The information that SiteMinder writes to the LDAP directory will be stored in the master LDAP directory and replicated to the slave LDAP directory according to the replication scheme of your network resources.

Read Referrals

In a large LDAP directory deployment, information may be divided among several LDAP directories. For example, one directory may contain enough user information to authenticate a user, while another directory may contain other important user attributes. The authentication directory can be configured to point to the directory containing user attributes. This process is called a read referral. If a directory connection exists for an LDAP directory that contains read referrals, SiteMinder is able to use the read referrals to retrieve information from the associated directories.

More information:

[Enhanced LDAP Referral Handling](#) (see page 152)

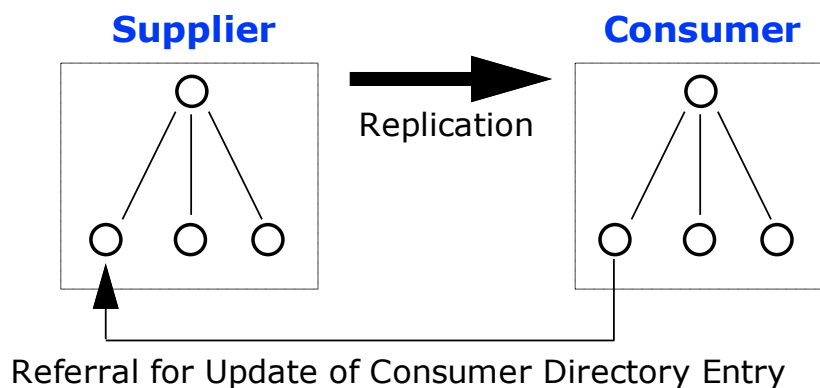
Directory Topology and LDAP Referrals

An LDAP directory's topology describes the division of a directory tree among physical servers. The logical sections of a directory tree are called partitions. LDAP directory topology varies widely between LDAP deployments, but regardless of the topology, the use of referrals between partitions allows the directory to function as a single service.

Three types of LDAP referrals can be employed in a directory topology. These types can be used in conjunction to create very complex directory structures. The types are:

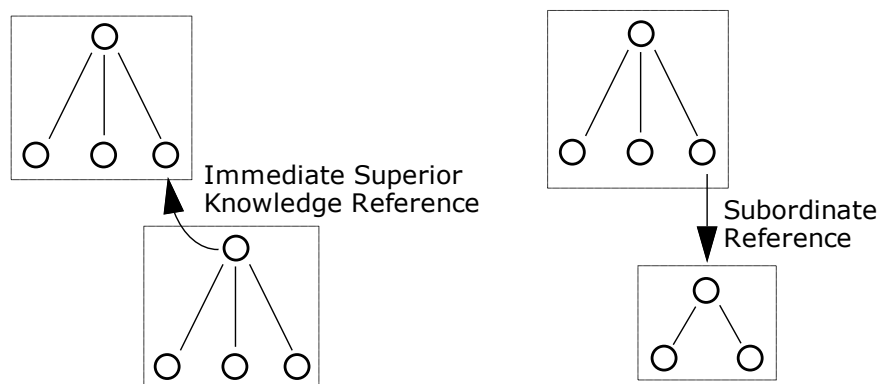
Replication Agreements

When a directory topology includes a replication agreement, all changes in a supplier directory are replicated (duplicated) in a second consumer directory. The consumer and supplier directories may be used to load balance requests, or may have a failover relationship. When an update request is received by the consumer directory, the consumer directory refers the request to the supplier directory where the update is completed. This is a very common type of LDAP referral.



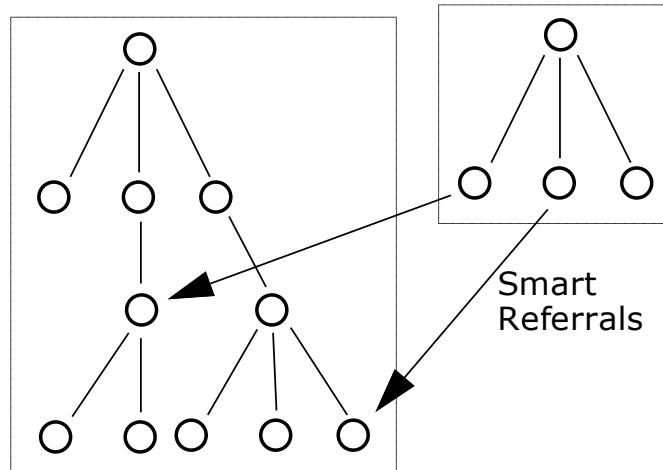
Knowledge References

Pointers from one directory partition to another are called knowledge references. Knowledge references that point to the node immediately upward toward the root in the DIT are considered *immediate superior knowledge references*. Knowledge references that point downward in the DIT to other partitions are considered *subordinate references*.



Smart Referrals

A pointer to a location in a portion of the directory that is not immediately above or below the original partition is called a *smart referral*. A smart referral contains enough information to see a node anywhere in the directory topology.



Enhanced LDAP Referral Handling

Enhancements have been made to the Policy Server's LDAP referral handling to improve performance and redundancy. Previous versions of the Policy Server supported automatic LDAP referral handling through the LDAP SDK layer. When an LDAP referral occurred, the LDAP SDK layer handled the execution of the request on the referred server without any interaction with the Policy Server.

SiteMinder includes support for non-automatic (enhanced) LDAP referral handling. With non-automatic referral handling, an LDAP referral is returned to the Policy Server rather than the LDAP SDK layer. The referral contains all of the information necessary to process the referral. The Policy Server can detect whether the LDAP directory specified in the referral is operational, and can terminate a request if the appropriate LDAP directory is not functioning. This feature addresses performance issues that arise when an LDAP referral to an offline system causes a constant increase in request latency. Such an increase can cause SiteMinder to become saturated with requests.

For example, a SiteMinder deployment includes two LDAP directories that are deployed in a supplier/consumer replication scheme with failover. All requests are made to the consumer directory. If the consumer directory is unavailable, the Policy Server uses the failover configuration to query the supplier directory.

If Password Services is enabled, an LDAP referral in the consumer directory takes place to ensure that password changes are written in the supplier directory. In previous versions of the Policy Server which only supported automatic LDAP referrals, if the supplier directory was unavailable, the Policy Server was unaware that the referral from the consumer directory required to write new password information into the supplier directory was not functioning, and would wait for a response from the supplier. This delay could cause the system to become saturated by pending requests.

If you configure your Policy Server with enhanced LDAP referral handling, the Policy Server is aware of the unavailable supplier LDAP directory and terminates requests that require password changes automatically, until the supplier directory is available to record password changes.

For information about configuring enhanced LDAP referral handling, see the *SiteMinder Policy Server Management Guide*.

How the Policy Server Binds to an LDAP User Store

The Policy Server opens three connections when connecting to an LDAP user store:

- The first connection verifies that the user store is up and running. By default, the Policy Server pings the user store every 30 seconds on this connection.
- The second connection is used for searches and updates. For example, the Policy Server uses this connection for user lookup and setting attributes on bind failures.
- The third connection is used for testing credentials. The Policy Server attempts to bind to the user store using the user's credentials. The result of the bind attempt identifies if the user's credentials are accepted or rejected.

ODBC Database Overview

SiteMinder can use a proprietary schema in an ODBC-compatible database as a user directory for authentication and authorization purposes. This option is useful at sites where user information, such as a user name, password, and group membership is stored in an ODBC database. At such sites, this feature allows the Policy Server to view a proprietary database schema as a user directory.

The Policy Server supports connections to the following types of ODBC-compatible databases:

- Microsoft SQL Server—Windows Policy Servers only
- Oracle RDBMS

Note: If your user directory data is stored in an Oracle database, we recommend that you use OCI, instead of ODBC, to connect to that user directory.

For the most current information about supported database versions, check the CA Support Site.

To configure the Policy Server to use a database as a user directory you must:

- Enter SQL query information in the SiteMinder SQL Query Scheme dialog. This dialog is accessible from the SiteMinder User Directory dialog. It contains fields for mapping information required by the Policy Server to fields in the ODBC-compatible database.

Note: Don't use the same data source with different query schemes. Create a unique data source for each query scheme.

- Define an ODBC data source that points to the database containing user information.

Note: Configure the ODBC data source as a System DSN. The data source must point to a Microsoft SQL Server database or an Oracle database. For information on configuring an ODBC data source, see your Windows operating system documentation.

More information:

[Configure a New SQL Query Scheme](#) (see page 197)

Windows Directory Overview

User directory connections can be set up with any of the following WinNT implementations:

- WinNT domain
- Individual WinNT computer in a domain
- Stand-alone WinNT computer

In order for the Policy Server to connect to your WinNT domain, it must meet the following requirements:

- A one way trust relationship between the domain containing the Policy Server, and the domain containing users, must exist.
- Every user in the domain that will be authenticated by SiteMinder needs the Access the computer from network user right for the machine where the policy server is running.
- The account that SiteMinder uses to access the User Directory object needs to have access to the IPC\$ share on the domain controller machine where the WinNT domain users are located.

Note: These requirements may be met by default if you use the default configuration for your WinNT domain. Your WinNT domain administrator should verify that the domain meets the above requirements.

The Policy Server authenticates against WinNT and can authorize users based on their individual identities and group membership.

When authenticating against a WinNT namespace, the Policy Server passes user credentials to WinNT for authentication. The credentials are the user's WinNT user name and password. In a SiteMinder environment, where multiple WinNT namespaces are defined, user authentication is faster if the user name supplied to SiteMinder includes the domain name (i.e. *domain\username*). In that case, SiteMinder skips all WinNT namespaces that do not match the specified domain name.

WinNT user names and passwords can be used as credentials.

Note: To authenticate users against a WinNT domain, the Policy Server must run on WinNT.

Active Directory Overview

SiteMinder supports user directories on the Microsoft Active Directory platform. Although the configuration for Active Directory (AD) and LDAP namespaces in the Policy Server User Interface is very similar, there are several functional differences.

The advantages of using the AD namespace when configuring an Active Directory user store include:

- SSL connectivity using a native Windows certificate database.
Note: Both the Policy Server and the systems hosting Active Directory user stores must have an established trust. For information about configuring Windows systems and Active Directory for SSL, see your Windows documentation.
- Support for native Windows SASL which allows for secure LDAP bind operations.

The disadvantages include:

- No support for enhanced LDAP referrals.
- No support for LDAP paging and sorting operations.

Note: For information about using the LDAP namespace for an Active Directory user store, see [Configure Active Directory Connections](#) (see page 171).

Custom Directory Overview

The Policy Server User Interface allows you to create custom user directory connections by creating shared libraries using the SiteMinder Directory API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information). Custom connections allow the Policy Server to interact with legacy directories. Once you create a directory connection using the SiteMinder APIs, you can configure a custom namespace in the User Directory Dialog.

Configure User Directory Connections

User directory connections allow the Policy Server to communicate with user stores in your enterprise. When configuring a user directory connection, some of the steps are identical regardless of the type of directory. The remainder of the requirements for directory connections depend on the type of directory.

To configure a user directory connection

1. Make sure that the Policy Server system can access the system that contains the directory or database.

Ping the user store system from the Policy Server system to verify a network connection.

Note: Some user stores may require the Policy Server to present credentials before allowing a connection.
2. Create a user directory object in the Policy Server User Interface using the User Directory Dialog.
3. In the User Directory Dialog, provide a Name and (optional) Description for the user directory object.

If you have multiple independent policy stores in your SiteMinder deployment, see [Define the Same User Directory Connection in Multiple Policy Stores](#) (see page 157).
4. See one of the following for information about configuring a connection to a specific type of user store:
 - [Configure CA, iPlanet, and IBM Directory Server LDAP Directory Connections](#) (see page 160)
 - [Domino Directory Connections Prerequisites](#) (see page 163)
 - [Configure Oracle Internet Directory Connections](#) (see page 164)
 - [Configuration of Novell eDirectory LDAP Directory Connections](#) (see page 166)
 - [Configure Active Directory Connections](#) (see page 171)
 - [Connect the Policy Server to an Active Directory Global Catalog](#) (see page 176)
 - [Configure an ADAM Directory Server](#) (see page 177)

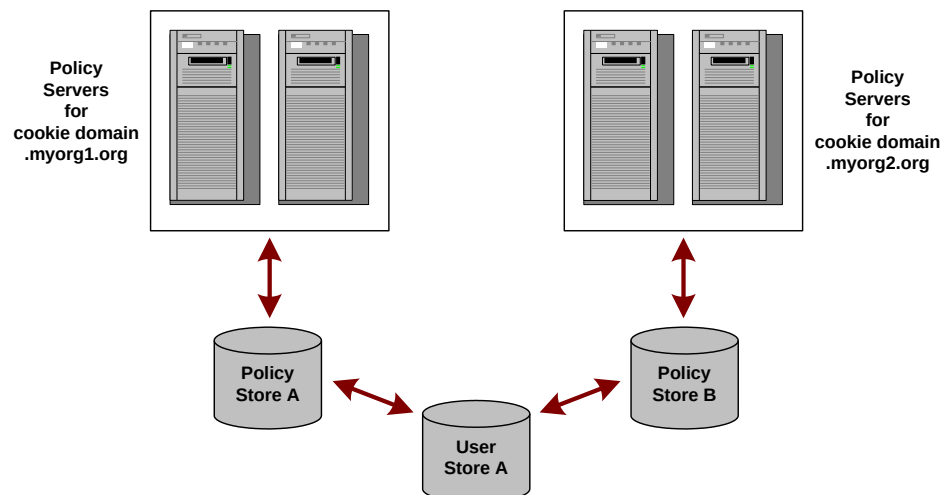
- [Configure ODBC Directory Connections](#) (see page 183)
- [Windows Directory Connections](#) (see page 204)
- [Configure Custom Directory Connections](#) (see page 209)

More information:

[User Directory Dialog](#) (see page 545)

Define the Same User Directory Connection in Multiple Policy Stores

Every Policy Server is connected to a policy store. Multiple Policy Servers may be configured to point to a single policy store. When you open an instance of the Policy Server User Interface, the objects that you add and modify are stored in the policy store associated with the Policy Server. As shown in the following figure, your SiteMinder environment may contain multiple independent policy stores for maintaining Policy Server data.



The Policy Servers for myorg1 are connected to Policy Store A. The Policy Servers for myorg2 are connected to Policy Store B. However, both organizations require data from User Store A.

To define a connection from multiple policy stores to a single user directory

1. Open a Policy Server User Interface associated with one of the policy stores in your SiteMinder deployment.
2. Configure a user directory connection.

When defining the user directory connection in the User Directory Dialog, note the value you supply in the Name field.

3. Open a Policy Server User Interface associated with another policy store in your SiteMinder deployment.
4. Configure the same user directory connection.

When defining the user directory connection in the User Directory Dialog, use the same Name that you used in Step 2.

For example, if you used a value of User Store A in the Name field when defining the user directory connection in the first policy store, to maintain single sign-on, you must configure the second policy store using a value of User Store A in the Name field of the User Directory Dialog.

5. Repeat this process for all independent policy stores in your SiteMinder deployment that will access the same user store.

If you use the same user directory name when defining the connections to the user store in each independent policy store, SiteMinder can maintain single sign-on for users who access resources protected by policies in the different policy stores.

More information:

[Configure User Directory Connections](#) (see page 156)

[User Directory Dialog](#) (see page 545)

Specify Directory Attributes

Some SiteMinder features require read or read/write access to directory attributes. This information must be stored in a user directory to which SiteMinder has access. As you configure user directory connections, you must provide the names of certain user profile attributes in the directories. You enter these attributes in the User Attributes tab of the User Directory Dialog.

The attributes you specify must have specific data types. Password Attribute and Password Data require binary attributes, and all others require string attributes.

The following figure lists the user directory profile attributes you must supply for certain SiteMinder features, specifies the types of user directories for which the attributes apply, and describes the SiteMinder features that require the attributes.

Note: The letter(s) in parentheses after the field name indicate whether the specified attribute requires read access (R) or read/write (RW) access.

SiteMinder Attribute Field	Directory Types	Description
Universal ID (R)	LDAP Database WinNT	Specifies the attribute that SiteMinder uses as the universal ID. The universal ID is a user identifier that can be passed to applications protected by SiteMinder to maintain a user's identity. This feature is used as a bridge between SiteMinder and legacy applications, which often use other attributes, to identify a user. The universal ID is also used in configuring Directory mapping.
Disabled Flag (RW)	LDAP Database	Specifies the attribute that SiteMinder uses to keep track of a user's account status. (See the <i>Policy Server Management Guide</i> .)
Password Attribute (RW)	LDAP Database	Specifies the attribute that contains the user's password. The attribute that you specify must correspond to the attribute in the directory that stores users' passwords.
Password Data (RW)	LDAP Database	Specifies the attribute that SiteMinder uses to track password policy information.
Anonymous ID (RW)	LDAP Database	Specifies the attribute that maintains the anonymous ID. For SiteMinder sites that allow anonymous login, users may access resources by authenticating with an anonymous authentication scheme. When a user logs in using the anonymous scheme, SiteMinder stores the specified DN in the attribute specified in the Anonymous ID field.
Email (R)	LDAP Database	This attribute is not currently used by a SiteMinder feature.
Challenge/Response (RW)	LDAP	Specifies the attribute that contains a question and answer pair, which is used by the Forgotten Password feature in Password Services. If a user forgets her password, SiteMinder uses the attribute specified in this field to determine the password hint that should be returned to the user.

Note: When configuring a user directory connection, you have the option to enter credentials that the Policy Server uses to access a directory. Make sure the credentials you enter have the same read/write access described in the previous table, otherwise the Policy Server will not be able to use the associated features.

For considerations when specifying user attributes according to the type of user store, see one of the following:

- [Configure CA, iPlanet, and IBM Directory Server LDAP Directory Connections](#) (see page 160)
- [Domino Directory Connections Prerequisites](#) (see page 163)
- [Configuration of Novell eDirectory LDAP Directory Connections](#) (see page 166)
- [Configure Active Directory Connections](#) (see page 171)
- [Configure ODBC Directory Connections](#) (see page 183)
- [Windows Directory Connections](#) (see page 204)
- Configure Custom Directory Connections (see page 209)

More information:

[Universal IDs](#) (see page 212)

[Password Policies](#) (see page 479)

[Anonymous Authentication Schemes](#) (see page 236)

[Directory Mappings Dialog](#) (see page 590)

Configure CA, Oracle, and IBM Directory Server LDAP Directory Connections

You can configure directory connections for CA, Oracle Directory Enterprise Edition (formerly Sun Java System Directory Server Enterprise Edition), and IBM Directory Server LDAP using the User Directory Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure directory connections

1. Open the User Directory Dialog.
2. In the Directory Setup tab, select LDAP from the Namespace drop-down list.
3. In the Directory Setup tab, enter connection information for your LDAP directory.
4. To configure the directory connection to include multiple servers for failover and load balancing, click the Configure button in the Directory Setup group box.

5. (Optional) In the Credentials and Connection tab, specify the administrator credentials that the Policy Server uses to connect to the LDAP directory, and specify whether the connection to the directory uses SSL.

Note: For Oracle Directory Enterprise Edition (formerly Sun Java System Directory Server Enterprise Edition) directories, we recommend using an administrator account other than cn=Directory Manager. Using cn=Directory Manager can cause performance issues due to security policies applied to this account. Instead, create a user with sufficient privileges to manage the directory and specify that user in the Connect Username field.

6. (Optional) In the User Attributes tab, specify the directory attributes which can be reserved for use by SiteMinder features.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

[Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)

[User Directory Dialog—LDAP Namespace—Directory Setup Tab](#) (see page 548)

[User Directory Dialog—LDAP Namespace—Credentials and Connection Tab](#) (see page 550)

[Specify Directory Attributes](#) (see page 158)

Enable User Store DSA Parameters

SiteMinder uses the Sun Java System LDAP SDK, which lets clients open one managed connection to the directory server and perform user binds under that connection. If you are using CA directory as a user store, the Policy Server connects to CA Directory by performing a bind request for each authentication request. Configure CA Directory to handle these requests, or CA Directory runs out of connections and authentication fails.

To enable user store DSA parameters

1. Open the .dxc file for the user store DSA.
2. Define the following at the bottom of the file:

```
#SiteMinder
set mimic-netscape-for-siteminder = true;
set concurrent-bind-user = DN;
set hold-ldap-connections = true;
```

3. Save and close the .dxc file.

The user store DSA parameters are enabled.

Note: The DN is in x500 format.

Example: <o acme><cn smadmin>

Enable Caching for a CA Directory User Store

You can improve SiteMinder authentication and authorization performance for large (for example, 5 MB) user stores by enabling the CA Directory DXcache feature

To enable caching

1. As the dsa user, edit the user store DSA's DXI file (for example, `<dxserver_install>\config\servers\CADsa.dxi`) and add the following lines to the end of the file:

```
# cache configuration
set max-cache-size = 100;
set cache-index = commonName, surname, objectClass;
set cache-attrs = all-attributes;
set cache-load-all = true;
set lookup-cache = true;
```

Note: The max-cache-size entry is the total cache size in MB. Adjust this value based on the total memory available on the CA Directory server and overall size of the user store. In addition, set the cache-index fields to those fields used by SiteMinder to perform a user search in the user store. For example, if users are authenticated and authorized based on their common name (cn=*), make sure that the commonName is set in the cache-index.

2. As the dsa user, stop and restart the user DSA to allow the DXcache configuration changes to take effect:

```
dxserver stop CADsa
dxserver start CADsa
```

Verify the CA Directory Cache Configuration

After configuring the CA Directory DXcache feature for the user store, you can verify that the cache is enabled using the DXmanager user interface.

To verify the cache

1. Using a Web browser, connect to the DXmanager Web interface.

For example:

```
http://<CA_host>:8080/dxmanager/ManagerServlet?hostgroup=All
```

2. Navigate to the DSA configuration page and verify that the DXcache Status field is set to Enabled for your policy store DSA.

Domino Directory Connections Prerequisites

In order to use SiteMinder with Domino LDAP, the following prerequisites must be met:

- In order for the Policy Server to recognize the groups in your Domino LDAP directory, groups must share the root DN you specify when configuring the directory in the Policy Server User Interface. For example, if you add a group to your address book (names.nsf) in Lotus Notes, and the group is called marketing/myorg.org, the root DN you enter in the Root field of the User Directory Dialog must be o=myorg.org.
- When defining users for Domino LDAP, it is recommended that you register new users. This additional step ensures consistency for user names. If a user has multiple entries in the User Name field of the Domino LDAP directory, the Policy Server uses the first entry to identify the user. If such a user tries to log in with a user name other than the first entry, the login fails. Note that users must have a value in the Internet Password field of the directory. When adding new users, be sure to provide an Internet password.

Note: For complete information about configuring Domino LDAP, see your Domino documentation.

Configure Domino Directory Connections

After meeting the prerequisites described in the previous section, you can configure the connections.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure Domino Directory connections

1. Open the User Directory Dialog.
2. In the Directory Setup tab, select LDAP from the Namespace drop-down list.
3. In the Directory Setup tab, enter connection information for your LDAP directory.

As you enter information in the Directory Setup tab, verify the following:

- The value in the Root field of the LDAP Search group box matches the organization name you assigned in Lotus Notes. The Root also includes a country if you specified a country in Lotus Notes. For example, if you have an organization called myorg, and it is located in the United states, the Root for this example must be entered as o=myorg,c=us.

- The values in the Start and End fields of the LDAP User DN Lookup group box form search strings using proper LDAP notation, rather than the shortened notation used in Lotus Notes. For information about search strings see LDAP Search Filters. For information about how the Start and End fields are used by the Policy Server to locate a user in the directory.
4. To configure the directory connection to include multiple servers for failover and load balancing, click the Configure button in the Directory Setup group box.
 5. (Optional) In the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the LDAP directory, and specify whether the connection to the directory will use SSL.

Note: For Oracle Directory Enterprise Edition (formerly Sun Java System Directory Server Enterprise Edition) servers, we recommend using an administrator account other than cn=Directory Manager. Using cn=Directory Manager may cause performance issues due to security policies applied to this account. Instead, create a new user with sufficient privileges to manage the directory and specify that user in the Connect Username field.

6. (Optional) In the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

[User Directory Dialog—LDAP Namespace—Directory Setup Tab](#) (see page 548)

[User Disambiguation in an LDAP Directory](#) (see page 143)

[Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)

[User Directory Dialog—LDAP Namespace—Credentials and Connection Tab](#) (see page 550)

[Specify Directory Attributes](#) (see page 158)

Configure Oracle Internet Directory Connections

After meeting the prerequisites, you can configure the Policy Server.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure the Policy Server to use an OID user directory

1. In OID:
 - a. Create another organizational unit (for example, OracleSchemaVersion) under a domain and follow the similar instructions from step 4 in the “Configuring the Policy Server to Use Oracle Internet Directory as a Policy Store” section in the “Configuring LDAP Directory Servers as a Policy Store” chapter of the *Policy Server Installation Guide*.
 - b. Enter a Distinguished Name. For example, **ou=people,cn=OracleSchemaVersion**.
2. To add users to the organizational unit under the same domain as mentioned in step 4 of the “Configuring the Policy Server to Use Oracle Internet Directory as a Policy Store” section in the “Configuring LDAP Directory Servers as a Policy Store” chapter of the *Policy Server Installation Guide*:
 - a. Repeat steps 1 and 2 from that section.
 - b. Instead of adding a domain, click Add and chose inetOrgPerson.
3. In the dialog box enter:
 - a. cn=user1
 - b. sn =user1
 - c. uid=user1
 - d. userpassword=user1
 - e. dn as cn=user1,ou=people,cn=OracleSchemaVersion
4. Open the User Directory Dialog.
5. In the Directory Setup tab, select LDAP from the Namespace drop-down list.
6. In the Directory Setup tab:
 - a. Enter connection information for your LDAP directory.
 - b. In the Root field, specify ou=people, cn=OracleSchemaVersion
7. To configure the directory connection to include multiple servers for failover and load balancing, click the Configure button in the Directory Setup group box.

8. Optionally, in the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the LDAP directory, and specify whether the connection to the directory will use SSL.

Note: For Oracle Directory Enterprise Edition (formerly Sun Java System Directory Server Enterprise Edition) directories, we recommend using an administrator account other than cn=Directory Manager. Using cn=Directory Manager may cause performance issues due to security policies applied to this account. Instead, create a new user with sufficient privileges to manage the directory and specify that user in the Connect Username field.

9. Optionally, in the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

[User Directory Dialog—LDAP Namespace—Directory Setup Tab](#) (see page 548)

[Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)

[User Directory Dialog—LDAP Namespace—Credentials and Connection Tab](#) (see page 550)

[Specify Directory Attributes](#) (see page 158)

Configuration of Novell eDirectory LDAP Server Connections

The following sections provide instructions for configuring user directory connections to a Novell eDirectory LDAP Server running on NetWare. Once you configure the Novell eDirectory as described in the following sections, you can configure a user directory connection in the Policy Server User Interface.

More information:

[Configure a Novell eDirectory LDAP Connection in Policy Server User Interface](#) (see page 169)

Assumptions for Novell eDirectory Configuration

The examples in the following sections assume the following:

- The Novell eDirectory contains an organization denoted by o=transpolar.
- The Novell eDirectory contains an administrator account called cn=admin.o=transpolar with a login name of admin.transpolar.

Configure NetWare

The goal of the configuration described in this section is to allow the Policy Server to log into the Novell eDirectory, view the contents of the directory, and retrieve directory attributes. For some advanced features of SiteMinder, you may also need to configure the Novell eDirectory to allow the Policy Server write-access to the directory.

If you installed LDAP as part of your Novell eDirectory installation, you should have a server in Novell eDirectory called LDAP Server and an LDAP group named LDAP Group. LDAP Server should be a member of the LDAP Group.

To create the LDAP Server and LDAP Group in Novell eDirectory

1. Create an LDAP Server in Novell eDirectory. (For this example, it is called LDAP Server.)
2. Create an LDAP Group in Novell eDirectory. (For this example, it is called LDAP Group.)
3. Assign LDAP Group to LDAP Server.

- a. In the NW Admin tool, right click on LDAP Server.

Note: If you are using the Netware ConsoleOne tool instead of the NW Admin tool to modify your Novell eDirectory, you must complete the same tasks using the tools available in ConsoleOne. The interface for the two tools is similar. See your Novell documentation for more information.

- b. From the popup menu, select Details.
- c. Enter LDAP Group in the LDAP Group field.
- d. Click OK.

More information:

[Specify Directory Attributes](#) (see page 158)

Configure Anonymous LDAP Access on Novell eDirectory

In order for the Policy Server to interact with an Novell eDirectory, you must create an account with enough administrative privileges to allow access to the directory.

The easiest configuration is to generate an anonymous user on the LDAP server and make this the proxy user. The user should be assigned enough power to perform all functions necessary for SiteMinder on the LDAP server.

The instructions below assign administrator privileges to an anonymous user, although you can configure the user with more limited privileges. The effect of this is that any anonymous access to the LDAP directory will gain the same privileges you give to SiteMinder.

To configure anonymous LDAP access

1. Create a user called LDAP_Anonymous.

The following procedure is an example which may differ based on your version of Novell products.

- a. From the menu bar of the NW Admin tool, select Object, Create, User.
- b. Add the name LDAP_Anonymous.
For example, use the login name LDAP_Anonymous.transpolar.
- c. Do not assign a password.
- d. In the right frame, select Security Equal To and add the admin user (e.g., Admin.transpolar).
- e. Click OK.

2. Set up a proxy account:

The following procedure is an example which may differ based on your version of Novell products.

- a. In the NW Admin tool, select LDAP Group.
- b. From the popup menu, select Details.
- c. Click Continue.
- d. In Proxy Username field, enter LDAP_Anonymous.transpolar.
- e. In right frame, select Access Control and click Add.
- f. In the LDAP ACL Name field, enter LDAP_Anonymous.
- g. Select the LDAP Distinguished Name check box and enter cn=LDAP_Anonymous.transpolar.
- h. Select the All Attributes and Object Rights check box.
- i. Click OK.
- j. In right frame, select Access Control and click Add.
- k. In box labeled LDAP ACL Name, enter Everyone.
- l. Select the Everything check box.
- m. Select the All Attributes and Object Rights check box.
- n. Click OK.
- o. Click OK.

To continue configuring your Novell eDirectory for use with the Policy Server, see Configure a Novell eDirectory LDAP Connection in Policy Server User Interface (see page 169).

More information:

[Configuration of Novell eDirectory LDAP Server Connections](#) (see page 166)

Special Access for the SiteMinder Administrator

The alternate instructions below allow special access only to the Policy Servers and may be more appropriate in some environments.

1. Create an Novell eDirectory user to represent the SiteMinder administrator (for example called SiteMinder_admin).
2. Give this user a password generated by the SiteMinder administrator which will be entered in the Policy Server User Interface.

Create a Novell eDirectory User Account for SiteMinder Administration

You can give the SiteMinder Administration a user account using the NW Admin tool.

To create a Novell eDirectory user account for SiteMinder administration

1. In the NW Admin tool, right click LDAP Group.
2. From the popup menu, select Details.
3. In the right panel, click Access Control.
4. Add an ACL.
5. Enter a name for the ACL, (e.g. SiteMinder).
6. In the Access By List screen, click Add.
7. In the Access By List panel, click LDAP Distinguished Name.
8. Enter cn=SiteMinder_admin.

By default, set the access level to Read, which is sufficient for SiteMinder's basic functions. Customers whose use active APIs or some of SiteMinder's advanced features (for example, Password Services, User Disablement, Registration Services) may require Write access.

Configure a Novell eDirectory LDAP Connection in Policy Server User Interface

The following procedure is an example which may differ based on your version of Novell products.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a Novell eDirectory LDAP connection

1. Log into the Policy Server User Interface.
2. Select System Configuration, User Directories, Create.
3. Enter a server name and optional description of your choice.
4. In the Namespace box, select LDAP.
5. In the Server field, enter the IP address of the NetWare machine on which the LDAP Server resides.
6. In the Search Root field, enter the root from which you want to search for users in your configuration.

For this example, o=transpolar.

Note: If you are using Novell eDirectory as a user directory, and your directory contains multiple organizations, you may need to leave the Root field blank in order for the Policy Server to search for users in multiple organizations.

7. In the User DN Lookup Start field, enter cn=.
8. In the User DN Lookup End field, enter ,o=transpolar.
9. If you have not configured Novell eDirectory for anonymous user access, select the Credentials and Connections tab.
10. Select the Require Credentials check box.
11. In the Connect User field, enter the SiteMinder_admin user name you created above.

With the assumed values of the administrator and organization chosen for this example, the user name should be in the following form:

cn=SiteMinder_admin,o=netegrity

12. In the Connect Password field, enter the password for the user.
13. Click Apply.
14. Click View Contents and you should see a list of groups, organizational units, and organizations defined in your configuration if the configuration is correct.
15. Click OK to close the Directory List Dialog.
16. Click OK to close the User Directory Dialog.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Configure Anonymous LDAP Access on Novell eDirectory](#) (see page 167)

Configure Active Directory Connections

Before you configure a connection to an Active Directory consider the following:

- For Windows deployments, SiteMinder establishes the Windows user context by passing the user's fully-qualified Windows id and password to IIS. SiteMinder obtains the fully qualified Windows id from the user's DN entry by concatenating the login name of the authenticated user with the first dc value found in the user's LDAP DN. The login name is usually the same value as the SM_USER header. For example, if the user DN is `iscn=name,cn=user_group,dc=server,dc=domain,dc=extension`, and the username is `user_name`, the resulting Windows id is `server\user_name`. IIS requires that the `user_name` be the same as the Windows user id and that `<server>` be the logon domain name.

- The AD namespace does not support multi-byte character sets. To use a multi-byte character set with Active Directory, configure your directory connection using the LDAP namespace.

Note: Regardless of the code page you are using, SiteMinder treats characters as they are defined in Unicode. Although your code page may reference a special character as single-byte, SiteMinder treats it as a multi-byte character if Unicode defines it as such.

- When authenticating against an AD namespace, the Policy Server binds to Active Directory using SASL. If a user's common name (CN) is different from the user's Windows logon name, the user can still authenticate even if the EnableSaslBind registry setting exists on the Policy Server machine.

The EnableSaslBind setting is a DWORD registry key that you can set to 0 or 1:

HKLM\Software\Netegrity\SiteMinder\CurrentVersion\Ds\LDAPProvider\EnableSaslBind

This setting disables or enables the SASL protocol while authenticating users. For example, if EnableSaslBind does not exist and you configure this setting to 1, the bind occurs with SASL. If EnableSaslBind exists and you configure this setting to 0, the bind occurs with Simple Authentication mechanism.

- In order for the Policy Server to identify the AD domain of an AD namespace, which is necessary to read account lock status, you must configure the LDAP search root of the user directory as the DN of the domain. If you set the LDAP search root to any other DN, the Policy Server is not able to identify the AD domain and is therefore unable to read the Windows lockout policy associated with the domain. This can lead users that are locked through the AD console to appear enabled when viewed in the Policy Server User Interface User Management dialog.

For example, if you have created five users through the AD console at DN `ou=People,dc=clearcase,dc=com` and locked two of those users. The SiteMinder User Management dialog will show locked users disabled only if you configure the LDAP search root as the DN of the AD domain (that is, `dc=clearcase,dc=com`). If you configure the LDAP search root as `ou=People,dc=clearcase,dc=com`, the locked users will incorrectly be shown as enabled.

- By default, SiteMinder re-prompts users for credentials when they are unauthorized due to being natively disabled in the directory server. This behavior does not occur for users stored in Active Directory. Rather, SiteMinder redirects natively disabled users to Password Services, even if Password services is not enabled for the authentication scheme protecting the resource. Create and enable `IgnoreDefaultRedirectOnADnativeDisabled` to prevent this Active Directory behavior.

IgnoreDefaultRedirectOnADnativeDisabled

Location:

`HKEY_LOCAL_MACHINE/SOFTWARE/Netegrity/Siteminder/CurrentVersion/Ds/LDAPProvider`

Values: 0 (disabled) or 1 (enabled)

Default: 0. If the registry key is disabled, the default behavior is in effect.

Note: If a password policy is in effect that specifies a redirect to Password Services, SiteMinder redirects the natively disabled users to Password Services regardless of the registry key's setting.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a connection to an Active Directory

1. Open the User Directory Dialog (see [Navigate to the User Directory Dialog](#) (see page 546)).
2. In the Directory Setup tab, select one of the following from the Namespace drop-down list:

- AD
- LDAP

Microsoft Active Directory is an LDAP-compliant user directory. You can configure your Active Directory connection using the AD namespace or the LDAP namespace.

3. In the Directory Setup tab, enter connection information for your Active Directory as described in [User Directory Dialog—AD Namespace Directory Setup Tab](#) (see page 567).

Note the following:

- The value in the Root field of the LDAP Search group box will typically have the following format:

`dc=<server>,dc=<domain>,dc=<extension>`

For example, `dc=server,dc=myorg,dc=org`

- For secure connections from the Policy Server to an AD namespace over SSL, you must specify the FQDN and port number in the Server field of the Directory Setup group box.

If you only specify the IP address, rather than the FQDN, the following error will appear in the logs: error 29, "User Directory Can Not be Contacted"

A report will also appear in the Windows Event Log stating the certificate does not match the name of the server.

4. To configure the directory connection to include multiple servers for failover and load balancing, click the Configure button in the Directory Setup group box. (See [Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)).
5. Optionally, in the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the Active Directory, and specify whether the connection to the directory will use SSL as described in [User Directory Dialog—AD Namespace Credentials and Connection Tab](#) (see page 569).

You can also specify whether or not the Policy Server should provide a Windows user security context. See [How a Windows User Security Context Is Obtained](#) for more information.

The administrator user name typically takes a form similar to the following:

`cn=<administrator>,cn=<administrator's group>,dc=<server>,dc=<organization>,dc=<com, net, etc.>`

6. Optionally, in the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features. See [Specify Directory Attributes](#) (see page 158).

More information:

[Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)
[Specify Directory Attributes](#) (see page 158)

Pros and Cons of Choosing an AD Namespace for an Active Directory Connection

SiteMinder supports user directories on the Microsoft Active Directory platform. Although the configuration for Active Directory (AD) and LDAP namespaces in the Policy Server User Interface is very similar, there are several functional differences.

The advantages of using the AD namespace when configuring an Active Directory user store include:

- SSL connectivity using a native Windows certificate database.

Note: Both the Policy Server and the systems hosting Active Directory user stores must have an established trust. For information about configuring Windows systems and Active Directory for SSL, see your Windows documentation.

- Support for native Windows SASL which allows for secure LDAP bind operations.

The disadvantages include:

- No support for enhanced LDAP referrals.
- No support for LDAP paging and sorting operations.

Note: For information about using the LDAP namespace for an Active Directory user store, see [User Directory Dialog—LDAP Namespace](#) (see page 547).

Pros and Cons of Choosing an LDAP Namespace for an Active Directory Connection

SiteMinder supports user directories on the Microsoft Active Directory platform. Although the configuring Active Directory (AD) and LDAP namespaces in the Policy Server User Interface is similar, there are several functional differences.

The advantages of using the LDAP namespace for an Active Directory user store include:

- Support for enhanced LDAP referrals.
- Support for LDAP paging and sorting.

The disadvantages include:

- No support for Windows SASL

The LDAP namespace does not support native Windows SASL, which allows native secure LDAP bind operations to support native Windows authentication methods such as NTLM.

- Indexing the Objectclass Attribute

Microsoft Active Directory uses a non-standard method for identifying object classes. As a result, the objectclass attribute in Active Directory is not indexed by default. This lack of indexing can cause the Policy Server User Interface to timeout when it searches through an Active Directory LDAP implementation that includes large numbers of users or groups.

For SiteMinder to run efficiently with an Active Directory user directory, you must index the objectClass attribute in Active Directory. For more information, see your Active Directory documentation.

- Active Directory and Password Services

Microsoft Active Directory requires an SSL connection to change stored user passwords. So that Password Services can work with Active Directory user directories, configure an SSL connection to any Active Directory LDAP user directory to which password policies are applied.

Additionally, define a specific Password Attribute, unicodePWD, to enable Password Services to work with Active Directory user directories.

Note: For complete information about configuring Microsoft Active Directory, see your Active Directory documentation.

- Using a Windows User Security Context

A SiteMinder Web Agent can run in a Windows user security context for accessing Web resources on IIS Web servers. So that SiteMinder can provide the Windows user security context, configure a session store and enable persistent sessions on a per realm basis. (See How SiteMinder Is Configured to Provide a Windows User Security Context.) Also enable this feature on the Credentials and Connection tab. (See [User Directory Dialog](#) (see page 545).)

- Installing a Certificate Authority

SiteMinder uses a Netscape LDAP SDK to communicate with LDAP directories. You might be required to install a Certificate Authority for your Active Directory instance. You can use a third-party certificate utility to manage your SSL certificates. One such utility is the Network Security Services (NSS) utility (version 3.2.2). You can download the respective file for your operating system from [Mozilla](#). Documentation for the utility is located on the Mozilla [NSS project page](#).

Complete the following using the utility:

- a. Create the cert7.db and key3.db database files.
- b. Add the root CA to the cert7.db database file and specify that the root CA can create server certifications.
- c. Add the server certification to the cert7.db database file.
- d. Validate that the certifications were added to the cert7.db database file.

Important! When using this tool on Windows Server 2003, specify the full path to the executable file.

■ To Configure a Certificate Database, select one of the following:

- In the SiteMinder Policy Server Management Console, specify the path to the Netscape Certificate Database file in the LDAP tab.

Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

- In the Netscape Certificate Database File field, specify the path to the file where the Netscape certificates are stored.

Example: C:\Program Files\Netscape\users\default\cert7.db

More information:

[LDAP Referrals](#) (see page 150)

Connect the Policy Server to an Active Directory Global Catalog

The Policy Server's user store supports the Global Catalog Support feature in Active Directory. However, since Global Catalog does not support writes to Active Directory, SiteMinder features that require writing to AD, such as Password Services, are not supported.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To connect the Policy Server to an Active Directory Global Catalog user store

1. In the Policy Server User Interface, select Edit, System Configuration, Create User Directory from the menu bar.
2. In the Directory Setup tab of the SiteMinder User Directory dialog, do the following:
 - a. In the Name field, enter the name of the user directory. This example uses adgc_user_dir.
 - b. Make sure LDAP is selected from the Namespace drop-down menu.
 - c. In the Server field, enter the IP Address and port number of the Active Directory Global Catalog. This example uses 172.25.135.180:3269.
 - d. In the Root field, enter the search base that covers all the domains in the global catalog. This example uses dc=com.
 - e. In the Start field, enter the starting LDAP user DN search criteria. This example uses (&cn=. Do not enter the period.

- f. In the End field, enter the ending LDAP user DN search criteria. This example uses `)(&objectclass=*)`. Do not enter the period.
 - g. Click the Credentials and Connections tab.
 3. In the Credentials and Connections tab, do the following:
 - a. Check Require Credentials.
 - b. In the Username field, enter the full DN of the Active Directory Global Catalog administrator. This example uses `cn=user1,cn=users,dc=universal,dc=com`.
 - c. Enter and reconfirm the password.
 - d. Check Secure Connection if you are using an SSL connection and go to the next section.

Configure an ADAM Directory Server

To create a user store in an ADAM Directory Server, see the following sections:

- [Create an ADAM Instance and User Store Connection](#) (see page 177)
- [Create a SiteMinder User For Connecting to the ADAM User Store](#) (see page 178)
- [Assign Administrator Privileges to the SiteMinder User](#) (see page 178)
- [Load Users into the ADAM User Store](#) (see page 179)
- [Configure the ADAM User Store in the Policy Server](#) (see page 179)

Create an ADAM Instance and User Store Connection

You use the Connection Settings dialog of ADAM ADSI Edit to create an ADAM instance and user store connection.

To create an ADAM instance and user store connection

1. Install an ADAM instance on a free port number. This example uses port 390.
2. Open the ADAM ADSI utility by going to Start, Program Files, ADAM, ADAM ADSI Edit.
3. Right-click ADAM ADSI Edit and select Connect to, which opens the Connections Setting dialog.
4. In the Connections Setting dialog:
 - a. Enter a name in the Connection name field. This example uses My Connection.
 - b. Select localhost in the Server name field.
 - c. Enter a free port number in the Port field. This example uses port 390.

- d. Select Distinguished name (DN) or naming context and enter O=userstore in the field.
- e. Click OK.

The connection (MyConnection [localhost:390] root element) you configured appears under ADAM ADSI Edit.

Create a SiteMinder User For Connecting to the ADAM User Store

You can create a SiteMinder user in the Connection Settings dialog shown in the previous topic.

To create a SiteMinder user for connecting to the ADAM user store

1. Under o=userstore, right-click CN=Roles and select New, Object.
2. In the Create Object dialog:
 - a. Select user and click Next.
 - b. In the Value field, enter a user name that SiteMinder uses to connect to the ADAM user store and click Next. This example uses admin.
 - c. Click Finish.
3. In the right column of ADAM ADSI Edit, right-click CN=admin and select Reset Password.
4. In the Reset Password dialog, enter and confirm the new password and click OK.

The SiteMinder user exists in ADAM and the next step is to give this user administrator privileges.

Assign Administrator Privileges to the SiteMinder User

You use ADAM ADSI Edit to assign administrator privileges to SiteMinder users.

To assign administrator privileges to the SiteMinder user

1. In the right column of ADAM ADSI Edit, right-click CN=Administrators and select Properties.
2. In the CN=Administrators Properties dialog, scroll down to the member attribute and click Edit.
3. Click Add ADAM Account.
4. In the Add ADAM Account dialog, add the SiteMinder user you created in Create a SiteMinder User For Connecting to the ADAM User Store.
5. Click OK.

This example uses CN=admin,CN=Role,O=userstore.

Load Users into the ADAM User Store

You use the ADAM Tools Command Prompt to load users into the ADAM user store.

To load users into the ADAM user store

1. Start the ADAM Tools Command Prompt by going to Start, Programs, ADAM, ADAM Tools Command Prompt.
2. Load the .ldif file that has the users for the store by running the following command:

```
ldifde -i -f <user_store>.ldif -s <IP_Address> -t <portnumber>
```

where <user_store>.ldif is the name of the file for your user store, <IP_Address> is the IP Address of the machine where ADAM is installed, and <portnumber> is the port number.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

Configure the ADAM User Store in the Policy Server

You use the SiteMinder User Directory dialog to set up the parameters necessary to connect to the ADAM user store.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure the ADAM user store in the Policy Server

1. In the Policy Server User Interface, select Edit, System Configuration, Create User Directory from the menu bar.

The system displays the SiteMinder User Directory dialog box.

2. In the Directory Setup tab of the SiteMinder User Directory dialog, do the following:
 - a. In the Name field, enter the name of the user directory. This example uses adam_user_store.
 - b. Make sure LDAP is selected from the Namespace drop-down menu.
 - c. In the Server field, enter the IP Address and port number of the ADAM user store instance. This example uses 172.26.13.128:390.
 - d. In the Root field, enter the root for the user store. This example uses O=userstore.

- e. In the Start field, enter the starting LDAP user DN search criteria. This example uses (CN=. Do not enter the period.
 - f. In the End field, enter the ending LDAP user DN search criteria. This example uses). Do not enter the period.
3. In the Credentials and Connections tab of the SiteMinder User Directory dialog, do the following:
 - a. Check Require Credentials.
 - b. In the Username field, enter the name of the SiteMinder administrator of the ADAM user store. This example uses CN=admin,CN=Roles,O=userstore.
 - c. Enter and reconfirm the password.
4. In the User Attributes tab of the SiteMinder User Directory dialog, do the following for Password Services:
 - a. In the Universal ID field, enter uid.
 - b. In the Disabled Flag field, enter carlicense.
 - c. In the Password Attribute field, enter userPassword.
 - d. In the Password Data field, enter audio.
 - e. Click OK.

Load Balancing/Failover Configuration for LDAP Directory Connections

The Policy Server can spread LDAP queries over multiple LDAP servers to enable failover and load balancing. If configured for failover, the Policy Server uses one LDAP server to fulfill requests until that server fails to respond. When the default server does not respond, the Policy Server routes the request to the next server specified for failover. This process can be repeated over multiple servers. Once the default server is able to fulfill requests again, the Policy Server routes requests to the original server.

If configured for load balancing, the Policy Server spreads requests over the specified LDAP servers. This distributes requests evenly across LDAP servers. Coupled with failover, load balancing provides faster, more efficient access to LDAP user directory information, with the added benefit of redundancy in the event of a server failure.

Port Numbers

In the SiteMinder LDAP Server dialog you can assign ports to individual LDAP servers and failover groups of servers, or let the Policy Server use default port numbers for LDAP servers.

The following guidelines apply when specifying port numbers:

If...	Then...
any server in a failover group other than the last (furthest to the right) server contains a port number	The Policy Server assumes all other servers in the group that do not have a specific port are using a default port (636 for SSL or 389 for non-SSL). For example, a failover group of servers includes the following: 123.123.12.12:350 123.123.34.34 Since the first server in the failover group includes port 350, communication with that server will take place on port 350. If the first server fails, the Policy Server will communicate with the second server using the default port 389, since no port was specified for the second server in the failover group.
you specify a port for the last server in a failover group of servers, but do not specify a port for any other servers in the group	The Policy Server uses the specified port for all servers in the failover group. For example, a failover group of servers includes the following: 123.123.12.12 123.123.23.23 123.123.34.34:350 All servers in the failover group use port 350.

More information:

[User Directory Dialog](#) (see page 545)

Add Additional Servers to the Setup for Load Balancing and Failover

The following procedure describes the process for adding additional servers to the setup for load balancing and failover.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add additional servers to the setup for load balancing and failover

1. In the Directory Setup group box of the User Directory Dialog, click Configure.
The LDAP Setup Dialog opens.
2. In the Add an IP Address or Name group box, select either IP Address or Domain Name.
3. If you selected the IP Address radio button, in the LDAP Server IP Address fields, enter the IP address for the LDAP user directory server.

If you do not know the IP address of the LDAP server, click the DNS Lookup button. Enter a host name in the DNS Lookup dialog and click OK. The DNS Lookup dialog closes and the IP address for the server appears in the LDAP Server IP Address fields in the SiteMinder LDAP Server dialog.
4. If you selected Domain Name, enter the domain name of the LDAP server in the Domain Name field.
5. In the Port field, enter a port number for the LDAP server.

If you do not specify a port number, the Policy Server uses the default port (636 for SSL and 389 for non-SSL).
6. Click either Add Load Balancing or Add Failover to add the IP address or domain name to the list in the Current Setup group box.
7. If you are adding another server for failover, select the IP address or server to which you want to add a failover LDAP server in the Current Setup group box.
8. Repeat steps 2 through 4 to enter another IP address or domain name.
9. Click Add Load Balancing or Add Failover.

Note: If you are adding a server for failover, the failover directory must use the same type of communication (SSL or non-SSL) as the primary directory, since both directories share the same port number.

The Policy Server User Interface displays the IP address or domain name in the Current Setup group box list below the existing server(s) for load balancing or beside the existing server(s) for failover.

Note: To remove an LDAP server (and any associated failover servers), select its IP address and click Remove.

Note: You can add the same server multiple times for load balancing to force more requests to be serviced by a specific system. For example, consider two servers in a group (Server1 and Server2). Server1 is a high-performance server and Server2 is a lesser system. You can add Server1 to the load balancing list twice so that it will process two requests for each request processed by Server2.

Configure Both Load Balancing and Failover for User Directories

To provide load balancing between two user directories that also failover to each other if one of the directories becomes unavailable, you must configure two failover groups. Each of the failover groups must be configured to load balance with the other group, since failover functionality is not a part of user directory load balancing.

For example, consider two user directories, A and B, that should be load balanced and also failover to each other. Where spaces imply failover and commas imply load balancing, groups should be setup as follows:

A B,B A

Note: This notation is similar to the notation used in the Server field of the User Directory dialog of the SiteMinder Admin UI, except that the letters representing the servers would be replaced by the IP addresses of the servers.

In the example, there are two failover groups: "A B" and "B A". These two groups are load balanced together. Therefore, if both directories are up, load balancing occurs between the first directories in each failover group, or A and B. If A becomes unavailable, failover occurs in the first group, and requests load balance between B and B (or always go to B) until A becomes available again.

Consider a similar example with 4 separate directories, A, B, C, and D, configured as follows:

A B,C D

In this case, requests are load balanced between A and C. If C becomes unavailable, requests are load balanced between A and D until C is once again available.

Configure ODBC Directory Connections

You can use the User Directory Dialog to configure a connection to your ODBC user directory.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Note: If you are using a SQL database for audit logs and caching is turned on (see the Policy Server Management Console chapter in the *SiteMinder Policy Server Management Guide*), under heavy load, SiteMinder performance may suffer as the Policy Server queues messages to be logged. To alleviate this situation, turn on asynchronous auditing for the realms associated with the resources being accessed by a high volume of users.

To configure an ODBC user directory connection

1. Open the User Directory Dialog.
2. In the Directory Setup tab, select ODBC from the Namespace drop-down list.
3. In the Directory Setup tab, enter connection information for your ODBC database directory.

One of the required fields in the Directory Setup tab, is the SQL Query Scheme field.

4. To configure the directory connection to include multiple servers for failover, click the Configure button in the Directory Setup group box.
5. Optionally, in the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the ODBC database.

The name you enter in the Username field must be the name of the user who owns the tables containing user directory data. For example, if you are using the SmSampleUsers schema, this user must be the owner of the SmUser, SmUserGroup and SmGroup tables.

Note: This administrator account must have read or read/write privileges for the user directory based on the features of SiteMinder included in your implementation.

6. Optionally, in the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features.

More information:

[Configure a Realm](#) (see page 320)

[Grouping Resources in Realms](#) (see page 315)

[Navigate to the User Directory Dialog](#) (see page 546)

[User Directory Dialog—ODBC Namespace—Directory Setup Tab](#) (see page 553)

[Configure SQL Query Schemes](#) (see page 197)

[Load Balancing/Failover Configuration for LDAP Directory Connections](#) (see page 180)

[User Directory Dialog—ODBC Namespace—Credentials and Connection Tab](#) (see page 555)

[Specify Directory Attributes](#) (see page 158)

Configure ODBC Data Sources for Failover

You can configure multiple ODBC data sources for failover. Once you configure the data sources for failover, if the primary data source fails to respond to a request, the next data source defined in the failover order responds to the request.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure failover for ODBC data sources

1. In the Directory Setup group box, click Configure.
The SiteMinder ODBC Directory Failover Setup dialog opens.
2. In the Add a Data Source group box, enter a data source name in the ODBC Data Source field.
3. Click Add Failover.
The Policy Server User Interface adds the data source name to the Failover Order list in the Current Setup group box.
4. Add any remaining data source names by repeating steps 2 through 3.
5. Click OK.
The Policy Server saves the list of ODBC data sources for failover. The data sources are displayed in a comma-delimited list in the Data Source field of the User Directory Dialog.

More information:

[ODBC Setup Dialog](#) (see page 574)

SQL Server User Store Case Insensitivity and Extra Trailing Spaces Password Issues

A SQL Server user store can be case insensitive and also ignore extra trailing spaces in users' passwords. This causes a problem because users entering passwords that are case insensitive or with extra trailing spaces can gain access to protected resources. For example, if a password policy is configured so that a user's password must be the case sensitive "ABCD", but the user enters "ABcd", SQL Server allows entry. In another example, if the password policy is configured so that a user's password must be " A B C", but the user enters " A B C ", SQL Server ignores the extra trailing spaces in the password and allows entry.

The following are solutions to these two issues.

First Issue: SQL Server User Store is Case Insensitive

The SQL Server database is not performing proper collation and does not recognize case sensitivity in passwords.

Solution 1

To impose case sensitivity to a SQL Server user store, select the proper collation during table creation when installing the database.

For more information about the collation, see the following:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/instsql/in_collation_30a6.asp

Solution 2

If you already installed SQL Server with the default collation and the database does not recognize case sensitivity in passwords, create the proper collation when creating tables for the SiteMinder user store.

To specify the collation, modify and then import one of the following scripts.

```
<siteminder_install>\db\SQL\smsampleusers_sqlserver.sql
```

```
<siteminder_install>\db\SQL\smsampleusers_sqlserver_upgrade.sql
```

Note: These two script files use the default US English localization.

smsampleusers_sqlserver.sql Script File

Change the following lines highlighted in bold in the smsampleusers_sqlserver.sql script file:

```
CREATE TABLE SmGroup (  
    GroupID          int NOT NULL,  
    Name nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL ,  
    PRIMARY KEY (GroupID)  
)  
DROP TABLE SmUser  
go  
CREATE TABLE SmUser (  
    UserID          int NOT NULL,  
    Name  nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL,  
    Password nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL,  
    LastName nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL,  
    FirstName nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL,  
    EmailAddress nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL,  
    TelephoneNumber nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL ,  
    Disabled nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL ,  
    PIN      nvarchar(255) COLLATE Latin1_General_CS_AS NOT NULL ,  
    Mileage int NOT NULL,  
    PasswordData varchar(2000) NOT NULL,  
    PRIMARY KEY (UserID)  
)  
go
```

After modifying the script, you need to import it into the SQL Server database.

Important! Back up any existing data as running this script removes existing data and creates new tables.

smsampleusers_sqlserver_upgrade.sql Script File

The following lines highlighted in bold are changes you need to make in the smsampleusers_sqlserver_upgrade.sql script file:

```
/* Upgrade table SmGroup*/
ALTER TABLE SmGroup ALTER COLUMN Name nvarchar(255) COLLATE Latin1_General_CS_AS NOT
NULL
go
/* Upgrade table SmUser*/
ALTER TABLE SmUser ALTER COLUMN Name nvarchar(255) COLLATE Latin1_General_CS_AS NOT
NULL
go
ALTER TABLE SmUser ALTER COLUMN Password nvarchar(255) COLLATE Latin1_General_CS_AS
NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN LastName nvarchar(255) COLLATE Latin1_General_CS_AS
NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN FirstName nvarchar(255) COLLATE Latin1_General_CS_AS
NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN EmailAddress nvarchar(255) COLLATE
Latin1_General_CS_AS NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN TelephoneNumber nvarchar(255) COLLATE
Latin1_General_CS_AS NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN Disabled nvarchar(255) COLLATE Latin1_General_CS_AS
NOT NULL
go
ALTER TABLE SmUser ALTER COLUMN PIN nvarchar(255) COLLATE Latin1_General_CS_AS NOT
NULL
go
```

After modifying the script, you need to import it into the SQL Server database.

Important! Back up any existing data as running this script removes existing data and creates new tables.

Second Issue: SQL Server Ignores Trailing Spaces in the Password

SQL Server pads the strings that are being compared and makes their lengths equal.

Solution

To have SQL Server recognize trailing white spaces in passwords stored in the database, modify the Authenticate User query in the ODBC Query Scheme object using the Policy Server User Interface. To have SQL Server compare strings without padding or trimming, incorporate the LIKE predicate instead of the = operator. When the right side of a LIKE predicate expression features a value with a trailing space, SQL Server does not pad the two values to the same length before the comparison occurs. An example authentication query is:

```
select Name from SmUser where Name = '%s' and Password LIKE '%s'
```

Important! Using the LIKE predicate expression in the password matching query can open a security hole. If a user enters a '%' in the password, SQL Server treats it as wild card character for the LIKE predicate and this user can authenticate using more than one password.

Note the following:

- If you are creating the ODBC query scheme object using the SDK, you can specify the authentication query using the pszQueryAuthenticateUser attribute of the Sm_PolicyApi_ODBCQueryScheme_t object.
- If you are creating the ODBC query scheme object using Perl Scripting Interface, you can specify the authentication query while calling the CreateODBCQueryScheme() API.

How to Configure an LDAP User Directory Connection over SSL

Configuring an LDAP user directory connection over SSL requires that you configure SiteMinder to use your certificate database files.

Complete the following steps to configure the connection over SSL:

1. Before you configure a connection over SSL.
2. Install the NSS utility.
3. Create the certificate database files.
4. Add the root Certificate Authority (CA) to the certificate database.
5. Add the server certificate to the certificate database.
6. List the certifications in the certificate database.
7. Configure the user directory connection for SSL.

8. Point the Policy Server to the certificate database.
9. Verify the SSL connection.

Before You Configure a Connection over SSL

Review the following before configuring an LDAP user directory connection over SSL:

- Ensure your directory server is SSL-enabled.

Note: For more information on configuring your directory server to communicate over SSL, refer to the vendor-specific documentation.

- SiteMinder uses a Netscape LDAP SDK to communicate with LDAP directories. As a result, SiteMinder requires that the database files be in a Netscape version file format (cert7.db).

Important! Do not use Microsoft Internet Explorer to install certificates into your cert7.db database file.

- A third-party certificate utility, which is compatible with Netscape, is required to manage your SSL certificates. We recommend the Mozilla® Network Security Services (NSS) utility, version 3.2.2.

Note: Version 3.2.2 is required to support the cert7.db format. Do not use later versions.

- (Active Directory) Considering the following:

- If the SiteMinder user directory connection was configured with the AD namespace, the following process does not apply. The AD namespace uses the native Windows certificate repository when establishing an SSL connection. When configuring the AD namespace to communicate over SSL:
 - Ensure that the SiteMinder user directory connection is configured for a secure connection. For more information, refer to [Configure the User Directory Connection for SSL](#) (see page 195).
 - On the machine hosting the Active Directory instance, ensure that the root CA certificate and the server certificate are added to the services' certificate store.

Note: For more information on configuring Active Directory to communicate over SSL, refer to the Microsoft documentation.

- If the SiteMinder user directory connection was configured with the LDAP namespace, complete the following process to configure the connection over SSL.

Install the NSS Utility

You install the NSS utility to manage your certificate database files.

Note: Install the utility on a system to which the Netscape Portable Runtime (NSPR) or the Policy Server is installed. Installing the utility to a system with either component ensures that the necessary DLLs or shared objects are available.

To install the NSS utility

1. Access the [Mozilla](#) NSS 3.2.2 FTP site.
2. Download the respective zip or tar for your operating system.
Note: A zip is not available for Windows Server 2003. Download the zip for Windows NT.
3. Extract the NSS utility to a temporary location on the system to which you are managing your certificate database files.

Create the Certificate Database Files

The Policy Server requires that the certificate database files be in the Netscape version file format (cert7.db). You may use the NSS utility to create the certificate database files.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the [NSS project page](#).

To create the certificate database files

1. From a command prompt, navigate to the bin directory in the location to which you extracted the NSS utility.

Example: C:\nss\bin

Note: Windows has a native certutil utility. Ensure you are working from the bin directory of the NSS utility or you may inadvertently run the Windows certutil utility.

2. Enter the following command:

```
certutil -N -d certificate_database_directory
```

-N

Creates the cert7.db, key3.db, and secmod.db certificate database files.

-d *certificate_database_directory*

Specifies the directory to which the NSS utility is to create the certificate database files.

Note: If the file path contains spaces, bracket the path in quotes.

The utility prompts for a password to encrypt the database key.

3. Enter and confirm the password.

NSS creates the required certificate database files:

- cert7.db
- key3.db
- secmod.db

Example: Create the Certificate Database Files

```
certutil -N -d C:\certdatabase
```

Add the Root Certificate Authority to the Certificate Database

You add the root Certificate Authority (CA) to make it available for communication over SSL.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the [NSS project page](#).

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

To add the root CA certificate to the certificate database

1. From a command prompt, navigate to the bin directory in the location to which you extracted the NSS utility.

Example: C:\nss\bin

Note: Windows has a native certutil utility. Ensure you are working from the bin directory of the NSS utility or you may inadvertently run the Windows certutil utility.

2. Run the following command to add the root CA to the database file:

```
certutil -A -n alias -t trust_arguments -i root_CA_path -d  
certificate_database_directory
```

-A

Adds a certificate to the certificate database.

-n *alias*

Specifies an alias for the certificate.

Note: If the alias contains spaces, bracket the alias with quotes.

-t *trust_arguments*

Specify the trust attributes to apply to the certificate when adding it to the certificate database. There are three available trust categories for each certificate, which are expressed in this order: "SSL, email, object signing". Specify the appropriate trust arguments so that the root CA is trusted to issue SSL certificates. In each category position, you may use zero or more of the following attribute arguments.

p

Valid peer.

P

Trusted peer. This argument implies p.

c

Valid CA.

T

Trusted CA to issue client certificates. This argument implies c.

C

Trusted CA to issue server certificates (SSL only). This argument implies c.

Important! This is a required argument for the SSL trust category.

u

Certificate can be used for authentication or signing.

-i *root_CA_path*

Specifies the path to the root CA file. Consider the following:

- The path must include the certificate name.
- Valid extensions for a certificate include .cert, .cer, and .pem.

Note: If the file path contains spaces, bracket the path in quotes.

-d *certificate_database_directory*

Specifies the path to the directory that contains the certificate database.

Note: If the file path contains spaces, bracket the path in quotes.

NSS adds the root CA to the certificate database.

Example: Adding a Root CA to the Certificate Database

```
certutil -A -n "My Root CA" -t "C,," -i C:\certificates\cacert.cer -d C:\certdatabase
```

Add the Server Certificate to the Certificate Database

You add the server certificate to the certificate database to make it available for communication over SSL.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the [NSS project page](#).

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

To add the server certificate to the certificate database

1. From a command prompt, navigate to the bin directory in the location to which you extracted the NSS utility.

Example: C:\nss\bin

Note: Windows has a native certutil utility. Ensure you are working from the bin directory of the NSS utility or you may inadvertently run the Windows certutil utility.

2. Run the following command to add the root certificate to the database file:

```
certutil -A -n alias -t trust_arguments -i server_certificate_path -d  
certificate_database_directory
```

-A

Adds a certificate to the certificate database.

-n *alias*

Specifies an alias for the certificate.

Note: If the alias contains spaces, bracket the alias with quotes.

-t *trust_arguments*

Specify the trust attributes to apply to the certificate when adding it to the certificate database. There are three available trust categories for each certificate, which are expressed in this order: "SSL, email, object signing". Specify the appropriate trust arguments so that the certificate is trusted. In each category position, you may use zero or more of the following attribute arguments:

p

Valid peer.

P

Trusted peer. This argument implies p.

Important! This is a required argument for the SSL trust category.

-i *server_certificate_path*

Specifies the path to the server certificate. Consider the following:

- The path must include the certificate name.
- Valid extensions for a certificate include .cert, .cer, and .pem.

Note: If the file path contains spaces, bracket the path in quotes.

-d *certificate_database_directory*

Specifies the path to the directory that contains the certificate database.

Note: If the file path contains spaces, bracket the path in quotes.

NSS adds the server certificate to the certificate database.

Example: Adding a Server Certificate to the Certificate Database

```
certutil -A -n "My Server Certificate" -t "P,," -i C:\certificates\servercert.cer -d C:\certdatabase
```

List the Certificates in the Certificate Database

You list the certifications to verify that they were added to the certificate database.

Note: The following procedure details the specific options and arguments to complete the task. For a complete list of the NSS utility options and arguments, refer to the Mozilla documentation on the [NSS project page](#).

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

To list the certifications in the certificate database

1. From a command prompt, navigate to the bin directory in the location to which you extracted the NSS utility.

Example: C:\nss\bin

Note: Windows has a native certutil utility. Ensure you are working from the bin directory of the NSS utility or you may inadvertently run the Windows certutil utility.

2. Run the following command:

```
certutil -L -d certificate_database_directory
```

-L

Lists all of the certificates in the certificate database.

-d *certificate_database_directory*

Specifies the path to the directory that contains the certificate database.

Note: If the file path contains spaces, bracket the path in quotes.

NSS displays the root CA alias, the server certificate alias, and the trust attributes you specified when adding the certificates to the certificate database.

Example: List the Certificates in the Certificate Database

```
certutil -L -d C:\certdatabase
```

Configure the User Directory Connection for SSL

You configure the user store connection to ensure that an SSL connection is used when the Policy Server and user store communicate.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure the user store connection for SSL

1. Log in to the Policy Server User Interface.
2. Click Infrastructure, Directory.
3. Click User Directory, Modify User Directory.

The Modify User Directory pane appears with a list of existing user directory connections.

4. Select the user directory connection you want, and click Select.

User directory settings appear.

5. Select the Secure Connection check-box, and click Submit.

The user directory connection is configured to communicate over SSL.

Point the Policy Server to the Certificate Database

You point the Policy Server to the certificate database to configure the Policy Server to communicate with the user directory over SSL.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To point the Policy Server to the certificate database

1. Start the Policy Server Management Console.

Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

2. Click the Data tab.
3. Enter the path to the Netscape certificate database file in the Netscape Certificate Database File field.

Example: C:\certdatabase\cert7.db

Note: The key3.db file must also be in the same directory as the cert7.db file.

4. Restart the Policy Server.

The Policy Server is configured to communicate with the user directory over SSL.

Verify the SSL Connection

You verify the SSL connection to ensure the user directory and the Policy Server are communicating over SSL.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To verify the SSL connection

1. Log in to the Policy Server User Interface.
2. Click Infrastructure, Directory.

3. Click User Directory, View User Directory.

The View User Directory pane appears with a list of existing user directory connections.

4. Select the connection you want, and click Select.

User directory settings appear.

5. Click View contents.

If SSL is properly configured, the Directory Content pane appears and lists the contents of the user directory.

Configure SQL Query Schemes

The Policy Server uses SQL Query Schemes to build queries that find user data in a relational database. You create and edit SQL Query Schemes using the SiteMinder SQL Query Scheme dialog.

Note: The “SM_” prefix in column names is reserved for additional special names required by SiteMinder. Column names in your user directory should not begin with the “SM_” prefix. Policy Server errors will occur during user lookups if this prefix appears in column names.

Configure a New SQL Query Scheme

You can configure a new SQL Query Scheme by using the SQL Query Scheme Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a New SQL Query Scheme

1. Open the SQL Query Scheme Dialog.
2. Update the contents of the query fields to correspond to your database schema.

Each of these queries listed in the table above must be configured to work with your relational database. You must replace the following database table and column names with the table and column names from your relational database:

- Name

When you configure the Policy Server to use a user store residing in a relational database, the Name parameter for a user must be unique so that SiteMinder can correctly identify the user. Thus, two users cannot have the same user name.

- SmUser—table

- SmGroup—table
- Password
- SmUserGroup—table
- Id
- UserID
- FirstName
- LastName
- TelephoneNumber
- EmailAddress
- Mileage
- PIN
- GroupID
- Disabled

Note: Once you configure a SQL query scheme you can associate it with a user directory connection.

3. Select a query type:

Use Literal Query

This is the default query type.

Use Bind Query

Replace each instance of '%s' with a question mark (?) to use bind-based parameters in the query. Any instance not changed should remain as '%s'.

Restrictions:

- Advanced Password Services (APS) does not support bind-based parameters.
- A query uses bind-based parameters if it contains a question mark (?) and literal parameters if it contains '%s'.
- A query cannot contain '%s' and question marks (?) together.

4. Click OK.
5. If you specify a new SQL query scheme for a user directory connection, restart the Policy Server using the Policy Server Management Console.

More information:

[SQL Query Scheme Dialog](#) (see page 576)

[Configure ODBC Directory Connections](#) (see page 183)

Edit an Existing SQL Query Scheme

Another feature of the SQL Query Scheme Dialog is its ability to edit existing schemes.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To edit an existing SQL Query Scheme

1. Open the SQL Query Scheme Dialog.
2. Update the contents of the query fields to correspond to your database schema.
3. If you specify a new SQL query scheme for a user directory connection, restart the Policy Server using the Policy Server Management Console.

More information:

[SQL Query Scheme Dialog](#) (see page 576)

[Configure a New SQL Query Scheme](#) (see page 197)

Select an SQL Query Scheme for an ODBC User Directory Connection

You can select an SQL Query Scheme using the User Directory Dialog.

To select an SQL Query Scheme

1. Open the User Directory Dialog for an existing user directory connection object.
2. Select the Directory Setup tab.
3. From the SQL Query Scheme drop-down list, select the SQL query scheme you want to associate with the user directory connection.
4. Click OK to save your selections and close the User Directory Dialog.
5. Restart the Policy Server using the Policy Server Management Console.

For information on the Console, see the *SiteMinder Policy Server Management Guide*.

Note: If you are using MS SQL Server, and your queries are returning names that include the apostrophe character (for example, O'Neil), you must replace any instance of 's' in the query strings to "%s". To avoid this problem, base your queries on user IDs that do not include apostrophes, or modify the query strings that include 's'.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

Configure SQL Query Schemes for Authentication via Stored Procedures

When stored procedures are required for authentication with ODBC user directories, configure the SQL query scheme to call the stored procedure as follows:

SQLServer

Syntax: Call *Procedure_Name* %s , %s

Example: Call EncryptPW %s , %s

Stored procedures in SQLServer must meet the following requirements:

- The first parameter must be the username, and the second parameter must be the password.
- All parameters must be defined using the keyword OUT.
- The stored procedure must return an integer value.

The following example shows how to create a stored procedure for a SQLServer user directory:

```
CREATE PROCEDURE EncryptPW
@UserName varchar(20) OUT ,
@PW varchar(20) OUT
AS
SELECT Smuser.name from Smuser where Smuser.name= @UserName and password = @PW
SELECT Smuser.password from Smuser where name= @UserName and password = @PW
return 0
```

MySQL

Syntax: Call *Procedure_Name* %s, %s

Example: Call EncryptPW %s, %s

Stored procedures in MySQL must meet the following requirements:

- The first parameter must be the username, and the second parameter must be the password.
- The stored procedure does not return a value.

The following example shows how to create a stored procedure for a MySQL user directory:

```
CREATE PROCEDURE EncryptPW(INOUT p_UserName varchar(20), INOUT p_PW varchar(20))
BEGIN
SELECT SmUser.Name into p_UserName from test.SmUser where SmUser.Name =
p_UserName and SmUser.Password = p_PW;
SELECT SmUser.Password into p_PW from test.SmUser where SmUser.Name =
p_UserName and SmUser.Password = p_PW;
END;
```

Oracle Functions

For Oracle user directories, you can create the following functions using the templates below:

- EncryptPW
- ChangePW

EncryptPW Function

The EncryptPW function must return an integer value, as follows:

- value = 0
Specifies success.
- value = 1
Specifies failure.

You can use the following template to create the EncryptPW function:

```
CREATE OR REPLACE FUNCTION EncryptPW(p_UserName IN OUT SmUser.Name%type, p_PW IN OUT
SmUser.Password%type)
RETURN INTEGER IS
nRet INTEGER :=1;
nCount NUMBER := 0;
BEGIN
select count(*) into nCount
from SmUser
where SmUser.Name = p_UserName and SmUser.Password = p_PW;
IF (nCount = 1) THEN
SELECT SmUser.Name into p_UserName
from SmUser
where SmUser.Name = p_UserName and SmUser.Password = p_PW;
SELECT SmUser.Password into p_PW
from SmUser
where SmUser.Name = p_UserName and SmUser.Password = p_PW;
```

```
RETURN 0;  
END IF;  
RETURN nRet;  
END EncryptPW;
```

ChangePW Function

The ChangePW function must return an integer value, as follows:

- value = 1
Specifies success.
- value = 0
Specifies failure.

You can use the following template to create the ChangePW function:

```
CREATE OR REPLACE FUNCTION ChangePW(p_PW IN SmUser.Password%type, p_UserName IN  
SmUser.Name%type)  
RETURN INTEGER IS  
nRet INTEGER :=1;  
nCount NUMBER := 0;  
BEGIN  
select count(*) into nCount  
from SmUser  
where SmUser.Name = p_UserName;  
IF (nCount = 1) THEN  
UPDATE SmUser  
SET SmUser.Password = p_PW  
where SmUser.Name = p_UserName;  
COMMIT;  
RETURN 0;  
END IF;
```

Asynchronous Call Support During Failover and Connection Pooling

Synchronous calls are reliable, returning only after the request is complete. Asynchronous calls return immediately. A caller can choose to abandon an asynchronous call and avoid delays associated with network failures.

Asynchronous Call Support Configuration

The following registry options are stored under the registry sub-key
Netegrity\SiteMinder\CurrentVersion\Database

Option	Legitimate Values	Default
AsynchronousCalls	0 = no 1 = yes	0
Determines whether or not database calls will be made asynchronously		
AsynchronousSleepTime	0 to n milliseconds	15 milliseconds
The amount of time between calls to wait before checking the status of an outstanding SQL call		
LoginTimeout	minimum 1 (second)	15
SQL Timeout: The amount of time to allow for a connection to log in to the database		
QueryTimeout	minimum 1 (second)	15
SQL Timeout: The amount of time to allow for a query to complete before canceling it.		

Note: When SQL Server is running on Windows NT, asynchronous call support causes a very small memory leak per abandoned connection. You may choose to extend the timeouts to reduce the number of fail-overs in an unreliable network by adjusting the settings discussed in the table above.

ODBC Connection Pooling

- Connections are pooled by ODBC data source. If a data source is used by one or more user directories AND (the policy store or another store on the policy server), the number of connections available may reach the total of the individual values specified for user directories and stores.
- SQL Server is more limited than Oracle in how connections may be shared. Two callers may not share an open result set while the results are being fetched to the client. Although the Policy Server uses server side cursors for SQL Server, there are limitations to concurrent activity, especially on multi-processor machines. Increasing the number of connections will generally improve concurrency.
- Oracle allows for multiple callers to share a connection, but may serialize calls internally. Again, you may see improved concurrency by increasing the number of connections allowed.
- If connections are shareable, the number of active requests will be balanced across the connections in a pool.
- Once a connection is opened, it is not closed until the Policy Server is shut down.

Windows Directory Connections

You can use the User Directory Dialog to configure a connection to your Windows domain. User directory connections can be set up with any of the following WinNT implementations:

- WinNT domain
- Individual WinNT computer in a domain
- Stand-alone WinNT computer

WinNT Domain Connection Requirements

In order for the Policy Server to connect to your WinNT domain, it must meet the following requirements:

- A one way trust relationship between the domain containing the Policy Server, and the domain containing users, must exist.
- Every user in the domain that will be authenticated by SiteMinder needs the Access the computer from network user right for the machine where the policy server is running.
- The account that SiteMinder uses to access the User Directory object needs to have access to the IPC\$ share on the domain controller machine where the WinNT domain users are located.

Note: These requirements may be met by default if you use the default configuration for your WinNT domain. Your WinNT domain administrator should verify that the domain meets the above requirements.

- For Windows deployments, SiteMinder establishes the Windows user context by passing the user's fully-qualified Windows id and password to IIS. SiteMinder obtains the fully qualified Windows id from the user's DN entry by concatenating the login name of the authenticated user with the first dc value found in the user's LDAP DN. The login name is usually the same value as the SM_USER header. For example, if the user DN is *iscn=name,cn=user_group,dc=server,dc=domain,dc=extension*, and the username is *user_name*, the resulting Windows id is *server\user_name*. IIS requires that the *user_name* be the same as the Windows user id and that <server> be the logon domain name.

The Policy Server authenticates against WinNT and can authorize users based on their individual identities and group membership.

When authenticating against a WinNT namespace, the Policy Server passes user credentials to WinNT for authentication. The credentials are the user's WinNT user name and password. In a SiteMinder environment, where multiple WinNT namespaces are defined, user authentication is faster if the user name supplied to SiteMinder includes the domain name (i.e. *domain\username*). In that case, SiteMinder skips all WinNT namespaces that do not match the specified domain name.

WinNT user names and passwords can be used as credentials.

Note: To authenticate users against a WinNT domain, the Policy Server must run on WinNT.

Configure a Windows Directory Connection

After meeting the prerequisites described in the previous section, you can configure the connection.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a Windows directory connection

1. Open the User Directory Dialog.
2. In the Directory Setup tab, select WinNT from the Namespace drop-down list.
3. In the Directory Setup tab, enter connection information for your Windows domain.

In the NT Domain Name field, enter the name of the WinNT domain, computer within the WinNT domain, or stand-alone WinNT computer:

WinNT domains

To define a namespace that represents global users and groups in a WinNT domain, specify the name of the domain in the NT Domain Name field. For user accounts in domains, SiteMinder supports both domain authentication as well as trusted domain authentication. In order for domain authentication to work, the system on which the Policy Server is installed must have a computer account in the appropriate domain. If this system does not have a computer account in all domains in which users need to be authenticated, the appropriate trust relationships must be established between domains.

You can change the NT account under which the Policy Server is running. To do so, open the Settings, Control Panel, Services dialog, and change the account under which the Policy Server is running to an account that has the necessary privileges to access the specified domain. This account must have the Act As Part of Operating System system privilege.

Individual WinNT computer in a domain

To define a Namespace that represents local users and groups in a computer that is a member of a domain, specify the name of the domain, followed by the name of the computer in the NT Domain Name field. The domain name and computer name must be separated by a forward slash (/). If you do not specify the name of the computer, performance during searches may suffer.

For example, if a domain called SampleDomain contains a computer called Comp1 which contains user information, you can enter the following in the NT Domain Name field:

SampleDomain/Comp1

Stand-alone WinNT computer

To define a namespace that represents local users and groups in a stand-alone computer, specify the name of the computer in the NT Domain Name field. The stand-alone computer must have a Policy Server installed on it in order for this namespace to be accessible. There may be an initial delay when the Policy Server accesses this namespace for the first time.

4. (Optional) In the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the Windows domain.

Note the following:

- Most user directories do not require credentials. The Policy Server runs as a system service on WinNT. By default a system service will not require credentials unless your network setup has been configured for greater security.

If you cannot access the user directory after supplying credentials, your network may require that you run the Policy Server from a privileged account.

- If you do specify an account, it must have read or read/write privileges for the user directory based on the SiteMinder features included in your implementation.
- If required by your Windows configuration, in the Connect Username field, enter the administrative username.

If your network requires credentials, enter the username for an account that has administrative privileges for the WinNT user directory. This account is used by SiteMinder to enable or disable users, and to access trusted domains. This account does not need to be the default administrator account, but it must be an account with the appropriate privileges.

If you specified a computer name in the NT Domain Name field, you must enter the computer name, followed by a backslash (\), followed by the name of the user account that has appropriate privileges.

For example, if you are entering credentials for a user called jsmith on a computer called Comp1, you enter the following:

Comp1\jsmith

For information about WinNT account privileges, see your Windows NT documentation.

- (Optional) Select Run in Authenticated User's Security Context to enable SiteMinder to provide a Windows user security context to Web applications that run on IIS Web servers and are protected by SiteMinder.

A SiteMinder Web Agent can provide a Windows user security context for accessing Web resources in IIS Web servers. In the Credentials and Connection tab, you can specify whether or not SiteMinder should provide a Windows user security context. This enables the IIS server to enforce Windows access control mechanisms.

- Before SiteMinder can provide the Windows user security context, you must configure a session store and enable persistent sessions on a per realm basis.
5. (Optional) In the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

[User Directory Dialog—WinNT Namespace—Directory Setup Tab](#) (see page 558)

[User Directory Dialog—WinNT Namespace—Directory Credentials and Connection Tab](#) (see page 559)

[Specify Directory Attributes](#) (see page 158)

[How SiteMinder Is Configured to Provide a Windows User Security Context](#) (see page 96)

LanMan User Directories

A LanMan user directory connection is an alternative method for distributing directory requests in a WinNT environment. The LanMan user directory connection option allows you to specify a failover list of Backup Domain Controllers (BDCs) used for each user directory lookup in the WinNT Registry. This list of BDCs takes precedence over the Primary Domain Controller (PDC), which handles directory requests in most WinNT deployments. Using a LanMan directory connection, the Policy Server sends WinNT directory requests to the first active BDC in the Registry list, rather than forcing requests to pass through the PDC.

More information:

[LanMan User Directories](#) (see page 933)

Failover for WinNT User Directories

Windows NT LanManager automatically provides failover for WinNT user directories by defining a list Primary Domain Controllers (PDCs) and Backup Domain Controllers (BDCs). SiteMinder does not contain additional features for WinNT directory failover. To configure failover, use Windows NT LanManager. For information about LanManager, see your Windows NT documentation.

WinNT Attributes in Responses

For the purpose of delivering information back to a Web application via responses, SiteMinder has access to the following default user attributes:

- Description
- FullName
- AccountExpirationDate
- UserFlags
- LoginWorkstations
- BadPasswordAttempts
- MaxLogins
- MaxStorage
- PasswordExpired
- LastLogin
- LastLogoff
- HomeDirectory
- Profile
- LoginScriptMinPasswordLength
- MaxPasswordAge
- MinPasswordAge
- PasswordHistoryLength

These user attributes can be accessed through the User Attribute feature in a Response dialog. A response can return the value of any of these attributes to a SiteMinder Agent. For Web implementations, SiteMinder returns these attributes as name/value pairs in HTTP header variables.

More information:

[Responses and Response Groups](#) (see page 353)

Configure Custom Directory Connections

You can use the User Directory Dialog to configure a connection to your custom user directory.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a custom user directory connection

1. Open the User Directory Dialog.
2. In the Directory Setup tab, select Custom from the Namespace drop-down list.
3. In the Directory Setup tab, enter connection information for your custom database directory.
4. (Optional) In the Credentials and Connection tab, specify administrator credentials that the Policy Server will use to connect to the custom directory.
5. (Optional) In the User Attributes tab, specify directory attributes that will be reserved for use by SiteMinder features.

You must use the Apply button to save your custom user directory before configuring user attributes. When you apply your changes, the Policy Server uses the shared library you specified in the Library field of the Directory Setup tab to determine which fields in the User Attributes tab are available for the custom directory.

More information:

[Navigate to the User Directory Dialog](#) (see page 546)

[User Directory Dialog—Custom Namespace—Directory Setup Tab](#) (see page 562)

[User Directory Dialog—Custom Namespace—Credentials and Connection Tab](#) (see page 563)

[Specify Directory Attributes](#) (see page 158)

Access the SiteMinder SQL Query Scheme Dialog

There are two methods for accessing the SiteMinder SQL Query Scheme dialog: from the SiteMinder Administration window and from the User Directory Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access SQL Query Schemes from the SiteMinder Administration window

1. From the menu bar of the SiteMinder Administration window, select View, SQL Query Schemes.

The Policy Server places a check mark in the View menu to the left of the menu selection and SQL Query Schemes appear in the System tab of the SiteMinder Administration window.

Note: If there is already a check next to SQL Query Schemes in the View menu, selecting the option again will remove SQL Query Schemes from the System tab.

2. Create, edit, or delete SQL Query Schemes.

Note: If you delete an SQL Query Scheme required by an existing user directory connection, the Policy Server will not be able to authenticate users against the relational database.

3. Restart the Policy Server using the Policy Server Management Console.

For information on the Console, see the *SiteMinder Policy Server Management Guide*.

To access SQL Query Schemes from the User Directory Dialog:

1. In the SQL Query Scheme group box, select a scheme from the drop-down list and click Configure, or click New.

The SiteMinder SQL Query Scheme dialog opens. If you are editing an existing scheme, the queries listed in the dialog reflect the current settings being used by the Policy Server to access the user directory. If you are creating a new SQL Query Scheme, the queries are samples that must be replaced with queries that will function with your database.

2. Make changes to the queries and click OK to save your changes.
3. Restart the Policy Server using the Policy Server Management Console.

For information on the Console, see the *SiteMinder Policy Server Management Guide*.

More information:

[Work with Policy Server Objects](#) (see page 75)

View User Directory Contents

The Policy Server User Interface allows you to view the contents of a user directory. By default you can only view user groups, not individual users. To view individuals, use a user directory search.

To view the contents of a user directory

1. Open the User Directory Dialog for an existing user directory connection.

Note: If you want to view the contents of a directory for which you have just entered new information, be sure to click Apply and save your changes, otherwise the Policy Server will not be able to connect to the user directory with the most recent settings. In the case of a previously unsaved user directory connection, the Policy Server will not be able to display any user directory information before you save your changes.

2. Click View Contents.

The Directory List Dialog opens and displays a list of user groups.

More information:

[Search Directories](#) (see page 211)

[Navigate to the User Directory Dialog](#) (see page 546)

[Directory List Dialog](#) (see page 582)

Search Directories

The Policy Server User Interface contains a user directory search feature. User directory searches allow you to view users or groups of users based on search expressions or directory attributes. User directory searches vary for each type of user directory, however they can be accessed from the User Directory Dialog.

Note: You can also access user directory searches from the Policy Users/Groups Dialog. You use this dialog when adding or removing users and groups in a policy. The Policy Users/Groups Dialog contains the same search icon used to access a user search as described below.

To search a user directory

1. Open the User Directory Dialog for the directory that you want to search.
2. Click View Contents.

The Directory List Dialog opens.

3. Click the Search button.

A search dialog opens. The contents of the dialog change based on the type of directory you selected.

See one of the following based on the directory type:

- [Search LDAP/AD Dialog](#) (see page 584)
- [Search Database Dialog](#) (see page 586)
- [Search WinNT Directory Dialog](#) (see page 587)

4. Enter search criteria.
5. Click OK.

The search window closes and the Policy Server User Interface displays the search results in the Directory List Dialog.

More information:

[Users/Groups Dialog](#) (see page 912)

[Directory List Dialog](#) (see page 582)

Universal IDs

A Universal ID (UID) is a customer-specific user identifier to any application that is under SiteMinder control. In many cases UIDs are different from user login names.

UIDs allow SiteMinder to bridge the gap between new applications and legacy applications, or to avoid changes in underlying user repositories. The goal is to make the process of delivering this ID to applications automatic, regardless of the number or type of application. Since the Policy Server uses a login name to identify a user in a directory, the UID provides another means for the Policy Server to identify the user. The UID is collected from a user directory for use by other applications.

When you configure a user directory connection in the Policy Server User Interface, you can specify a UID in the User Attributes tab of the User Directory Dialog.

More information:

[Configure User Directory Connections](#) (see page 156)

How SiteMinder Uses UIDs

When you configure a user directory connection with a UID, once a user logs into SiteMinder, the Policy Server fetches the UID from the designated attribute in the user's directory profile. This value is placed in the session ticket (SESSIONSPEC) and returned to the requesting SiteMinder Agent. Web Agents make this value available to web-based applications in a header variable (HTTP_SM_UNIVERSALID). This value can be passed to applications or objects designed using the Agent API to validate the session ticket or to ask for an authorization. In either case the UID is returned as part of successful outcome.

Chapter 8: Directory Mapping

This section contains the following topics:

- [Directory Mapping Overview](#) (see page 213)
- [Configure a Directory Mapping](#) (see page 215)
- [Delete a Directory Mapping](#) (see page 217)
- [Configure Authentication and Validation Directory Mapping](#) (see page 217)
- [Directory Mapping Examples](#) (see page 219)
- [Directory Mapping and Responses](#) (see page 222)

Directory Mapping Overview

SiteMinder assumes that a user will be authenticated and authorized against the same user directory. Although this default behavior is sufficient in many cases, SiteMinder also provides the ability to authenticate users against one directory, and authorize users against a separate directory. This feature is called directory mapping. It is especially useful when authentication information is stored in a central directory, but authorization information is distributed in separate user directories that are associated with particular network applications.

Note: Impersonation is not supported by directory mapping. The impersonatee, the user being impersonated, must be uniquely present in the authentication directories associated with the domain or the impersonation fails.

Mapping from an authentication directory to an authorization directory is a three-step process.

1. Set Up User Directory Connections

Directory connections you want to specify as authentication or authorization directories in a mapping must be configured using the User Directory Dialog.

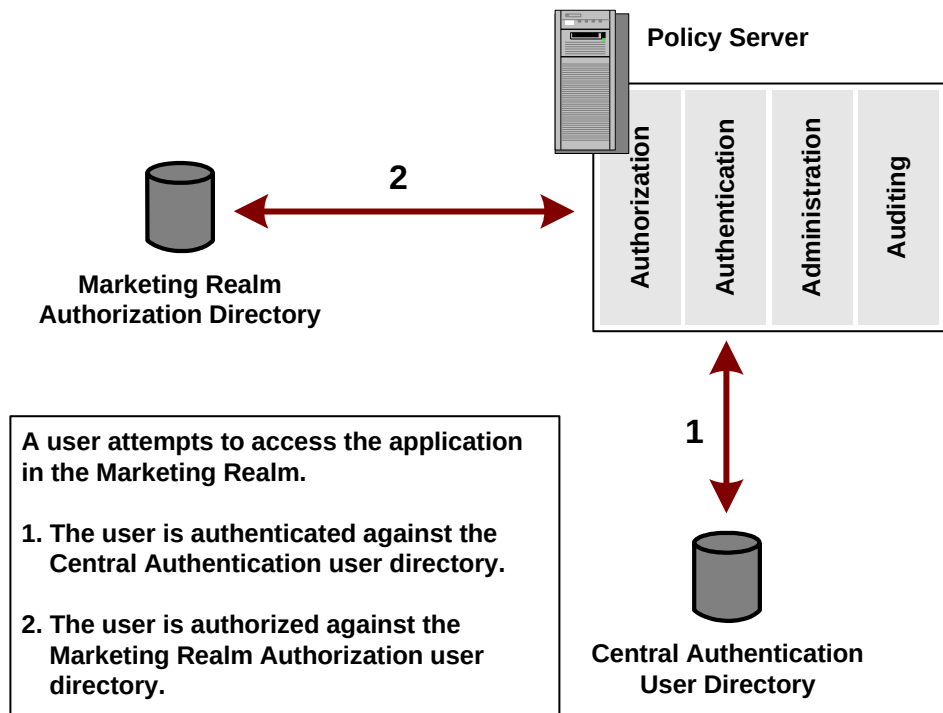
2. Configure a Directory Mapping

The Policy Server uses directory mappings to locate authenticated users in separate authorization directories.

3. Assign a Directory Mapping to a Realm

By associating a directory mapping with a specific realm, you can define the directory against which a user will be authorized for specific resources in a network.

For example, in the following diagram, all of the users in a company are authenticated against a single central user directory, but the marketing organization has a separate user directory that contains authorization data for Marketing staff. Using the Policy Server, you can configure a directory mapping to the Marketing authorization user directory, then you can create a realm for the Marketing application that uses the authorization directory specified in the mapping. Whenever a user tries to access the Marketing application, the Policy Server authenticates the user against the central user directory, but authorizes the user against the Marketing user directory.



More information:

[Configure User Directory Connections](#) (see page 156)

[Configure a Directory Mapping](#) (see page 215)

[Grouping Resources in Realms](#) (see page 315)

Configure a Directory Mapping

In order to configure a directory mapping, the user directory connections for both the authentication and authorization directories must be configured in the Policy Server User Interface. For information on configuring a user directory connection, see one of the following:

- [Configure User Directory Connections](#) (see page 156)
- [Configure SQL Query Schemes](#) (see page 197)
- [Windows Directory Connections](#) (see page 204)
- [Configure Custom Directory Connections](#) (see page 209)

The following table describes supported types of directory mapping, and the method that can be used to map the authentication directory to the authorization directory.

Authorization Directory			
Authentication Directory	LDAP	Relational Database	WinNT
LDAP	Identical DN Universal ID	Universal ID	N/A
AD	Identical DN Universal ID	Universal ID	N/A
Relational Database	Universal ID	Identical DN Universal ID	N/A
WinNT	Universal ID	Universal ID	Identical DN

To configure a directory mapping

1. From the menu bar of the SiteMinder Administration window, select Advanced, Directory Mapping.
The Directory Mappings dialog opens.
2. To add a new directory mapping, click Add.
The Add Directory Mapping dialog opens.
3. From the Authentication Directory drop-down list, select the directory against which users will be authenticated.
4. From the Authorization Directory drop-down list, select the directory against which users will be authorized.

5. In the Mapped DNs group box, select one of the following radio buttons:

Identical DN

Maps the user's distinguished name (DN) exactly from the authentication directory to the authorization directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the authorization directory. Therefore, the authorization directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the authorization directory to identify the user.

6. Click OK.

The Policy Server saves your changes and closes the Add Directory Mapping dialog. The new directory mapping appears in the Directory Mapping dialog.

7. Click OK to close the Directory Mapping dialog.

More information:

[Universal IDs](#) (see page 212)

Assign an Authorization Directory to a Realm

The Policy Server assumes a user is authenticated and authorized in a single user directory. In order for the Policy Server to authenticate a user in one directory and authorize a user in another directory, you must assign a directory mapping to specific realms. Once a directory mapping is assigned to a realm, the Policy Server uses the authorization directory specified in the realm to authorize users.

To assign a directory mapping to a realm

1. Open the SiteMinder Realm dialog for the realm to which you want to assign a directory mapping.
2. In the Advanced group box, click on the Directory Mapping tab.
3. From the drop-down list, select the user directory that the realm should use for authorizing an authenticated user.

A value of Default indicates that there is no directory mapping; the authentication directory will be used as the authorization directory when a user attempts to access a resource in the realm.

Note: This list contains only those user directories that have been configured as authorization directories in an existing directory mapping.

4. Click OK.

The Policy Server saves your changes and closes the SiteMinder Realm dialog. Users that try to access the realm will authenticate normally. Once they are authenticated, the existence of the authorization directory for the realm forces the Policy Server to search for an appropriate directory mapping. The Policy Server looks for a directory mapping that uses the authorization directory indicated by the realm and the user directory against which the user is authenticated.

More information:

[Configure a Realm](#) (see page 320)

Delete a Directory Mapping

You can delete a directory mapping on the Directory Mappings dialog. There are two ways to navigate to the Directory Mappings dialog. One way is to select Advanced on the menu bar and Directory Mapping... on the drop-down list. Another way is to select the Advanced tab on the Realm Properties dialog and to click Edit Mappings... on the Advanced tab.

To delete a directory mapping

1. Navigate to the Directory Mappings dialog.

The Directory Mappings dialog lists the current directory mappings.

2. Select a directory mapping and click Delete.

Important! Deleting a Directory Mapping that is in use may leave some Realms and Policies in an inconsistent state.

3. Click Yes to continue.
4. Click OK to delete the selected directory mapping.
5. Click OK to close the Directory Mappings dialog.

Configure Authentication and Validation Directory Mapping

Before you can configure an AuthValidate directory mapping, you must configure connections from the Policy Server to the authentication and validation user directories in the Policy Server User Interface. When you configure the AuthValidate directory mapping, you must satisfy the following conditions:

- The authentication and validation user directories must each be associated with a different policy store. When the two user directories are associated with the same policy store, AuthValidate mapping fails.

- The authentication and validation user directories must have different names. When they have the same name, AuthValidate mapping fails.
- The policy store associated with the validation user directory must not contain any user directory with the same name as the authentication user directory. If the authentication user directory's name is the same as any user directory in the policy store associated with the validation user directory, AuthValidate mapping fails.

In the following table, you can see the supported directory mapping types and methods:

Authentication Directory	Validation Directory		
	LDAP	Relational Database	WinNT
LDAP	Identical DN Universal ID	Universal ID	N/A
AD	Identical DN Universal ID	Universal ID	N/A
Relational Database	Universal ID	Identical DN Universal ID	N/A
WinNT	Universal ID	Universal ID	Identical DN

To configure a directory mapping

1. From the menu bar of the SiteMinder Administration window, select Advanced, AuthValidate Dir Mapping.

The AuthValidate Directory Mappings dialog opens.

2. Click Add.

The Add AuthValidate Directory Mapping dialog opens.

3. Enter the directory against which users are authenticated in the Authentication Directory field.

Select the directory against which users are validated from the Validation Directory drop-down list.

4. In the Mapped DNs group box, select one of the following option buttons:

Identical DN

Maps the user's distinguished name (DN) exactly from the authentication directory to the validation directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the validation directory. Therefore, the authorization directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the validation directory to identify the user.

5. Click OK.

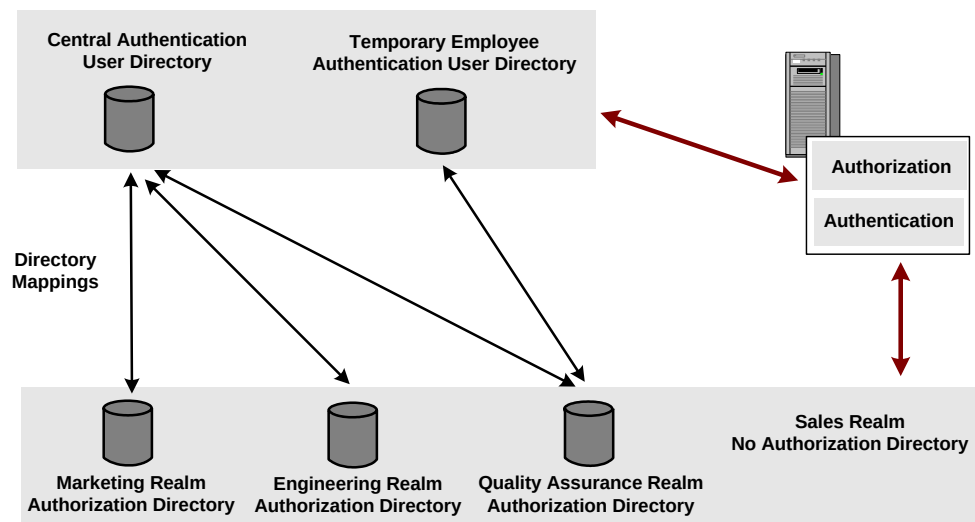
The Policy Server saves your changes and closes the Add AuthValidate Directory Mapping dialog. The new directory mapping appears in the AuthValidate Directory Mappings dialog.

6. Click OK

The AuthValidate Directory Mappings dialog closes and your mapping is saved.

Directory Mapping Examples

Multiple directory mappings may be needed in order to correctly authenticate and authorize users that have access to network resources. The following figure illustrates a simple sample case where multiple directory mappings may be required.



In the example, there are three realms that have separate authorization user directories and one realm that does not have its own authorization user directory. User authentication information is maintained in two distinct authentication directories. The Policy Server uses one of the authentication directories based on where the user's information is stored and one of the authorization directories if the requested resource is in the Marketing, Engineering, or Quality Assurance realms.

More information:

[Grouping Resources in Realms](#) (see page 315)

Employee Accesses an Engineering Realm Resource

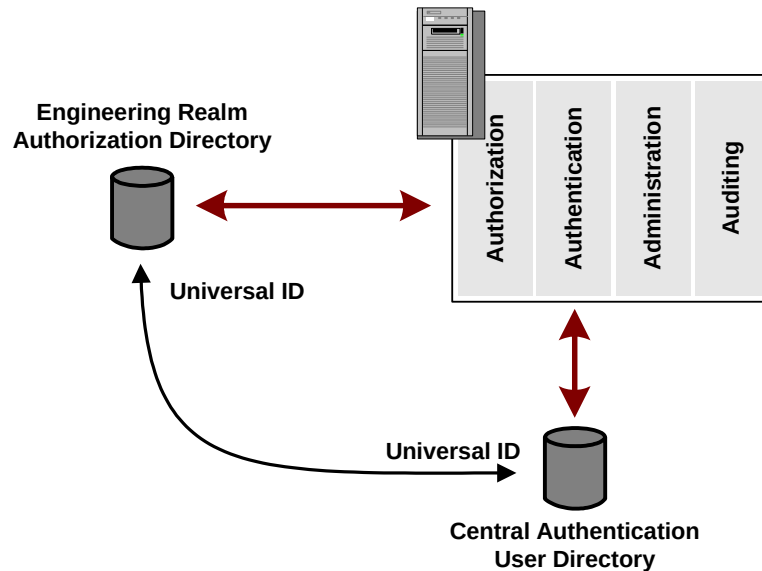
In the SiteMinder protected network described in the previous figure, a regular employee's authentication data resides in the Central Authentication user directory, and the employee attempts to access a resource in the Engineering Realm. When the employee is properly authenticated, the Policy Server recognizes that the Engineering realm uses its own authorization directory. The Policy Server looks for the directory mapping between the Central Authentication user directory and the Engineering Realm authorization user directory, then maps the user's identity to the authorization directory. Once this is done, the Policy Server can verify if the employee has access to the requested realm.

Temporary Employee Accesses a Quality Assurance Realm Resource

In the SiteMinder protected network described in the previous figure, a temporary employee's authentication data resides in the Temporary Employee Authentication user directory, and the employee attempts to access a resource in the Sales Realm. The Sales Realm is not associated with its own authorization directory. SiteMinder authenticates the employee in the Temporary Employee Authentication directory, then checks to see if a directory mapping is set up. Since the Sales Realm does not have its own authorization directory, SiteMinder attempts to authorize the employee using information in the same directory where the employee was authenticated.

Directory Mapping by Universal ID

In the SiteMinder protected network described in the previous figure, an employee's authentication data resides in the Central Authentication user directory, and the employee attempts to access a resource in the Engineering Realm. When the employee is properly authenticated, the Policy Server uses a directory mapping to find the employee in the Engineering Authorization directory, based on the employee's Universal ID (see the following figure).



In the diagram above, assume the directory mapping uses a Universal ID. The Policy Server uses the attribute in the authentication directory that is defined as the universal ID to find a matching universal ID in the authorization directory. Once the universal ID is located in the authorization directory, the SiteMinder can finish processing policies to determine whether or not the user can access a protected resource.

Directory Mapping Case Sensitivity

Case-sensitive directories, such as an Oracle database, treat the values "ROBIN" and "robin" as two different usernames. Other directories, such as an LDAP directory, are not case-sensitive and treat the values "Robin", "ROBIN", "robin", and "RobIn" as the same username. This can be a problem when a user is authenticated using a directory that is not case-sensitive, but authorized using a directory that is case-sensitive.

When authentication fails because the authentication directory is case-sensitive, the user can recover by reentering the username in the format required by the directory. If the directory requires the username to be in the format "Name", for example, the user can reenter the name correctly as "Robin". When authorization fails because the authorization directory is case-sensitive, however, the Policy Server has no way to recover.

When the authorization directory is case-sensitive, you can change the format of the authenticated username, so that it matches the format required by the authorization directory. If the authenticated username is "RoBiN", but the authorization directory requires the username to be in the format "Name", you can first change "RoBiN" to "Robin" and then authorize the user.

Directory Mapping and Responses

SiteMinder allows you to configure responses that collect the information in user directory attributes. If you use directory mappings, you must consider the effects that the mappings will have on some responses. For example, the directory mapping described in the diagram above would retrieve values from the Central Employee Authentication directory for an OnAuth event and from the Engineering Realm Authorization directory for an OnAccess event. For a description of events associated with responses, see [Responses and Response Groups](#) (see page 353).

Chapter 9: Policy Server Administrators

This section contains the following topics:

[Administrator Concepts](#) (see page 223)

[Administrator Scope and Tasks](#) (see page 225)

[Configure an Administrator](#) (see page 226)

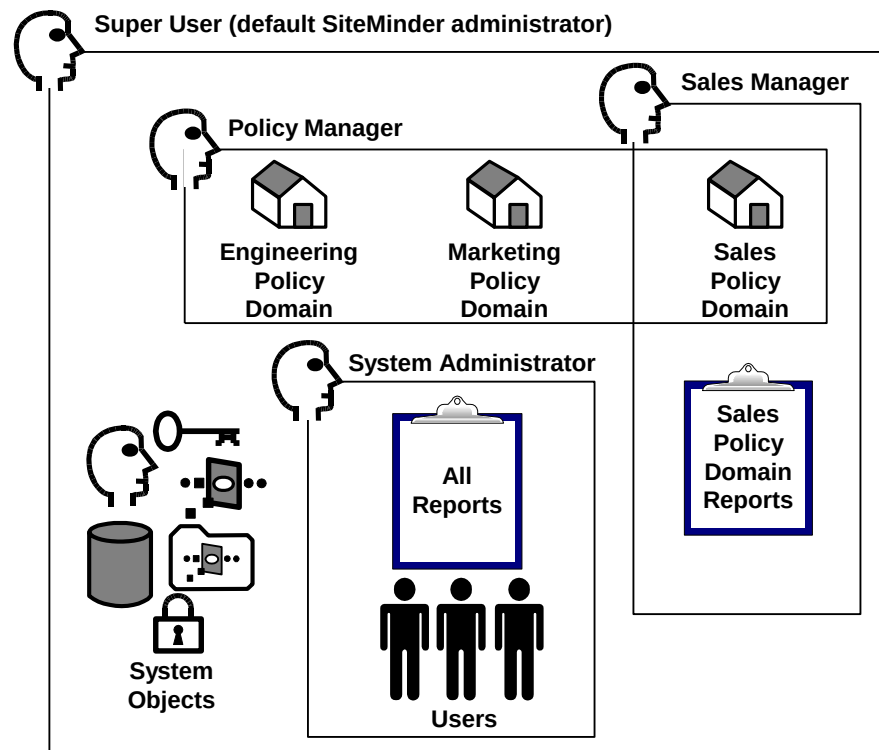
[Precautions on Deleting or Editing an Administrator](#) (see page 229)

Administrator Concepts

Everyone who has access to Policy Server objects and tools is considered an administrator. Depending on their role in your organization, Policy Server administrators have access to different network resources and Policy Server features, and are responsible for different tasks.

When you install the Policy Server, the installation sets up a default administrator account automatically. This account has maximum privileges, and with it you can create additional administrator accounts for those who need to add or make changes to Policy Server objects.

The following diagram illustrates the types of administrator privileges associated with certain job responsibilities.



In the previous figure, certain administrative responsibilities overlap. For example, the Policy Manager and the Sales Manager may both make changes to the Sales policy domain and the objects (rules, policies, responses, etc.) contained in the policy domain. The Super User may access all objects, manage users, and view all reports. You can create administrators and assign privileges to the administrators to match the administrative roles that exist in your organization.

Administrator Scope and Tasks

Administrators have a scope (system objects and policy domains or selected policy domains) and tasks that determine their privileges. These privileges allow administrators to use a set of Policy Server features.

The following table describes the privileges associated with each combination of administrator scope and task.

Administrator		
Scope	Tasks	Administrative Privilege
System	Manage System & Domain Objects	■ Create/edit/delete Agents, Agent Configuration Objects, Agent groups, Agent types, Host Configuration Objects, user directories, policy domains, authentication schemes, directory mappings, certificate mappings, registration schemes, and SQL Query Schemes. ■ Note: You cannot create or edit Trusted Host objects with this privilege, only delete them. To register Trusted Hosts, you must have Register Trusted Host privilege. ■ Create/delete parent realms in all domains. ■ Create/edit/delete administrators. ■ Flush all caches, including cached resources. ■ Change global settings. ■ All privileges for Manage Domain Objects listed below.
Domains	Manage Domain Objects	■ In managed domains: create/edit/delete rules, rule groups, responses, response groups, policies. ■ Edit top-level realms in managed domains (not resource filters). ■ Create/edit/delete nested realms in managed domains. ■ Flush specific realms from the resource cache, and flush all resources (in privileged domains) from the cache.
System	Manage Keys and Password Policies	■ Create/edit/delete password policies. ■ Manage keys.
Domains	Manage Password Policies	■ Create/edit/delete password policies for users in directories attached to managed domains.

Administrator

Scope	Tasks	Administrative Privilege
System	Manage Users	■ Flush all user session caches, or flush the user session cache of any individual user cache from any directory. ■ Enable/disable users in any directory. ■ Force password change on any user in any directory.
Domains	Manage Users	■ Flush user session caches for individual users in directories attached to managed domains. ■ Enable/disable users in directories attached to managed domains. ■ Force password change on users in directories attached to managed domains.
System	Register Trusted Hosts	■ Register Trusted Hosts

Configure an Administrator

When you create an administrator in the Policy Server User Interface, you must then configure it through the following four procedures.

To configure an administrator

1. Identify the administrator
2. Specify how that administrator is authenticated when they log in
3. Identify the scope of an administrator's privileges
4. Select the individual tasks the administrator is allowed to complete.

Note: An administrator, Administrator A, cannot create another administrator, Administrator B, with more privileges than Administrator A.

Configure Administrator Identity and Authentication Details

You can use the Policy Server User Interface to configure the identity of the administrator.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure an administrator

1. Using the Policy Server User Interface, create or open an existing administrator object for editing in the Administrator Properties dialog.
2. Provide a Name and, optionally, a Description of the administrator.
3. In the Administrator Lookup group box, specify whether the administrator is to authenticate against the SiteMinder Database (policy store) or against an External Directory.
4. Provide details specific to the administrator's lookup method:
 - If the administrator is to authenticate against the SiteMinder Database, provide a Password for that user and then Confirm that password in the Password Group box.
 - If the administrator is to authenticate against an External Directory, select user directory and authentication scheme in the Authenticate group box.
5. Proceed to configure the scope of the administrator's privileges to manage system and/or domain objects and the specific tasks that they can perform.

More information:

[Administrator Scope and Tasks](#) (see page 225)

Administrator Scope and Tasks

The Scope specified for an administrator determines the range of Policy Server objects they can create, edit, and delete:

System Administrator

A system administrator can manage all system objects and domain objects in all policy domains

Domain Administrator

A domain administrator can only manage objects in one or more specified policy domains.

An administrator's scope also determines the range of tasks that they can be configured to perform on those tasks.

Configure a System Administrator

A system administrator is an administrator that can manage all system and domain objects.

To configure a system administrator

1. Complete the steps outlined in [Configure Administrator Identity and Authentication Details](#) (see page 226).
2. In the Scope group box, select the System radio button.
3. In the Tasks group box, select one or more of the following check boxes to assign privileges to the administrator:
 - Manage System and Domain Objects
 - Manage Users
 - Manage Keys and Password Policies
 - View Reports
 - Register Trusted Hosts

More information:

[Configure Administrator Identity and Authentication Details](#) (see page 226)

Configure a Domain Administrator

A domain administrator can only manage objects in one or more specified policy domains.

To configure a domain administrator

1. Complete the steps outlined in [Configure Administrator Identity and Authentication Details](#) (see page 226).
2. In the Scope group box, select the Domain radio button.
The Scope group box changes to a list of selected policy domains.
3. If the list of domains is empty or incorrect, click Add/Remove....
The Domain Manager for These Domains dialog opens.
4. In the Available Members list, highlight the policy domains that the new administrator can manage, then click the Left Arrow button.
Adds the selected policy domains to the Current Members list.

5. Click OK.

The Domain Manager for These Domains dialog closes and the selected policy domains appear in the list in the Scope group box of the SiteMinder Administrator dialog.

6. In the Tasks group box, select one or more of the following check boxes to assign privileges to the administrator:
 - Manage Domain Objects
 - Manage Users
 - Manage Password Policies
 - View Reports

More information:

[Configure Administrator Identity and Authentication Details](#) (see page 226)

[Domain Manager For These Domains Dialog](#) (see page 604)

Precautions on Deleting or Editing an Administrator

Important! Proceed with caution when deleting or editing an administrator.

When deleting an administrator, be careful not to delete all administrators, including the one you are logged on as. If you delete all administrators you will have to use the smreg or tool (for Solaris) or the smreg.exe tool (for WinNT) included on the installation CD to restore a default Administrator using the following command:

```
smreg -su <password>
```

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

This command resets the password of the SiteMinder administrator account to the value you enter for <password>.

Note: For more information about this tool, run smreg.exe from a command prompt with the -? switch.

When editing, be careful not to modify the settings for the administrator you are logged on as. If you change your administrative settings, you may lose access to SiteMinder features. If you lock yourself out of a feature, an administrator with full access to the Policy Server must restore your privileges.

Chapter 10: Authentication Schemes

This section contains the following topics:

- [Authentication Schemes Overview](#) (see page 231)
- [Basic Authentication Schemes](#) (see page 241)
- [Basic Over SSL Authentication Schemes](#) (see page 243)
- [HTML Forms Authentication Schemes](#) (see page 244)
- [Windows Authentication Schemes](#) (see page 258)
- [MS Passport Authentication Schemes](#) (see page 261)
- [RADIUS CHAP/PAP Authentication Schemes](#) (see page 265)
- [RADIUS Server Authentication Schemes](#) (see page 267)
- [SafeWord Server Authentication Schemes](#) (see page 268)
- [SafeWord Server and HTML Forms Authentication Schemes](#) (see page 269)
- [SecurID Authentication Schemes](#) (see page 270)
- [X.509 Client Certificate Authentication Schemes](#) (see page 277)
- [X.509 Client Certificate and Basic Authentication Schemes](#) (see page 279)
- [X.509 Certificate or Basic Authentication Schemes](#) (see page 281)
- [X.509 Client Certificate and HTML Forms Authentication Schemes](#) (see page 283)
- [X.509 Client Certificate or HTML Forms Authentication Schemes](#) (see page 286)
- [Anonymous Authentication Schemes](#) (see page 288)
- [Custom Authentication Schemes](#) (see page 290)
- [Federation Security Services Authentication Schemes](#) (see page 291)
- [TransactionMinder Authentication Schemes](#) (see page 291)
- [Impersonation Authentication Schemes](#) (see page 292)
- [Certificate Mapping](#) (see page 293)

Authentication Schemes Overview

In most cases, when a user attempts to access a network resource, the owner of the network wants to verify the identity of the user. Company employees can be identified to determine which resources they can use. Customers can be identified for personalization of content as they access resources. Even anonymous users can be tracked uniquely, so that their history can be used to provide a quality experience when they once again access the network. To identify a user, SiteMinder employs authentication schemes.

Authentication schemes provide a way to collect credentials and determine the identity of a user. SiteMinder supports various authentication schemes. These schemes range from basic user name/password authentication and HTML forms-based authentication to digital certificates. Simple schemes can be used for low risk network resources, while complex schemes can be employed to help ensure added security for critical network resources.

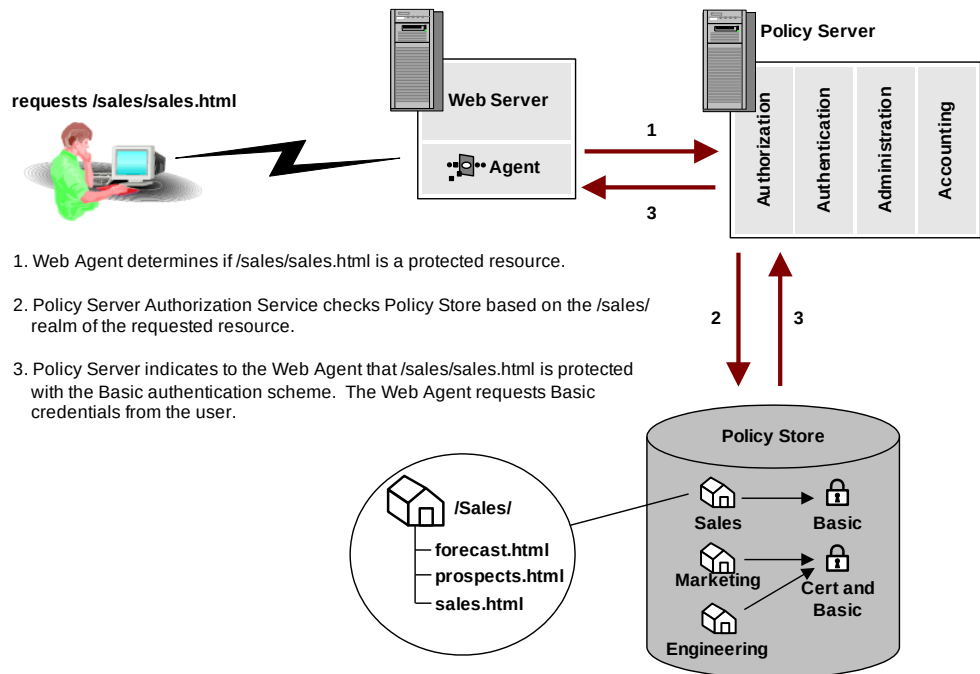
Authentication schemes must be configured using the Policy Server User Interface. During authentication, SiteMinder Web Agents communicate with the Policy Server to determine the proper credentials that must be retrieved from a user who is requesting resources.

This chapter discusses general information for working with authentication schemes in the Policy Server User Interface, then provides separate sections that explain how to configure each supported scheme using authentication scheme templates. These templates provide the Policy Server with most of the information required to process a scheme. An administrator must complete the configuration of an authentication scheme by supplying implementation-specific information, such as server IP addresses, or shared secrets required to initialize a scheme.

Overview of Authentication Scheme Processing

When a user attempts to access a protected network resource, the Policy Server uses the authentication scheme associated with the resource's realm to determine how to identify the user. The authentication scheme specifies the credentials that the user must supply for authentication, as well as the method used by the Policy Server to validate the user's identity.

You can use the Policy Server User Interface to configure authentication schemes and assign the schemes to realms. The following diagram illustrates how an authentication scheme is called when a user attempts to access a protected resource.



In the example above, the user requests the protected resource `sales.html` from the `/Sales/` realm. This realm requires Basic authentication. The Policy Server informs the Web Agent that the resource is protected and requests Basic credentials from the user via the Web Agent, which prompts the user for a user name and password.

More information:

[Configure a Realm](#) (see page 320)

Limit Policy Server Search to One User Store during Authentication

A single user can be stored in more than one user directory or database associated with a policy domain. This user has the same password in each user store. During authentication, if the Policy Server finds that the user is disabled in one user store, then by default, it continues searching for the user in all stores associated with the policy domain. The user fails authentication only if the Policy Server finds the user disabled in all associated user stores. The user is authenticated if it is enabled in any associated user store.

This default Policy Server behavior is configurable. To configure the Policy Server to stop searching when it first finds the user disabled in a user store, add the following registry key and set its value to one: `ReturnOnDisabledUser`.

To limit Policy Server search to one user store during authentication

1. Manually add the registry key `ReturnOnDisabledUser`:

Windows

Add the registry key `ReturnOnDisabledUser` to the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion
\PolicyServer
```

Solaris

Add the following lines to the `sm.registry` file:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion
\PolicyServer
ReturnOnDisabledUser=0x1; REG_DWORD
```

2. Assign `ReturnOnDisabledUser` the value of one.

Types of Authentication Schemes

The Authentication Schemes supported by SiteMinder fall into a number of categories. These categories represent the general characteristics of available authentication methods. Details are provided in the sections of this chapter that discuss each specific authentication scheme.

Basic Authentication Schemes

Basic authentication identifies a user based on a user name and password. The user's identity is stored in a user directory. With a basic authentication scheme, the Policy Server locates a user in a directory based on the user name, then verifies that the password matches the one saved in the user directory (binds the user to the directory). If the user name and password supplied by the user match the data in the user directory, SiteMinder authenticates the user.

The Policy Server User Interface provides authentication scheme templates for the following basic schemes:

- Basic (HTTP Basic)
- Basic over SSL

HTML Forms-based Authentication Schemes

SiteMinder supports the use of customized HTML forms to collect authentication information. In a forms-based authentication scheme, a user can be required to enter additional information such as an organization, account number, etc. The Policy Server verifies the additional information against user directory attributes before authenticating the user.

Windows Authentication Scheme

Integrated Windows Authentication (IWA) is a proprietary mechanism developed by Microsoft to validate users in pure Windows environments. IWA enforces Single Sign-On by allowing Windows to gather user credentials during the initial interactive desktop login process and subsequently transmitting that information to the security layer. SiteMinder, using the Windows Authentication scheme, secures resources by processing user credentials obtained by the Microsoft Integrated Windows Authentication infrastructure.

Previous versions of SiteMinder supported Windows authentication through the NTLM authentication scheme. However, this support was limited to environments with NT Domains or where the Active Directory service is configured to support legacy NT Domains in mixed mode.

The Windows authentication scheme allows SiteMinder to provide access control in deployments with Active Directories running in native mode, as well as Active Directories configured to support NTLM authentication. The Windows Authentication scheme replaces SiteMinder's previous NTLM authentication scheme. Existing NTLM authentication schemes continue to be supported and can be configured using the new Windows Authentication scheme.

The NTLM authentication scheme can be used for resources that are protected by Web Agents on IIS Web servers, and whose users access resources via Internet Explorer Web browsers. This scheme relies on a properly-configured IIS Web server to acquire and verify a user's credentials. The Policy Server bases authorization decisions on the user's identity as asserted by the IIS server.

X.509 Client Certificate Authentication Schemes

SiteMinder supports the use of X.509 V3 client certificates. Digital certificates act as cryptographic proof of a user's identity. Once a certificate is installed on a client, that certificate can be used to verify the identity of a user who is accessing a resource. Certificate authentication uses SSL communication and can be combined with basic authentication to provide an even higher level of access security.

The Policy Server User Interface provides authentication scheme templates for the following certificate-based authentication schemes:

- X.509 Client Certificates
- X.509 Client Certificates and Basic
- X.509 Client Certificates or Basic
- X.509 Client Certificates and HTML Forms
- X.509 Client Certificates or HTML Forms

Note: In the case of certificate-only authentication schemes, the web agent returns HTTP Error 403: Access Denied/Forbidden for any failed authentication or authorization attempt. This is because there is no way for the web agent to challenge the user for a new certificate.

Proxy Authentication Schemes

Proxy authentication schemes are schemes that use the Policy Server as a proxy or substitute for the server required by a third party authentication product. With a proxy scheme, the Policy Server performs the authentication function of the third party server using scheme specific libraries.

SiteMinder supports the following proxy authentication schemes:

- RSA ACE/Server (used with SecurID tokens)
- RSA ACE/Server (used with SecurID tokens) with HTML forms support for resetting passwords
- Secure Computing Safeword Server
- RADIUS Server

Digest Authentication Schemes

A digest authentication scheme reads an encrypted user attribute string stored in a directory, then compares the string to the encrypted string it receives from the user. If the encrypted strings match, the Policy Server authenticates the user. A digest scheme compares a string encrypted on a client workstation to an encrypted string on a server without using an encrypted transmission.

SiteMinder supports the following digest authentication schemes:

- RADIUS CHAP
- RADIUS PAP

Anonymous Authentication Schemes

An anonymous authentication scheme allows non-registered users to access specific Web content. When a user accesses a resource that has anonymous authentication, SiteMinder assigns the user a Global User Identification (GUID). SiteMinder places this GUID in a persistent cookie on the user's browser so that the user can access specific resources without being challenged to authenticate.

Custom Authentication Schemes

If SiteMinder does not provide a method of authentication that you want to use, you can use CA's APIs to develop a custom authentication scheme.

Note: If you have installed the Software Development Kit, see the *Developer's Guide for C* or the *Developer's Guide for Java* for more information about creating custom authentication schemes.

Authentication over SSL

Authentication can be configured to run over a Secure Sockets Layer (SSL) connection. SSL is a method of establishing an encrypted connection between a client and a server using digital certificates to initiate the connection, and to establish a proof of identity.

The following authentication scheme types are configured to use an SSL connection:

- Basic*
- HTML Forms*
- X.509 Client Certificates
- X.509 Client Certificates and Basic
- X.509 Client Certificates or Basic

- X.509 Client Certificate or HTML Forms
- X.509 Client Certificate and HTML Forms

* SSL connection is optional.

Protection Levels

Authentication schemes require a protection level. This level ranges from 0 to 1000. A higher number indicates that the scheme provides higher level of protection. Protection levels allow single sign-on for Authentication Schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Note: Anonymous authentication schemes always have a protection level of zero. A custom authentication scheme may have a protection level of 0-1000. All other authentication schemes may have a protection level of 1-1000.

For example, if you have a set resources that is available to all network users, you can assign a Basic (user name and password) authentication scheme with a low protection level such as 5. For revenue information that is available only to corporate executives, you can assign an X.509 client certificate scheme with a high protection level such as 15.

When users authenticate successfully against a scheme, they can access any resource with a protection level equal to or below the current authentication scheme without being challenged to authenticate a second time. However, users must still be authorized for a resource to gain access.

In the example above, a user who authenticated with a user name and password (protection level 5) would have to authenticate a second time with a digital certificate if he or she attempted to access the revenue information that was assigned the X.509 client certificate authentication scheme with the protection level of 15. However, if the user attempted to access resources for another department, which were also protected by a scheme of level 5, the user would not be challenged to authenticate a second time.

More information:

[Domains](#) (see page 307)

[Policies](#) (see page 375)

Supported Authentication Schemes

The following table lists all supported authentication schemes and their credential requirements:

Credential Requirements					
Authentication Schemes	Directory User Name	Directory Password	Code from Token	X.509 Certificate	User Profile Attributes
Anonymous					
Basic	yes	yes			
Basic over SSL	yes	yes			
Custom	optional	optional	optional	optional	optional
HTML Forms (over SSL optional)	custom credentials	custom credentials			optional
Impersonation	yes				optional
MS Passport	yes	yes			yes
NTLM or Windows	yes*	yes*			
RADIUS CHAP/PAP	yes	yes			
RADIUS Server	yes	yes			
SafeWord Server	yes	yes			
SafeWord and Forms	yes	yes			optional
SecurID	yes		yes		
SecurID and Forms	yes		yes		optional
X.509 Client Certificate				yes	
X.509 Client Certificate and Basic (uses SSL)	yes	yes		yes	
X.509 Client Certificate or Basic (over SSL optional)	yes for Basic	yes for Basic		yes for Certificate	
X.509 Client Certificate and HTML Forms	custom credentials	custom credentials		yes	optional

Credential Requirements					
Authentication Schemes	Directory User Name	Directory Password	Code from Token	X.509 Certificate	User Profile Attributes
X.509 Client Certificate or HTML Forms	custom credentials for HTML Forms	custom credentials for HTML Forms		yes for Certificate	optional for HTML Forms

*For NTLM or Windows, the user, when trying to access a resource, is not prompted for a user name and password by SiteMinder. This scheme relies on a properly-configured IIS Web server to acquire and verify a user's credentials. The Policy Server bases authorization decisions on the user's identity as asserted by the IIS server.

Each of these schemes is discussed in detail later in this chapter.

Supported Authentication Schemes and Password Policies

Some authentication scheme types support Password Policies, while others do not. You can view whether a particular type of authentication scheme supports Password Policies by opening the Authentication Scheme Properties dialog in the Policy Server User Interface. To view a particular authentication scheme type, select it from the drop-down list on the Scheme Common Setup group box and observe the Password Policies Enabled for this Authentication Scheme check box. If the authentication scheme does not support Password Policies, the check box description is dimmed and the check box is unavailable.

To view supported authentication scheme types and whether they support Password Policies without accessing the Policy Server User Interface, see the following table:

Authentication Scheme Type	Type Supports Password Policies?
Anonymous	No
Basic	Yes
Basic over SSL	Yes
Custom	Yes
HTML Forms	Yes
Impersonation	No
MS Passport	No

RADIUS CHAP/PAP	Yes
RADIUS Server	Yes
SafeWord	No
SafeWord and HTML Forms	No
SecurID	No
SecurID and HTML Forms	No
X.509 Client Certificate	No
X.509 Client Certificate and Basic	Yes
X.509 Client Certificate or Basic	Yes
X.509 Client Certificate and HTML Forms	Yes
X.509 Client Certificate or HTML Forms	Yes
Windows Authentication	Yes

Set Up an Authentication Scheme Object in the Policy Server User Interface

To setup a new authentication scheme in the Policy Server User Interface, components should be configured in the following order:

1. Web Server (only for certificate, SSL, and HTML forms-based schemes)
2. Policy Server (including Certificate Mapping for X.509 certificate schemes)

More information:

[Certificate Mapping](#) (see page 293)

Web Server

In order for a SiteMinder Web Agent to support any SSL-based Authentication Scheme, a Web Server must be configured to support SSL.

Note: See the *Policy Server Installation Guide* for instructions on Web server configuration.

Policy Server

After you configure your Web servers to support authentication schemes, you can configure the Policy Server to support the schemes.

More information:

[Authentication Schemes Overview](#) (see page 231)

Multiple Instances of a Single Authentication Scheme Configuration

You can configure multiple instances of most authentication schemes in the Policy Server User Interface. For example, you might create multiple HTML forms-based schemes to process login, forgotten password requests, logout, etc. If you create multiple instances of a scheme type, be sure to set protection levels to reflect your security requirements.

Basic Authentication Schemes

The Policy Server installation process automatically configures a Basic authentication scheme. This scheme verifies a user's identity according to a user name and password that are passed to a user directory service for authentication. For LDAP user directories, this is done via an LDAP "Bind" operation. For Windows NT directories, this is done via the LogonUser() function call. The HTTP Basic Authentication protocol is used to deliver credentials from the browser to the Web server protected by the SiteMinder Web Agent. Basic authentication schemes are supported only with ASCII characters.

As discussed in [Overview of Authentication Scheme Processing](#) (see page 232), when a user attempts to access a resource protected by Basic authentication, the SiteMinder Agent prompts the user to enter a user name and password. When the user enters a name and password, the Agent passes the credentials to the Policy Server over an encrypted connection, and the Policy Server matches the name against the users contained in the directories attached to the policy domain that contains the resource. When the Policy Server finds a matching user name, it compares the password in the user directory to the password supplied by the user. If the passwords match, the user is authenticated, and the Policy Server sends a message to the Web Agent indicating that the Agent may proceed. If the authentication fails, the user is challenged to re-enter credentials.

Note: This scheme does not provide encrypted credential delivery by default. Instead, the user name and password are delivered from the browser to the Web server protected by the Web Agent via the standard HTTP Basic protocol, unless every protected URL on the Web server is set up to require SSL. However, communication between the Web Agent and the Policy Server always takes place over an encrypted connection. For an encrypted authentication scheme based on simple user names and passwords, see [Basic Over SSL Authentication Schemes](#) (see page 243).

Realms that you create in the Policy Server User Interface use the Basic authentication scheme by default. You can change the authentication scheme when you create a new realm or modify an existing realm.

For an additional level of security with Basic authentication, you can create password policies. This SiteMinder feature allows you to manage password rules.

More information:

[Domains](#) (see page 307)

[Grouping Resources in Realms](#) (see page 315)

[Password Policies](#) (see page 479)

Basic Scheme Prerequisites

In order to use the Basic authentication scheme, the following prerequisites must be met:

- Client user name and password information must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.

More information:

[User Directories](#) (see page 141)

Configure a Basic Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a Basic authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select Basic Template.
4. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)

[Realms](#) (see page 315)

Basic Over SSL Authentication Schemes

The Basic Over SSL Authentication Scheme verifies a user's identity by passing a user name and password credentials to a user directory in a process similar to Basic authentication. However, credential delivery is always done over an encrypted Secure Sockets Layer (SSL) connection even if the protected URLs are not setup to require SSL. The SiteMinder Web Agent accomplishes this by redirecting the user's browser to establish an SSL connection prior to credential delivery. After the credentials are delivered the Web Agent redirects the browser back to the original URL.

For an additional level of security with Basic over SSL authentication, you can create password policies. This SiteMinder feature allows you to manage password rules.

More information:

[Basic Authentication Schemes](#) (see page 234)

[Password Policies](#) (see page 479)

Basic over SSL Scheme Prerequisites

In order to use the Basic over SSL authentication scheme, the following prerequisites must be met:

- Client user name and password information must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser using the HTTPS protocol.
- A SiteMinder Web Agent must be installed on the Web server to which requests are redirected for SSL. The Web Agent enables the server to handle the .scc MIME type required by the authentication scheme. See the following sections for information about specifying the SSL server.

More information:

[User Directories](#) (see page 141)

Configure a Basic Over SSL Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a Basic Over SSL authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select Basic Over SSL Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—Basic over SSL Template Scheme Setup Tab](#) (see page 611)..
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

HTML Forms Authentication Schemes

HTML Forms authentication schemes provide a method for authentication based on credentials gathered in a custom HTML form. This flexible means of credential collection allows you to:

- Provide a “branded” look, perhaps including a company logo.
- Substitute custom labels for user name and password collection (for example, if users think in terms of an account number and a PIN rather than a name and password).
- Provide authentication based on credentials other than a user name and password (via user directory attributes). In this case, an authentication scheme library on the Policy Server machine maps the user’s data to a DN. By mapping the user to a DN, the Policy Server can match an attribute list to the appropriate values in a user directory. This process is called back-end mapping.

- Provide authentication based on credentials that include user attributes in addition to the user name and password. This is considered additional attribute verification. A custom authentication scheme library is not required for additional attribute verification.

For example, a custom form can be used to collect a name and a secret phrase for users who forget their password.

- Provide multiple HTML forms for login, logout, forgotten passwords, etc.

Note: HTML Forms authentication schemes are supported with multi-byte characters.

Multiple *Forms-based* Authentication Schemes can be configured in a Policy Server installation. Each scheme consists of the following components:

Forms Credential Collector (FCC)

The FCC process files are composed in a simple mark-up language that includes HTML and some custom notation.

Each HTML Forms scheme must have its own .fcc file. This file contains the custom form definition and additional information that the FCC uses to process HTML Forms authentication. The FCC extracts credentials that a user enters in the custom form generated from the .fcc file.

For the HTML Forms authentication scheme, the default extension for .fcc files is .fcc. If you want to use a different extension:

- For Apache or Oracle iPlanet web servers, configure your web server to use that extension.
- For Domino or IIS Web servers, specify that extension in the FCCCEXTENSIONS parameter of your Web Agent configuration file or object. For more information on Web Agent configuration parameters, see the *SiteMinder Web Agent Guide*.

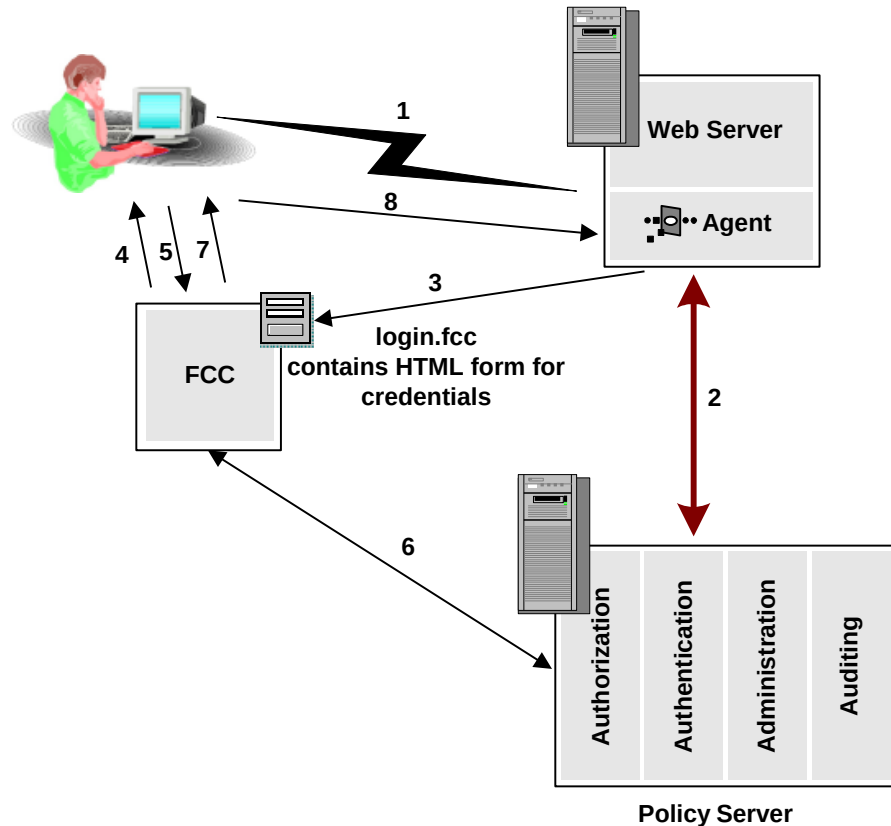
.unauth file

SiteMinder displays the contents of this file to users who exceed the maximum number of failed authentication attempts specified by the authentication scheme. A .unauth file should exist for each .fcc file. For example, if you have a login.fcc file on a Web server, you should also have a login.unauth file in the same location.

If an smerrorpage variable has been defined in the .fcc file, the .unauth file is not required.

Authentication Scheme Library

This is a shared library that runs on the Policy Server machine and performs authentications.



The previous diagram describes the process for HTML Forms authentication.

1. A user requests a resource contained in a realm protected by HTML Forms authentication.
2. The Web Agent contacts the Policy Server and determines that the user's request must be redirected to the credential collector.
3. The Web Agent redirects the request to the URL of the credential collector file.
4. The credential collector displays the form described in the .fcc file in the user's browser.
5. The user fills out the custom form and Posts (submits) the form. The credential collector processes the credentials.
6. The credential collector (FCC) logs the user into the Policy Server. The Policy Server returns user session data to the credential collector.

7. If the user is authenticated, the credential collector creates a session cookie, passes the session cookie to the browser and redirects the user to the resource that he or she originally requested.
8. The user uses the session cookie to authenticate. Then, the Web Agent handles user authorization.

More information:

[SiteMinder FCC Files](#) (see page 249)

HTML Forms Scheme Prerequisites

To use the HTML Forms authentication scheme, the following prerequisites must be met:

- A customized .fcc file must reside on a Web Agent server in the cookie domain in which you want to implement HTML Forms authentication. CA provides sample .fcc files under the Samples/Forms subdirectory, where you installed your Web Agent. For example:

`<SM_Web_Agent_Install_Dir>\Samples\Forms`

- A customized .unauth file must reside on the Web Agent server unless the .fcc file uses the smerrorpage directive.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- The default HTML forms library must be installed to handle HTML Forms authentication processing:
 - SmAuthHTML.dll on Windows
 - smauthhtml.so on Solaris

These files are installed automatically when you configure a Web Agent. For more information, see the *Policy Server Installation Guide*.

- (For Oracle iPlanet Web Servers [formerly Sun Java Systems]) Increase the value of the StackSize parameter in the magnus.conf file to a value greater than 131072. Failing to change the value causes the web server to dump its core and restart each time SiteMinder makes an authentication request using forms.

More information:

[SiteMinder FCC Files](#) (see page 249)

[User Directories](#) (see page 141)

Custom Authentication Scheme Library Writing and Installation

The user name and password data collected by the FCC are passed to the Policy Server, which passes them to the Authentication Scheme library.

Unless back-end mapping is required, the SmAuthHTML Authentication Scheme library can be used; it is distributed with the Policy Server and already installed on the Policy Server machine.

Note: Back-end mapping requires a custom Authentication Scheme library. If you have installed the Software Development Kit, see the *Developer's Guide for C*.

If you have written a custom Authentication Scheme and want to gather more data than the username and password, the FCC should pack that data into the username and password fields (each of which must be less than 511 characters long). The custom Authentication Scheme library must then be able to unpack the data and map it to user name and password.

The FCC can be installed on the same machine as the Policy Server.

HTML Forms Authentication Templates

All templates used for forms authentication are text files, called .fcc files, that use an extended version of HTML.

For the HTML Forms authentication scheme, the default extension for .fcc files is .fcc. If you want to use a different extension:

- For Apache Web servers, configure your Web server to use that extension (see the *SiteMinder Web Agent Installation Guide*).
- For Domino or IIS Web servers, specify that extension in the FCCExtensions parameter of your Web Agent configuration file or object. For more information on Web Agent configuration parameters, see the *SiteMinder Web Agent Guide*.

The SiteMinder Web Agent includes sample English, French and Japanese .fcc files, which you can customize. These files are located in the following directories:

Language	Directory
English	Windows: Web Agent\Samples\Forms UNIX: webagent/samples/forms
French	Windows: Web Agent\Samples\Formsfr UNIX: webagent/samples/formsfr

Language	Directory
Japanese	Windows: Web Agent\Samples\Formsja UNIX: webagent/samples/formsja

By default, SiteMinder uses the English .fcc files.

To use these sample .fcc files, create a Forms Authentication Scheme that specifies the appropriate target directory.

SiteMinder FCC Files

The SiteMinder Forms Credential Collector (FCC) incorporated into SiteMinder Web Agents reads template files called .fcc files. The .fcc files are written using standard HTML tags and a small amount of proprietary notation required by SiteMinder to verify attributes and take advantage of custom features described later in this section.

Important! If you create or edit an .fcc file on a Windows system and move that file to a UNIX system, your UNIX system may append ^M at the end of lines of text. These characters, which identify the file as a Windows text file, cause .fcc files to fail during authentication. When moving files from Windows text editors to UNIX systems, be sure to examine the files and remove the appended characters. To avoid this situation, create and edit .fcc files that will be used in a UNIX environment on a UNIX system.

For the HTML Forms authentication scheme, the default extension for .fcc files is .fcc. If you want to use a different extension:

- For Apache or Oracle iPlanet web servers, configure your Web server to use that extension.
- For Domino or IIS Web servers, specify that extension in the FCCExtensions parameter of your Web Agent configuration file or object. For more information on Web Agent configuration parameters, see the *SiteMinder Web Agent Guide*.

When a user requests a resource protected by an HTML Forms scheme, the Web Agent redirects the user to an .fcc file. The .fcc file invokes the FCC on the Web server in order to collect credentials from the user via a customized form. The FCC generates a browser page and builds a user name and password based on the contents of the .fcc file. The file resides in a Web server's name space and is accessed like any HTML file. The .fcc file may contain two parts. Both parts are optional.

The first part of the FCC contains directives that are used when executing a POST operation on the .fcc file. The directives are never passed to the client. They must be at the beginning of the file and are of the form: @name=value

The *name* is the name of a variable. The *value* is the variable's value. The *value* may contain strings of the form: %name1%. This will be replaced by the value of the variable associated with name1.

The second part of the .fcc file contains HTML code that is returned when a GET operation is performed on the .fcc file. This part may include text in the form "\$\$name\$\$", including the quotation marks (") that will be replaced by the value associated with name. The name is not case sensitive.

The hidden inputs listed in the following figure are used to hold state for the credential collectors:

Name	To dynamically set, use the value:*	Data preserved
target	\$\$target\$\$	Resource that a user wants to access
smauthreason	\$\$smauthreason\$\$	Reason for a login failure
postpreservationdata	\$\$postpreservationdata\$\$	Data that a user submits through a post request.
smagentname	\$\$smagentname\$\$	Agent name used for logging user in.

*Be sure to enter the quotation marks (").

At a minimum, an .fcc file must collect the following:

- User name
- Password
- Target

Important! If users will be submitting post requests to a resource protected by an authentication scheme that uses a credential collector (see the following figure), use the postpreservationdata input. Otherwise, data that users attempt to post to the requested resource will be lost.

Schemes

[Basic Over SSL Authentication Schemes](#) (see page 243)

[HTML Forms Authentication Schemes](#) (see page 244)

[X.509 Client Certificate Authentication Schemes](#) (see page 277)

[X.509 Client Certificate and Basic Authentication Schemes](#) (see page 279)

[X.509 Certificate or Basic Authentication Schemes](#) (see page 281)

[X.509 Client Certificate and HTML Forms Authentication Schemes](#) (see page 283)

Schemes

[X.509 Client Certificate or HTML Forms Authentication Schemes](#) (see page 286)

The following is an example of a valid (though simple) .fcc file:

Creates a user name based on form data	<code>@username=uid=%USER%,ou=%GROUP%,o=%ORG% @smretries=3 <html> <head><title>Sample Login Form</title></head> <body> <h3> Please enter your login credentials</h3> <form method=post><table> <tr> <td>User Name:</td> <td><input type=text name=USER></td> </tr> <tr> <td>Password:</td> <td><input type=password name=PASSWORD></td> </tr> <tr> <td>Group: </td><td><input type=text name=GROUP></td> </tr>
 <tr> <td>Organization: </td> <td><Select Name=ORG SIZE=1> <OPTION>Company A <OPTION>Company B <OPTION>Company C <OPTION>Company D </SELECT></td> <input type=hidden name=target value="\$\$target\$\$"> <input type=hidden name=smauthreason value="\$\$smauthreason\$\$"> <tr><td><input type=submit value=LOGIN></td></tr> </table></form></body> </html></code>
Collects the user's password	
Collects the user's organization	

The file above is the usermap.fcc sample file included with the default installation of SiteMinder Web Agents.

The .fcc file above creates a distinguished name (DN) for the user based on the information the user enters in the User Name field and the Organization drop-down list of the HTML form. This DN is the user name authentication credential. The user's password is collected from the Password field of the HTML form. The hidden realm and target input values are also collected so that the user can be directed to the appropriate resource when authentication is complete.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

How Name/Value Pairs are Generated in FCC Files

The LoginFCC generates a name space for use in `$$name$$` expansions and for use by special name value pairs. If a name is entered more than once the last value wins.

Name/value pairs are added to this name space in the following order.

1. Name/Values from the query string (on Get only).
2. The name `smtarget` is created by copying the value of the target (on Get only).
3. Name/Values from the post data.
4. Name/Values from the cookies.
5. Name/Values from the `@` directives (on Post only).
6. SiteMinder responses named in the “`smheaders`” name value pair (on Post only).

Special Name/Value Pairs

SiteMinder’s FCC can interpret a number of special name/value pairs (`@`directives) that invoke non-standard processing. The following are the special `@`directives and their meanings:

username

Name to use to as the login user name.

password

Password to use to perform the login.

target

Resource to access after login.

smheaders

Colon separated list of SiteMinder response names to include in the namespace. The colon separated list must contain an entry for each header that you want to include in a SiteMinder transaction. For example, if you want to pass the value of `header1` and `header2` as part of a SiteMinder transaction, the following would be included in your FCC:

```
@smheaders=header1:header2
```

smerrorpage

If there is an error on a Post to the custom form, the user’s browser will be redirected to this page. If this special value is not specified in a `.fcc` file, SiteMinder uses the `.unauth` file associated with the `.fcc` file as the error page.

smretries

Specifies the maximum number of login attempts allowed. If you set this directive to 0, the number of retries is unlimited. If you set the number to 1 or greater, that is the number of retries allowed.

Note: If users log in using a POST to an .fcc form, it may appear that the user is given additional attempts to log in beyond the value of the smretries directive. However, the user is allowed access only if valid credentials are entered in the number of attempts specified by smretries.

smpasswordfcc

Determines whether data is posted from the Password Services FCC file or from a different FCC file.

Default: 1

Important! It is recommended that the default value remain unchanged. The Safeword authentication scheme may not work properly if the default value is changed.

smusrmsg

Text that describes why the user was challenged / failed to login.

smauthreason

Reason code associated with a login failure.

smsavecreds

Set to Yes to save user credentials in a persistent cookie on the user's browser.

smsave

Colon separated list of names to be saved as persistent cookies.

save

Another name for smsave.

smtransient

Colon separated list of names to be saved as transient cookies.

smagentname

Specifies the agent name that will be supplied to the Policy Server when a user enters credentials and submits the form for authentication. If the Agent parameter, FCCCompatMode=NO, you must specify a value using this directive.

smlogout

Logs a user out of SiteMinder, similar to logoffuri. By placing @smlogout=true in your .fcc template, the FCC will log a user out of SiteMinder and redirect the user to the target. As such, the @smlogout directive is typically used in conjunction with the @target directive (@target=<yoururlhere>).

urlencode(*name*)

Replaced by the URL encoded value of the named variable.

Note: If you expect the additional attributes or the Password to contain special characters (" . & = + ? ; / : @ = , \$ %), you need to URL-encode each additional attribute value in the .fcc file. SiteMinder uses US-ASCII encoding.

urldecode(*name*)

Replaced by the URL decoded value the named variable.

Note: The “sm” prefix for name/value pairs is reserved for additional special names required by SiteMinder. When creating names for your login page do not use the “sm” prefix.

More information:

[Include Header Values in the FCC Namespace](#) (see page 255)

Collect Additional Attributes

In addition to username and password, you can collect additional attributes from users, such as a users email address or job title.

To collect additional attributes

1. Specify the attribute(s) by configuring the HTML Forms authentication scheme and the associated .fcc file.
2. Configure the HTML Forms authentication scheme by specifying new attributes in the Additional Attribute List fields of the Scheme Setup dialog.

Note: If you expect the additional attributes or the Password to contain special characters (" . & = + ? ; / : @ = , \$ %), you need to URL-encode each additional attribute value in the .fcc file. SiteMinder uses US-ASCII encoding.

3. Modify the .fcc file to collect additional attributes.

To modify the .fcc file, add the following line at the beginning of the file:

```
@password=PASSWORD=%PASSWORD%&newattr1=%newattr1%&newattr2=%newattr2%
```

If the additional attributes have special characters, the line should look like the following:

```
@password=PASSWORD=%urlencode(PASSWORD)%&newattr1=%urlencode(newattr1)%&newattr2=%urlencode(newattr2)%
```

where newattr1=%newattr1% and newattr2=%newattr2% represent the additional attributes. The value before the equals sign is the attribute name and the value between the percent sign (%) sign is the attribute value.

The FCC parses the names of the new attributes from the attribute values.

Note: Append additional attributes to the @password directive with the ampersand (&) character.

When you add attributes to the FCC, keep the following in mind:

- The attribute name, which is identified by the `%attribute_name%` format, must match the input name entry that you also add to the FCC. For example, if you add the new attribute address:

```
@password=PASSWORD=%PASSWORD%&mail=%address%
```

or

```
@password=PASSWORD=%urlencode(PASSWORD)%&mail=%urlencode(address)%
```

you would add a line to the .fcc file similar to the following:

```
<input name="address" type="text">
```

- The name of the additional attribute must match the name of the attribute in the user directory. For example, to collect email addresses from an LDAP directory, specify one of the following:

```
@password=PASSWORD=%PASSWORD%&mail=%address%
```

or

```
@password=PASSWORD=%urlencode(PASSWORD)%&mail=%urlencode(address)%
```

where mail is the name of the LDAP attribute that stores email addresses.

- Do not add additional spaces after the value you specify for the @password directive. Additional spaces may be interpreted as meaningful characters and prevent users from being authenticated.
- The name of the attribute in the HTML Forms Authentication scheme must match the name of the additional attribute in the .fcc file.

For example, to add the attribute mail (from the example above) to the authentication scheme, enter the following in the Additional Attributes List field:

```
AL=PASSWORD,mail
```

Note: The additional attribute names are case-sensitive.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

Include Header Values in the FCC Namespace

As described in [Generate Name/Value Pair in FCC Files](#) (see page 252), the FCC name space can include headers. To use a header in a name space, you must use the @smheaders directive in order to copy the headers into the name space. If you do not include a header name in @smheaders, it will not be available in the name space.

For example, if you want to retrieve the HTTP headers called HEADER1 and HEADER2, and put them into hidden form fields in your FCC, you the following must appear in the FCC:

```
@smheaders=HEADER1:HEADER2
<input type="hidden" name="HEADER1" value="$$HEADER1$$">
<input type="hidden" name="HEADER2" value="$$HEADER2$$">
```

Without the @smheaders directive the output (as seen in your web browser) would be:

```
<input type="hidden" name="HEADER1" value="">
<input type="hidden" name="HEADER2" value="">
```

When you add the @smheaders directive as described above, the output is:

```
<input type="hidden" name="HEADER1" value="Contents of header HEADER1">
<input type="hidden" name="HEADER2" value="Contents of header HEADER2">
```

The following are example JSP/ASP examples for assigning the header values to hidden form fields:

```
%<key_name>% == request.getHeader() or request.ServerVariables() $$<key_name>$$ ==
request.getParameter() or request.QueryString()
```

For example, to assign the value of a header, "HEADER1" to a hidden FCC form field:

```
<input type=hidden name=HEADER1 value=%HEADER1%>
```

To get the same value from the query-string:

```
<input type=hidden name=HEADER1 value=$$HEADER1QueryStringParam$$>
```

Tell Users Why Login Failed

The default behavior of forms-based authentication is to redirect unauthenticated or unauthorized users back to the original login form. Although you can configure the smretries directive (@smretries) to provide users with additional login attempts, the default behavior does not let you display a message that informs users why the login failed.

The SiteMinder Web Agent is shipped with the DynamicRetry.fcc and DynamicRetry.unauth files. This sample pair of .fcc files changes the behavior of the redirect. The login page (DynamicRetry.fcc) is configured to send users to the unauthorized page (DynamicRetry.unauth) after one failed login attempt. The unauthorized page is a different HTML page than the login page. As a result, the unauthorized page can contain a message stating why the login failed. By default, the unauthorized page is configured with a message that informs users that they have entered invalid credentials for the resource they are attempting to access.

Note: You can change this message by opening DynamicRetry.unauth and updating the text in between the h3 tags.

To tell users why login failed, specify the target path to the DynamicRetry.fcc file when configuring the authentication scheme. The default path to the DynamicRetry.fcc file is *agent_home*\samples\forms\DynamicRetry.fcc

agent_home

Specifies the SiteMinder Web Agent installation path.

Consider the following limitations when using the DynamicRetry pair of .fcc files:

- You cannot count user login attempts based on the SMTRYNO cookie.
- You cannot limit the number of login attempts.

Note: You can use this method in combination with Password Services to disable users with a password policy after a specified number of failed attempts. More information on password policies exists in the Password Policies.

Configure an HTML Form Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an HTML Form authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select HTML Form Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—HTML Form Template Scheme Setup Tab](#) (see page 612).

Ensure that the .fcc file you specify in the Target field complies with the guidelines described in [HTML Forms Authentication Templates](#) (see page 248).

5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)

[HTML Forms Authentication Templates](#) (see page 248)

[Realms](#) (see page 315)

Enable Non-browser Client Support

You can configure HTML Form schemes that collect Basic (username and password) credentials to authenticate users using non-browser HTTP clients. These clients can be developed using Perl scripts, C++, and Java programs that communicate using HTTP protocol. Custom clients must send the basic credentials in with the initial request via an HTTP Authorization header or SiteMinder does not authenticate the users. If the credentials are not sent via an HTTP Authorization header, SiteMinder redirects to the HTML Form scheme without non-browser client support.

To enable non-browser client support

1. Open the HTML Authentication Scheme dialog for the scheme that is configured for the realm.
2. Select the Support Non-browser clients check box.

Non-browser client support is enabled.

Windows Authentication Schemes

Integrated Windows Authentication (IWA) is a proprietary mechanism developed by Microsoft to validate users in pure Windows environments. IWA enforces Single Sign-On by allowing Windows to gather user credentials during the initial interactive desktop login process and subsequently transmitting that information to the security layer. SiteMinder, using the Windows Authentication scheme, secures resources by processing user credentials obtained by the Microsoft Integrated Windows Authentication infrastructure.

Previous versions of SiteMinder supported Windows authentication through the NTLM authentication scheme. However, this support was limited to environments with NT Domains or where the Active Directory service is configured to support legacy NT Domains in mixed mode.

The Windows authentication scheme allows SiteMinder to provide access control in deployments with Active Directories running in native mode, as well as Active Directories configured to support NTLM authentication. The Windows Authentication scheme replaces SiteMinder's previous NTLM authentication scheme. Existing NTLM authentication schemes continue to be supported and can be configured using the new Windows Authentication scheme.

Note: In some circumstances, you may want use Windows User Security Context functionality combined with other authentication schemes instead of using the Windows authentication scheme.

The Windows authentication scheme can be used for resources that are protected by Web Agents on IIS Web servers, and whose users access resources via Internet Explorer Web browsers. This scheme relies on a properly-configured IIS Web server to acquire and verify a user's credentials. The Policy Server bases authorization decisions on the user's identity as asserted by the IIS server.

More information:

[How SiteMinder Is Configured to Provide a Windows User Security Context](#) (see page 96)

Windows Authentication Scheme Prerequisites

In order to use the Windows Authentication scheme, the following prerequisites must be met:

- For legacy WinNT directories or Active Directory in mixed mode:
 - The user directory connection you create in the Policy Server User Interface must specify the WinNT namespace.
 - The requested resources can be located on any type of web server, but the authentication server and the Web Agent protecting those resources must be on a Microsoft IIS web server.
- For Active Directories running in native mode:
 - User data must reside in an Active Directory.
 - User directory connections must specify either an LDAP or AD namespace.
 - The requested resources can be located on any type of web server, but the authentication server and the Web Agent protecting those resources must be on a Microsoft IIS web server.
 - Client and server accounts must be enabled for delegation.
- Users must log in using Internet Explorer Web browsers (4.0 or later).
- To work on IIS6 in Windows 2003, the Verified that file exists option in the Wildcard Application Maps must not be set.
- Windows Authentication schemes also require that any virtual directory on the IIS web server that contains the creds.ntc file remain unprotected.
- Internet Explorer browser options must be setup to allow automatic logon with a user's current username and password.

For Internet Explorer 5.x and 6.x Browsers

1. From the menu bar in Internet Explorer, select Tools, , Internet Options.
The Internet Options dialog opens.
2. Click the Security tab to bring it to the front.

3. Select your Internet zone and click Custom Level.
4. The Security Settings dialog appears.
5. Scroll down to User Authentication, Logon.
6. Select the Automatic logon with current username and password radio button.
7. Click OK.

For Internet Explorer 4.x Browsers

1. From the menu bar in Internet Explorer, select View, Internet Options.
The Internet Options dialog opens.
2. Click the Security tab to bring it to the front.
3. Select your Internet zone from the drop down list.
4. In the Internet zone group box, select the and click Custom radio button and click Settings.
5. The Security Settings dialog appears.
6. Scroll down to User Authentication, Logon.
7. Select the Automatic logon with current username and password radio button.
8. Click OK.

Windows Authentication Scheme Considerations

The IIS Web server, not the Policy Server, performs authentication based on credentials it receives from the Internet Explorer Web browser. Therefore, you cannot use the OnAuthAttempt authentication event to redirect users who do not exist in the user store.

Configure a Windows Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a Windows authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select the Windows Authentication Template.

4. Complete the Protection Level field and Password Policies Enabled for this Authentication Scheme check box.

Note: For more information, see [Scheme Common Setup Group Box](#) (see page 609).

5. Complete the fields on the Scheme Setup tab as described in [Authentication Scheme—Windows Template—Scheme Setup Tab](#) (see page 616).
6. Click OK to save your changes and close the Authentication Scheme Dialog.

The authentication scheme may now be assigned to a realm.

More information:

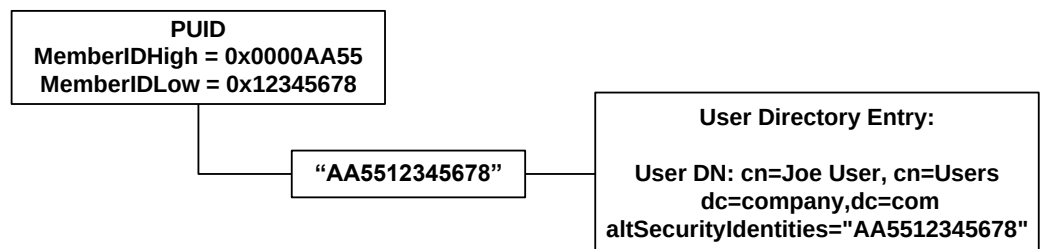
[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

MS Passport Authentication Schemes

Microsoft® .NET Passport is an online service that provides common Internet authentication across participating Web sites. Using a .NET Passport account, a user can move among participating sites without the need to authenticate with each of them. These sites become participating .NET Passport sites by implementing support for the .NET Passport authentication service through the .NET Passport single sign-in (SSI). This implementation includes a link to login through Passport using a common interface that supports co-branding. Users that are already logged into Passport will automatically be authenticated with the site, by means of a browser redirect to acquire Passport identity, and creation of site cookies containing the Passport identity. However, Passport does not authorize or deny a specific user's access to participating sites or resources.

The Passport identity does not contain any data for authorization. The only field that can be used to derive identity and subsequently permissions is the Passport Unique ID (PUID). SiteMinder uses this PUID to map to a local identity that is used to personalize the site and to authorize access to resources protected by SiteMinder policies.

The PUID is converted to a string and is mapped to a SiteMinder identity through a user directory attribute. In the following diagram, the user DN for "Joe User" is mapped to a Passport PUID through the directory attribute "altSecurityIdentities".



Since Passport authentication in SiteMinder is based on mapping the Passport identity to a SiteMinder user, registration is a required component for using Passport for authentication or personalization.

Registration is controlled through a registration URL that is configured in the authentication scheme. The configuration parameter is `?registrationurl=?` followed by a registration page or the keyword "FORM=" and SiteMinder form URL. This model provides two methods for registration of Passport users.

The first method uses a site-provided web application to link the Passport identity with a SiteMinder user account through the attribute configured in the authentication scheme. This requires the site to develop the web pages to accept registration data and to provide a service for setting the user directory attribute. SiteMinder can be used to provide the interfaces to the user directory and the secure tunnel behind the firewall using a tunnel service.

The second method of registration leverages the SiteMinder FCC and the forms authentication model. When the `registrationurl` includes the keyword `FORM=`, the subsequent URL is treated as a forms redirect. SiteMinder provides the `passport.fcc` file to use as a template for developing a Passport registration page using forms.

SiteMinder 5.5 provides a Passport Authentication Scheme template. The library name is `smauthmspp`. This authentication scheme can be used on both Windows and UNIX Policy Servers.

Passport Authentication Support in the Policy Server

The Policy Server and SiteMinder Web Agents support the following implementations of Passport authentication:

- Passport authentication established through a common registration URL
- Passport authentication using SiteMinder HTML forms for registration
- Anonymous logins when no Passport identity exists

For information about deploying Passport authentication in your enterprise, see:

<http://www.microsoft.com>

Set Protection Levels for Passport Authentication

Since the process of establishing a Passport identity does not include any authorization for access to participating sites or resources, the Passport authentication scheme should be assigned a relatively low protection level. We recommend using Passport authentication for personalization, and enforcing an authentication scheme with a higher protection level for sensitive resources. For example, Passport users could be authenticated, and their identities established using a SiteMinder protection level of 1. When the users request sensitive financial information, they might be forced to reauthenticate using an HTML forms authentication scheme with a protection level of 10.

More information:

[Protection Levels](#) (see page 237)

Passport Authentication Prerequisites

In order to use the Passport authentication scheme, the following prerequisites must be met:

- Passport SDK (version 1.4 for WinNT, or version 2.1 for Win2000) must be configured
- Passport Partner site must be configured
- SiteMinder 5.x QMR1 or QMR v2 Web Agents (when available) must be installed and running on Internet Information Server (IIS) on Windows systems
- Policy Server version 5.5 must be installed

Map Search Specifications for Passport Authentication

In order for SiteMinder to recognize the identity established through Microsoft® .NET Passport, a search specification must be configured to allow SiteMinder to locate the Passport Unique ID (PUID) in a particular user directory type.

You can configure a search specification for the following namespaces:

- ODBC
- Custom

Note: SiteMinder does not support Passport authentication using a WinNT user directory.

To map a search specification for a namespace

1. Configure the authentication scheme as described in [MS Passport Authentication Schemes](#) (see page 261).
2. In the Scheme Setup tab, locate the Map Search Root Specification to Namespace group box.
3. Select the appropriate namespace from the list.

Note: The authentication scheme can be configured to provide a search specification for one or more directory types.

4. Click Edit.

The Namespace Mapping dialog opens.

5. In the Search Specification field, enter the appropriate mapping.

Mappings should be configured as follows:

- An ODBC search specification is configured as follows:
PassportUID=PUID:%s@company.local
- A custom directory specification is configured as follows:
PassportUID=PUID:%s@company.local

6. Click OK.

The Namespace Mapping dialog closes and the search specification appears in the Map Search Specification to Namespace group box.

Configure an MS Passport Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an MS Passport authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select MS Passport Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—MS Passport Template Scheme Setup Tab](#) (see page 619).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)

[Realms](#) (see page 315)

RADIUS CHAP/PAP Authentication Schemes

The RADIUS protocol can be used to implement CHAP or PAP based authentication.

PAP Overview

The Password Authentication Protocol (PAP) provides a simple method for a user to authenticate using a 2-way handshake. PAP only executes this process during the initial link to the authenticating server. With this scheme, an Id/Password pair is repeatedly sent by the user's machine to the authenticating server until authentication is acknowledged or the connection is terminated.

This authentication method is most appropriately used where a plain text password must be available to simulate a login at a remote host. In such use, this method provides a similar level of security to the usual user login at the remote host.

CHAP Overview

CHAP (Challenge-Handshake Authentication Protocol) is a more secure authentication scheme than PAP. In a CHAP scheme, the following takes place in order to establish a user's identity:

1. After the link between the user's machine and the authenticating server is made, the server sends a challenge message to the connection requester. The requester responds with a value obtained by using a one-way hash function.
2. The server checks the response by comparing it against its own calculation of the expected hash value.
3. If the values match, the authentication is acknowledged; otherwise the connection is usually terminated.

At any time, the server can request the connected party to send a new challenge message. Because CHAP identifiers are changed frequently and because authentication can be requested by the server at any time, CHAP provides more security than PAP.

RADIUS CHAP/PAP Scheme Overview

The RADIUS CHAP/PAP scheme authenticates users by computing the digest of a user's password, and then comparing it to the CHAP password in the RADIUS packet. The digest consists of the user's hashed password, which is calculated using a directory attribute specified during the configuration of the RADIUS CHAP/PAP authentication scheme.

RADIUS CHAP/PAP Scheme Prerequisites

In order to use the RADIUS CHAP/PAP authentication scheme, the following prerequisites must be met:

- The field in the user directory specified for the clear text password must contain a value.

Configure a RADIUS CHAP/PAP Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a RADIUS CHAP/PAP authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select RADIUS CHAP/PAP Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—RADIUS CHAP/PAP Scheme Setup Tab](#) (see page 622).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

RADIUS Server Authentication Schemes

SiteMinder supports the RADIUS protocol by using the Policy Server as the RADIUS server and an NAS client as the RADIUS client. RADIUS Agents allow the Policy Server to communicate with the NAS client devices. In the RADIUS Server authentication scheme the Policy Server acts as a RADIUS server attached to the SiteMinder protected network.

This scheme accepts user name and password as credentials. Multiple instances of this scheme can be defined. This scheme does not interpret RADIUS attributes that may be returned by the RADIUS server in the authentication response.

For more information on RADIUS Server authentication with SiteMinder, see the material on using the Policy Server as a RADIUS server in the *SiteMinder Policy Server Management Guide*.

RADIUS Server Scheme Prerequisites

In order to use the RADIUS Server authentication scheme, the following prerequisites must be met:

- The RADIUS server must be on a network accessible by the Policy Server.

Configure a RADIUS Server Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a RADIUS Server authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select RADIUS Server Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—RADIUS Server Template Scheme Setup Tab](#) (see page 623).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

SafeWord Server Authentication Schemes

This Authentication Scheme authenticates users against a SafeWord Server, including users who are logging in via the SafeWord hardware tokens. You can define multiple instances of this scheme. The exact configuration parameters of the SafeWord Server are specified by the SafeWord configuration file.

Note: Safeword authentication schemes (smauthenigma and smauthenigmahtml) are only supported on Windows and Solaris platforms after the 6.0 SP3 CR03 release.

SafeWord Server Scheme Prerequisites

In order to use the SafeWord authentication scheme, the following prerequisites must be met:

- The SafeWord Server must be installed on a network accessible by the Policy Server.
- The exact location of the SafeWord Server must be specified in the SafeWord configuration file.

Configure a SafeWord Server Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a SafeWord Server authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select SafeWord Server Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—SafeWord Template Scheme Setup Tab](#) (see page 624).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

SafeWord Server and HTML Forms Authentication Schemes

This Authentication Scheme authenticates users against a SafeWord Server in conjunction with a custom HTML form, including users who are logging in via the SafeWord hardware tokens. You can define multiple instances of this scheme. The exact configuration parameters of the SafeWord Server are specified by the SafeWord configuration file.

SafeWord Server and HTML Forms Scheme Prerequisites

In order to use the SafeWord authentication scheme, the following prerequisites must be met:

- The SafeWord Server must be installed on a network accessible by the Policy Server.
- The exact location of the SafeWord Server must be specified in the SafeWord configuration file.
- A customized .fcc file must reside on a Web Agent server in the cookie domain in which you want to implement HTML Forms authentication. CA provides sample .fcc files under the Samples/Forms subdirectory, where you installed your Web Agent. For example:

`<SM_Web_Agent_Install_Dir>\Samples\Forms`

- A customized .unauth file must reside on the Web Agent server unless the .fcc file uses the smerrorpage directive.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- The default HTML forms library must be installed to handle HTML Forms authentication processing:
 - SmAuthHTML.dll on Windows
 - smauthhtml.so on Solaris

These files are installed automatically when you configure a Web Agent. For more information, see the *Policy Server Installation Guide*.

- (For Oracle iPlanet Web Servers [formerly Sun Java Systems]) Increase the value of the StackSize parameter in the magnus.conf file to a value greater than 131072. Failing to change the value causes the web server to dump its core and restart each time SiteMinder makes an authentication request using forms.

More information:

[SiteMinder FCC Files](#) (see page 249)

[User Directories](#) (see page 141)

Configure a SafeWord Server and HTML Forms Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a SafeWord Server and HTML Forms authentication scheme:

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select SafeWord Server and HTML Form Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—SafeWord HTML Form Template Scheme Setup Tab](#) (see page 625).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

SecurID Authentication Schemes

The RSA Ace/SecureID authentication schemes authenticate users logging in with ACE credentials, which include the user names, PINs, and TOKENCODEs. ACE usernames and passwords reside in the ACE/Server user store and can be changed by ACE/Server administrator. Single-use TOKENCODEs are generated by SecureID tokens.

SiteMinder supports the following SecurID authentication schemes:

SecurID Authentication

This Authentication Scheme authenticates users who are logging in via RSA SecureID hardware tokens. The scheme accepts a user name and password. The password is the user's token PIN followed by the dynamic code.

Note: The SecurID Template authentication scheme will fail to authenticate a user if the user does not use the user directory attribute named 'uid' to map to the userid of the user in the Ace Server. Workaround: Use the SecurID HTML Forms Template when mapping LDAP directory attributes other than uid to the userid of the user in the Ace Server.

SecurID Authentication with HTML Forms Support

SiteMinder supports a scheme for SecurID authentication that includes HTML forms support. The HTML forms provide a method for users to identify themselves and reactivate their accounts after they have been disabled because they entered their PIN or token information incorrectly.

The RSA ACE/SecurID scheme authenticates users who are not allowed to change their ACE PIN. However, users allowed or required to change their PIN are authenticated through the RSA ACE/SecurID scheme with HTML form. Both schemes are based on ACE/Server - Ace/Agent model and require RSA/SecurID (tokens) and RSA/ACE (server software) products.

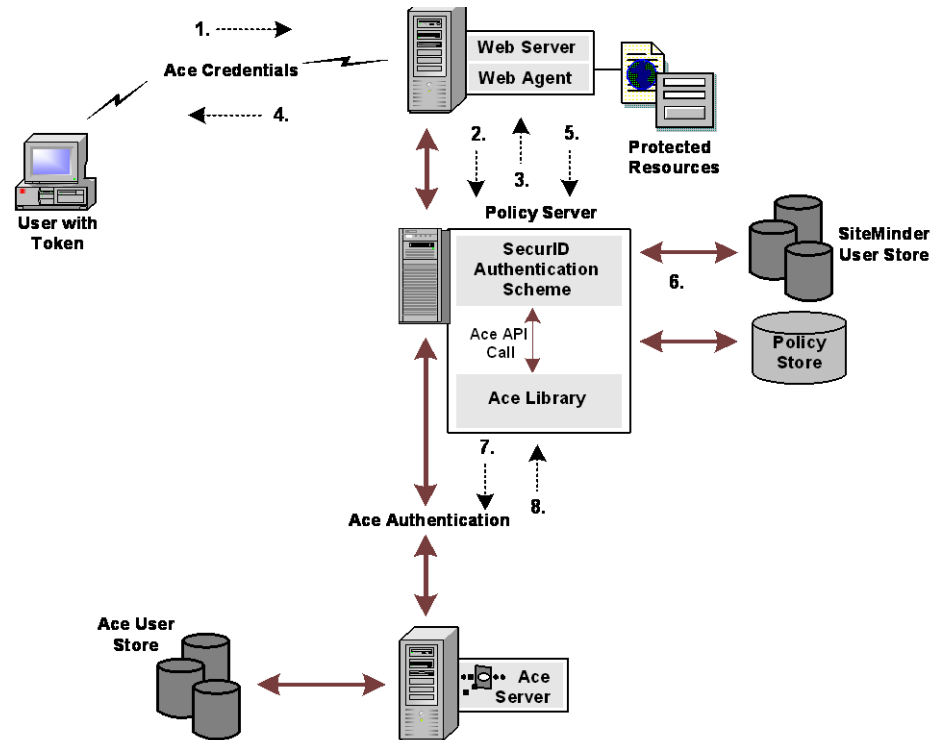
The RSA ACE/Server works with RSA SecurID tokens to authenticate users' identities through valid RSA Ace/Agents. Further, RSA supports Web Agents and custom Agents. SiteMinder SecurID authentication schemes act as custom ACE/Agents and use ACE/SDK and libraries.

The ACE/Server, however, has its own policy in handling users' credentials and stores this information in the ACE user store. The policy includes several requirements and limitations for each user, which are defined by the ACE administrator that can be changed at any time. The administrator configures users to authenticate by either PIN or TOKENCODE. If users are configured to authenticate by PIN, the system does not require a TOKENCODE. If users are configured to authenticate with TOKENCODE, both the TOKENCODE and PIN are required. In addition, users could be required to set a new PIN while trying to authenticate. Under this "new PIN mode", the old and new PINs are required.

To successfully authenticate users, SiteMinder SecureID authentication schemes require mapping between the ACE and SiteMinder users. This mapping is represented as an authentication scheme object attribute in the SiteMinder policy store.

In 5.5 SP1, the enhancement to the RSA ACE/SecurID Scheme with HTML form affects the "new PIN mode".

The following figure shows how the RSA ACE/Server interacts with SiteMinder:



1. A user requests a resource such as a Web page.
2. The SiteMinder Web Agent determines if the resource is protected.
3. The Policy Server indicates the requested resource is protected by the RSA ACE/SecurID Authentication Scheme with HTML form.
4. The Web Agent prompts the user for credentials.
5. The user enters credentials and the Agent sends them to the Policy Server.
6. The Policy Server performs user disambiguation and maps the SiteMinder user name to RSA/ACE user name. At this point, the Policy Server does not apply SiteMinder password policies.
7. The Policy Server forwards the authentication request to the ACE/Server by making a sequence of ACE API calls. It also send the user's credentials to the ACE/Server.
8. The ACE/Server indicates that the user is required to change the PIN and then returns control back to the Policy Server.
9. The Policy Server returns control to the Web Agent and then the Agent redirects the user to a CGI or JSP.
10. The CGI or JSP generates the applicable HTML form, presents it to the user, and then prompts for a user name, old PIN, new PIN, and new PIN confirmation.

11. The Web Agent sends a new set of credentials to the Policy Server.
12. The Policy Server makes a new sequence of API calls to the ACE/Server.
13. If new the PIN is accepted, then the ACE API call returns success. From here, the user is asked to login with the new PIN and the authentication process is complete.
14. If new PIN is rejected, Step 10 to Step 12 are repeated.

The PIN can be rejected if it:

- Is too long
- Is too short
- Contains alphabetic characters

In the process illustrated in the previous figure authentication includes presenting back-end PIN policy-related messages on the HTML form. The PIN policy validation is done by the SecurID Authentication scheme before sending a new PIN to the ACE/Server. The ACE SDK has a set of functions that can retrieve the following PIN attributes:

- Maximum length
- Minimum length
- Alphabetic and numeric or only numeric characters

Based on this information, the PIN is validated and the appropriate error messages are constructed and presented to the user, and the validation is done in the following order:

- The PIN is not too long
- The PIN is not too short
- The PIN does not't have invalid characters

In the 5.5 SP1, only the relevant portion of the policy is made available to users in these new error messages. The following shows an example of how these new messages might appear:

If a sample user, Joe, has a PIN that is 5 to 8 digits long but enters a new PIN as "poem", then he would see the following error message:

Your new PIN is too short. PIN must contain at least 5 character(s).

If the next PIN he entered was "novel", he would get:

Your new PIN may not have alphabetic characters.

If the next PIN he entered was "123412341234," he would see.

Your new PIN is too long. PIN must contain 8 or fewer character(s).

It is possible that, despite the successful PIN validation by the SecureID Authentication scheme, the ACE/Server will reject a new PIN. In this case, the user is asked for a new set of credentials, but no reason is given. Further, in the enhanced SecurID Authentication scheme, the user is automatically granted access to the target Web page after a successful PIN change.

SecurID Scheme Prerequisites

To use the SecurID authentication scheme, the following prerequisites must be met:

- On Windows Policy Servers, the RSA ACE/Client software must be installed on the same machine as the Policy Server. For information about supported RSA ACE/Client versions, see the Platform Support Matrix on the [CA Support](#) site.

Note: The SM_ACE_FAILOVER_ATTEMPTS environment variable, which is used to set the failover attempts to the ACE server, has been removed.

Configure a SecurID Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a SecurID authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).

From the Authentication Scheme Type drop-down list, select SecurID Template.

3. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—SecurID Template Scheme Setup Tab](#) (see page 627).
4. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

SecurID with HTML Forms Support Scheme Prerequisites

In order to use the SecurID with HTML support authentication scheme, the following prerequisites must be met:

- The RSA ACE/Client software must be installed on the same machine as the Policy Server.
- For Policy Servers on Windows, the ACE configuration information file (sdconf.rec) must reside in the winnt\system32 directory and point the VAR_ACE and USR_ACE variables to the appropriate ACE agent data and prog directories respectively.
- For Policy Servers on UNIX, the sdconf.rec file should be in the *<policy server installation dir>/bin* directory. In addition, the VAR_ACE and USR_ACE variables should point to the *<policy server installation dir>/bin*, allowing the Policy Server to use SiteMinder API libraries and not the ACE agent.
- A customized .fcc file must reside on a Web Agent server in the cookie domain in which you want to implement HTML Forms authentication. CA provides sample .fcc files under the Samples/Forms subdirectory, where you installed your Web Agent. For example:

```
<SM_Web_Agent_Install_Dir>\Samples\Forms
```
- A customized .unauth file must reside on the Web Agent server unless the .fcc file uses the smerrorpage directive.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- The default HTML forms library must be installed to handle HTML Forms authentication processing:
 - SmAuthHTML.dll on Windows
 - smauthhtml.so on Solaris

These files are installed automatically when you configure a Web Agent. For more information, see the *Policy Server Installation Guide*.

More information:

[SiteMinder FCC Files](#) (see page 249)

[User Directories](#) (see page 141)

Configure a SecurID and HTML Forms Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a SecurID and HTML Forms authentication scheme

1. Open the Authentication Scheme Dialog.

2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select SecurID and HTML Form Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—SecurID HTML Form Template Scheme Setup Tab](#) (see page 628).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Forms Support for Re-activating and Verifying SecurID Users

If you protect a realm with the SecurID and HTML Forms scheme, users who are suspended due to improper logins can attempt to activate their accounts using a number of customizable HTML forms provided with SiteMinder. You can modify the layout and wording of these forms, but you must not modify the tags that gather user information.

The forms provided with SiteMinder include the following:

PWLogin.template

This form is where users enter a username and passcode to login.

PWNextToken.template

This template requests multiple token codes to confirm that the user is in possession of a working SecurID token.

Forms Support for Activating New User Accounts

The following forms are used by SiteMinder when an administrator creates a new user account, and that user logs in. Through the forms, a user creates a PIN, or has SiteMinder generate a random PIN.

PWSystemPIN.template

For new users, or users whose accounts have been suspended (due to too many invalid login attempts), this template prompts the user to acquire a new PIN. This template accepts the user's original username and passcode, but instead of granting access to a protected resource, it redirects the user to another form where the user can receive or create a new PIN.

PWNewPINSelect.template

This template allows a user to indicate if the system should generate a new PIN, or if the user wants to enter a new PIN.

PWUserPIN.template

This template allows a user to enter a new PIN. It requires that the user provide a valid username and passcode along with the new PIN. In this template, \$USRMSG\$ is replaced with instructions for creating a new PIN number. For example:

PINs must be between 4 and 8 characters in length.

PWPINAccept.template

This template indicates that a new system-generated PIN has been created. In this template, \$USRMSG\$ is replaced by the system generated PIN.

When a user clicks Continue, the user is immediately prompted to log in using the new PIN.

X.509 Client Certificate Authentication Schemes

X.509 client certificates provide cryptographic proof of a user's identity. The certificate, supplied by a certificate vendor, is unique, and can be used to identify the user who attempts to access a protected resource. It contains the following information:

- Name of the subscriber (user)—This is a unique name called the distinguished name.
- Public key of the subscriber
- Operational period for the certificate
- Name of the Certificate Authority that issued the certificate
- Certificate serial number

The X.509 Client Certificate authentication scheme is the method SiteMinder uses to implement certificate authentication. SiteMinder's X.509 Client Certificate authentication consists of two parts:

- Establishing an SSL connection and collecting the client certificate information.
- Identifying a user in a directory service based on the client certificate and optionally verifying this certificate with the directory.

How SiteMinder Establishes an SSL Connection for Certificates

The SiteMinder Web Agent establishes an SSL connection by redirecting the user's browser to an HTTPS:// URL pre-configured by SiteMinder. The SSL connection to this URL is set up to require a client certificate to be submitted by the browser. The SiteMinder component invoked at this URL extracts information from the certificate delivered as part of the SSL connection and conveys this information to the SiteMinder Web Agent.

More information:

[Authentication over SSL](#) (see page 236)

How SiteMinder Identifies Users Based on Certificates

Once SiteMinder establishes an SSL connection between the SSL certificate server and the user, the SiteMinder Web Agent collects the user's certificate information and passes it to the Policy Server for verification. The certificate is then mapped to a user DN in a directory. For LDAP directories, you can configure the Policy Server to verify that the same certificate is associated with the user DN in the directory and to check the appropriate Certificate Revocation List (CRL) to ensure the certificate has not been revoked. For WinNT directories you can configure the Policy Server to check CRLs.

X.509 Client Certificate Scheme Prerequisites

In order to use the X.509 Client Certificate authentication scheme, the following prerequisites must be met:

- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser (HTTPS protocol).
- X.509 client certificates must be installed on client browsers.
- Trust must be established between client certificates and server certificates.
- Certificate must be issued by a valid and trusted Certification Authority (CA).
- The issuing CA's public key must validate the issuer's digital signature.
- Client and server certificates must not have expired.
- The user's public key must validate the user's digital signature.

Note: For Apache Web servers where Certificates are required or optional, the SSL Verify Depth 10 line in the httpd.conf file must be uncommented.

Configure an X.509 Certificate Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an X.509 Certificate authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select X509 Cert Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—X509 Client Cert Template Scheme Setup Tab](#) (see page 631).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

X.509 Client Certificate and Basic Authentication Schemes

The X.509 Client Certificate and Basic authentication scheme combines Basic authentication and X.509 Client Certificate authentication. This authentication scheme provides an extra layer of security for critical resources. In order for a user to authenticate successfully, the following two events must occur:

- The user's X.509 client certificate must be verified.
- AND
- The user must provide a valid user name and password.

For X.509 Client Certificate authentication, SiteMinder processes authentication using the following steps:

1. The Policy Server instructs the SiteMinder Web Agent to redirect the user to an SSL server and map the user's certificate to the server.
2. SiteMinder verifies the user exists.
3. SiteMinder verifies the user's basic credentials.
4. SiteMinder verifies that the certificate credentials and the basic credentials represent the same user.

More information:

[Basic Over SSL Authentication Schemes](#) (see page 243)

[X.509 Client Certificate Authentication Schemes](#) (see page 235)

X.509 Client Certificate and Basic Scheme Prerequisites

In order to use the X.509 Client Certificate and Basic authentication scheme, the following prerequisites must be met:

- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser (HTTPS protocol).
- X.509 client certificates must be installed on client browsers.
- Trust must be established between client certificates and server certificates.
- Certificate must be issued by a valid and trusted Certification Authority (CA).
- The issuing CA's public key must validate the issuer's digital signature.
- Client and server certificates must not have expired.
- The user's public key must validate the user's digital signature.
- Client user name and password information must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.

Note: For Apache Web servers where Certificates are required or optional, the SSL Verify Depth 10 line in the httpd.conf file must be uncommented.

More information:

[User Directories](#) (see page 141)

Configure an X.509 Certificate and Basic Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an X.509 Certificate and Basic authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).

3. From the Authentication Scheme Type drop-down list, select X509 Cert and Basic Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—X509 Client Cert and Basic Template Scheme Setup Tab](#) (see page 632).
5. Click OK to save your changes and close the Authentication Scheme Dialog.
The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

X.509 Certificate or Basic Authentication Schemes

The X.509 Client Certificate or Basic authentication scheme allows either Basic authentication or X.509 Client Certificate authentication to establish a user's identity. For a user to authenticate successfully, one of the following two events must occur:

- The user's X.509 client certificate must be verified.
- OR
- The user must provide a valid user name and password.

With this scheme, when a user requests a protected resource, the Web Agent challenges the user's browser to present a certificate. If the user does not have a certificate or chooses not to provide one (by clicking Cancel), the Web Agent challenges the user with the HTTP Basic protocol to obtain a user name and password.

This scheme is useful when you must gradually deploy X.509 certificates. For example, in a company with 50,000 users, it is a challenge to issue and deploy 50,000 certificates simultaneously. This scheme allows you to issue certificates as you see fit (500 or 5,000 at a time). During this transition period, your resources can be protected with certificates for those users who already have them, allowing other authorized users to access resources based on directory user names and passwords.

This scheme gives you the option of configuring the Basic authentication exchange to require an SSL connection.

Note: If you implement multiple certificate-based authentication schemes that include a mixture of X.509 Certificate OR Basic schemes, a browser caching limitation can cause unexpected behavior. When a user chooses not to use the certificate-based authentication for accessing a resource in a realm protected by a Certificate or Basic authentication scheme, the browser automatically caches this decision not to send the certificate. If the same user (using the same browser session) then attempts to access a resource that is protected by an authentication scheme with a mandatory certificate portion, such as X.509 Certificate, X.509 Certificate and Basic, or X.509 Certificate and Form, the user receives a "Forbidden" error message.

Because the user did not send a certificate for the certificate-based authentication when accessing the first resource, and the browser cached that decision, the user is automatically rejected when accessing the realm that requires the certificate.

Users who have valid certificates can be encouraged to use them when accessing resources in a deployment that includes a mixture of realms protected by certificate-based authentication schemes, including X.509 Certificate or Basic schemes and other certificate-based schemes that do not give the user a choice about send a certificate for authentication.

More information:

[Basic Authentication Schemes](#) (see page 234)

[X.509 Client Certificate Authentication Schemes](#) (see page 235)

X.509 Client Certificate or Basic Scheme Prerequisites

In order to use the X.509 Client Certificate or Basic authentication scheme, the following prerequisites must be met:

- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser (HTTPS protocol).
- X.509 client certificates must be installed on client browsers.
- Trust must be established between client certificates and server certificates.
- Certificates must be issued by a valid and trusted Certification Authority (CA).
- The issuing CA's public key must validate the issuer's digital signature.
- Client and server certificates must not have expired.
- The user's public key must validate the user's digital signature.
- Client user name and password information must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.

Note: For Apache Web servers where Certificates are required or optional, the SSL Verify Depth 10 line in the httpd.conf file must be uncommented.

More information:

[User Directories](#) (see page 141)

Configure an X.509 Certificate or Basic Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an X.509 Certificate or Basic authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select X509 Cert or Basic Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—X509 Client Cert or Basic Template Scheme Setup Tab](#) (see page 634).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

X.509 Client Certificate and HTML Forms Authentication Schemes

The X.509 Client Certificate and HTML Forms authentication scheme combines HTML Forms authentication and X.509 Client Certificate authentication. This authentication scheme provides an extra layer of security for critical resources. In order for a user to authenticate successfully, the following two events must occur:

- The user's X.509 client certificate must be verified.
- AND
- The user must provide the credentials requested by an HTML form.

For X.509 Client Certificate authentication, SiteMinder processes authentication using the following steps:

1. The Policy Server instructs the SiteMinder Web Agent to redirect the user to an FCC on an SSL-enabled web server.
2. The Web Agent presents the form.
3. The FCC passes the certificate and form back to the Policy Server.
4. The Policy Server verifies that the user in the certificate mapping exists.
5. The Policy Server verifies the user's HTML form credentials.
6. SiteMinder verifies that the certificate credentials and the HTML Forms credentials represent the same user.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

[X.509 Client Certificate Authentication Schemes](#) (see page 235)

X.509 Client Certificate and HTML Forms Scheme Prerequisites

In order to use the X.509 Client Certificate and HTML Forms, the following prerequisites must be met:

- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser (HTTPS protocol).
- X.509 client certificates must be installed on client browsers.
- Trust must be established between client certificates and server certificates.
- Certificate must be issued by a valid and trusted Certification Authority (CA).
- The issuing CA's public key must validate the issuer's digital signature.
- Client and server certificates must not have expired.
- The user's public key must validate the user's digital signature.
- Form credentials information must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- (For Oracle iPlanet Web Servers [formerly Sun Java Systems]) Increase the value of the StackSize parameter in the magnus.conf file to a value greater than 131072. Failing to change the value causes the web server to dump its core and restart each time SiteMinder makes an authentication request using forms.

Note: For Apache Web servers where Certificates are required or optional, the SSL Verify Depth 10 line in the httpd.conf file must be uncommented.

The certificate and forms data are collected and passed to the Policy Server together.

if...	then...
There is no certificate	SiteMinder issues error 500
The certificate and forms credentials are not accepted	SiteMinder issues error 500

Configure an X.509 Certificate and HTML Forms Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an X.509 Certificate and HTML Forms authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select X509 Cert and Form Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—X509 Client Cert and Form Template Scheme Setup Tab](#) (see page 637).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Agent API Support

The X.509 Client Certificate and HTML Forms uses the Sm_AuthApi_Cred_SSLRequired and the Sm_AuthApi_Cred_FormRequired bits.

X.509 Client Certificate or HTML Forms Authentication Schemes

The X.509 Client Certificate or HTML Forms authentication scheme allows either HTML Forms authentication or X.509 Client Certificate authentication to establish a user's identity. For a user to authenticate successfully, one of the following two events must occur:

- The user's X.509 client certificate must be verified.
- OR
- The user must provide the credentials requested by an HTML form.

With this scheme, when a user requests a protected resource, the Web Agent challenges the user's browser to present a certificate, after which the scheme has the following effect:

If...	then...
A certificate is presented	SiteMinder processes the certificate
A certificate is rejected	SiteMinder issues error 500
No certificate is presented	SiteMinder presents a form
The form is rejected	SiteMinder prompts again for a form

This scheme is useful when you need to gradually deploy X.509 certificates. For example, in a company with 50,000 users, it is a challenge to issue and deploy 50,000 certificates simultaneously. This scheme allows you to issue certificates as you see fit (500 or 5,000 at a time). During this transition period, your resources can be protected with certificates for those users who already have them, allowing other authorized users to access resources based on HTML forms credentials.

Note: If you implement multiple certificate-based authentication schemes that include a mixture of X509 Certificate OR Forms schemes, a browser caching limitation can cause unexpected behavior. When a user chooses not to use the certificate-based authentication for accessing a resource in a realm protected by a Certificate or Forms authentication scheme, the browser automatically caches this decision not to send the certificate. If the same user (using the same browser session) then attempts to access a resource that is protected by an authentication scheme with a mandatory certificate portion, such as X509 Certificate, X509 Certificate and Basic, or X509 Certificate and Form, the user receives a "Forbidden" error message.

Since the user chose not to send a certificate for the certificate-based authentication when accessing the first resource, and the browser cached that decision, the user is automatically rejected when accessing the realm that requires the certificate.

Users who have valid certificates can be encouraged to use them when accessing resources in a deployment that includes a mixture of realms protected by certificate-based authentication schemes, including X509 Certificate or Forms schemes and other certificate-based schemes that do not allow a user to choose whether to send a certificate for authentication.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

[X.509 Client Certificate Authentication Schemes](#) (see page 235)

X.509 Client Certificate or HTML Forms Scheme Prerequisites

In order to use the X.509 Client Certificate or HTML Forms authentication scheme, the following prerequisites must be met:

- An X.509 Server Certificate must be installed on the SSL Web server.
- The network must support an SSL connection to the client browser (HTTPS protocol).
- X.509 client certificates must be installed on client browsers.
- Trust must be established between client certificates and server certificates.
- Certificates must be issued by a valid and trusted Certification Authority (CA).
- The issuing CA's public key must validate the issuer's digital signature.
- Client and server certificates must not have expired.
- The user's public key must validate the user's digital signature.
- User attributes requested by the HTML form must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- (For Oracle iPlanet Web Servers [formerly Sun Java Systems]) Increase the value of the StackSize parameter in the magnus.conf file to a value greater than 131072. Failing to change the value causes the web server to dump its core and restart each time SiteMinder makes an authentication request using forms.

More information:

[User Directories](#) (see page 141)

Configure an X.509 Certificate or HTML Forms Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an X.509 Certificate or HTML Forms authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select X509 Cert or Form Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template Scheme Setup Tab](#) (see page 639).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

Note: For Apache Web servers where Certificates are required or optional, the SSL Verify Depth 10 line in the httpd.conf file must be uncommented.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Agent API Support

In the Agent API, the value `Sm_Api_Cred_CertOrForm` has been added to the enumerated type `Sm_Api_Credentials_t`. `Sm_Api_Credentials_t` specifies the credentials, if any, that are required for a user to access the realm referenced by the structure `Sm_AgentApi_Realm_t`. The enumerated type applies to the `nRealmCredentials` field of the structure.

The new value specifies that user authentication requires either an X.509 certificate or a forms-based authentication scheme.

Anonymous Authentication Schemes

The Anonymous authentication scheme allows SiteMinder to provide access privileges to users who are not yet identified in your network. Assigning an Anonymous authentication scheme to a realm does not provide access control, but it does allow SiteMinder to personalize content for the user.

When a user accesses a resource in a realm that uses the Anonymous scheme, the Policy Server assigns a Global Unique Identifier (GUID). This GUID is stored on the user's browser and provides a method for identifying the anonymous user.

When you create an Anonymous authentication scheme, you must specify a guest distinguished name (DN). You can bind policies to this guest DN that provide personalized content.

Note: Personalized content in a realm protected by an Anonymous scheme is based on the guest DN, not the GUID of the user. Anonymous users view content according to policies that include the guest DN. Identified users have a distinct DN, so an identified user who accesses the same resource (protected by an anonymous scheme) views the content of the resource based on their unique DN rather than the guest DN.

More information:

[Grouping Resources in Realms](#) (see page 315)

Anonymous Scheme Prerequisites

In order to use the Anonymous authentication scheme, the following prerequisites must be met:

- A guest DN for anonymous users must exist in a user directory.
- A connection to the user directory must be configured using the SiteMinder User Directory dialog.
- If you want to track users according to GUIDs assigned by Anonymous authentication, you must enable user tracking in the SiteMinder Global Settings dialog (see the *SiteMinder Policy Server Management Guide*).

More information:

[User Directories](#) (see page 141)

Configure an Anonymous Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an Anonymous authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).

3. From the Authentication Scheme Type drop-down list, select Anonymous Template.
4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—Anonymous Template Scheme Setup](#) (see page 642).
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Custom Authentication Schemes

If you want to use an authentication method that is not provided by SiteMinder, you can create a custom authentication scheme. Once you create a Custom scheme, you must configure the scheme in the SiteMinder Authentication dialog.

Note: For information on configuring an smauthetsso custom authentication scheme, which is needed for enabling single sign-on from CA Single Sign-On to SiteMinder, see [CA SSO/WAC Integration](#) (see page 977).

Note: If you have installed the Software Development Kit, see the *Developer's Guide for C* for information about creating a custom authentication scheme.

Custom Scheme Prerequisites

The prerequisites of a Custom authentication scheme must be determined when you create the scheme using CA's APIs. Prerequisites will differ between authentication schemes.

Configure a Custom Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a Custom authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select Custom Template.

4. Complete the fields in the Scheme Setup tab as described in [Authentication Scheme Dialog—Custom Template Scheme Setup Tab](#) (see page 644).
5. Click OK to save your changes and close the Authentication Scheme Dialog.
The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Federation Security Services Authentication Schemes

If you have purchased and installed Federation Security Services, the following authentication schemes are added to the existing SiteMinder schemes:

- SAML Artifact Template
- SAML POST Template
- SAML 2.0 Template
- WS-Federation Template

For more information about these authentication schemes, see the *Federation Security Services Guide*.

TransactionMinder Authentication Schemes

If you have purchased and installed TransactionMinder, the following authentication schemes are added to the existing SiteMinder schemes:

- XML Document Credential Collector (XML DCC)
- XML Digital Signature (XML-DSIG)
- SAML Session Ticket
- WS-Security

For more information about these authentication schemes, see the *TransactionMinder Operations Guide*.

Impersonation Authentication Schemes

By configuring a series of Policy Server objects, you can allow privileged users to impersonate other users. This feature is useful in situations where a helpdesk or customer service representative must troubleshoot problems for a customer, or when an employee is out of the office. Part of the impersonation process requires an impersonation authentication scheme which allows a privileged user to begin the impersonation process, identify the user to be impersonated (impersonatee), and establish an impersonation session. This authentication scheme is similar to the HTML Forms authentication scheme.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)
[Impersonation](#) (see page 457)

Impersonation Scheme Prerequisites

To use the Impersonation authentication scheme, the following prerequisites must be met:

- A customized .fcc file must reside on a Web Agent server in the cookie domain in which you want to implement impersonation. CA provides sample .fcc files under the /forms subdirectory, where you installed your Web Agent. For example, on Windows systems:

 <SM_Web_Agent_Install_Dir>\samples\forms\

For general details about composing .fcc files, see [SiteMinder FCC Files](#) (see page 249). For information about specific .fcc file requirements for impersonation, see [Enable Impersonation through an .fcc File](#) (see page 466).
- Connections to the user directories containing impersonators and impersonatees must be configured using the SiteMinder User Directory dialog.
- The default HTML forms library must be installed to handle authentication processing:
 - smauthimpersonate.dll on Windows
 - smauthimpersonate.so on Solaris

These files are installed automatically when you configure a Web Agent. For more information, see the *Policy Server Installation Guide*.

Note: Impersonation is not supported by directory mapping. The impersonatee, the user being impersonated, must be uniquely present in the authentication directories associated with the domain or the impersonation fails.

Note: If a password policy applies to an impersonatee and is in effect, the Impersonation authentication scheme fails.

More information:

[User Directories](#) (see page 141)

Configure an Impersonation Authentication Scheme

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure an Impersonation authentication scheme

1. Open the Authentication Scheme Dialog.
2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).
3. From the Authentication Scheme Type drop-down list, select Impersonation Template.
4. Complete the fields in the Scheme Setup tab as described in Authentication Scheme Dialog—Impersonation Template Scheme Setup Tab.
5. Click OK to save your changes and close the Authentication Scheme Dialog.

The Authentication scheme may now be assigned to a realm.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)
[Realms](#) (see page 315)

Certificate Mapping

X.509 client certificates provides strong user authentication. However, in order for SiteMinder to use a certificate to identify a user, the certificate must be compared to a user's information in a directory. SiteMinder uses a certificate mapping to determine how to compare a user's certificate with the information stored in the user directory.

SiteMinder supports certificate mapping for users whose authentication information is stored in a WinNT, Microsoft SQL Server, Oracle, or LDAP user directory. A certificate mapping defines how data in the certificate is mapped to form a user Distinguished Name (DN). The Policy Server uses this user DN to authenticate the user.

If certificates are stored in an LDAP directory, a certificate mapping can direct the Policy Server to verify that the certificate presented by the user matches the certificate associated with the user DN in the LDAP directory.

More information:

[User Directories](#) (see page 141)

Configure a Certificate Mapping

After meeting the prerequisites described in the previous section, you can configure the scheme.

To configure a certificate mapping

1. Navigate to the Certificate Mappings Dialog.
2. Complete the fields described in Certificate Mappings Dialog Fields and Controls.
3. Once you have configured the certificate mapping, click Apply to save your settings.
4. Click Test to test the mapping.
5. Click OK to save your changes and close the dialog.

Note: When specifying the Issuer DN, prefix the following reserved characters with a backslash or escape character (\):

- Comma (,)
- Semicolon (;)
- Quotes (")
- Backslash (\)
- Plus sign (+)
- Greater than sign (>)
- Less than sign (<)

For more information about reserved characters in LDAP distinguished names, see the following link: <http://www.faqs.org/rfcs/rfc2253.html>.

More information:

[Navigate to the Certificate Mappings Dialog](#) (see page 666)

[Test a Certificate Mapping](#) (see page 295)

Test a Certificate Mapping

Once you configure a certificate mapping, you can test the mapping using the Test button located in the Mapping group box of the Certificate Mapping dialog. This opens the Certificate Map Test dialog which displays the search string that the Policy Server will use to map client certificates to user directory attributes.

To test a certificate mapping

1. Navigate to the Certificate Mappings Dialog.
2. In the Mapping group box, click the Test button.

The Certificate Map Test dialog opens.

3. From the Directory drop-down list, select a user directory connection.

The user directory connections available in the drop-down list include all existing directory connections of the type you selected in the Map from IssuerDN in Certificate to Directory Type group box of the Certificate Mapping dialog.

The contents of the Directory Information group box change based on the type of user directory connection you are testing. For WinNT, ODBC and OCI user directory connections, the group box displays the Directory Type you are testing. For LDAP directory connections, the group box displays the Directory Type, as well as the Lookup Start and Lookup End values used to locate a user's DN within the LDAP directory.

When you select a user directory connection, the Certificate Map Test dialog displays information about the certificate mapping.

4. Click OK to close the Certificate Map Test dialog.

More information:

[Navigate to the Certificate Mappings Dialog](#) (see page 666)

[Certificate Map Test Results](#) (see page 296)

Certificate Map Test Results

When you select a user directory connection in the Certificate Map Test dialog, the Policy Server tests the mapping and provides the following set of results:

Mapping Information group box

Contains mapping information that you specified in the Mapping group box of the Certificate Mapping dialog.

Mapping Type

Displays the mapping type you selected for the directory mapping. Possible types include Single Attribute, Custom, and Exact.

Mapping

Displays the attribute or custom mapping you specified in the Certificate Mapping dialog.

String Sent to Directory for Authentication group box

Displays the string that the Policy Server compares to the user directory for authentication.

How SiteMinder Confirms the Validity of Certificates

SiteMinder can confirm the validity of certificates using either of the following two methods:

Certificate Revocation List (CRL) checking

A certificate revocation list (CRL) is a list of revoked X.509 client certificates published by the CA to an LDAP user directory. Comparing certificates against CRLs is one method of ensuring that certificates are valid.

Note: The Policy Server can support CRLs greater than 1.7 MB in size.

Online Certificate Status Protocol (OCSP) checking

The OCSP protocol enables OCSP-enabled applications (such as SiteMinder) to determine the (revocation) state of an identified X.509 client certificate in a more timely manner than is possible with CRLs. When a user with such a certificate tries to access a protected resource, SiteMinder issues a status request to an OCSP responder and suspend acceptance of the certificate in question until the responder provides a response.

Certificate Revocation Lists can get large and their propagation can therefore become slow. OCSP checking provides real-time status information and lessens network traffic significantly, but requires access to an OCSP Responder.

Configure Certificate Revocation List Checking

If SiteMinder is configured to perform CRL checking, when a user with an invalid client certificate tries to access a protected resource, SiteMinder finds the user's certificate in the CRL and rejects the authentication.

Note: The Policy Server cannot verify the status of a certificate using a Certificate Revocation List (CRL) that is larger than 64 KB. This limit is due to the third party libraries that are used to parse CRLs.

SiteMinder compares certificates against CRLs stored in an LDAP directory. SiteMinder verifies the signature of the CRL by retrieving the Certificate Authority's (CA) public certificate from the LDAP directory. SiteMinder supports the following RSA algorithms for signature verification:

- MD5
- MD2
- SHA1

CRLs must be kept up-to-date by CA administrators. If SiteMinder retrieves an expired CRL, all certificates from the CA with the expired CRL will be denied access. If multiple CRLs exist, SiteMinder will search for and use the most recent CRL. If a CA's public certificate is not available or your CRL is signed with an unsupported algorithm, you can turn off signature checking during the CRL verification process.

Note: If signature checking is turned off, make sure that the LDAP directory is protected appropriately.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure CRL checking

1. In the Certificate Revocation List Checking group box of the Certificate Mapping dialog, select the Perform CRL Checks check box.

CRL checking is an additional level of security that involves retrieving a CRL from an LDAP directory, verifying the CRL, and finally validating that the user's certificate has not been revoked. If the Perform CRL Checks check box is marked, the Policy Server processes CRL checking.

2. From the CRL Directory drop-down list, select the name of the LDAP user directory that contains the CRL.

The LDAP user directory connection must be defined using the SiteMinder User Directory dialog. If the user directory connection has not been defined, click Create to open the SiteMinder User Directory dialog and add the new directory connection.

3. In the DN in CRL Directory field, enter the DN of the CA if the DN is different from the DN you entered in the Issuer DN field.

It is possible to have a CA whose DN in the LDAP directory for CRL checking is different from the Issuer DN in the client certificate. In such a case, you must specify the DN of the CA for the LDAP CRL directory in this field.

4. Select the Verify Signature check box to enable signature verification.

When you enable signature verification, the Policy Server checks the CA's public certificate against a signature stored in the policy database.

5. If your CRL uses distribution points, select the Use Distribution Points check box.

Large CRLs may contain multiple distribution points that can be used to locate a revoked user. Distribution points indicate a starting point in the CRL LDAP directory. The distribution point provides a starting point for a CRL check and saves the processing time that it would take to search the entire CRL for a particular user.

When this check box is marked, SiteMinder looks at a user's certificate, and retrieves the distribution point from the certificate, then uses it to find the appropriate LDAP directory entry point for the CRL.

6. If you want SiteMinder to cache CRL entries, select the Cache check box.

When you mark this check box, SiteMinder checks the NextUpdate field in the CRL for the date when the cached CRL information should be deleted and replaced with updated CRL information. SiteMinder uses the cached CRL information until the date specified in the NextUpdate field of the CRL.

The NextUpdate field in the CRL is optional. If there is no value for NextUpdate, SiteMinder will not cache CRL information.

7. Click Apply to save your changes, or click OK to save your changes and return to the Certificate Mappings dialog.

The new mapping appears in the Current Mappings list of the Directory Mappings dialog.

More information:

[Configure User Directory Connections](#) (see page 156)

Configure Online Certificate Status Protocol Checking

SiteMinder can check the status of a certificate using Online Certificate Status Protocol (OCSP). During certificate checking, the Policy Server looks for the existence of an Issuer DN in a configuration file (smocsp.conf). If the Issuer DN is found, a certificate status check is made using a certified OCSP 1.0 Responder also specified in the smocsp.conf file. If the Issuer DN is not found in the configuration file, the certificate is considered to have passed OCSP checking.

To use OCSP checking, you will require:

- An established CA environment
- An OCSP Responder
- One or more LDAP directories containing OCSP Responder and Certificate Authority public certificates.

To configure OCSP certificate status checking

1. Turn off CRL checking. In the Certificate Revocation List Checking group box of the Certificate Mapping dialog, make sure that the Perform CRL Checks check box is not checked.
2. Create an smocsp.conf file in the Policy Server config directory (siteminder_installation_dir/config)

The smocsp.conf file must be an ASCII file containing one or more OCSPResponder records, each with the following format:

```
[
OCSPResponder IssuerDN C=US,O=U.S. Government,OU=DoD,OU=PKI,CN=DOD CLASS 3 CA-9
CACertDir localhost:389
CACertEP cn=DOD CLASS 3 CA-9,ou=PKI,ou=DoD,o=U.S. Government,c=US ResponderCertDir
localhost:389
ResponderCertEP cn=OCSP,ou=PKI,ou=DoD,o=U.S. Government,c=US ResponderCertAttr
cacertificate
ResponderLocation aristotle.jfcom.mil:80
]
```

where:

- Each OCSPResponder record must start with the tag name OCSPResponder.
- Attribute names are not case sensitive.
- The user directory is not the IP Address of the physical directory, but is the name of the user directory connection listed in the Policy Server User Interface.
- The existence of a value for IssuerDN indicates the existence of an OCSP Responder record | object.

- CACertDir and ResponderCertDir directories should exist as user directories in the Policy Server database.
- All attributes except for AIAExtension and ResponderLocation are mandatory. In the case of these two, ResponderLocation should exist or AIAExtension should be YES.
- The default (unspecified) for AIAExtension is NO.
- The priority of AIAExtension and ResponderLocation is as follows:
 - If AIAExtension is YES:
 1. AIAExtension is used for validation, if it is found in the certificate.
 2. ResponderLocation is used.
 - If AIAExtension is NO:

ResponderLocation will be used regardless of the value of AIAExtension in the userCertificate in the request.

```
[
OCSPResponder
IssuerDN <IssuerDN>
[AltIssuerDN <IssuerDN>]
CACertDir <Name of User Dir containing CA cert>
CACertEP <Entry point in CACertDir containing CA cert>
ResponderCertDir <Name of User Dir containing Responder cert>
ResponderCertEP <Entry point in ResponderCertDir containing Responder cert>
ResponderCertAttr <Directory attribute of Responder cert>
ResponderLocation <Server-name of Responder:port #>
AIAExtension<YES|NO>
]
```

More information:

[Configure Certificate Revocation List Checking](#) (see page 297)

Custom Mapping Expressions

You can use custom mapping expressions for complex multiple attribute mapping. This allows you to specify multiple user attributes that should be extracted from a user DN to establish a certificate mapping.

Note: Custom mapping expressions are also useful when simulating certificate-based authentications through authentication the SiteMinder Test Tool.

The syntax for a custom mapping expression is a parsing specification designed to enable full mapping flexibility. It indicates which information to take from the certificate and where it should be applied to in the user directory.

The basic syntax is as follows:

```
UserAttribute=%{CertificateAttribute},  
UserAttribute2=%{CertificateAttribute}
```

More information:

[Certificate-based Authentication Tests](#) (see page 531)

EnableCustomExprOnly Registry Key

When you create a custom certificate mapping for an LDAP user directory, the resulting search query string includes the LDAP User DN Lookup Start and End strings in addition to the Mapping Expression that you specify on the Create Certificate Mapping pane. The resulting query is invalid, as seen in the following example:

LDAP User DN Lookup Start

```
(samAccountName=
```

LDAP User DN Lookup End

```
)
```

Certificate Mapping Expression

```
(mail=%{E})
```

Resulting Search Query

```
(samAccountName=(mail=%{E}))
```

To omit the User DN Lookup Start and End strings from the search query, navigate to `\Netegrity\SiteMinder\CurrentVersion\PolicyServer\` and set the `EnableCustomExprOnly` registry key to 1. The resulting search query string is valid, as seen in this example:

Certificate Mapping Expression

```
mail=%{E}
```

Resulting Search Query

```
mail=%{E}
```

Note: If the `EnableCustomExprOnly` registry key is 0 (the default) or the key does not exist, the User DN Lookup Start and End strings are included in the resulting search query.

Enable LegacyCertMapping Registry Key

Using LDAP syntax to create search filters that contain logic operators requires you to enable the LegacyCertMapping registry key. Enabling the registry key allows legacy behavior in certificate mapping, which ensures that users are authenticated using the specified LDAP search criteria.

LegacyCertMapping

KeyType: DWORD

Values: 0 (disabled) and 1 (enabled)

Default: 0

To enable the registry key on Windows

1. Navigate to
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\
PolicyServer, and open LegacyCertMapping.
2. Edit the KeyType value to REG_DWORD.
3. Edit the Values value to 1.

Note: If a value other than 0x1 is set, or the registry value does not exist, the registry key is disabled.

4. Save the registry key.

LegacyCertMapping is enabled, and LDAP search filter syntax can be used with custom mapping.

To enable the registry key on UNIX

1. Open the sm.registry file.
2. Add the following lines to the file:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\  
PolicyServer=XXXXX
```

```
LegacyCertMapping=0x1 REG_DWORD
```

3. Save the file.

LegacyCertMapping is enabled, and LDAP search filter syntax can be used with custom mapping.

Example 1

If a user's certificate contains:

SubjectDN: CN=John Smith, UID=JSMITH, OU=development, O=CompanyA

You can specify the following custom mapping:

CN=%{UID}, OU=%{OU}, O=%{O}

The resulting UserDN is:

CN=JSMITH, OU=development, O=CompanyA

Example 2

The custom mapping syntax also handles more complex mappings, as illustrated in the example:

If the user's certificate contains:

Subject DN: CN=John Smith + UID=jsmith +EMAIL=jsmith@companyA.com,
ou=development, o=companyA

You can specify the following custom mapping:

CN=%{CN.CN}+UID=%{CN.UID}, OU=%{O}

The resulting UserDN is:

CN=John Smith+UID=JSMITH, OU=companyA

In the above example, the CN contained multiple attributes. The syntax indicated which components of the CN to take and apply to the UserDN's CN. This was done by specifying "CN.CN or CN.UID" This syntax indicates that the custom expression uses both the CN and UID parts of the CN.

Note: You cannot use the "+" operator to disambiguate multiple attributes in a user directory. The "+" operator is used like any other character in the user DN for a user that is present in the user directory.

Example 3

Static text can be represented in a custom expression by leaving it outside of the bracket notation as shown below.

The user's certificate contains:

Subject DN: CN=John Smith, UID=JSMITH, OU=development

You can specify the following mapping:

CN=%{UID}, OU=%{OU}, O=companyA

The resulting UserDN is:

CN=JSMITH, OU=development, O=CompanyA

For more information, see the next section.

Template String Usage

The template string is composed of text and hash-bracketed expressions `%{...}`. All text outside the brackets is returned unchanged. The hash-bracketed expressions are evaluated based on the following rules:

- Undistinguished variable names (e.g. DN) are resolved before being returned.
- Distinguished variable names (e.g. DN.UID) are resolved to the variable component before being returned.

Map to the Certificate Serial Number or IssuerDN

Certificate Mapping supports mapping of the CertSerialNumber and IssuerDN attributes, which are not part of the subjectDN. These attribute(s) in the subjectDN of user certificates can be mapped to default or custom user-attribute(s), such as UID or CN in the user directory.

To map these attributes, add the following in the Mapping Expressions field in the Certificate Mappings Properties dialog:

- CustomAttributeinLDAP1 = %{CertSerialNumber}

Custom Certificate Mapping for Multiple Attributes of the Same Type

Some certificates may have multiple attributes of the same type in their Subject DN. SiteMinder supports a simple method for using a custom certificate mapping to see attributes other than the first attribute of a particular type. The syntax is as follows:

`%{attribute_name}` for the first occurrence of attribute_name

`%{attribute_nameN}` for the Nth occurrence of attribute_name

If the Subject DN of the certificate contained `CN=user,ou=dev,sn=1234,sn=2345,sn=3456,o=company,c=us`, you could set up a custom certificate mapping to any of the sn attributes. For example, to map to the first sn, enter `%{sn}` as the custom mapping. To map to the second sn, you could enter `%{sn2}` as the custom mapping.

Map to Non-Required Attributes

Sometimes certificates for individuals may be slightly different. For example, some users may have two employee, while others have a single number. In these cases, you may want to map to the second of the numbers when a second attribute exists. You can do so using the following notation:

`%{attribute_name2/attribute_name}`

Using the example from above, you could enter `%{EN2/EN}` as a custom mapping to indicate that the second number in the Subject DN should be used if it exists, otherwise, the first occurrence of the Employee attribute should be used.

This notation can also be used to specify two different attributes that are acceptable for a certificate mapping. For example, to indicate that the EN should be used, but a CN may be used if the EN does not exist, you could enter `%{EN/CN}`.

Chapter 11: Domains

This section contains the following topics:

[Policy Domain Overview](#) (see page 307)

[Domains and User Membership](#) (see page 308)

[Configure a Policy Domain](#) (see page 309)

[Add Identity Manager Environments to a Policy Domain](#) (see page 312)

[Affiliate Domains](#) (see page 313)

[Domain Editing](#) (see page 313)

[Domain Deletion](#) (see page 313)

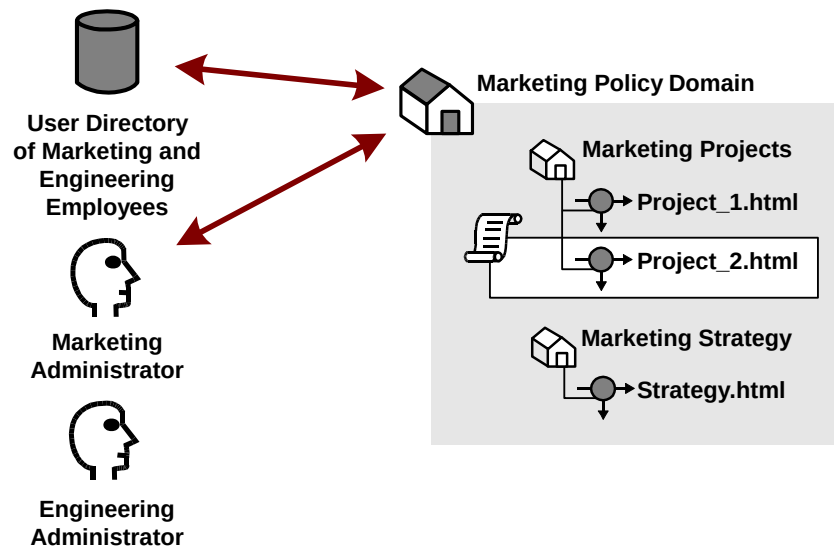
Policy Domain Overview

A policy domain is a logical grouping of resources associated with one or more user directories. In addition, policy domains require one or more administrator accounts that can make changes to the objects within the policy domain. Policy domains contain realms, rules, responses, and policies (and optionally, rule groups and response groups). An administrator with the Manage System and Domain Objects privilege can grant control over a policy domain to other administrators. For information about administrator privileges, see [Policy Server Administrators](#) (see page 223).

The resources in a policy domain can be grouped in one or more realms. A realm is a set of resources with a common security (authentication) requirement. Access to resources is controlled by rules, which are associated with the realm that contains the resource. The following diagram illustrates a small policy domain which contains realms and their associated rules, as well as a rule group, response group, and a pair of responses.

By grouping realms and rules in a policy domain, you can provide organizations with a secure domain for their resources. In the policies section, you learn how to create policies within a policy domain to control access to the policy domain's resources.

In the sample diagram below, a Marketing policy administrator who is specified in the Marketing policy domain can manage the Marketing Strategy and Marketing Projects realms. The policy domain ensures that the Engineering administrator, who does not have administrative privileges to manage the Marketing policy domain, cannot control resources belonging to the Marketing department. However, the Marketing policy domain is associated with a user directory that contains engineering users. If the administrator for the Marketing department creates a policy within the Marketing policy domain that allows Engineering staff to access the resource Project 2.html, engineering users may access the resource.



More information:

[Policies](#) (see page 375)

Domains and User Membership

Besides acting as a container for domain objects, policy domains also connect to user directories. The Policy Server authenticates users based on the requirements of the realm in which the target resource resides. In order to authenticate a user, the Policy Server must find the user directory where a user is defined. The Policy Server does this by locating the policy domain to which a realm belongs. From the policy domain, the Policy Server queries the user directories specified in the policy domain's search order.

The search order is defined when you add user directory connections to a policy domain. The order in which you add directory connections determines the order that the Policy Server uses to search for a user. For example, if you set up policy domain for a company migrating user data from a WinNT directory to an LDAP directory, and you want the Policy Server to search in the new LDAP directory first, then look in the WinNT user directory, add the LDAP directory connection to the policy domain first, then add the WinNT user directory connection. For more information about adding user directory connections to a policy domain, see [Assign User Directories](#) (see page 310).

More information:

[Authentication Schemes](#) (see page 231)

Configure a Policy Domain

When you configure a policy domain, you assign user directories and administrators. You can also create realms as described in the next chapter.

Note: You can edit a policy domain's properties if you need to add an administrator or realm in the future.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a new policy domain

1. Log into the Policy Server User Interface.
2. From the menu bar of the SiteMinder Administration window, select Edit, System Configuration, Create Domain.
The SiteMinder Domain dialog opens.
3. In the Name field, enter a name for the policy domain.
4. In the Description field, enter a brief description of the policy domain.

More information:

[Grouping Resources in Realms](#) (see page 315)

[Start the Policy Server User Interface](#) (see page 64)

[Domain Dialog](#) (see page 675)

Disable Global Policy Processing for a Domain

Global policies allow you to associate responses with particular resources and events across all domains. By default, global policies apply to all of the resources in a policy domain.

To disable global policies for a specific domain

1. In the SiteMinder Domain dialog, deselect the Global Policies Apply check box.
2. Click Apply.

More information:

[Global Policies, Rules, and Responses](#) (see page 435)
[Domain Dialog](#) (see page 675)

Assign User Directories

In the User Directories tab of the SiteMinder Policy Domain dialog you specify the user directories associated with a policy domain. When the Policy Server attempts to authenticate users, it compares their credentials to the directories you specify in this tab.

To assign user directories to a policy domain

1. Select the User Directories tab.
2. From the drop-down list box at the bottom of the tab, select a user directory you want to include in the policy domain.
3. Click the Add button.

The Policy Server User Interface adds the directory to the list displayed in the User Directories tab.

4. Repeat steps 2 and 3 for all user directories you want to associate with the policy domain.
5. Optionally, you can create a user directory from this dialog by clicking Create.

The User Directory dialog opens. Create the user directory as explained in User Directories on page 161. When you save the new User Directory and close the User Directory dialog, the directory you created appears in the User Directories tab in the SiteMinder Policy Domain dialog.

6. Click Apply.

The Policy Server User Interface saves your changes.

More information:

[User Directories Tab](#) (see page 677)

Assign an Administrator

When you create a new policy domain, you can assign administrators to the policy domain. These administrators may create, edit, and delete Policy Server objects within the domain.

To assign an Administrator to a policy domain

1. Select the Administrators tab.
2. From the drop-down list box at the bottom of the tab, select an administrator.
3. Click Add.

The Policy Server User Interface adds the administrator to the list in the top portion of the Administrators tab. When you save the policy domain, administrators included in the list on this tab can manage objects within the policy domain.

Note: Adding an administrator in the Administrators tab is equivalent to adding the policy domain to an administrator with a Scope of Some Domains and a Task of Manage Domain Objects.

4. (Optional) Create an administrator by clicking Create.

The Administrator dialog opens. Create the administrator as explained in [Configure an Administrator](#) (see page 226). When you save the new administrator and close the SiteMinder Administrator dialog, the Policy Server User Interface displays the administrator in the Administrators tab.

5. Click Apply.

The Policy Server User Interface saves your changes.

More information:

[Policy Server Administrators](#) (see page 223)

[Administrators Tab](#) (see page 677)

[Administrator Scope and Tasks](#) (see page 225)

Create a Realm

Policy domains contain zero or more realms. If you have not already done so, save the new policy domain by typing a name in the Name field and a brief description in the Description field, then clicking Apply.

Note: You cannot create a realm from within the Policy Domain dialog until the policy domain has been saved.

To create a realm

1. Select the Realms tab.
2. Click the Create button.

The Realm dialog opens. Create the realm as explained in [Configure a Realm](#) (see page 320). When you save the new realm and close the Realm dialog, the Policy Server User Interface displays the realm you created in the Realm tab.

3. Click Apply to save your changes, or click OK to save your changes and close the SiteMinder Policy Domain dialog.

More information:

[Grouping Resources in Realms](#) (see page 315)
[Realms Tab](#) (see page 678)

Add Identity Manager Environments to a Policy Domain

If you have purchased CA Identity Manager (Identity Manager), you may add Identity Manager Environments to a policy domain. Identity Manager Environments are views of management namespaces. These namespaces are constructed by Identity Manager administrators to define default roles, tasks, administrators, and properties for Identity Manager .

Note: You may only add an Identity Manager Environment to a policy domain if you have also added the corresponding user directory to the policy domain. For information about adding user directories to policy domains, see [Assign User Directories](#) (see page 310).

To add an Identity Manager Environment to a policy domain

1. In the Policy Domain dialog, select the Identity Manager Environments tab.
2. Select the Identity Manager Environment that you want to associate with the policy domain from the drop-down list at the bottom of the tab.

3. Click Add.

The Policy Server User Interface adds your selection to the list of Identity Manager Environments at the top of the tab.

Note: If the user directory associated with the Identity Manager Environment has not been added to the policy domain you will receive an error that instructs you to add the user directory before adding the Identity Manager Environment.

4. Click OK to save your selections and close the dialog.

The Identity Manager Environments that you selected will be available when you create policies. For information about using Identity Manager Environments with policies, see *CA Identity Manager Operations Guide*.

Affiliate Domains

If you have purchased SiteMinder Federation Security Services, you can configure Affiliate domains. An affiliate domain is a logical grouping of grouping of SAML 1.x consumers or SAML 2.0 Service Providers associated with one or more user directories. For more information, see the *SiteMinder Federation Security Services Guide*.

Domain Editing

You can change the name, description, user directory connections and administrators associated with a policy or affiliate domain. All other features of a domain are a result of peripheral configuration.

Domain Deletion

Important! When deleting a domain, you must remember that deletion of a domain *will also destroy* all of the domain connections (user directories, and administrators) and objects (rules, rule groups, realms, responses, response groups, and policies, or affiliates) contained in the domain.

Note: It may take a short amount of time for all deleted objects to be removed from caches.

Chapter 12: Grouping Resources in Realms

This section contains the following topics:

[Overview](#) (see page 315)

[Realms](#) (see page 315)

[Configure a Realm](#) (see page 320)

[Realm Editing and Deletion](#) (see page 325)

[Flush a Realm from the Resource Cache](#) (see page 326)

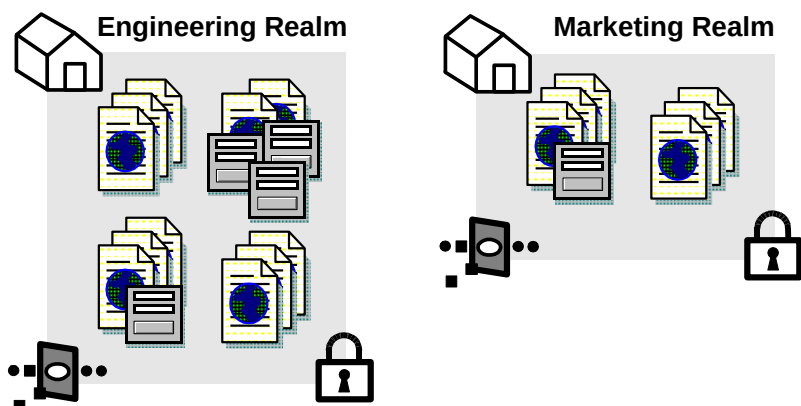
Overview

In order to manage access control for a complex set of resources that are available on the Internet or corporate intranet, that set of resource must be logically grouped so that security policies can be created. SiteMinder uses several object types to allow you to group resources. When grouping resources in SiteMinder, the basic groupings for your resources are realms.

Realms

A realm is a cluster of resources within a policy domain grouped together according to security requirements. A realm is usually defined for resources that reside in a common location on your network. For example, marketing information that resides in a /marketing directory on your network might be configured as a realm in a policy domain controlled by an administrator in your company's marketing organization.

The following diagram shows the contents of two realms.



Each of the realms contains resources, such as HTML files, forms, or applications. In addition, each realm is associated with an Agent and an authentication scheme.

- For information about Agents and Agent groups, see [Agents and Agent Groups](#) (see page 99).
- For information about authentication schemes, see [Authentication Schemes](#) (see page 231).

How Resources Are Identified by Agent, Realm, and Rule

The resources protected by SiteMinder are identified by the following:

[Agent] [Realm Resource Filter] [Rule Resource]

Agent

An Agent monitors a server that contains one or more realms of protected resources.

Realm Resource Filter

A string, such as a relative path to a directory, that specifies the resources covered by the realm. If the realm is a top-level realm, specify the resources relative to the server that serves up the files or application. If the realm is nested, specify the resources relative to the parent realm.

For example, the realm might cover the contents of a directory that is immediately below the document root of a Web server, such as:

```
<document_root>/HR
```

Here, you could specify the realm resource filter as:

```
/HR
```

Rule Resource

A string or regular expression that specifies the resources to which the rule applies. Specify the resources relative to the realm containing the resource. For example, if the realm resource filter ends with a directory name, the rule resource might include a subdirectory of the realm directory and even the name of a file in that subdirectory, such as:

```
/Managers/PayRanges.html
```

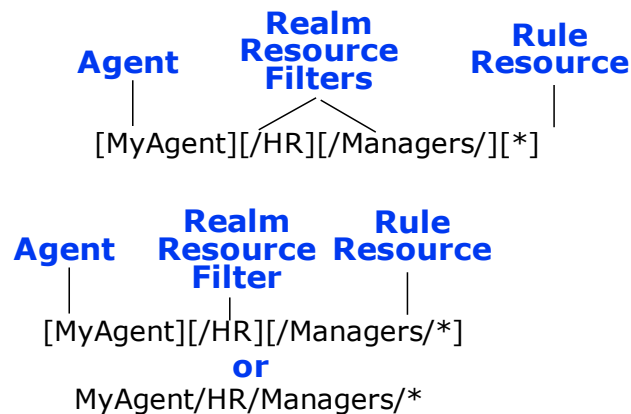
You can use wildcards to broaden the specification of a rule. For example:

```
/Managers/*
```


Combining the three elements, suppose that:

- The agent called MyAgent protects a Web server on host MyHost, in domain myorg.org.
- You want the realm to cover the contents of the following Web Server directory:
<document_root>/HR
- The HR directory contains a subdirectory called Managers, and you want to protect all files in the subdirectory.

For the Policy Server, the following figure shows the effective resource.



You could configure the directory called Managers as a nested realm under the /HR realm.

To access the protected page PayRanges.html, under the Managers subdirectory, a user would need to:

1. Specify the resource:
http://MyHost.myorg.org/HR/Managers/PayRanges.html
2. Provide credentials for a user authorized to access the resource. Administrators use policies to specify which users are authorized to access a resource.

More information:

[Agents and Agent Groups](#) (see page 99)

[Configure a Realm](#) (see page 320)

[Rules](#) (see page 327)

Unprotected Realms, Rules, and Policies

By default a realm is created in a Protected state. In most cases, you should use protected realms instead of changing a realm to an Unprotected state. In a protected realm, all resources are protected against access. To allow access, a rule must be defined, then included in a policy.

When you create a realm in an Unprotected state, you must configure rules before SiteMinder protects the resources in the realm. If you create a rule for resources in the unprotected realm, only the specified resources are protected. Once the resource is protected, the rule must be added to a policy to allow users to access the resource. You may want to use an unprotected realm if only a subset of the resources in a realm need to be protected from unauthorized access.

The following is an example of the actions required when setting up an Unprotected realm:

Action	Protection State
Create unprotected realm called Realm1 with the Resource Filter: /dir.	Resources contained in /dir and subdirectories are not protected.
Create Rule1 in Realm1 for the resource: file.html.	The file /dir/file.html is protected, but the rest of the contents of /dir are not protected.
Create Policy1 and bind Rule1 and User1 to the Policy.	User1 can access /dir/file.html. All other users cannot access the protected file.

Note: If you want to track users for a realm with an Anonymous authentication scheme, the realm must be a protected realm. For information about the Anonymous scheme, see [Anonymous Authentication Schemes](#) (see page 288).

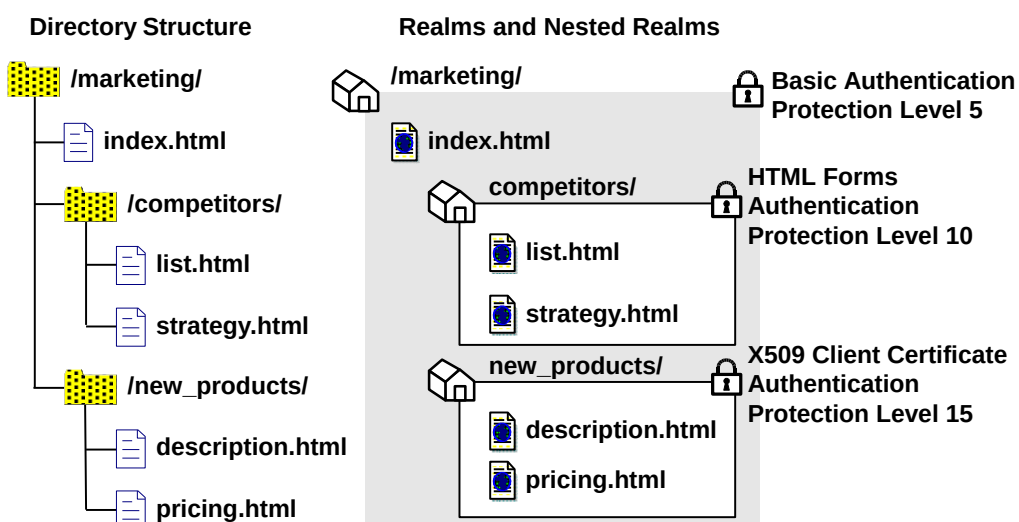
Nested Realms

In SiteMinder, realms represent groups of resources in much the same way that directories of files and folders represent a file system's contents. Nested realms allow you to increase the protection level of resources that are lower in a directory tree. Below any existing realm, you can create a nested realm. You can then assign an authentication scheme with a higher protection level to the nested realm.

By default, to access resources in the child realm, a user must be authorized for resources in the parent realm *and* for resources in the child realm. You can globally change the default behavior of the Policy Server and always allow access to the resources in the child realm for users who are authorized either for the parent realm *or* the child realm. However, we do *not* recommend changing from the *and* logic to the *or* logic, which is less secure. To change to the *or* logic, remove the check from the Enable Nested Security checkbox. To access the checkbox, select Global Settings from the Tools menu in the Policy Server User Interface.

Note: Do not assign the anonymous authentication realm to any realm in a nested structure, including the top-level realm. You can't authorize specific users for resources protected by an anonymous authentication scheme, so the *and* logic will fail.

The following example illustrates how nested realms can be used to provide increasing levels of security.



In the realm structure shown in the previous figure, the realms mimic the file structure of the resources. Each of the nested realms has a different authentication scheme than its parent realm. Since the authentication scheme for each child realm has a higher protection level than that of the parent realm, users will need to re-authenticate when they try to access resources at lower levels of the tree. To implement this example, for each realm, you need to create a rule. Then, you need to create corresponding policies so that each policy contains a rule and users that need to access resources in a child realm can also access resources in the parent realm.

Note: Only administrators with the Manage System and Domain Objects privilege may create, edit, and delete realms. However, administrators with the Manage Domain Objects privilege may create, edit, and delete nested realms underneath existing realms in their policy domains.

More information:

[Rules](#) (see page 327)

[Policies](#) (see page 375)

[Policy Server Administrators](#) (see page 223)

Realms in Request Processing

When a user requests a resource, the Policy Server uses the longest matching realm to determine if a resource is protected, and if so, which authentication scheme must be used to establish the user's identity. The longest matching realm consists of the resource filter that can be located in the deepest level of any group of nested realms (or a single realm if nested realms are not used) that matches the requested path to a resource.

Examples

Using the example from the previous section, a file called list2.html in the location /marketing/competitors/list2.html matches the nested realm /marketing/competitors/. When the Policy Server processes authentication for list2.html, the user authenticates via HTML Forms, since that is the authentication scheme associated with the /marketing/competitors/ realm.

In the same example, a file called current_budget.html in the location /marketing/budgets/current_budget.html. Since the /budget directory is not specifically called out in a nested realm, the longest matching realm for this resource is /marketing/. Therefore, the user authenticates via the Basic user name and password authentication scheme.

Configure a Realm

Realms are groupings of resources in a specific location on your network. The contents of a realm are protected by Agents. When users request resources within a realm, the associated Agent handles authentication and authorization of the user. The realm determines the method of authentication.

You can configure realms for any type of SiteMinder Agent. Realms for Affiliate Agents require far less information than realms for other types of Agents. The process for configuring a nested realm differs slightly, since you must begin the configuration from an existing realm in the SiteMinder Administration window. The following sections discuss the procedures for configuring realms for different Agent types and nested realms.

More information:

[Agents and Agent Groups](#) (see page 99)

Configure a Realm Protected by a SiteMinder Web Agent

The Policy Server allows you to configure a realm of resources that a user accesses via a Web Server.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a realm protected by a SiteMinder Web Agent

1. Create a realm object in the Policy Server User Interface using the Realm Dialog.
2. In the Realm Dialog, provide a Name and (optional) Description for the realm object.
3. Configure resource properties in the Resource tab.

In this tab you select a Web Agent and an authentication scheme, specify a resource filter, and select a default protection state for the realm.

4. Configure session properties in the Session tab.

The Session tab allows you to configure session timeouts, session type (persistent or non-persistent), and whether synchronous auditing is enabled for the realm.

5. Configure advanced properties in the Advanced tab.

The Advanced tab of the SiteMinder Realms dialog allows you to select registration schemes, authorization directory mappings, and which types of events (authentication and authorization) are processed for the realm. Processed events enable you to associate rules with the indicated types of events.

Registration schemes allow new user accounts to be created in a user directory. If the authentication scheme associated with a realm requires that users have identities in a user directory, assigning a registration scheme to the realm provides a mechanism for adding new accounts in the user directory.

Directory mappings allow users who are authenticated in one directory to be authorized in another directory.

SiteMinder supports the definition of rules that are triggered by authentication attempts and authorization attempts. If you define rules based on these events you must select the appropriate check box.

6. Click Apply to save the realm or click OK to save the realm and return to the SiteMinder Administration window.

More information:

[Realm Dialog](#) (see page 681)

[Realm Dialog—Resource Tab](#) (see page 682)

[Realm Dialog—Session Tab](#) (see page 684)

[Realm Dialog—Advanced Tab](#) (see page 687)

[User Sessions](#) (see page 85)

[Rules](#) (see page 327)

Configure a Realm Protected by a RADIUS Agent

The Policy Server User Interface allows you to configure realms protected by a RADIUS Agent. These realms do not require all of the same information that is required for a SiteMinder Web Agent realm.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a realm on a RADIUS Agent

1. Create a realm object in the Policy Server User Interface using the Realm Dialog.
2. In the Realm Dialog, provide a Name and (optional) Description for the realm object.
3. Configuring resource properties in the Resource tab.
 - In the Resource group box specify a RADIUS Agent and authentication scheme.
 - Enter a RADIUS Agent name or an Agent group name in the Agent field, or choose from a list of existing Agents and Agent Groups using the Lookup button.

Note: SiteMinder uses the selected scheme to establish the identity of a user who attempts to access a resource in the realm.

4. Configure session properties in the Session tab.
5. Click Apply to save the realm or click OK to save the realm and return to the SiteMinder Administration window.

More information:

[Realm Dialog](#) (see page 681)

[Realm Dialog—Resource Tab](#) (see page 682)

[Authentication Schemes](#) (see page 231)

[Realm Dialog—Session Tab](#) (see page 684)

Configure a Realm for a 4.x Affiliate Agent

Note: This section describes 4.x Affiliate Agents, which are not related to the affiliate objects described in Affiliates chapter of this guide. The Affiliates chapter describes the SAML Affiliate Agent, which is the Agent for SiteMinder's Federation Security Services.

Although SiteMinder does not "protect" resources on an affiliate's Web site, a realm must be defined so that you can configure responses to pass information to an Affiliate Web Agent.

Realms for SiteMinder Affiliate Agents require far less configuration than SiteMinder Web Agents. Consequently, the procedure for configuring a realm that uses an Affiliate Agent is relatively simple.

Note: Affiliate Agents must be installed on servers at affiliate sites to receive user information from a SiteMinder protected site.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a realm on an Affiliate Agent

1. Create a realm object in the Policy Server User Interface using the Realm Dialog.
2. In the Realm Dialog, provide a Name and (optional) Description for the realm object.
3. Configure resource properties in the Resource tab.

The Resource tab of the SiteMinder Realm dialog allows you to specify an Agent, resource filter, authentication scheme, and default protection for the realm.

Enter an Affiliate Agent name or an Agent group name in the Agent field, or use the Lookup button to choose from a list of existing Agents and Agent Groups:

Note: You specify the resources protected by a realm for an Affiliate Agent when you configure the Affiliate Agent on the Web Server. For more information, see the *SiteMinder Web Agent Guide*.

4. Click Apply to save the realm or click OK to save the realm and return to the SiteMinder Administration window.

More information:

[Responses and Response Groups](#) (see page 353)

[Realm Dialog](#) (see page 681)

[Realm Dialog—Resource Tab](#) (see page 682)

Configure a Nested Realm

You can configure realms within other realms. Administrators with the Manage Domain Objects privilege may create nested realms under existing realms for policy domains within their scope.

Note: You can only create nested realms under realms protected by SiteMinder Web Agents.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a Realm on a SiteMinder Web Agent

1. Log into the Policy Server User Interface.
2. If you have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Expand the policy domain to which you want to add a realm by clicking on the + symbol to the left of the policy domain icon.
3. If you have the Manage Domain Objects privilege:

In the Object pane, expand the policy domain to which you want to add a realm by clicking on the + symbol to the left of the policy domain icon.
4. In the expanded policy domain, expand the realm icon by clicking on the + symbol to the left of the realm.
5. Right click on the realm under which you want to add a nested (child) realm.
6. From the pop-up menu, select Create Realm under Realm.

The SiteMinder Realm dialog opens.
7. Configure the nested realm with the following considerations:
 - In the Resource Filter field, enter the path for the resource filter.
 - This resource filter is added to the resource filter of the parent realm(s). For example, if the filter for the parent realm is /marketing, and you enter a filter of /data for the nested realm, the entire filter is:

[agent of the parent realm]/marketing/data]

Note: Asterisk (*) and question mark (?) characters are treated as literal characters in resource filters (not wildcards).

8. Click Apply to save the realm or click OK to save the realm and return to the SiteMinder Administration window.

When you look at nested realms in the SiteMinder Administration window, the Resource column displays the resource filters for parent realms. Green text indicates a resource filter inherited from parent realms.

More information:

[Nested Realms](#) (see page 318)

[Policy Server Administrators](#) (see page 223)

[Start the Policy Server User Interface](#) (see page 64)

[Realm Dialog—Resource Tab](#) (see page 682)

Realm Editing and Deletion

The following sections discuss issues that should be considered before you edit or delete a realm or nested realm.

Realm Editing

When you edit an existing realm you cannot make changes to the following:

Agent

The Agent that protects a server where an existing realm is located cannot be changed. If you need to change the Agent, you must delete the realm and recreate it with the new Agent.

Resource Filter

The resource filter of an existing realm cannot be changed. If you need to change the resource filter, you must delete the realm and recreate it with the new resource filter.

Realm Deletion

When you delete a realm, all nested realms associated with the realm are also deleted. In addition, all rules associated with the deleted realm and its nested realms are also deleted.

Note: It may take a short amount of time for all deleted objects to be removed from caches.

Flush a Realm from the Resource Cache

SiteMinder caches realm information when users access protected resources. This allows SiteMinder to improve network performance by keeping track of recently used resources. However, if you make a change to the security requirements or contents of a realm, you may want to flush the realm from the SiteMinder resource cache.

Note: If you have the Manage System and Domain Objects administrative privilege, you can flush all realms from the resource cache using the Cache Management dialog (see the *Policy Server Management* guide).

To flush a single realm from the resource cache

1. Login to the SiteMinder Administration User Interface.
 2. If you have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Expand the policy domain that contains the realm you want to flush from the resource cache by clicking on the + symbol to the left of the policy domain icon.
 3. If you have the Manage Domain Objects privilege:

In the Object pane, expand the policy domain that contains the realm you want to flush from the resource cache by clicking on the + symbol to the left of the policy domain icon.
 4. Expand the Realm object by clicking on the + symbol to the left of the realm icon.
 5. Right click on the realm you want to flush from the resource cache.
 6. From the pop-up menu, select Flush Resources in Realm.
- SiteMinder flushes the realm from the resource cache.

More information:

[Start the Policy Server User Interface](#) (see page 64)

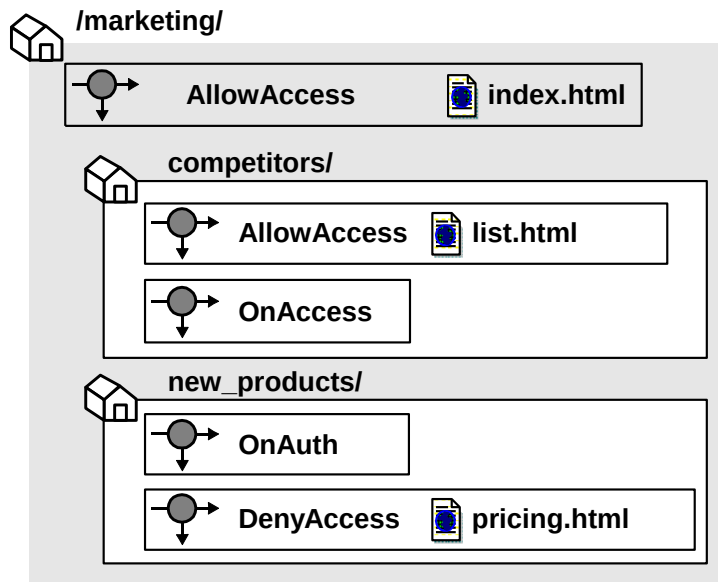
Chapter 13: Rules

This section contains the following topics:

- [Rules Overview](#) (see page 327)
- [Configure a Rule for Web Agent Actions](#) (see page 334)
- [Configure a Rule for Authentication Event Actions](#) (see page 336)
- [Configure a Rule for Authorization Event Actions](#) (see page 337)
- [Configure a Rule for Impersonation Event Actions](#) (see page 339)
- [Configure a Rule for an Affiliate Agent](#) (see page 340)
- [Resource Matching and Regular Expressions](#) (see page 341)
- [Enable and Disable Rules](#) (see page 343)
- [Advanced Rule Options](#) (see page 344)
- [Rule Deletion](#) (see page 347)

Rules Overview

Rules identify specific resources and either allow or deny access to the resources. Rules can also be used to trigger responses when authentication or authorization events take place. When you create rules, you must associate rules with specific realms. The following diagram illustrates a number of realms and nested realms and their associated rules.



In the diagram above, different realms and nested realms have specific rules associated with the resources in the realm. It is also possible to have a single rule associated with all of the resources in a realm, or a subset of resources in the realm. This is done by using resource matching or regular expressions to specify resources.

More information:

[Resource Matching and Regular Expressions](#) (see page 341)

How Rules Work as Part of a Policy

Policies protect resources by binding together rules, users, and responses. Rules are the parts of policies that determine precisely which resources are protected, and which types of actions cause a rule to fire. For example, a rule can specify all HTML files in a realm are protected for a GET action, which a Web server uses to respond to a request for an HTML page. When a user's browser attempts to access the resource, the rule fires and the policy containing the rule determines whether or not the user can view the selected resource.

How the Policy Server Processes Rules

The Policy Server evaluates rules according to the relationships between users, rules, and responses defined in policies. When a user accesses a protected resource, the Policy Server must process rules included in policies to determine whether or not the user is authorized for the resource, if any authentication and authorization events must be processed, and if any responses should be generated and returned to SiteMinder Agents.

When the Policy Server processes an authorization event, it looks for the realm with the longest resource filter matching the protected resource. Then, the Policy Server fires only those rules associated with that realm. In this example, the user is a manager, who wants to access the following protected resource:

/company/employees/managers/performance/

The following realms have resource filters that match the protected resource:

Realm Name	Realm Description	Resource Filter
Company	Customers, Employees, Vendors	/company/
Company Employees	All employees	/company/employees/
Company Managers	All managers	/company/employees/managers/

Performance Management	Managers with team members	/company/employees/managers/performance/

The realm with the longest matching resource filter is Performance Management. In response to the authorization event, the Policy Server fires all rules associated with the Performance Management realm.

In a deployment of nested realms, the Policy Server keeps a ranked list of matching realms for use during processing. If any matching rules deny access to a resource, processing stops, and the Policy Server returns any responses associated with the deny access rule to the SiteMinder Agent.

The Policy Server collects responses from all matching rules that fire. When the Policy Server finishes collecting responses based on rules, it deletes any duplicate responses.

In a deployment that uses nested realms, the Policy Server collects the entire list of accumulated responses for all matching rules. For OnAuthAccept rules, the Policy Server returns the entire list of responses to the SiteMinder Agent. For OnAuthReject rules, the Policy Server only returns the responses associated with the rule in the deepest nested realm to the SiteMinder Agent. OnAuthReject rules fire only for users bound to the policy.

More information:

[Realms](#) (see page 315)

[Responses](#) (see page 353)

Rules and Nested Realms

Nested realms are realms created within an existing realm. A nested realm has a parent, or top level realm, and is considered a child of the parent realm. When you create nested realms, you can also create separate rules to protect the resources in the child realms. You may also copy an existing rule, attach the rule to another realm, and rename the rule.

More information:

[Nested Realms](#) (see page 318)

Rule Actions

A rule's action determines what must take place for the rule to fire. A rule fires when the Policy Server determines that an action specified in a rule occurs. The rule must be contained in an existing, enabled policy. For example, if a policy contains a rule that allows access to an HTML page, and the policy specifies users who exist in a particular directory, when one of the users listed in the directory attempts to access a resource, the Policy Server determines that the rule must fire in order to process the request.

When a rule fires, the Policy Server processes the action specified in the rule based on the way the policy that contains the rule is configured. For example, if a user is not in a group specified in a policy, a rule that allows an HTTP Get action for an HTML page will not allow the user to access the resource.

More information:

[Policy Overview](#) (see page 375)

Web Agent Actions

Rules with a Web Agent action either allow or deny access to the resource(s) specified by a rule when one of the HTTP actions specified in the rule occur.

When a rule that specifies Allow Access fires, if a user authenticates successfully, SiteMinder allows the user to access the specified resource. If a rule specifies Deny Access, SiteMinder denies access to the successfully authenticated user. Deny access rules may be added to policies to provide an additional layer of security by rejecting specific individuals or groups who should not have access to a resource. Allow Access is the default.

Deny access rules take precedence over allow access rules. If a deny access rule and an allow access rule fire when a user attempts to access a resource, the presence of the deny access rule overrides all allow access rules.

The Web Agent rule actions are:

Get

Retrieves a resource for viewing via HTTP.

Put

Supports legacy HTTP actions.

Post

Posts information supplied by a user via HTTP.

XML Agent Actions

If you have purchased TransactionMinder, two additional Web Agent rule actions are available for XML Agent use:

ProcessSOAP

Supports incoming XML messages wrapped with a SOAP envelope.

ProcessXML

Supports incoming raw XML messages (not wrapped with a SOAP envelope).

For more information, see the *TransactionMinder Operations Guide*.

Affiliate Agent Actions

Affiliate Agents are SiteMinder Agents that communicate with SiteMinder Web Agents installed on the Web servers of a portal Web site.

Affiliate Agent rules are simple, because they do not protect the resources of an affiliate Web site. The Affiliate Agent processes responses sent from the portal site, so that applications on the affiliate Web site may take advantage of the information gathered about users on the SiteMinder protected portal Web site.

Affiliate Agent actions are only available (in place of Web Agent actions) for realms associated with an Affiliate Agent. There is only one possible action:

Visit

Allows a SiteMinder portal site to interact with an affiliate Web site.

More information:

[Configure a Realm for a 4.x Affiliate Agent](#) (see page 323)

Authentication Events

Authentication events occur as SiteMinder tries to establish a user's identity. As a rule action, an authentication event causes the Policy Server to fire a rule at a particular point in the authentication process.

Authentication events occur when a user accesses a resource protected by a rule that includes an On-Auth event. Unlike Web Agent actions or authorization events, authentication events always apply to the entire realm. You can't create an On-Auth rule that applies to a portion of a realm.

The following is a list of possible On-Auth events:

OnAuthAccept

Occurs if authentication was successful. This event may be used to redirect a user after a successful authentication.

OnAuthReject

Occurs if authentication failed for a user that is bound to a policy containing an On-Auth-Reject rule. This event may be used to redirect the user after a failed authentication.

OnAuthAccept and OnAuthReject events fire both at authentication time (when the user enters his / her username and password) and at validation time (when the user's cookie is read for user information). However, there are certain special actions that only occur at authentication time:

Realm timeout override (unless EnforceRealmTimeouts is used).

Unless you have a version of the Web Agent that supports the EnforceRealmTimeouts option and that option is enabled, the Idle and Max Timeouts for the user will stay at the values for the realm in which the user last authenticated (only changes if the user has to reenter credentials). See section 3.3 of the *SiteMinder 4.x Web and Affiliate Agent Quarterly Maintenance Release 4 Release Notes* for more information on EnforceRealmTimeouts.

Redirects.

Redirects are only allowed at authentication time for a number of reasons, but one of the most practical is that it would be very easy to configure an infinite loop of redirection if OnAuth redirection were allowed at validation time as well.

Access to the user's password.

The password is not stored in the SMSESSION cookie, so the only time it is available is when the user actually enters it (authentication time).

Note: OnAuth event results are per realm, so for example, if a user goes from realm A to realm B and had an OnAuthAccept header in realm A, it will not be available in realm B. When the user goes back to realm A, the header will be set again.

OnAuthAttempt

Occurs if the user was rejected because SiteMinder does not know this user (an unregistered user, for example, can be redirected to register first).

OnAuthChallenge

Occurs when custom challenge-response authentication schemes are activated (for example, a token code).

OnAuthUserNotFound

This event is only used to trigger Active Responses. This event should not be used to trigger any response other than an Active Response.

A rule with an authentication event action may be coupled with a SiteMinder response in a policy. When a user is authenticated (or rejected), the Policy Server passes any response associated with the applicable On-Auth rule back to the requesting Agent.

Note: To optimize SiteMinder performance and limit the number of times the Web Agent must retrieve static information from the Policy Server, you can set up a rule based on the OnAuthAccept authentication event, then create a response that returns the static information. When you bind the rule and response in a policy, the rule fires for users specified in the policy, and the static response is only returned to users who successfully authenticate.

More information:

[Responses](#) (see page 353)

Authorization Events

Authorization events occur as SiteMinder verifies whether or not a user is authorized to access a resource. As a rule action, an authorization event causes the Policy Server to fire a rule at a particular point in the authorization process.

The following is a list of possible authorization events:

OnAccessAccept

Occurs as the result of successful authorization. This event may be used to redirect users who are authorized to access a resource.

OnAccessReject

Occurs as the result of failed authorization. This event may be used to redirect users who are not authorized to access a resource.

A rule with an authorization event action may be coupled with a SiteMinder response in a policy. When a user is authorized (or rejected), the Policy Server passes any responses associated with the applicable On-Access rule back to the requesting Agent.

More information:

[Responses](#) (see page 353)

Impersonation Events

Impersonation provides a method for a privileged user to assume the role of another user without ending the privileged user's session. Impersonation events are used to start impersonation sessions when resources are accessed.

Possible impersonation events:

ImpersonationStart

When included in an appropriate policy, a rule that includes this event allows an impersonation session to begin

ImpersonationStartUser

When included in an appropriate policy, a rule that includes this event allows a set of users to be impersonated.

Advanced Rule Options

Advanced options allow you to define additional rule settings:

Time restrictions

Define when a rule should and should not fire.

Active rules

Allow dynamic authorization based on external business logic.

More information:

[Add Time Restrictions for Rules](#) (see page 344)

Configure a Rule for Web Agent Actions

The following procedure describes the process for creating a rule that specifies a Web Agent action.

Note: You should also follow this procedure when configuring rules for a TransactionMinder XML Agent, while considering the additional rule actions available for that agent type. See the *eTrust SOA Security Manager Operations Guide*.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a Rule for Web Agent Actions

1. Using the Policy Server User Interface, create a rule object.
The Rule Dialog opens.
2. Provide a Name and, optionally, a Description of the new rule.
3. In the Realm and Resources group box, select the name of the realm from the Realm drop-down list box that includes the resources to which this rule will apply.
4. In the Resource field, enter the Resource that this rule will protect.
Note: When you look at rules associated with a realm in the SiteMinder Administration window, the Resource column displays the resource specified in each rule. Green text indicates a resource filter inherited from a realm or group of nested realms.
5. Select the Perform Regular Expression Matching check box if you want the rule's resource to use regular expressions.
6. In the Action group box, select the Web Agent Actions radio button.
7. In the Action group box, select one or more of the following HTTP actions from the list to the right of the radio buttons:
 - Get
 - Put
 - Post
8. In the Allow/Deny and Enable/Disable group box, select either the Allow Access or the Deny Access radio button.

When an Allow Access rule fires, SiteMinder allows properly authenticated users who are associated with the policy that contains the rule to access the resource specified by the rule.

When a Deny Access rule fires, SiteMinder denies access to properly authenticated users who are specified in the policy that contains the rule.

Note: Deny Access rules take precedence over Allow Access rules. You can use a Deny Access rule to make sure that a user or group of users cannot access a resource, regardless of any other policies and rules that exist in your SiteMinder implementation.
9. If you do not want the rule to be active, unset the Enabled check box.
10. If required, set time restrictions and or active rule settings in the Advanced group box.
11. Click Apply to save the rule, or click OK to save the rule and return to the SiteMinder Administration window.

When you look at rules associated with a realm in the SiteMinder Administration window, the Resource column displays the resource specified in each rule. Green text indicates a resource filter inherited from a realm or group of nested realms.

More information:

[Rule Dialog](#) (see page 691)

[Regular Expressions for Resource Matching](#) (see page 342)

[Policy Overview](#) (see page 375)

[Enable and Disable Global Rules](#) (see page 441)

[Advanced Rule Options](#) (see page 334)

Configure a Rule for Authentication Event Actions

Rules that include SiteMinder authentication events allow you to control actions that occur when users authenticate to gain access to a resource.

Note: In order to configure rules for authentication events, you must configure the realm in which the rule is to be created to process these events. To do this, ensure that the Process Authentication Events option in the Advanced tab of the SiteMinder Realm dialog is set.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a rule for authentication event actions

1. Using the Policy Server User Interface, create a rule object.

The Rule Dialog opens.

2. Provide a Name and, optionally, a Description of the new rule.
3. In the Realm and Resources group box, select the name of the realm from the Realm drop-down list box that includes the resources to which this rule will apply.
4. In the Action group box, select the Authentication Events radio button.

Note: The Resource field and Perform regular expression pattern matching options are disabled, because Authentication Events always apply to the entire realm. The Allow Access and Deny Access options are also disabled as they are not applicable for these events.

5. In the Action group box, select one or more of the following authentication events from the drop-down list:
 - OnAuthAccept
 - OnAuthAttempt
 - OnAuthChallenge
 - OnAuthReject
 - OnAuthUserNotFound

6. If you do not want the rule to be active, unset the Enabled check box.
7. If required, set time restrictions and or active rule settings in the Advanced group box.
8. Click Apply to save the rule, or click OK to save the rule and return to the SiteMinder Administration window.

More information:

[Authentication Events](#) (see page 331)

[Configure a Realm](#) (see page 320)

[Rule Dialog](#) (see page 691)

[Enable and Disable Rules](#) (see page 343)

[Advanced Rule Options](#) (see page 334)

Configure a Rule for Authorization Event Actions

Authorization events occur after a user is authenticated. Rules that include authorization events allow SiteMinder to call responses based on whether a user is or is not authorized for the resource the user requested. When the user has been granted or denied access based on their privileges, the appropriate event is triggered.

Note: In order to configure rules for authorization events, you must configure the realm in which the rule is to be created to process these events. To do this, ensure that the Process Authorization Events option in the Advanced tab of the SiteMinder Realm dialog is set.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a rule for authorization event actions

1. Using the Policy Server User Interface, create a rule object.
The Rule Dialog opens.
2. Provide a Name and, optionally, a Description of the new rule.
3. In the Realm and Resources group box, select the name of the realm from the Realm drop-down list box that includes the resources to which this rule will apply.
4. In the Resource field, enter the resource that this rule will protect.

Note: When you look at rules associated with a realm in the SiteMinder Administration window, the Resource column displays the resource specified in each rule. Green text indicates a resource filter inherited from a realm or group of nested realms.

5. Mark the Perform Regular Expression Matching check box if you want the rule's resource to use regular expressions.
6. In the Action group box, select the Authorization Events radio button.
Note: The Allow Access and Deny Access options are disabled as they are not applicable for these events.
7. In the Action group box, select one or more of the following authorization events from the drop-down list:
 - OnAccessAccept
 - OnAccessReject
8. If you do not want the rule to be active, unset the Enabled check box.
9. If required, set time restrictions and or active rule settings in the Advanced group box.
10. Click Apply to save the rule, or click OK to save the rule and return to the SiteMinder Administration window.

More information:

[Authorization Events](#) (see page 333)

[Configure a Realm](#) (see page 320)

[Rule Dialog](#) (see page 691)

[Regular Expressions for Resource Matching](#) (see page 342)

[Enable and Disable Global Rules](#) (see page 441)

[Advanced Rule Options](#) (see page 334)

Policy Considerations When Configuring OnAccessReject Rules

When you create rules that include OnAccessReject events, sometimes referred to as OnAccessReject rules, you must take into account how the Policy Server processes policies. This section outlines the special circumstances created by OnAccessReject rules.

An OnAccessReject rule will not fire if it is in the same policy as a GET / POST rule. When a user is authenticated, SiteMinder resolves the identity of the user. Therefore, if the OnAccessReject rule and the GET / POST rule are in the same policy, then a user who is allowed access to a resource is the same user who should be redirected on an OnAccessReject event. Since the user is allowed access, the reject event never applies.

To resolve this discrepancy, create a separate policy for the OnAccessReject rule (may also include other event rules) and specify the users for which it should apply.

For example, in an LDAP user directory, User1 should have access to a resource and everyone else in the group, ou=People, o=company.com, should be redirected to an OnAccessReject page. Two policies are required:

Policy1

Includes a GET / POST rule that allows access for User1.

Policy2

Includes the OnAccessReject rule and a Redirect response, and specifies the group, and ou=People, o=company.com.

Since User1 is authorized, the OnAccessReject rule will not fire when User1 access the resource. However, the OnAccessReject rule will fire for all other users in the group, ou=People, o=company.com, because they are not authorized to access the resource.

More information:

[Policy Overview](#) (see page 375)

Configure a Rule for Impersonation Event Actions

Impersonation events are used to start impersonation sessions when resources are accessed.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a rule for impersonation events

1. Using the Policy Server User Interface, create a rule object.
The Rule Dialog opens.
2. Provide a Name and, optionally, a Description of the new rule.
3. In the Realm and Resources group box, select the name of the realm from the Realm drop-down list box that includes the resources to which this rule will apply.
4. In the Resource field, enter the resource that this rule will protect.
Note: When you look at rules associated with a realm in the SiteMinder Administration window, the Resource column displays the resource specified in each rule. Green text indicates a resource filter inherited from a realm or group of nested realms.
5. Mark the Perform Regular Expression Matching check box if you want the rule's resource to use regular expressions.

6. In the Action group box, select the Authorization Events radio button.
Note: The Allow Access and Deny Access options are disabled as they are not applicable for these events.
7. In the Action group box, select one or more of the following impersonation events from the drop-down list:
 - ImpersonationStart
 - ImpersonationStartUser
8. If you do not want the rule to be active, unset the Enabled check box.
9. If required, set time restrictions and or active rule settings in the Advanced group box.
10. Click Apply to save the rule, or click OK to save the rule and return to the SiteMinder Administration window.

More information:

[Impersonation](#) (see page 457)

[Rule Dialog](#) (see page 691)

[Regular Expressions for Resource Matching](#) (see page 342)

[Impersonation Realms and Events](#) (see page 461)

[Enable and Disable Rules](#) (see page 343)

[Advanced Rule Options](#) (see page 334)

Configure a Rule for an Affiliate Agent

Affiliate Agents are SiteMinder Agents that communicate with SiteMinder Web Agents installed on the Web servers of a portal Web site. Affiliate Agent rules are very simple, since they do not protect the resources of an affiliate Web site. The Affiliate Agent processes responses sent from the portal site, so that applications on the affiliate Web site may take advantage of the information gathered about users on the SiteMinder protected portal Web site.

The only action available in a rule for an Affiliate Agent is Visit. This is a special action for Affiliate Agents that allows a SiteMinder portal site to interact with an affiliate Web site.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a rule for an Affiliate Agent

1. Using the Policy Server User Interface, create a rule object.
The Rule Dialog opens.

2. Provide a Name and, optionally, a Description of the new rule.
3. In the Realm and Resources group box, select the name of the realm from the Realm drop-down list box that includes the resources to which this rule will apply.

Note: The Resource field and Perform regular expression pattern matching options are disabled, because the resources protected by an Affiliate Agent are specified when you configure the Affiliate Agent on the Web Server. In the Action group box, the Affiliate Agent actions radio button and its single associated action (Visit) are automatically selected and all other event types disabled.

4. Click Apply to save the your changes, or click OK to save your changes and close the SiteMinder Rule dialog.

More information:

[Configure a Realm for a 4.x Affiliate Agent](#) (see page 323)
[Rule Dialog](#) (see page 691)

Resource Matching and Regular Expressions

Rules may use resource matching and regular expression matching to specify resources in a realm.

Standard Resource Matching

By default, resource matching for a rule is done with wildcards.

The following table describes the characters that are supported for resource matching

Character	Use
*	The wildcard (*) is used to match all characters in the string. The expression *.html will match all files with a .html file extension, such as index.html, out.html, and so forth.
?	The question mark (?) will match a single character of the string. The expression lmn?p will match the sub-string lmnop, lmnp, and so forth.

Regular Expressions for Resource Matching

Regular expressions allow for greater flexibility in resource matching. To enable regular expression matching, in the SiteMinder Rule dialog, select the Perform Regular Expression Pattern Matching check box.

Regular expressions are text patterns used for string matching. The following table contains syntax examples for setting up regular expressions.

Characters	Results
\	Used to quote a meta-character (like '*')
\\	Matches a single '\' character
(A)	Groups subexpressions (affects order of pattern evaluation)
[abc]	Simple character class (any character within brackets matches the target character)
[a-zA-Z]	Character class with ranges (any character range within the brackets matches the target character)
[^abc]	Negated character class
.	Matches any character other than newline
^	Matches only at the beginning of a line
\$	Matches only at the end of a line
A*	Matches A 0 or more times (greedy)
A+	Matches A 1 or more times (greedy)
A?	Matches A 1 or 0 times (greedy)
A{n}	Matches A exactly <i>n</i> times (greedy)
A{n,}	Matches A at least <i>n</i> times (greedy)
A{n,m}	Matches A at least <i>n</i> but not more than <i>m</i> times (greedy)
A*?	Matches A 0 or more times (reluctant)
A+?	Matches A 1 or more times (reluctant)
A??	Matches A 0 or 1 times (reluctant)
AB	Matches A followed by B
A B	Matches either A or B

Characters	Results
\1	Backreference to 1st parenthesized subexpression
\n	Backreference to <i>n</i> th parenthesized subexpression

Limit: Each regular expression can contain no more than 10 subexpressions, including the expression itself. The number of subexpressions equals the number of left or opening parentheses in the regular expression plus one more left parenthesis for the expression itself.

Enable and Disable Rules

The Policy Server User Interface allows you to enable and disable rules. If a rule is enabled, SiteMinder automatically protects the resource(s) specified in the rule. If a rule is disabled, SiteMinder does not automatically protect the resource(s) specified in the rule. By default, when you create a rule, it is enabled.

If a rule is enabled, no one may access the protected resource(s) unless a policy that contains the rule has been created, and the user attempting to access the rule is part of a group specified in the policy. To allow access to resources before a policy is put into place, you can disable the rule.

To enable or disable a rule

1. Log into the Policy Server User Interface.
2. Depending on your administrative privileges, do one of the following:
3. If you have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Expand the policy domain that contains the rule you want to disable or enable.
 - c. Select the realm or nested realm that contains the rule you want to disable or enable.
 - d. In the List pane, select the rule you want to disable or enable.
4. You have the Manage Domain Objects privilege:
 - a. In the Object pane, expand the policy domain that contains the rule you want to disable or enable
 - b. Select the realm or nested realm that contains the rule you want to disable or enable.
 - c. In the List pane, select the rule you want to disable or enable.

5. From the menu bar, select Edit, Properties of Rule.
The SiteMinder Rule dialog opens.
6. In the Allow/Deny and Enable/Disable group box, select the Enabled check box.
Selecting this check box toggles the check mark. If the box contains a check mark, the rule is enabled. If it does not, the rule is disabled.
7. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Start the Policy Server User Interface](#) (see page 64)
[Administrator Scope and Tasks](#) (see page 225)

Advanced Rule Options

The Advanced group box in the SiteMinder Rule Dialog is where you define additional rule settings. This group box contains a tab for time restrictions and a tab for active rules. Time restrictions and active rules are discussed in the following sections.

Add Time Restrictions for Rules

The Advanced group box of the SiteMinder Rule dialog contains a Time Restrictions tab. This region contains two buttons and a message that indicates whether or not time restrictions have been applied to the current rule.

A time restriction indicates when a rule fires. For example, if you create an Allow Access rule with a time restriction that limits access to a resource to 9am - 5 pm, Monday - Friday, the rule will only fire and allow users to access the resource during the specified time. The resource will not be available outside the times indicated.

Note: For information about how SiteMinder handles time across multiple time zones, see [How the Web Agent and Policy Server Calculate Time](#) (see page 83).

To add a time restriction to a rule

1. In the Advanced group box of the SiteMinder Rule dialog, click the Time Restrictions tab.
The Time Restrictions tab moves to the front.
2. In the Time Restrictions tab, click Set.
The Time dialog opens.

3. Click on the Select button to the right of the Effective Starting Date field.

The area to the right of the Effective Starting Date field displays the Set Starting Date region. The contents of this region change based on the radio button you select.

Select Now if you want the Time Restriction to take effect as soon as you close the Time Restriction dialog. Click on Select Date and use the up and down arrow buttons to select a day, month, and year if you want to assign a specific starting date for the Time Restriction.

4. Click on the Select button to the right of the Expiration Date field.

The area to the right of the Expiration Date field displays the Set Expiration Date region. The contents of this region change based on the radio button you select.

Select Never if you want the Time Restriction to remain in effect for an undetermined period. Click on Select Date and use the up and down arrow button to select a day, month, and year if you want to assign a specific expiration date for the Time Restriction.

5. Click on blocks in the time table to change the blocks from Blue (Fire Rule), to Yellow (Do Not Fire).

Each block in the table represents an hour. When a block is yellow, the rule will not fire during that hour, and the rule will not apply to the resources governed by it.

You can select an entire row (day) by clicking on one of the Day buttons (Monday, Tuesday, etc.). You can select an entire column (hour across every day of the week) by clicking on one of the Hour buttons.

To change all Yellow blocks (Do Not Fire) to Blue (Fire Rule), click on Always Fire. To change all Blue blocks to Yellow, click on Never Fire. To reset Time Restrictions to their last saved values, click Reset.

6. Click OK to save your settings and return to the SiteMinder Rule dialog.

Under the Set and Remove buttons in the Time Restriction group box, a note indicates that Time Restrictions have been applied to the rule.

Note: If you add time restrictions to a rule, the resources protected by that rule will be subject to the rule only during the specified time periods. For example, if you create a rule that Allows Access during specific period of time, anyone who tries to access the protected resources outside of that time period, regardless of their credentials, will not be able to access the protected resource.

Configure an Active Rule

An Active Rule is used for dynamic authorization based on external business logic. The Policy Server invokes a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure an Active Rule

1. In the Advanced group box of the SiteMinder Rule dialog, click the Active Rule tab.
The Active Rule tab moves to the front.

2. In the Active Rule tab, click Set.
The SiteMinder Active Rule Editor dialog opens.

3. In the Variable Name field, enter the variable name to be generated.

4. In the Library Name field, enter the name of the shared library that supports the Active Rule.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default path for your system to search for the shared library at run-time.

You should not include a file extension for the shared library. For example, if you are using a Windows platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory, you simply need to enter lib in the Library Name field.

5. In the Function Name field, enter the name of the actual function in the shared library that implements the Active Rule.
6. In the Parameter(s) field, enter the (optional) list of parameters to be passed to the function in the shared library.

As you enter information in the fields, the system displays the generated string in the Generated Script field.

7. Click OK to save the Active Rule and return to the SiteMinder Rule dialog.

Rule Deletion

If you delete a rule, the rule is automatically removed from any policies that include the rule. The policies remain on your system. You should verify that the policies function without the deleted rule.

Note: Policies must contain at least one rule.

When you delete a rule that is included in a rule group, it may take several seconds before the deleted rule is removed from the rule group.

Note: It may take a short amount of time for all deleted objects to be removed from caches.

Chapter 14: Rule Groups

This section contains the following topics:

[Rule Group Overview](#) (see page 349)

[Create a Rule Group](#) (see page 350)

[Add a Rule To or Remove a Rule From a Rule Group](#) (see page 350)

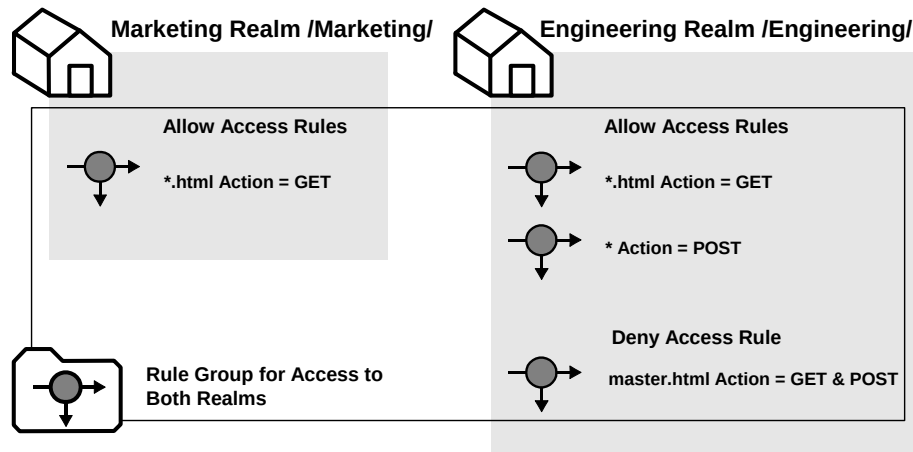
[Rule Group Property Editing](#) (see page 351)

[Rule Group Deletion](#) (see page 351)

Rule Group Overview

A rule group is a set of rules that can be bound to SiteMinder policies. You can use a rule group to combine groups of rules you will be applying to the same policy. For example, if you have a number of rules that allow a GET action for different resources of a Web site, you could then create a rule group that contains all of the resources. When you configure the policy that will include the rules, you can add a single rule group to the policy, rather than add all of the rules individually.

When you include a rule group in a policy, each rule in the group is evaluated and applied independently of other rules in the group.



The previous diagram illustrates a rule group that contains rules for both the Marketing realm and the Engineering realm. The rule group can be used in a policy rather than including all four rules separately.

More information:

[Policy Overview](#) (see page 375)

Create a Rule Group

You can use a rule group to combine groups of rules you will be applying to the same policy.

To create a rule group

1. If necessary, configure the Policy Server User Interface to display Rule Group objects. To do this, select View, Rule Groups from the menu bar.

The Policy Server User Interface places a check mark in the View menu to the left of the menu selection and *Rule Groups* appear in the System tab of the SiteMinder Administration window.

Note: If there is already a check next to *Rule Groups* in the View menu, selecting the option again will remove *Rule Groups* from the System tab.

2. Create a rule group object. The Rule Group Properties Dialog opens.
3. Provide a Name and, optionally, a Description of the new rule group.
4. In the Agent Type group box, identify the type of agent for which the rule group is to contain rules by clicking on the appropriate radio button and selecting from the drop-down list box.

The Agent Type for a rule group must correspond to the Agent Type of the rules within the rule group. When you add rules to the rule group, only rules with the same Agent type appear in the list of available rules. For more information, see the next section.

5. Continue by adding rules to your rule group.

More information:

[Rule Group Properties Dialog](#) (see page 698)

[Add a Rule To or Remove a Rule From a Rule Group](#) (see page 350)

Add a Rule To or Remove a Rule From a Rule Group

You can add any number of rules of the same Agent type to a rule group. All of the rules must exist in the same policy domain.

To add rules to a rule group

1. In the Group Members group box of the SiteMinder Rule Group Dialog, click Add/Remove.

The Rule Group Items dialog opens.

2. To add a rule to the rule group, select a rule from the Available Members list and click on the Left Arrow button, which points to the Current Members list. The opposite procedure removes rules from the Current Members list.

You can select multiple rules by holding the CTRL key and clicking on rules in one of the Members lists. When you select multiple rules and click one of the Arrow buttons, all of the selected rules are moved.

3. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Rule Group dialog.

More information:

[Rule Group Items Dialog](#) (see page 702)

Rule Group Property Editing

All of the properties of a rule group, except the Agent Type (and vendor for RADIUS Agents), can be edited. If you want to change the Agent Type (or vendor) of a rule group, you must delete the rule group and create a new one with the desired Agent Type.

Rule Group Deletion

When you delete a rule group, remember that you are only removing the grouping. The rules that were contained in the rule group still exist.

Chapter 15: Responses and Response Groups

This section contains the following topics:

[Responses](#) (see page 353)

[How SiteMinder Processes Responses](#) (see page 354)

[Response Groups](#) (see page 372)

Responses

A response passes user attributes, DN attributes, static text, or customized active responses from the Policy Server to a SiteMinder Agent. Responses can be used by servlets, Web applications, or other custom applications to display customized content, change SiteMinder settings, or redirect users to different resources. When working with Web applications, responses can be used as privileges or entitlements for fine-grained access control.

A policy contains rules and responses which are bound to users and user groups. In a policy, responses are bound to specific rules or rule groups. When a rule fires, the associated response returns information to a SiteMinder Agent.

Responses take the form of name/value pairs. When a rule is triggered, the Policy Server returns the paired response to the SiteMinder Agent.

For example, if a user attempts to access a protected Web page, but is not authorized to view the contents of the page, a response can redirect the user to an HTML page that indicates the user does not have access, and provide details for contacting a system administrator.

For Web Agents, SiteMinder adds response attributes to HTTP header variables or HTTP cookie variables so that the responses are available to the Web resource or application named in the rule. In a RADIUS environment, the response is returned to the RADIUS client.

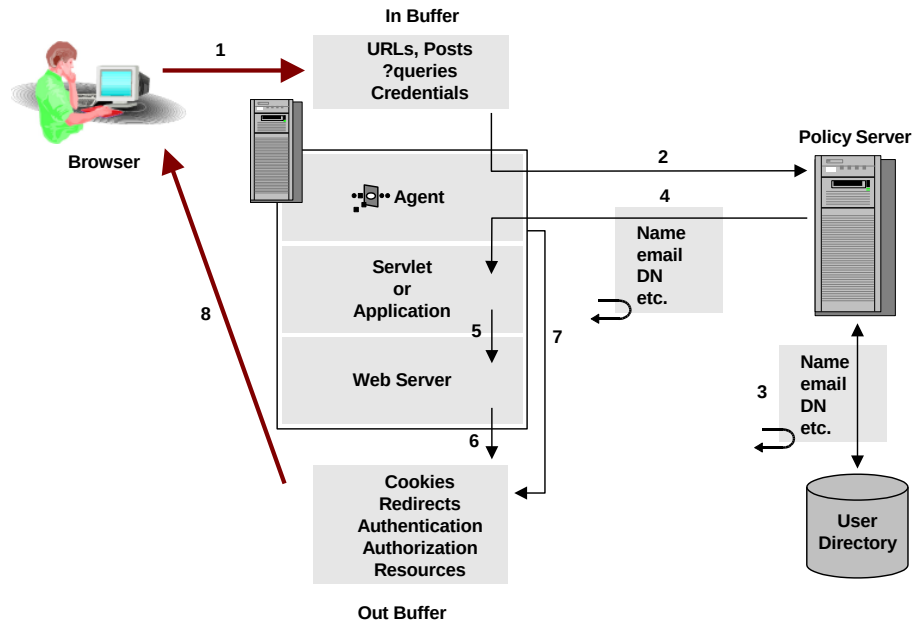
More information:

[Policy Overview](#) (see page 375)

[Rules Overview](#) (see page 327)

How SiteMinder Processes Responses

The following diagram illustrates how SiteMinder uses responses when processing a user's request for resources.



In the previous diagram, SiteMinder processes responses using the following steps:

1. A user requests a resource that is protected by a SiteMinder Agent.
The In Buffer represents the Web Server buffer where the requested URL, Post data or query strings reside during Web Server processing.
2. The SiteMinder Agent intercepts requests for protected resources, and communicates with the Policy Server to authenticate and authorize the user.
Part of the authentication process binds the user to a record in a user directory.
3. The Policy Server uses the binding to retrieve attributes specified in a SiteMinder response from the user's entry in the user directory.
4. The Policy Server passes user attributes specified in the response back to the Web Agent.
5. The attributes returned to the Web Agent may be used by a servlet or application that has been customized to use the attributes specified in the response.
The servlet or application executes its processes and passes its results to the Web Server.
6. The Web Server's Out Buffer contains the resulting information that must be returned to the user.

7. The Web Agent adds any SiteMinder specific information to the Web Server's Out Buffer.

The Web Agent may pass any of the following to the Out Buffer: SiteMinder cookies, URLs for redirection, and successful /unsuccessful authentications or authorizations.

8. The Web Server passes the contents of the Out Buffer to the user.

Types of Responses

A response is a container for one or more response attributes. The response attributes are what a SiteMinder Agent receives after the Policy Server processes a response. The available response attributes differ based on the type of response.

There are three types of SiteMinder responses:

- Web Agent responses
- Affiliate Agent responses
- RADIUS responses.

The following sections discuss the different response types.

Note: You can also create response types for custom Agents and response attributes using the CA APIs (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

Web Agent Responses

Web Agent responses are SiteMinder responses that provide name/value pairs usable by a SiteMinder Web Agent. These responses can contain attributes for HTTP header variables, cookie variables, and URLs for redirections.

More information:

[Web Agent Response Attributes](#) (see page 356)

Affiliate Agent Responses

Affiliate Agent responses are SiteMinder responses that provide name/value pairs usable by a SiteMinder Affiliate Agent. These responses can contain attributes for HTTP header variables or HTTP cookie variables.

More information:

[Affiliate Agent Response Attributes](#) (see page 357)

RADIUS Responses

RADIUS responses are SiteMinder responses that provide values usable by a RADIUS Agent. These responses can contain response attributes for all supported RADIUS attributes.

More information:

[RADIUS Agent Response Attributes](#) (see page 357)

Response Attributes

Each SiteMinder response contains one or more response attributes. These attributes differ based on the type of response. The following sections discuss the response attributes that are available for each type of response.

Web Agent Response Attributes

Web Agent response attributes are response attributes that SiteMinder Web Agents can interpret and pass on to other applications. The following is a list of generally available Web Agent response attributes:

- WebAgent-HTTP-Authorization-Variable
- WebAgent-HTTP-Cookie-Variable
- WebAgent-HTTP-Header-Variable
- WebAgent-OnAccept-Redirect
- WebAgent-OnAccept-Text
- WebAgent-OnAuthAccept-Session-Idle-Timeout
- WebAgent-OnAuthAccept-Session-Max-Timeout
- WebAgent-OnReject-Redirect
- WebAgent-OnReject-Text

Note: For complete descriptions of these response attributes, see the *SiteMinder Web Agent Guide*.

Affiliate Agent Response Attributes

Affiliate Agent response attributes are response attributes that SiteMinder Affiliate Agent can interpret and pass on to other applications at an affiliate Web site. The following is a list of Affiliate Agent response attributes:

- AffiliateAgent-HTTP-Header-Variable
- AffiliateAgent-HTTP-Cookie-Variable

Note: For complete descriptions of the response attributes, see the *SiteMinder Web Agent Guide*.

RADIUS Agent Response Attributes

RADIUS Agent response attributes are response attributes that RADIUS Agents can interpret. All of the response attributes supported by SiteMinder correspond to the attributes described in the Request for Comments (RFC) 2138, which describes attributes supported by the RADIUS protocol.

Responses and Directory Mappings

Directory mappings allow you to specify a separate authorization user directory for a realm. When you define a separate authorization directory, a user is authenticated based on the information contained in one directory, but authorized based on the information contained in another directory.

When you create a response and associate it with a authentication (OnAuth) event, any information retrieved from a user directory is retrieved from the authentication directory. If you create an authorization (OnAccess) event, any information retrieved from a user directory is retrieved from the authorization directory.

More information:

[Directory Mapping Overview](#) (see page 213)

Configure a Response

When you configure a SiteMinder response, you must create the response in the SiteMinder Response dialog, then add response attributes to the response.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a response

1. Log into the Policy Server User Interface.
2. If you have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Select the policy domain to which you want to add a response.
3. You have the Manage Domain Objects privilege:

In the Object Pane, select the policy domain to which you want to add a response.
4. From the menu bar, select Edit, <domain name>, Create Response.

The SiteMinder Response dialog opens.
5. In the Name field, enter the name of the new response.
6. Optionally, in the Description field, enter a description for the new response.
7. In the Agent Type group box, select one of the following radio buttons:

SiteMinder

Indicates the response passes information to a SiteMinder Web Agent or Affiliate Agent.

RADIUS

Indicates the response passes information to a RADIUS Agent.

8. If you selected the SiteMinder radio button, select a SiteMinder Agent type from the drop-down list in the Agent Type group box.

Your choices include Web Agent or Affiliate Agent. Select the Agent type to which the response must pass information.
9. If you selected the RADIUS radio button, select a RADIUS Agent type from the drop-down list in the Agent Type group box.

Your choices include a list of RADIUS vendors, as well as an option for Generic RADIUS. Select the Agent type to which the response must pass information.
10. Click Apply to save your changes.

You can now add response attributes to your response. See the next section for details about adding response attributes for each Agent type.

More information:

[Start the Policy Server User Interface](#) (see page 64)
[Response Dialog](#) (see page 705)

Response Attributes

Each SiteMinder response may contain one or more response attributes. Response attributes identify the pieces of information that the Policy Server passes to a SiteMinder Agent. Each SiteMinder Agent type can accept different response attributes.

Note: For information on configuring an smetssocookie Web Agent active response attribute, which is needed for enabling single sign-on from SiteMinder to CA Single Sign-On, see [Configure an smetssocookie Web Agent Active Response Attribute](#) (see page 990).

Response Attribute Types

SiteMinder supports different types of response attributes. The types of response attributes determine where the Policy Server finds the proper values for the response attributes.

You can specify the following types of response attributes when you add response attributes to a SiteMinder response:

Static

Returns data that remains constant.

Use a static attribute to return a string as part of a SiteMinder response. This type of response can be used to provide information to a Web application. For example, if a group of users has specific customized content on a Web site, the static response attribute, `show_button = yes` could be passed to the application.

User Attribute

Returns profile information from a user's entry in a user directory.

This type of response attribute returns information associated with a user in a directory. A user attribute can be retrieved from an LDAP, WinNT, Microsoft SQL Server or Oracle user directory.

Note: In order for the Policy Server to return values from user directory attributes as response attributes, the user directories must be configured in the SiteMinder User Directory dialog.

DN Attribute

Returns profile information from a directory object in an LDAP, Microsoft SQL Server or Oracle user directory.

This type of response attribute is used to return information associated with directory objects to which the user is related. Groups to which a user belongs, and Organizational Units (OUs) that are part of a user DN, are examples of directory objects whose attributes can be treated as DN attributes.

For example, you can use a DN attribute to return a company division for a user, based on the user's membership in a division.

Note: In order for the Policy Server to return values from DN attributes as response attributes, the user directories must be configured in the SiteMinder User Directory dialog.

Active Response

Returns values from a customer supplied library that is based on the SiteMinder Authorization API.

An Active Response is used to return information from an external source. An Active Response is generated by having the Policy Server invoke a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

Note: It is up to you to make sure the value returned by an active response is valid. For example, if an active response returns a numeric type, the library and function must return a string whose value is a number.

When you configure a response attribute, the correct Value Type for the response attribute is displayed in the upper right corner of the SiteMinder Response Attribute Editor dialog.

More information:

[User Directory Dialog](#) (see page 545)

[LDAP Overview](#) (see page 142)

[Response Attributes](#) (see page 356)

Configure a Web Agent Response Attribute

Web Agent responses can have a wide range of response attributes. These attributes include HTTP header variables, cookie variables, redirections to other resources, text, and timeout values. For details about most specific Web Agent response attribute types, see the *SiteMinder Web Agent Guide*.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a Web Agent response attribute

1. In the SiteMinder Response dialog, make sure the SiteMinder radio button is selected in the Agent Type group box and Web Agent is selected in the drop-down list.

2. Click Create.

The SiteMinder Response Attribute Editor dialog opens.

From the Attribute drop down list, select the type of response attribute you want to add to the Web Agent response. For complete descriptions of response attributes, see the *SiteMinder Web Agent Guide*.

3. In the Attribute Setup tab, select a radio button for the type of response attribute in the Attribute Kind group box.

You can select one of the following radio buttons: Static, User Attribute, DN Attribute, or Active Response.

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return a response with an attribute contained in a group that is nested in a group specified by a policy. Nested groups often occur in complex LDAP deployments.

Your selection from the Attribute drop-down list and the response attribute type radio button you select determine the available fields in the Attribute Fields group box.

4. If you selected the Static radio button, in the Attribute Fields group box, complete the Variable Name and Variable Value fields (for header variables) or the Cookie Name and Cookie Value fields (for cookie variables).

For example, to return the name/value pair show_content=yes to the Web Agent, enter show_content in the Variable Name field, and yes in the Variable Value field.

5. If you selected the User Attribute radio button, in the Attribute Fields group box, complete the Variable Name (or Cookie Name for cookie variables) and Attribute Name fields.

For example, to return the value of a user's email attribute to the Web Agent, enter email_address in the Variable Name field, and email (where email is the name of the attribute in the user directory) in the Attribute Name field.

Note: For a list of automatically generated SiteMinder user attributes that can be used in responses, see [SiteMinder Generated User Attributes](#) (see page 366).

6. If you selected the DN Attribute radio button, in the Attribute Fields group box, complete the Variable Name (or Cookie Name for cookie variables), DN Spec, and Attribute Name fields.

If the response attribute will be retrieved from an LDAP user directory that contains nested groups (groups that contain other groups), and you want the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind group box.

7. If you selected the Active Response radio button, in the Attribute Fields group box, complete the Variable Name field (or Cookie Name for cookie variables), Library Name and Function Name fields. Optionally, complete the Parameters field.

8. If you selected the Variable radio button, in the Attribute Fields group box, complete the Variable Name (or Cookie Name for cookie variables) and Variable fields.
9. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

[Response Attribute Types](#) (see page 359)

Advanced Tab

See the Advanced tab section in [Response Attribute Editor Dialog](#) (see page 707).

Configure an Affiliate Agent Response Attribute

Affiliate Agent response attributes can be HTTP header variables or cookie variables. For details about specific Web Agent response attribute types, see the *SiteMinder Web Agent Guide*.

To configure an Affiliate Agent response attribute

1. In the SiteMinder Response dialog, make sure the SiteMinder radio button is selected in the Agent Type group box and Affiliate Agent is selected in the drop-down list.
2. Click Create.

The SiteMinder Response Attribute Editor dialog opens.

The process for configuring an Affiliate Agent response attribute is identical to the process for configuring a Web Agent response attribute. The only difference is that Affiliate Agents only support HTTP header variables or HTTP cookie variables as response attributes.

3. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

[Configure a Web Agent Response Attribute](#) (see page 360)

Configure a RADIUS Response Attribute

RADIUS response attributes can be any of the attributes supported by the RADIUS protocol.

To configure a RADIUS response attribute

1. In the SiteMinder Response dialog, make sure the RADIUS radio button is selected in the Agent Type group box and the appropriate vendor is selected in the drop-down list.

2. Click Create.

The SiteMinder Response Attribute Editor dialog opens.

The process for configuring a RADIUS response attribute is identical to the process for configuring a Web Agent response attribute. However, you do not need to supply a Variable Name for a RADIUS response attribute, since the attributes use a default name.

A small number of RADIUS response attributes use the *Enumerate* Value Type. For these attributes, you select from a drop-down list, rather than entering information.

3. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

[Configure a Web Agent Response Attribute](#) (see page 360)

Configure Response Attribute Caching

Responses return values to a requesting Agent. The data returned to the Agent can be a fixed value, or it may change over time. When you use a SiteMinder Agent to protect a resource, Agents can cache a value for fixed data, so that the value does not need to be recalculated each time the associated policy fires.

For example, a customer's account number is a fixed value, while the customer's account balance changes after each transaction. It would be more efficient to retrieve the account number once and then cache it. However, you probably want the balance to be recalculated at a regular interval to make sure the information is current.

Note: SiteMinder does not cache RADIUS response attributes.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure response attribute caching

1. In the SiteMinder Response Attribute Editor dialog, select one of the following radio buttons in the Attribute Caching region:

Cache value

Indicates the value of the response attribute should be calculated once and then cached for later use by the Policy Server. Use this radio button for attributes that remain static for extended periods of time.

Recalculate value every <number> seconds

Indicates the value of the response attribute should be recalculated at the interval of seconds that you enter in the field. Set this value to 1 second for information that should be recalculated each time the rule associated with the response fires.

Note: Setting this interval to zero is equivalent to selecting the Cache Value radio button.

2. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

Use Variable Objects in Responses

You can create responses that included variable objects. Variable objects can be used in response attributes to include dynamic information evaluated during the authorization of a request.

Note: Variable Objects included in responses are only evaluated during the authorization of a request, and not during the authentication process. Responses that include variables are limited to authorization events.

Responses can contain any number of response attributes. To use variable objects to obtain response values, you must add response attributes that see variable objects.

Note: HTTP header variables and HTTP cookie variables must not be confused with SiteMinder variable objects. HTTP header variables are name value pairs contained in an HTTP request or response. HTTP cookie variables are name value pairs contained in an HTTP cookie. SiteMinder variable objects are objects that define a value which is obtained at runtime and may be included as the value of an HTTP header variable or an HTTP cookie variable.

Configure a Response Attribute that Contain Variables

A response may contain one or more response attributes whose values are determined by variable objects.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a response attribute that contains a variable

1. Follow the instructions in [Configure a Response](#) (see page 357) to create a response.
2. In the SiteMinder Response dialog, make sure the SiteMinder radio button is selected in the Agent Type group box and Web Agent is selected in the drop-down list.
3. Click Create.
The SiteMinder Response Attribute Editor dialog opens. The Attribute Kind group box contains a Variable Definition radio button.
4. From the Attribute drop-down list, select a type of response attribute.
5. In the Attribute Kind group box, select the Variable Definition radio button.
6. In the Attribute Fields groups box, enter a name in the Variable Name field.
This name will be used to identify the response attribute.
7. In the Variable field, enter the name of the variable object that will be evaluated to provide the value for the response attribute, or click the Lookup button to open a dialog where you can select an existing variable object.
If you click the Lookup button, the SiteMinder Variable Lookup dialog opens.
8. Click OK to save your changes and close the Response Attribute dialog.

More information:

[Response Attributes](#) (see page 356)

[Select a Variable Using Variable Lookup](#) (see page 365)

Select a Variable Using Variable Lookup

The SiteMinder Variable Lookup dialog allows you to quickly select existing variable objects from a tree structure. The tree contains branches for each type of variable object that you can configure. By expanding a branch, you can view a list of variable objects of the associated type that have been configured in the current policy domain.

To select a variable from the SiteMinder Variable Lookup dialog

1. Click on the + symbol next to the variable type to expand the list of existing variables.
2. Select a variable.
3. Click OK.

The variable object you selected will be substituted in the field associated with the Lookup button you clicked in the Response Attribute dialog.

Response Property Editing

All of the properties of a response, except the Agent Type, can be edited. If you want to change the Agent Type, you must delete the response and create a new one.

SiteMinder Generated User Attributes

The following list contains user attributes that SiteMinder generates automatically. These attributes can be specified as response attributes for Web Agent responses.

SM_USER

The Web Agent places the username in an SM_USER http header variable for all requests. In cases where the user does not provide a username, such as certificate-based authentication, or in the case where the username is not known, the value of the SM_USER header variable is not set.

SM_USERDN

For an authenticated user, the Web Agent populates this http header variable with the DN as determined by the Policy Server. In the case of certificate-based authentication, this attribute can be used to identify a user.

SM_USERNAME

For an authenticated user, this attribute holds the user DN as disambiguated by SiteMinder. For an unauthenticated user, this attribute holds the user ID as specified by the user in the login attempt.

SM_USERIMPERSONATORNAME

If the authentication scheme performs impersonation, this attribute holds the user DN that is authenticated by SiteMinder.

SM_USERLOGINNAME

This attribute holds the user ID as specified by the user in the login attempt.

SM_USERIPADDRESS

This attribute holds the user's IP address at the time of authentication or authorization.

SM_USERPATH

For an authenticated user, this attribute holds a string that represents the directory namespace and directory server (both as specified in the user directory definition), and user DN (as disambiguated by SiteMinder). For example:

“LDAP://123.123.0.1/uid=scarter,ou=people,o=airius.com”

For an unauthenticated user, this attribute holds the same value as SM_USERNAME.

SM_USERPASSWORD

This attribute holds the password as specified by the user in the login attempt. This attribute is only available after a successful authentication through the OnAuthAccept event. The value is returned only on authentication, not on authorization.

SM_TRANSACTIONID

This attribute holds the transaction ID that is generated by the agent.

SM_USERSESSIONSPEC

The user’s session ticket.

SM_USERSESSIONID

This attribute holds the session ID of a user who has already been authenticated, or the session ID that will be assigned to the user upon successful authentication.

SM_USERSESSIONIP

This attribute holds the IP address that was used during the original user authentication (upon establishment of a session).

SM_USERSESSIONUNIVID

This attribute holds the user’s universal ID. If no universal ID directory attribute is specified in the user directory definition, the value defaults to the user’s DN.

SM_USERSESSIONDIRNAME

This attribute holds the name of the user directory that the Policy Server is configured to use.

SM_USERSESSIONDIROID

This attribute holds the object ID of the user directory that the Policy server is configured to use.

SM_USERSESSIONTYPE

This attribute holds the user’s session type. The value is one of the following:

- 2 - session
- 1 - identity

SM_USERLASTLOGINTIME

This attribute holds the time, using GMT, that the user last logged in and was authenticated. This response attribute is only available for an OnAuthAccept authentication event. For this attribute to be populated, both of the following conditions must be true:

- Password Services is enabled
- The user has logged in through SiteMinder at least once

SM_USERPREVIOUSLOGINTIME

This attribute holds the time, using GMT, of the successful login prior to the last (which is represented by SM_USERLASTLOGINTIME). This response attribute is only available for an OnAuthAccept authentication event. For this attribute to be populated Password Services must be enabled.

SM_USERGROUPS

This attribute holds the groups to which the user belongs. If the user belongs to a nested group, this attribute contains the group furthest down in the hierarchy. For all nested groups to which the user belongs, use SM_USERNESTEDGROUPS.

For example, if user JSmith belongs to the group Accounts Payable, which is contained in group Accounting, SM_OUSERNESTEDGROUPS[contains Accounts Payable. If you want both Accounting and Accounts Payable, use SM_USERNESTEDGROUPS.

SM_USERNESTEDGROUPS

This attribute holds the nested groups to which the user belongs. For only the group furthest down in the hierarchy, use SM_OUSERNESTEDGROUPS[.

For example, if user JSmith belongs to the group Accounts Payable, which is contained in group Accounting, SM_USERNESTEDGROUPS contains Accounting and Accounts Payable. If you want only Accounting, use SM_OUSERNESTEDGROUPS[.

SM_USERSCHEMAATTRIBUTES

This attribute holds the user attributes associated with the DN, or properties associated with the user. If the user directory is a SQL database, then SM_USERSCHEMAATTRIBUTES holds the names of the columns in the table where user data is stored. For example, using the SmSampleUsers schema, SM_USERSCHEMAATTRIBUTES holds the names of the columns in the SmUser table.

SM_USERPOLICIES

When a user is authorized for a resource, this attribute holds the names of the policies that give the user authorization. For example, suppose that to purchase an item, you must be one of the users associated with the Buyer policy. If the Policy Server authorizes me to buy an item, then SM_USERPOLICIES will contain Buyer.

SM_USERPRIVS

When a user is authenticated or is authorized for a resource, SM_USERPRIVS holds all of the response attributes for all policies that apply to that user, in all policy domains.

SM_USERREALMPRIVS

When a user is authenticated or is authorized for a resource under a realm, SM_USERREALMPRIVS holds all the response attributes for all rules under that realm.

For example, suppose that there is a realm called Equipment Purchasing. Under that realm, there is a CheckCredit rule. Associated with the CheckCredit rule is a response that returns the buyer's credit limit, such as limit = \$15000, as a response attribute. If the buyer attempts to purchase equipment worth \$5000, the CheckCredit rule fires. SM_USERREALMPRIVS would contain all of the response attributes for all of the rules under the Equipment Purchasing realm.

SM_AUTHENTICATIONLEVEL

When a user is authenticated for a resource, this attribute holds an integer number (of 0 to 1000) that represents the protection level of the authentication scheme under which the user was authenticated.

SM_USERDISABLEDSTATE

This attribute holds a decimal number that represents a bit mask of reasons that a user is disabled. The bits are defined in SmApi.h under the Sm_Api_DisabledReason_t data structure, which is part of the SDK.

For example, a user may be disabled as a result of inactivity, Sm_Api_Disabled_Inactivity. In Sm_Api_DisabledReason_t, the reason Sm_Api_Disabled_Inactivity, corresponds to the value 0x00000004. So, in this case, SM_USERDISABLEDSTATE is 4.

A user can be disabled for multiple reasons.

For more information on Sm_Api_DisabledReason_t, see the *Developer's Guide for C* (available only if the Software Development Kit is installed).

SM_USER_APPLICATION_ROLES

If you have purchased CA Identity Manager (Identity Manager), this attribute may be used in responses. It contains a list of all roles assigned or delegated to a user. If an application name is specified, only the roles associated with the application are returned in the response attribute.

The syntax of the response attribute in the Attribute Name field of the Response Attribute Editor dialog is as follows:

`SM_USER_APPLICATION_ROLES[:application_name]`

where is an optional name of an application defined in Identity Manager .

Note: The value for *application_name* must be communicated to the Policy Server administrator. Application names are not automatically passed to the Policy Server User Interface.

For information about Identity Manager roles, see *CA Identity Manager Operations Guide*.

SM_USER_APPLICATION TASKS

If you have purchased CA Identity Manager (Identity Manager), this attribute may be used in responses. It contains a list of all tasks assigned or delegated to a user. If an application name is specified, only the tasks associated with the application are returned in the response attribute.

The syntax of the response attribute in the Attribute Name field of the Response Attribute Editor dialog is as follows:

SM_USER_APPLICATION_TASKS[:*application_name*]

where is an optional name of an application defined in Identity Manager .

Note: The value for *application_name* must be communicated to the Policy Server administrator. Application names are not automatically passed to the Policy Server User Interface.

For information about Identity Manager tasks, see *CA Identity Manager Operations Guide*.

More information:

[Configure a Web Agent Response Attribute](#) (see page 360)

Availability of SiteMinder-generated Response Attributes

The following table shows the availability of SiteMinder generated response attributes during authentication, authorization and impersonation events.

Response Attribute	Authentication and Authorization Events					Impersonation Events
	GET/PUT	On Auth Accept	On Auth Reject	On Access Accept	On Access Reject	Impersonation Start User
SM_USERNAME	Yes	Yes	Yes	Yes	Yes	No
SM_USERPATH	Yes	Yes	Yes	Yes	Yes	No
SM_USERIPADDRESS	Yes	Yes	Yes	Yes	Yes	No

Response Attribute	Authentication and Authorization Events					Impersonation Events
	GET/PUT	On Auth Accept	On Auth Reject	On Access Accept	On Access Reject	Impersonation Start User
SM_USERPASSWORD	No	Yes	Yes	No	No	No
SM_TRANSACTIONID	Yes	No	No	Yes	Yes	No
SM_USERSESSIONID	Yes	Yes	No	Yes	Yes	No
SM_USERSESSIONSPEC	Yes	No	No	Yes	Yes	No
SM_USERSESSIONIP	Yes	Yes	Yes	Yes	Yes	No
SM_USERSESSIONUNIVID	Yes	Yes	No	Yes	Yes	No
SM_USERSESSIONDIRNAME	Yes	Yes	No	Yes	Yes	No
SM_USERSESSIONDIROID	Yes	Yes	No	Yes	Yes	No
SM_USERSESSIONTYPE	Yes	Yes	No	Yes	Yes	No
SM_USERLASTLOGINTIME	No	Yes	No	No	No	No
SM_OUSERNESTEDGROUPS	Yes	Yes	No	Yes	Yes	No
SM_USERNESTEDGROUPS	Yes	Yes	No	Yes	Yes	No
SM_USERSCHEMAATTRIBUTES	Yes	Yes	Yes	Yes	Yes	No
SM_USERLOGINNAME	No	Yes	Yes	No	No	No
SM_USERIMPERSONATORNAME	No	No	No	No	No	Yes
SM_USERDISABLEDSTATE	Yes	Yes	No	Yes	Yes	No
SM_USERPOLICIES	No	No	No	Yes	No	No
SM_USERREALMPRIVS	Yes	No	No	No	No	No
SM_USERPRIVS	Yes	No	No	No	No	No

Active Responses Created for Custom Agents

The Policy Server does not send an active response to a custom agent when the value of the active response is null. You can override this default behavior by modifying the DWORD registry key named EnableNullValueResponse at the following location:

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion
\PolicyServer

To Enable Null Value Active Responses

1. From the Windows Start menu, select Run.
2. Type regedit in the Open field on the Run dialog, and click OK.
3. In the Registry Editor, navigate to:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion
\PolicyServer
4. Modify the DWORD registry value, as follows:
Key Name: EnableNullValueResponse
Key Type: DWORD
Key Value: 1 (enable null value active responses)
Default Value: 0 (disable null value active responses)
5. Restart the Policy Server.

Note: The registry key named EnableRedirectNullValueResponse is deprecated.

Response Deletion

You can delete a response from the Policy Server User Interface. When you delete a response, the response is removed from any policies that contained the response.

Note: It may take a short amount of time for all deleted objects to be removed from caches.

Response Groups

A response group is a collection of responses that are logically grouped so they can be applied to a single rule within a policy. All relevant responses in a response group will fire when a rule paired with the response group fires.

Response groups allow you to combine multiple responses in a single object. When you create policies, you can more easily associate multiple responses with a single rule within those policies.

Configure a Response Group

Once you create more than one response, you can combine the responses in a single response group.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a response group

1. Log into the Policy Server User Interface.
2. From the menu bar of the SiteMinder Administration window, select View, Response Groups.

The Policy Server User Interface places a check mark in the View menu to the left of the menu selection and *Response Groups* appear in the System tab of the SiteMinder Administration window.

Note: If there is already a check next to *Response Groups* in the View menu, selecting the option again will remove *Response Groups* from the System tab.

3. If You have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Select the policy domain to which you want to add a response group.
4. If you have the Manage Domain Objects privilege:

In the Object Pane, select the policy domain to which you want to add a response group.
5. From the menu bar, select Edit, <domain name>, Create Response Group.

The SiteMinder Response Group dialog opens.
6. In the Name field, enter the name of the new response group.
7. Optionally, in the Description field, enter a brief description in the new response group.
8. In the Agent Type group box, select one of the following radio buttons:

SiteMinder

Indicates the response group passes information to a SiteMinder Agent.

RADIUS

Indicates the response group passes information to a RADIUS Agent.

9. If you selected the SiteMinder radio button, select a SiteMinder Agent type from the drop-down list in the Agent Type group box.

Your choices include Web Agent or Affiliate Agent. Select the Agent type to which the response group must pass information.

10. If you selected the RADIUS radio button, select a RADIUS Agent type from the drop-down list in the Agent Type group box.

Your choices include a list of RADIUS vendors, as well as an option for Generic RADIUS. Select the Agent type to which the response group must pass information.

11. Click Apply to save your changes.

You can now add responses to your response group. See the next section for details about adding responses.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Response Dialog](#) (see page 705)

Response Group Property Editing

You can change all of the properties of a response group, except Agent Type. To change the Agent Type of a response group, you must delete the response group and create a new group of the appropriate Agent Type.

Response Group Deletion

You can delete a response group from the Policy Server User Interface, however, when you delete a response group, the Policy Server User Interface only deletes the grouping, not the individual responses contained in the group.

Chapter 16: Policies

This section contains the following topics:

- [Policy Overview](#) (see page 375)
- [Configure a Policy](#) (see page 378)
- [Enable and Disable Policies](#) (see page 390)
- [Advanced Policy Options](#) (see page 391)
- [Policy Binding Establishment](#) (see page 396)
- [Policy Deletion](#) (see page 405)
- [Policy Processing](#) (see page 405)
- [Expressions in Policies](#) (see page 413)

Policy Overview

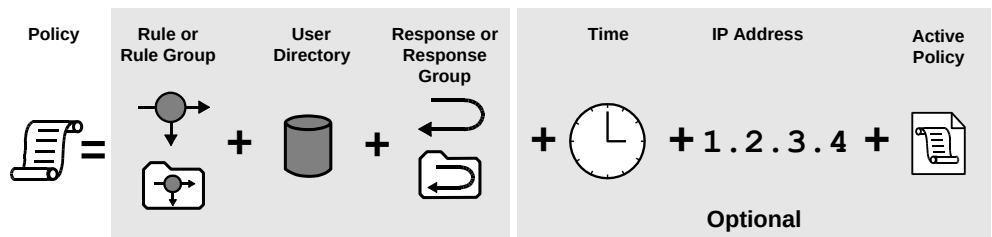
Policies define how users interact with resources. When you create policies in the Policy Server User Interface, you link together (bind) objects that identify users, resources, and actions associated with the resources.

Policies are stored in policy domains. When you configure a policy, you can select users and groups from the user directories available in the policy domain.

SiteMinder identifies resources through rules. When you create a policy, you can select rules that specify the resources you want to include in a policy.

Once you identify users and resources in a policy, you can specify actions that should take place when those users access the specified resources. These actions take the form of responses. Policies can include responses that allow or deny access to a resource, customize a user's session time, redirect the user to other resources, or customize the content the user receives based on attributes contained in a user directory.

The following diagram illustrates all of the possible parts of a policy. These parts are described briefly following the diagram, and in more detail throughout the rest of this chapter.



Rules/Rule Groups

A policy must contain at least one rule or rule group. A rule identifies a specific resource or resources that are included in the policy.

Users

A policy must specify the users or groups of users that are affected by the policy. Connections to these users or groups of users must be configured in the SiteMinder User Directory dialog. Only users or user groups for directories that are included in the policy domain in which the policy is located may be associated with a policy.

Responses

A response defines the action that is triggered when a user accesses a resource specified in a rule. Responses can return attributes from a user directory for use by other applications or to customize the appearance of a resource. Responses can also trigger actions based on authentication and authorization events.

(Optional) IP Addresses

A policy may be limited to specific user IP addresses. Once you add an IP address restriction to a policy, if a user attempts to access a resource from an IP address not specified in the policy, the policy will not fire for the user, and therefore will not allow/deny access or process any responses.

(Optional) Time Restrictions

A policy may be limited to specific days or ranges of hours. A policy with a time restriction will not fire outside specified times, and therefore will not allow/deny access to protected resources or process any responses.

(Optional) Active Policies

An Active policy allows business logic external to SiteMinder to be included in a policy definition. Active policies allow SiteMinder to interact with custom software created using the SiteMinder APIs.

More information:

[Domains](#) (see page 307)

[Rules](#) (see page 327)

[Responses and Response Groups](#) (see page 353)

[User Directories](#) (see page 141)

[Specify Allowable IP Addresses for Policies](#) (see page 391)

[Add and Remove Time Restrictions for Policies](#) (see page 394)

[Configure an Active Policy](#) (see page 395)

Policies Explanation

Policies bind other Policy Server objects together into a logical group that determines how the objects should interact. By linking together users that are accessible through directory connections, rules that point to specific resources, and responses that define actions, policies define who is authorized to access resources. Responses included in policies can also provide personalization by retrieving directory attributes when a user accesses a resource.

When one of the users specified in a policy attempts to access a resource identified in one of the policy's rules, the Policy Server uses the information contained in the policy to resolve whether or not the user can access the resource, and if any personalization should take place.

More advanced policies can be restricted to certain time periods or certain user IP addresses. This allows administrators of a group of resources a finer control over their resources.

Policy Bindings

A policy binding is the method used to link a user with a policy. The Policy Server only resolves policies for users who are part of a policy binding created by the users or groups contained in a policy.

Before the Policy Server can resolve a user's attempt to access a protected resource, the user must be authenticated. When SiteMinder authenticates a user, it establishes a context for the user. The user context provides information about who the user is and what privileges the user has when accessing resources.

For example, if a user is part of the group in a user directory called Employees, when the user authenticates, the Policy Server creates a policy binding for the user's membership in the group Employees. When the user attempts to access a resource protected by a rule in a policy that allows access for Employees group members, the user's policy binding allows SiteMinder to authorize the user.

More information:

[Authentication Schemes](#) (see page 231)

[Policy Binding Establishment](#) (see page 396)

More information:

[CA Arcot WebFort and RiskFort](#) (see page 1015)

[CA Arcot A-OK](#) (see page 1023)

Confidence Levels in Policies

The CA Arcot integration makes available a confidence level for use in policies. Confidence levels extend policies to include the results of the risk evaluation that CA Arcot completes when authenticating a user.

The Policy Server can use these results when making authorization decisions.

Configure a Policy

The following procedure describes the process for configuring a policy in the Policy Server User Interface.

Note: You can also create policies using the Scripting Interface for Perl. For more information, see the *SiteMinder Scripting Guide for Perl*.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a policy

1. Log into the Policy Server User Interface.
2. Open the Policy Dialog.
3. Enter a Name and optional Description for the new policy.
4. Add users and groups to the policy.
5. Add one or more rules to the policy.
6. (Optional) Associate responses with rules.
7. (Optional). Configure advanced policy options.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Policy Dialog](#) (see page 733)

[Add Users to a Policy](#) (see page 379)

[Add Rules to a Policy](#) (see page 380)

[Associate a Rule with a Response or Response Group](#) (see page 380)

[Associate a Rule with a Global Response](#) (see page 381)

[Advanced Policy Options](#) (see page 391)

Add Users to a Policy

The next step in creating a policy is to add users or groups of users to the policy. When you add a user or group to the policy, you create a policy binding between the selected users and the policy. When a user tries to access a protected resource, the policy verifies that the user is part of its policy binding, and then fires the rules included in the policy to see if the user is allowed to access the resource.

To add users to a policy

1. In the Policy Dialog, click on the Users tab.

If the current policy domain contains more than one user directory, the directories appear as tabs contained in the User tab.

2. Click the Add/Remove button.

The Policy Users/Groups Dialog opens.

3. To add users to the policy, select an entry from the Available Members list and click on the Left Arrow button, which points to the Current Members list.

The opposite procedure removes users from the Current Members list.

You can select multiple entries by holding the CTRL or SHIFT key and clicking on entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the Policy Server User Interface moves all of the selected entries.

Individual users are not displayed automatically. However, you can use the Search utility to find a specific user within one of the listed groups. Different types of user directories must be searched differently.

4. Click OK to save your changes and return to the Policy Dialog.

More information:

[Policy Dialog](#) (see page 733)

[Users/Groups Dialog](#) (see page 912)

[View User Directory Contents](#) (see page 210)

[Policy Binding Establishment](#) (see page 396)

Add Rules to a Policy

Rules indicate the specific resources included in a policy, and whether to allow or deny access to the resources when the rule fires. Responses indicate the actions that should take place when the rule fires.

You must add at least one rule or rule group to a policy.

To add rules or rule groups to a policy

1. In the Policy Dialog, click on the Rules tab.
2. Click the Add/Remove Rules button. A list of available rules and rule groups is displayed in the Rule Items Dialog.
3. To add a rule to the policy, select a rule from the Available Members list and click on the Left Arrow button, which points to the Current Members list.
4. The reverse procedure removes rules from the Current Members list.
5. You can select multiple rules by holding the CTRL or SHIFT key and clicking on rules in one of the Members lists. When you select multiple rules and click one of the Arrow buttons, all of the selected rules are moved to the other list.
6. Click OK to save your changes and return to the SiteMinder Policy dialog.

More information:

[Rule Items Dialog](#) (see page 765)

Associate a Rule with a Response or Response Group

You can associate a response or response group with a rule in a policy. When the rule fires, the associated response also fires.

To associate a rule with a response or response group

1. In the Policy Dialog, click on the Rules tab.
2. Select the rule to which you want to associate a response, and click the Set Response button.

The Set Response dialog opens. This dialog contains a list of the responses that have been configured for the selected policy domain.

3. Click on a response and click OK.

The Policy Server associates the response with the rule you selected in the Rule tab, and the Set Response dialog closes.

Associate a Rule with a Global Response

You can associate a rule with an existing global response.

To associate a rule with a global response

1. In the Policy Dialog, click on the Rules tab.
2. In the Rules tab, select a rule and click the Set Global Response button.

The Available Global Responses dialog opens.

3. Click on a global response and click OK.

The Policy Server associates the global response with the rule you selected in the Rule tab, and the Available Global Responses dialog closes.

More information:

[Global Policies, Rules, and Responses](#) (see page 435)

Exclude a User or Group from a Policy

The Policy Server User Interface allows you to exclude a user or group of users from a policy. This feature is very useful if you have a large user group that should be included in a policy, but you want to exclude a small subset of the group from the policy.

To exclude a user or group from a policy

1. Open the Policy Dialog.
2. Select the Users tab.
3. Click the Add/Remove button.
The Policy Users/Groups Dialog opens.
4. In the Policy Users/Groups dialog, select a user or group from the Current Members list.
5. To exclude the selected user or group, click Exclude.

The symbol to the left of the user or group in the Current Members list changes to the following to indicate that the user or group is excluded from the policy.



When you exclude a group from a policy, the exclusion indicates that anyone included in the policy who is a member of the excluded group (or the specifically excluded user), is not included in the policy. For example, if a policy contained the group Employees, and the excluded group Marketing, anyone who is a member of the Employees group, and not part of the Marketing group is included in the policy.

6. Click OK to return to the SiteMinder Policy dialog.

More information:

[Policy Dialog](#) (see page 733)

Allow Nested Groups in Policies

Your LDAP user directories may contain groups that contain other groups. In very complex directories, groups nested in a hierarchy of other groups is one way to organize tremendous amounts of user information. If you enable a policy to search for users in nested groups, any groups contained in a group you add to a policy will be searched when the Policy Server processes the policy. If you do not enable a policy to search nested groups, the Policy Server will only search in the group you specify in the policy, regardless of any nested groups that may exist.

To allow nested groups in a policy that contains an LDAP user directory

1. In the Policy Dialog, click on the Users tab.

If the current policy domain contains more than one user directory, the directories appear as tabs contained in the User tab. In the image above, three user directories (Sample WinNT, Sample ODBC, and Sample LDAP) have been included in the policy domain that contains the policy.

2. Select the Allow Nested Groups check box to enable nested groups searching for the policy.

More information:

[Policy Dialog](#) (see page 733)

AND Users/Groups Check Box

The AND Users/Groups check box lets you restrict authorization to users who are members of more than one user group or to a particular user who is a member of one or more user groups. When adding individual users and user groups in a user directory to a policy, you can specify AND relationships between them by selecting the check box. Alternately, you can specify OR relationships between them by clearing the check box.

When you specify AND relationships and apply the resulting policy to a user, the user must meet the following requirements to be authorized:

- The user must be a member of each user group that is bound to the policy.
- If an individual user is bound to the policy, the user must be that individual.

Note: A user who is excluded from the policy or is a member of a group that is excluded from the policy cannot be authorized.

Example: Assume that User1, Group1, and Group2 are all bound to a policy and that AND relationships are specified. In this case, test_user must be User1 and a member of Group1 and Group2 to be authorized.

Example: Assume that User1, User2, and Group1 are all bound to a policy and that AND relationships are specified. In this case, test_user cannot be both User1 and User2. Therefore, test_user cannot be authorized.

Important! Do not add two or more individual users to a policy and specify AND relationships. Because no single user can be more than one individual, the policy always fails.

To specify both AND and OR relationships, choose one of the following configurations:

- A Single Policy and Multiple User Directories

In this configuration, two or more user directories are available to a single policy. The relationship between individual users and user groups in a single directory can be AND or OR. The relationship between individual users and user groups in different directories is always OR.

Example: There are two user directories and a single policy. In each directory, there are two user groups, and an AND relationship is specified. Assume that Directory1 contains Group1 and Group2 and that Directory2 contains Group3 and Group4. In this case, test_user must be a member of Group1 and Group2 or a member of Group3 and Group4 to be authorized.

This can be expressed logically as follows:

Directory1(Group1 AND Group2) OR Directory2(Group3 and Group4)

Use Case: There are two user directories and a single policy. Directory1 contains the user groups Facilities and Human_Resources, and an AND relationship is specified. Directory2 contains the user groups Marketing and Sales, and an OR relationship is specified. In this case, the user must be a member of Facilities and Human_Resources or a member of Marketing or a member of Sales to be authorized. This can be expressed logically as follows:

Directory1(Facilities AND Human_Resources) OR Directory2(Marketing OR Sales)

- Multiple Policies and a Single User Directory

In this configuration, two or more policies in a shared domain have access to a single user directory. The relationship between individual users and user groups in the user directory can be AND in one policy and OR in another policy. The relationship between different policies in a shared domain is always OR.

Example: There are two policies and one user directory. The user directory contains four user groups. Assume that Group1 and Group2 are bound to Policy1 and that Group3 and Group4 are bound to Policy2. AND relationships are specified between the user groups in both policies. In this case, test_user can be authorized by the application of Policy1 or Policy2. This can be expressed logically as follows:

Policy1(Group1 AND Group2) OR Policy2(Group3 AND Group4)

Use Case: There are two policies and one user directory. The user groups Human_Resources, Marketing, and Sales are bound to Policy1, and an OR relationship is specified. The user groups Facilities and Human_Resources are bound to Policy2, and an AND relationship is specified. In this case, the user must be a member of Human_Resources, Marketing, or Sales or a member of Facilities and Human_Resources to be authorized. The second policy only authorizes members of Facilities who are also members of Human_Resources.

This can be expressed logically as follows:

Policy1(Human_Resources OR Marketing OR Sales) OR Policy2(Facilities AND Human_Resources)

Specify AND/OR Relationships between Users/Groups

The AND Users/Groups check box lets you restrict authorization to users who are members of more than one user group or to a particular user who is a member of one or more user groups. When adding individual users and user groups in a user directory to a policy, you can specify AND relationships between them by selecting the check box. Alternately, you can specify OR relationships by clearing the check box.

When you specify AND relationships and apply the resulting policy to a user, the user must meet the following requirements to be authorized:

- The user must be a member of each user group that is bound to the policy.
- If an individual user is bound to the policy, the user must be that individual.

Important! Do not add two or more individual users to a policy and specify AND relationships. Because no single user can be more than one individual, the policy always fails.

To specify AND relationships between a user and one or more user groups or between multiple user groups in one user directory

1. If the policy contains more than one user directory, select the tab corresponding to the user directory that you want on the Policy Properties pane.
2. Select the AND Users/Groups check box on the user directory's tab.

3. Click Apply.

The icons next to the individual users and user groups in the selected user directory are updated to show AND relationships.

Note: When you clear the check box and click Apply, the icons are updated to show OR relationships between the individual users and user groups in the selected user directory.

Add Users by Manual Entry

In addition to the using the Available Members list in the Policy Users/Groups Dialog to specify users and groups to be included in a policy, you can specify a user or search string in the Manual Entry group box.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add a user by manual entry

1. In the Policy Users/Groups Dialog, do one of the following:

For LDAP directories, enter a valid DN in the Manual Entry field.

When entering a manual entry for an LDAP user directory, you can also select one of the following from the Action drop down list:

Search Users

Indicates that the LDAP search specified in the Entry field is limited to matches in user entries.

Search Groups

Indicates that the LDAP search specified in the Entry field is limited to matches in group entries.

Search Organizations

Indicates that the LDAP search specified in the Entry field is limited to matches in organization entries.

Search Any Entry

Indicates that the LDAP search specified in the Entry field is limited to matches in user, group, and organization entries.

For Microsoft SQL Server, Oracle and WinNT directories, enter a user name in the Manual Entry field.

For an Microsoft SQL Server or Oracle, you can enter a SQL query, instead. For example:

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

The Policy Server will perform the query as the database user specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, you need to be familiar with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and want to add specific users, you could select from the SmUser table.

Note: For an LDAP directory, you can enter all in the Manual Entry field to bind the policy to the entire LDAP directory.

2. Click Add to Current Members.

The Policy Server User Interface adds the user or query to the Current Members list.

3. Click Apply to save your changes, or click OK to save your changes and return to the Policy Dialog.

More information:

[Users/Groups Dialog](#) (see page 912)

Enhance Policy Server's LDAP Authorization Performance

You can enhance the Policy Server's authorization performance for users stored in LDAP user directories by limiting the role-based authorization to a specific user record and not the user's role by doing the following.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To enhance the policy server's performance

1. On the Domains tab of the Policy Server User Interface, select the properties of a policy.
2. In the SiteMinder Policy Dialog, click the Users tab.
3. Click the Add/Remove button.
4. In the Entry field of the Users/Groups dialog, enter the attribute/value pair required for authorization.
5. Set the Action field to Validate DN.
6. Click Add Current Members. The search filter now appears in the Available Members list as a user attribute.

After making this change, the Policy Server's LDAP search is done within the context of the current user and not in the LDAP server's base DN. This optimization decreases the load on the LDAP server and Policy Server, which allows quicker authorization responses.

Add LDAP Expressions to Policies

If you create a policy in a policy domain that contains connections to an LDAP user directory, you can use the Policy Server User Interface's Expression Editor to bind an LDAP search expression to a policy. Search expressions can bind users to a policy based on attributes that appear in user, group, and organization profiles.

To add an LDAP expression using the Expression Editor

1. Open the Policy Dialog.
2. From the Users tab on the Policy Dialog, select the tab associated with the LDAP directory and click the Add/Remove button.

The Policy Users/Groups Dialog opens.

3. In the Policy Users/Groups Dialog, click the Create New Expression button.

The Expression Editor dialog opens.

4. Build an LDAP expression that binds a particular user, group, or organization attribute to your policy.
5. Click OK to save your expression and close the Expression Editor.

Your expression appears in the Current Members list of the SiteMinder Policy Users/Groups dialog. The expression also appears in the Manual Entry field.

You can use the expression in the manual entry field as a template for creating an additional LDAP expression.

The following sections describe other features available in the Expression Editor dialog that allow you to create extensive LDAP expression that can be added to your policies.

More information:

[Policy Dialog](#) (see page 733)

[Expression Editor](#) (see page 738)

Add Roles to a Policy

If you have purchased CA Identity Manager (Identity Manager), you can add Identity Manager access roles to your policies. Identity Manager access roles define roles that may be assigned to users in Identity Manager. Roles may be associated with users or groups of users that share common access requirements. For more information about access roles in Identity Manager, see *CA Identity Manager Operations Guide*.

You can add roles to policies using a procedure similar to adding groups to policies. When a user who has been assigned the appropriate access role in Identity Manager tries to access a protected resource, the Policy Server verifies that the user has been assigned the Identity Manager role, and then fires the rules included in the policy to see if the user is allowed to access the resource.

To add Identity Manager roles to a policy

1. In the SiteMinder Policy dialog, click on the Users tab.
The Users tab contains sub-tabs for each user directory and IdentityMinder Environment included in the policy domain.
2. Click on the tab for the IdentityMinder Environment that contains the roles you want to add to your policy.
3. Click the Add/Remove button.
The SiteMinder Policy Identity Manager Role dialog opens.
4. To add roles to the policy, select an entry from the Available Members list and click on the Left Arrow button, which points to the Current Members list.
The opposite procedure removes roles from the Current Members list.
You can select multiple entries by holding the CTRL or SHIFT key and clicking on entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the Policy Server User Interface moves all of the selected entries.
5. Click OK to save your changes and return to the SiteMinder Policy dialog.

Exclude Roles in a Policy

If you have purchased CA Identity Manager (Identity Manager), you can exclude Identity Manager access roles from your policies. Identity Manager access roles define roles that may be assigned to users in Identity Manager. Roles may be associated with users or groups of users that share common access requirements. For more information about access roles in Identity Manager, see *CA Identity Manager Operations Guide*.

You can exclude roles from policies using a procedure similar to excluding groups from policies. When a user who has been assigned the excluded access role in Identity Manager tries to access a protected resource, the Policy Server verifies that the user has been assigned the excluded Identity Manager role, and blocks access to the resource.

To exclude Identity Manager roles from a policy

1. In the SiteMinder Policy dialog, click on the Users tab.
The Users tab contains sub-tabs for each user directory and IdentityMinder Environment included in the policy domain.
2. Click on the tab for the IdentityMinder Environment that contains the roles you want to exclude from your policy.
3. Click the Add/Remove button.
The SiteMinder Policy Identity Manager Role dialog opens.
4. To add roles to the policy, select an entry from the Available Members list and click on the Left Arrow button, which points to the Current Members list.
The opposite procedure removes roles from the Current Members list.
You can select multiple entries by holding the CTRL or SHIFT key and clicking on entries in one of the Members lists. When you select multiple entries and click one of the Arrow buttons, the Policy Server User Interface moves all of the selected entries.
5. In the Current Members list, select the roles you want to exclude, and click the Exclude button located under the list.
A red circle with a slash appears to the left of the excluded roles.
6. Click OK to save your changes and return to the SiteMinder Policy dialog.

Add a Confidence Level to a Policy

You add a confidence level to a policy to apply the results of a risk evaluation to an authorization decision.

To add a confidence level to a policy

1. In the Policy Dialog, click the Advanced tab.
The Advanced tab appears.
2. Enter a name for the active expression and click Edit.
The SiteMinder Active Policy Editor screen appears.
3. Do the following:
 - a. Enter the following library name:
smriskactiveexpr

- b. Enter the following function name:
CheckConfidenceLevel
 - c. Enter a confidence level in the Function Parameters field. A valid value is 1-1000.
4. Click OK.
The Advanced tab appears.
5. Click OK.

Enable and Disable Policies

The Policy Server User Interface allows you to enable and disable policies. By default, when you create a policy, the policy is enabled. When a policy is enabled, rules contained in the policy fire when users attempt to access the resources specified in the rules.

If you disable a policy, the rules contained in the policy still fire, but no user will be authorized by the policy. Any resources specified in rules contained in the policy are still protected. Until you enable the policy, no users may access resources associated with the rules specified in the policy. However, if another enabled policy allows access to a resource in the disabled policy, users associated with the enabled policy may access the resource.

To enable or disable a policy

1. Open the Policy Dialog for the policy you want to enable or disable.
2. In the Policy Dialog, select the Enabled check box.

If the check box is marked with a check, the policy is enabled. If the check box is not marked with a check, the policy is disabled. If you disable a policy, the policy does not fire.
3. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Policy Dialog](#) (see page 733)

Advanced Policy Options

There are a number of advanced features you can use when setting up policies in the Policy Server User Interface. These features include the following:

IP Addresses

This feature allows you to specify certain IP addresses that a user must be using in order for a policy to fire.

Time Restrictions

This feature allows you to specify times during which the policy fires. If you add a time restriction to a policy, the policy is effectively disabled outside of the specified times.

Active Policies

This feature allows you to have a function call invoked in a shared library created with the SiteMinder API. The function call determines whether or not the policy fires.

More information:

[Enable and Disable Policies](#) (see page 390)

Specify Allowable IP Addresses for Policies

The Policy Server User Interface allows you to specify an IP address, range of IP addresses, or a subnet mask that users must use in order for a policy to fire. For example, if you include a rule that allows access to resources in the policy, and then you specify a range of IP addresses, only those users who login in from one of the specified IP addresses will be allowed access to the protected resources.

To specify IP addresses

1. Open the Policy Dialog.
2. Select the IP Addresses tab.
3. Click on the Add... button.
The Add an IP Address dialog opens.
4. Select one of the radio buttons to indicate whether you want to add a Single Host, Host Name, Subnet Mask, or Range of IP addresses to the policy.
5. Fill in the appropriate information based on your selection. See one of the following:
 - [Specify a Single IP Address](#) (see page 392)
 - [Add a Host Name](#) (see page 392)

- [Add a Subnet Mask](#) (see page 393)
- [Add a Range of IP Addresses](#) (see page 394)

6. Click OK to save IP address information and return to the SiteMinder Policy dialog.

More information:

[Policy Dialog](#) (see page 733)

[Add an IP Address Dialog](#) (see page 916)

Specify a Single IP Address

You can specify a single IP address for a policy. If you specify a single IP address, the policy only fires for users who access the policy's resources from the specified IP address.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To specify a single IP address to a policy

1. In the Add an IP Address Dialog, click on the Single Host radio button.

The dialog changes to display the fields associated with your selection.

2. Enter an IP Address in the fields provided.

If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog.

Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog closes and the IP address for the host name you entered appears in the SiteMinder Policy dialog.

3. Click OK to save the IP address and return to the SiteMinder Policy dialog.

The IP address appears in the list of addresses in the Policy Dialog.

More information:

[Add an IP Address Dialog](#) (see page 916)

Add a Host Name

You can specify a host name for a policy. If you specify a host name, the policy only fires for users who access the policy's resources from the specified host.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add a host name to a policy

1. In the Add an IP Address Dialog, click on the Host Name radio button.
The dialog changes to display the fields associated with your selection.
2. In the Host Name field, enter the host name.
3. Click OK to save your settings and return to the Policy Dialog.
The host name appears in the list of addresses in the SiteMinder Policy dialog.

More information:

[Add an IP Address Dialog](#) (see page 916)

Add a Subnet Mask

You can specify a subnet mask for a policy. If you specify a subnet mask, the policy only fires for users who access the policy's resources from the specified subnet mask.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add a subnet mask to a policy

1. In the Add an IP Address Dialog, click on the Subnet Mask radio button.
The dialog changes to display the fields associated with your selection.
2. Enter an IP Address in the fields provided.

If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog.

Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog closes and the IP address for the host name you entered appears in the SiteMinder Policy dialog.
3. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
4. Click OK to save your settings and return to the Policy Dialog.
The subnet mask appears in the list of addresses in the SiteMinder Policy dialog.

More information:

[Add an IP Address Dialog](#) (see page 916)

Add a Range of IP Addresses

You can specify a range of IP addresses for a policy. If you specify a range of IP addresses, the policy only fires for users who access the policy's resources from one of the IP addresses included in the range of addresses.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add a range of IP addresses to a policy

1. In the Add an IP Address Dialog, click on the Range radio button.

The dialog changes to display the fields associated with your selection.

2. Enter a starting IP Address in the From fields.

If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog.

3. Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog closes and the IP address for the host name you entered appears in the SiteMinder Policy dialog.

4. Enter an ending IP address in the To fields.

5. Click OK to save the IP address and return to the Policy Dialog.

The range of IP addresses appears in the list of addresses in the SiteMinder Policy dialog.

More information:

[Add an IP Address Dialog](#) (see page 916)

Add and Remove Time Restrictions for Policies

The Policy Server User Interface allows you to add time restrictions to a policy. When you add a time restriction, the policy only fires during the period specified by the time restriction. If a user attempts to access a resource outside of the period specified by the time restriction, the policy does not fire.

For example, if you create a time restriction for a policy that secures access to a resource, and specifies that the policy will only fire from 9am - 5 pm, Monday - Friday. A user will only be authenticated and authorized during the times indicated in the time restriction. The resources protected by the policy will not be available outside the times indicated.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To add a time restriction to a policy

1. Open the Policy Dialog.
2. Select the Time tab.
3. In the Time Restrictions group box, click Set.
The Time Dialog opens.
4. Add time restrictions as required.
5. When you finish entering time restrictions, click OK to save your changes and return to the Policy Dialog.

More information:

[How the Web Agent and Policy Server Calculate Time](#) (see page 83)

[Policy Dialog](#) (see page 733)

[Time Dialog](#) (see page 905)

[Add Time Restrictions for Rules](#) (see page 344)

How Rule and Policy Time Restrictions Interact

If you specify a time restriction for a policy, and that policy contains a rule with a time restriction, the policy fires during the times that are intersection of the two restrictions.

For example, if a policy has a time restriction of 9AM to 5PM, and a rule has a time restriction of Monday through Friday, then the policy only fires between 9AM and 5PM, Monday through Friday.

Configure an Active Policy

An active policy is used for dynamic authorization based on external business logic. An active policy is included in the authorization decision by having the Policy Server invoke a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

To configure an Active Policy

1. Open the Policy Dialog.
2. Select the Advanced tab.

3. In the Active Policy group box, click Edit.
The Active Policy Editor opens.
4. Enter information in the dialog as described in [Active Policy Editor Fields and Controls](#) (see page 911).
5. Click OK to save the Active Policy and return to the Policy Dialog.

More information:

[Policy Dialog](#) (see page 733)

[Active Policy Editor Dialog](#) (see page 910)

Policy Binding Establishment

The following sections describe the methods for establishing different types of policy bindings. Supported policy binding types differ based on the type of user directory in which user information is located.

Policy Bindings for LDAP Directories

When SiteMinder authenticates a user, it establishes a user context. Subsequently, access control policy decisions are based on the user context matching one of the criteria shown in the table below.

Description	
User	The user's Distinguished Name (DN) must match the DN specified in the policy.
User Attribute	The search expression specifying conditions related to user attributes must be true.

Description

User Group	The user's DN must be a member of the user group specified in the policy.
Group Attribute	The search expression specifying conditions related to the group attribute must be true.
Organizational Role	The user must occupy the organizational role specified in the policy.
Organization Unit	The user must be a member of the organizational unit specified in the policy. The Organizational Unit must be a part of a user's DN, group, or role (group and role are not used by default).
Organization	The user must be a member of the organization specified in the policy. The Organization must be a part of a user's DN, group, or role (group and role are not used by default).
Organization Attribute	The search expression specifying conditions related to the organization attribute must be true.
Custom Object Classes	SiteMinder can be configured to associate Policies with custom directory objects.

Generally, you bind users or user attributes to policies via the SiteMinder Policy dialog in the Policy Server User Interface by selecting an entry from the list of available directory entries. Individual users are not visible in the list of available directory entries. However, you can search for specific users within a directory and add the users directly to the policy.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Manual Entry Field

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Manual Entry Field

1. From the Users tab of the SiteMinder Policy dialog, click Add/Remove.

The SiteMinder Policy Users/Groups dialog opens.

2. In the Manual Entry field, specify a user DN.

For example: uid=JSmith, ou=people, o=myorg.org

Note: This user DN must match exactly the user's distinguished name. This feature will not match a subset of information contained in the user's DN.

3. Click the Add to Current Members button.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Search Feature

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Search feature

1. From the Users tab of SiteMinder Policy dialog, click Add/Remove.

The SiteMinder Policy Users/Groups dialog appears.

2. Click the Search button located below the Available Members list.

3. Enter search criteria and click OK.

The search criteria must be in the form of Attribute=Value. For example, to search for all users with a Common Name that begins with J, enter cn=J*.

The users that match your criteria appear in the Available Members list.

4. Add users to the Current Member list.
5. Click Apply to save your changes, or click OK to save changes and return to the SiteMinder Policy dialog.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

[Search Directories](#) (see page 211)

Bind Policies to User Attributes

To bind a policy to user attributes, specify an LDAP search expression that defines conditions related to user attributes that must be true. For example, to bind a policy to all people whose location (*l*) is *westcoast* or whose mail address (*mail*) ends with *string.com*, insert the following search expression (using a pipe (*|*) at the beginning) in the Manual Entry field:

```
(|(l=westcoast)(mail=*string.com))
```

More information:

[Add LDAP Expressions to Policies](#) (see page 387)

Bind Policies to User Groups

User Groups appear in the list of Available Members in the SiteMinder Policy Users/Groups dialog. To bind a policy to a User Group, move the User Group into the Current Members list in the dialog and save your changes.

Bind Policies to Organizational Roles

SiteMinder allows you to bind policies to an Organizational Role. When you bind a policy to an organizational role, users must be a member of the role in order for the policy to fire.

Bind Policies to Group Attributes

To bind a policy to group attributes, specify an LDAP search expression that defines conditions related to group attributes that must be true. For example, to bind a policy to all groups located in the state of Massachusetts in the USA, insert the following search expression in the Manual Entry field:

```
(&(c=USA)(s=Massachusetts))
```

More information:

[Add LDAP Expressions to Policies](#) (see page 387)

Bind Policies to Organization Units

To bind a policy to an organizational unit, specify an LDAP search expression that defines an organizational unit. For example, to bind a policy to all people whose organization unit (*ou*) is *marketing*, insert the following search expression in the Manual Entry field: `ou=Marketing`

Bind Policies to Organizations

To bind a policy to an organization, specify an LDAP search expression that defines an organization. For example, to bind a policy to all people whose organization (*o*) is *myorg.org*, insert the following search expression in the Manual Entry field: `o=myorg.org`

Bind Policies to Organization Attributes

To bind a policy to organization attributes, specify an LDAP search expression that defines conditions related to organization attributes that must be true. For example, to bind a policy to all organizations located in the state of Massachusetts in the USA, insert the following search expression in the Manual Entry field:

`(&(c=USA)(s=Massachusetts))`

More information:

[Add LDAP Expressions to Policies](#) (see page 387)

Binding Policies to Custom Object Classes

SiteMinder can be configured to bind policies to custom object classes. If you have the Software Development Kit installed, see the *Developer's Guide for C* for more information.

Policy Bindings for WinNT User Directories

When SiteMinder authenticates a user, it establishes a user context. Subsequently, access control policy decisions are based on the user context matching one of the criteria shown below.

User Namespace	Description
User	The user's user name must match the user name specified in the policy.
User Group	The user must be a member of the user group specified in the policy.

Generally, you bind users to policies via the Policy dialog in the Policy Server User Interface by selecting an entry from the list of available directory entries. However, individual users are not visible in the list of available directory entries.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Manual Entry Field

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Manual Entry field

1. From the Users tab of the SiteMinder Policy dialog, click Add/Remove.
The SiteMinder Policy Users/Groups dialog opens.
2. In the Manual Entry field, specify a user DN.
For example: uid=JSmith, ou=people, o=myorg.org
3. Click the Add to Current Members button.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Search Feature

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Search Feature

1. From the Users tab of SiteMinder Policy dialog, click Add/Remove.
The SiteMinder Policy Users/Groups dialog appears.
2. Click the Search button located below the Available Members list.
3. Enter search criteria and click OK.
The users that match your criteria appear in the Available Members list.

4. Add users to the Current Member list.
5. Click Apply to save your changes, or click OK to save changes and return to the SiteMinder Policy dialog.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

[Search Directories](#) (see page 211)

Policy Bindings for Microsoft SQL Server and Oracle User Directories

When SiteMinder authenticates a user, it establishes a user context. Subsequently, access control policy decisions are based on the user context matching one of the criteria shown in below.

User Namespace	Description
User	The user's name must match the user name specified in the policy.
User Group	The user must be a member of the user group specified in the policy.
User Attribute	The search expression specifying conditions related to user attributes must be true.
SQL query	The SQL query specifying conditions related to the user must be true.

Generally, you bind users or user attributes to policies via the Policy Users/Groups dialog in the Policy Server User Interface by selecting an entry from the list of available directory entries. However, individual users may not be visible in the list of available directory entries (depending on the setup of Query Enumerate in the SQL query scheme for the user directory).

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Manual Entry Field

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Manual Entry field

1. From the Users tab of the SiteMinder Policy dialog, click Add/Remove.

The SiteMinder Policy Users/Groups dialog opens.

2. In the Manual Entry field, specify a user name or a SQL query.

You can specify an individual user: Lisac

To add all users whose names begin with L, you could enter the query:

```
SELECT Name FROM SmUser WHERE Name LIKE 'L%'
```

3. Click the Add to Current Members button.

The Policy Server adds the query to the Current Members list.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

Bind Policies to Users with the Search Feature

There are two ways to bind individual users to a policy. The first is by using the Manual Entry field in the SiteMinder Policy Users/Groups dialog. The second is by using the Search feature in the SiteMinder Policy Users/Groups dialog.

To bind a policy to a user with the Search feature

1. From the Users tab of SiteMinder Policy dialog, click Add/Remove.

The SiteMinder Policy Users/Groups dialog appears.

2. Click the Search button located below the Available Members list.

3. Enter search criteria and execute the search.

For example, to search the SmSampleUsers schema for all users whose names begin with L, enter:

Search—Users

Attribute—Name

Value—L*

The users that match your criteria appear in the Available Members list.

4. Add users to the Current Member list.
5. Click Apply to save your changes, or click OK to save changes and return to the SiteMinder Policy dialog.

More information:

[Add Users by Manual Entry](#) (see page 385)

[Add Users to a Policy](#) (see page 379)

[Search Directories](#) (see page 211)

Bind Policies to User Attributes

To bind policies to user attributes, specify a search expression that defines conditions related to user attributes that must be true. For example, to bind a policy to all people whose area code is 555, insert the following expression in the Manual Entry field: (areacode='555').

Bind Policies to User Groups

User Groups appear in the list of Available Members in the SiteMinder Policy Users/Groups dialog. To bind a policy to a User Group, move the User Group into the Current Members list in the dialog and save your changes.

More information:

[Add Users to a Policy](#) (see page 379)

Bind Policies to SQL Queries

SiteMinder allows you to use the Manual Entry field to specify a SQL query to bind policies to users.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To use the Manual Entry Field to Bind Policies to a SQL Query

1. From the SiteMinder Policy dialog, click Add/Remove.

The SiteMinder Policy Users/Groups dialog opens.

2. In the Manual Entry field, enter an SQL query.

Specify an SQL query that finds all the users to which the policy should be bound. For example, to bind a policy to all people whose account balance is greater than \$1000, a query might look like this:

```
SELECT name FROM customers WHERE balance, 1000
```


The contents of the SQL query depend on your database schema.

3. Click the Add to Current Members button.

More information:

[Configure a Policy](#) (see page 378)

Policy Deletion

When you are deleting a policy, remember that all of the elements in the policy still exist, it is only the grouping (or binding) of those elements that you remove when you delete the policy.

Policy Processing

The Policy Server use policies to authorize access to resources and provide entitlements to authorized users. The resources protected by policies are usually stored in a hierarchy that mimics security, organizational, and business requirements and takes advantage of the SiteMinder's hierarchical or "nested" realms. The remainder of this chapter discusses how SiteMinder processes policies and gathers entitlements in a hierarchical structure.

More information:

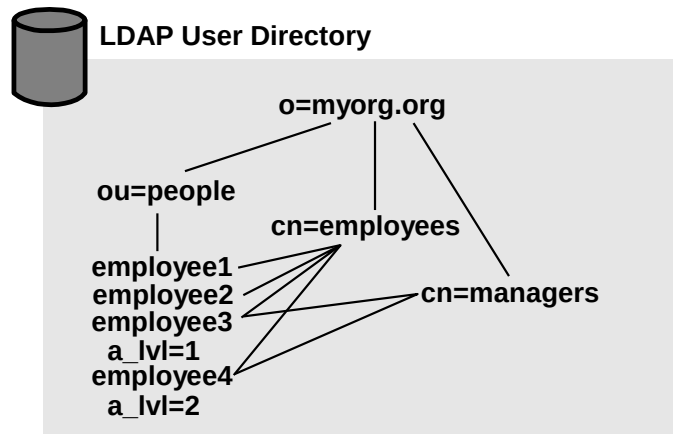
[Nested Realms](#) (see page 318)

Sample SiteMinder Configuration with Nested Realms

This section explains the sample configuration that will be used in examples in the remainder of this chapter.

User Directory

Assume that the policy domain that contains the policies and other relevant Policy Server objects includes a connection to the LDAP user directory in the following diagram.



The sample user directory contains the following:

o=myorg.org

This is an organization.

ou=people

This is an organizational unit that contains information for all employees.

employee<n>

These are directory entries for each employee. Note that `a_lvl` is a user attribute that indicates an access level. For the purpose of the examples in this section, assume that `employee1` and `employee2` have an access level of zero (`a_lvl=0`).

cn=employees

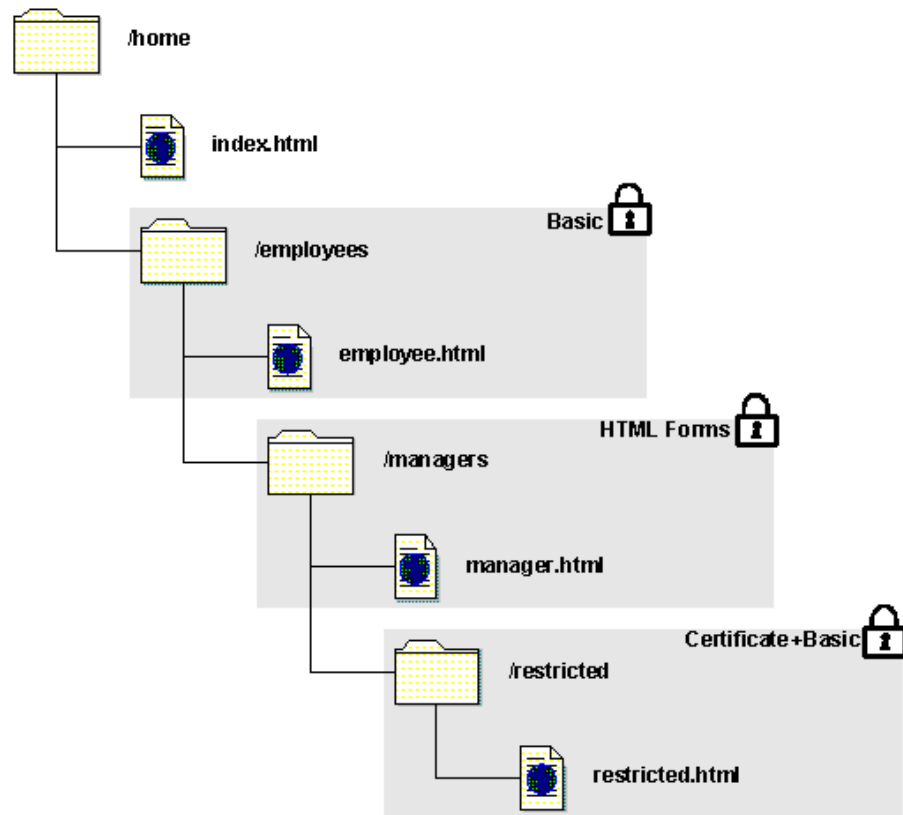
This is a group that contains all company employees as its members.

cn=managers

This is a group that contains all employees with a managerial title as its members. Note that `employee3` and `employee4` are the only employees in this group, and their respective access levels are greater than zero.

Nested Realms and Resources

The structure of nested realms and resources used in the examples in this section are illustrated in the following diagram. The shaded areas indicate realms that are protected by an authentication scheme.



The nested realms are made up of the following directories and resources:

/home

This directory is the top level of the sample nested realm. This realm specifies `/home/` as its resource filter, and contains the unprotected resource of `index.html`. The `index.html` file is not protected by an authentication scheme.

/employees

This directory, contained in the `/home` directory, is protected by a Basic authentication scheme, which requires a user name and password as credentials. The realm associated with this directory uses a resource filter of `employees/` and contains the `employee.html` file. The total Resource Filter for this realm is `/home/employees/`.

/managers

This directory, contained in the /employees directory, is protected by an HTML Forms authentication scheme which requires a user name, password, and additional personal information as credentials. The realm associated with this directory uses a resource filter of managers/ and contains the manager.html file. The total Resource Filter for this realm is /home/employees/managers/.

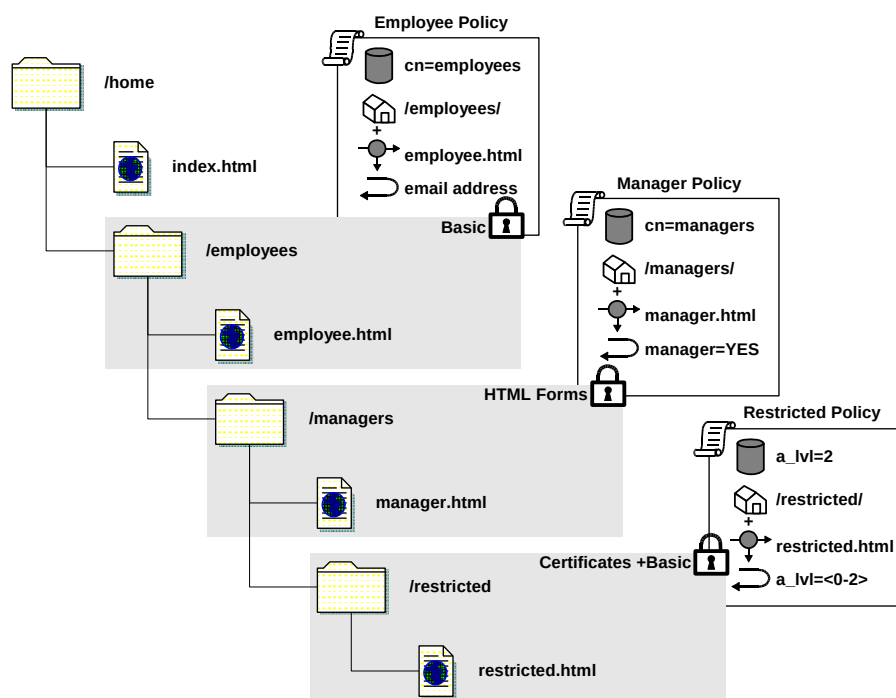
/restricted

This directory, contained in the /managers directory, is protected by an X.509 Certificate + Basic authentication scheme which requires user name, password, and a valid X509 certificate as credentials. The realm associated with this directory uses a resource filter of restricted/ and contains the restricted.html file. The total Resource Filter for this realm is /home/employees/managers/restricted/.

In these examples the protection level of the /employees realm is less than the protection level of the /managers realm, which is less than the protection level of the /restricted realm.

Policies and Responses

The policies and responses used in the examples in the remainder of the chapter are illustrated in the following diagram and described below.



The following is a description of each of the sample policies and the objects contained in each policy.

Employee Policy

This policy contains a Get rule that protects the employee.html resource. This resource is located in the /employees realm. The policy binds the user group cn=employees, so that all employees in the LDAP directory can access the resource once they are successfully authenticated. When an authenticated user is authorized by this policy, SiteMinder returns a response of the user's email address.

For example, if employee1 attempts to access /home/employees/employee.html and is successfully authenticated, the Policy Server allows employee1 to access the resource and returns the email address: employee1@myorg.org

A Web application can use this response for customization when accessing other company resources.

Manager Policy

This policy contains a Get rule that protects the manager.html resource. This resource is located in the /manager realm. The policy binds the user group cn=managers so that only employees contained in cn=managers group can access the resource once they are successfully authenticated.

When an authenticated manager is authorized by this policy, SiteMinder returns a static response. In the example, if employee3 attempts to access /home/employees/managers/manager.html and is successfully authenticated, the Policy Server allows employee3 to access the resource and returns the following response:

manager=YES

An application can use this response to activate features that are only available to company managers.

Restricted Policy

This policy contains a Get rule that protects the restricted.html resource. This resource is located in the /restricted realm.

The policy binds only the employees in the directory who have an access level user attribute of two (a_lvl=2). Managers with the correct access level can access the resource once they are successfully authenticated. When a user attempts to access the restricted.html resource, SiteMinder returns a response of a_lvl=<0-2>. For example, if employee4 attempts to access /home/employees/managers/restricted/restricted.html and is successfully authenticated, the Policy Server allows employee4 to access the resource and returns the following response:

a_lvl=2

An application can use this response to activate features that are only available employees with an access level of two.

Authentication Processing for Hierarchical Policies

Policies must contain rules. Rules can include authentication and authorization events. Based on how rules are configured, one of four authentication events can occur when the Policy Server attempts to identify a user based on credentials.

OnAuthAccept

The authentication is successful.

OnAuthAttempt

The authentication fails because the user is not found in any directory in the policy domain's search order.

OnAuthReject

The user *is found in a directory*, but the authentication fails because the credentials supplied by the user are incorrect. If this occurs, the Policy Server looks in the next directory in policy domain's directory search order. If the user's credentials cannot be verified in any of the directories in the search order, the Policy Server processes OnAuthReject events.

The user must be found in a directory for an OnAuthReject rule to fire. If the user is not found in any directory, an OnAuthReject rule will not fire.

OnAuthUserNotFound

The authentication server has a valid session ticket, yet it cannot find the user directory. This situation should only occur in a single sign-on environment that uses multiple directories, though it may take place if a user was idle long enough to be removed from the Agent's cache, and the user was removed from the directory. When this event occurs, the Policy Server evaluates the user's existence in the directory rather than relying on the session ticket.

The Policy Server attempts to authenticate users based on the longest matching realm. For example, if a user attempts to access `/home/employees/managers/manager.html`, the Policy Server uses the `/managers` realms to determine the required credentials. In the example in the previous figure, the user must complete a browser-based form required by the HTML Forms authentication scheme associated with the realm.

Note: The longest matching realm also determines the time-outs for the user's session. If a timeout is associated with the realm in which the user successfully authenticated, that timeout is used. You can use responses to override a realm timeout for a specific resource or group of resources.

More information:

[Configure Response Attribute Caching](#) (see page 363)

Successful Authentications

OnAuthAccept rules fire when all of the following are satisfied:

- A user enters credentials that satisfy the requirements of the longest matching realm.
- The Policy Server finds the user in a user directory.
- The Policy Server verifies the user's credentials.

At this point, the Policy Server collects OnAuthAccept entitlements from the longest matching realm, and any realms above it in the hierarchy of nested realms. In the example in the previous figure, if a user is successfully authenticated for the resource `/home/employees/managers/manager.html`, the Policy Server collects any OnAuthAccept entitlements for the `/managers` realm, then the `/employees` realm, and finally the `/home` realm.

Rejected Authentication Attempts

Policy domains are configured with a directory search order. When the Policy Server attempts to authenticate a user, it searches each user directory in the search order until it finds the user and verifies the supplied credentials. If the Policy Server locates a user in a directory, but the credentials supplied by the user do not match, the Policy Server looks at the next directory in the search order. If the Policy Server does not find a match for the user in any directory, the user's authentication attempt fails in the context of the realm that contains the requested resource.

For example, if a user attempts to access `/home/employees/managers/manager.html`, and the user is located in a user directory, but fails to provide valid credentials for any directory in the search order, the authentication event fails in the `/managers` realm. The Policy Server then processes any events for a rejected authentication attempt in that realm (OnAuthReject).

More information:

[Domains and User Membership](#) (see page 308)

Unsuccessful Authentication Attempts

If the Policy Server cannot find the user who attempts to authenticate in any of the directories in the directory search order, the authentication fails in the context of the realm that contains the requested resource. The Policy Server then processes any events for a failed authentication attempt (OnAuthAttempt).

Authorization Processing for Hierarchical Policies

Policies can contain rules that allow access to resources and may also include rules that trigger SiteMinder events. The possible authorization events include the following:

OnAccessAccept

This type of event occurs when a user is successfully authorized.

OnAccessReject

This type of event occurs when an authenticated user is denied access to a resource.

If a rule does not specify an authorization event, the rule either allows or denies access to the resource.

Policy Processing for Authorized Users

For the Policy Server to authorize a user to access a resource in a nested realm, the user must be authorized in each realm from the top of the hierarchy to the realm that contains the resource. Once a user is authenticated, the Policy Server checks the policies governing each realm above the resource to make sure the user is authorized. In the previous example, if employee3 wants to access `/home/employees/managers/manager.html`, the Policy Server verifies that employee3 is authorized in `/home`, `/employees`, and finally, `/managers`.

Once the Policy Server determines that the user is authorized, it collects entitlements for the user from each of the nested realms. Next, the Policy Server processes `OnAccessAccept` events starting at the top of the realm hierarchy and moving through the longest matching realm. In the previous example, the responses returned by the Policy Server would include employee3's email address, as well as the static response `manager=YES`.

Policy Processing for Unauthorized Users

Policy processing for unauthorized users is the same, whether or not the user is explicitly denied access by a policy, or simply not contained in a policy that allows access. Any entitlements collected so far for that user are lost and the Policy Server processes `OnAccessReject` events in the context of the rejecting realm where the requested resource is located.

In the previous example, if employee1, who is not a member of the `cn=managers` group, attempts to access `/home/employees/managers/manager.html`, the Policy Server only processes `OnAccessReject` events for the `/managers` realm.

Directory Mapping for Hierarchical Policies

In a SiteMinder configuration that contains nested realms, each realm may contain a directory mapping which specifies an authorization directory to be used which is different from the authentication directory. This allows companies to use a single directory for authenticating users, while distributing authorization directories throughout an enterprise. When the Policy Server processes entitlements in a group of nested realms, the it checks for directory mappings for each realm in the hierarchy. If there is no directory mapping, the Policy Server uses the authentication directory as the authorization directory.

More information:

[Directory Mappings Dialog](#) (see page 590)

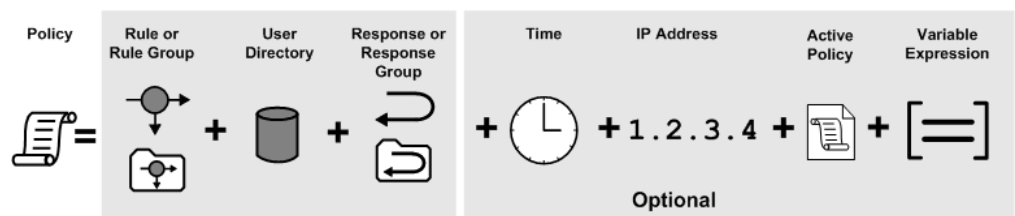
Expressions in Policies

eTelligent Rules makes available a set of variables for use in policy expressions. To use eTelligent rules, you must also purchase and install the Web Agent Option Pack. For information on installing the Option Pack, see the *Policy Server and Web Agent Option Pack Guide*.

Variable expressions extend policies to include dynamic information evaluated at runtime. Expressions can be added to policies that make use of variable objects. Variable objects may be used in expressions to create a boolean set of conditions that determines entitlements for the resources protected by the policy.

To use variable objects in an active expression within a policy, you must configure a policy object and build the appropriate boolean expression using the SiteMinder Policy Expression dialog. The interface is similar to the LDAP Expression editor described in [Add LDAP Expressions to Policies](#) (see page 387).

Note: Expressions may be added in addition to all other data supported by policy objects as shown in the following figure.



More information:

[Variables](#) (see page 421)

Add an Expression to a Policy

Expressions may be added to a policy in addition to all other data supported by policy objects.

To add an expression to a policy

1. In the Policy Dialog, select the Expression tab.
2. In the Expression group box, click the Edit button.

The Policy Expression Dialog opens.

3. Add and manipulate conditions to create a boolean expressions that incorporate variable objects in policy decisions.
4. Click OK to save your changes and return to the SiteMinder Policy dialog.

More information:

[Policy Dialog](#) (see page 733)

[Policy Expression Dialog](#) (see page 745)

Chapter 17: Certificate Authorities and Web Services Variables

This section contains the following topics:

[Overview](#) (see page 415)

[Keys and Certificates in the Key Database](#) (see page 415)

[The smkeydatabase.properties File](#) (see page 416)

[Create and Manage the Key Database Using Smkeytool](#) (see page 417)

Overview

To use SSL connections while resolving Web Services variables, you must configure a list of trusted Certificate Authorities (CAs) that can be used when the Policy Server establishes a connection to a Web Service. To accomplish this, you must set up a key database for each Policy Server that is responsible for connecting to a Web Service.

The key database is a flat-file key and certificate database that lets you store, manage, and retrieve keys and certificates required to sign and validate messages with WS-Security tokens. The service is also responsible for decrypting symmetric XML encryption keys that have been encrypted using the site's public key.

The SiteMinder smkeytool utility lets you create a new key database or delete an existing one and create a new one. There can only be one key database per Policy Server.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

Keys and Certificates in the Key Database

The following gets stored in the key database:

- An enterprise private key and certificate for signing WS-Security documents and generating X509v3 tokens.
- Public key certificates that correspond to the private keys used for signing the WS-Security documents and generating the tokens. These certificates are used to validate the WS-Security documents.

A given Policy Server may sign and/or verify WS-Security documents. Keys and certificates for signing and validation can be added to the same key database, depending on what the Policy Server is doing.

The table in the following figure shows which objects you need to add to the key database to handle particular WS-Security signing and validation requirements.

Function	WS-Security Token Type	Required Database Objects
Signing	All	Private key and certificate of Web service host enterprise.
Generating X509 Tokens	X509v3	Private key and certificate of Web service host enterprise.
Signature Validation	SAML Assertion; Sender Vouches	Certificate of issuing Web service consumer application.
	SAML Assertion; Holder-of-key	Certificates of XML request subject and issuing Web service consumer application.
	X.509v3; Username (if signed)	Certificate of trusted issuer.

The smkeydatabase.properties File

The key database properties file, smkeydatabase.properties, defines the configuration parameters required to access and manage the key database.

You should not have to modify this file, with the exception of possibly changing the directory where the database resides or the name of the database.

The properties file is installed in the following location:

- <policy_server_home>\config\properties (Windows)
- <policy_server_home>/config/properties (UNIX)

For reference, the following table describes the parameters in the file.

Entry	Default Value	Description
DBLocation	<code><policy_server_home>\smkeydatabase\</code>	Path to the directory in which the database resides. Specify the location that smkeytool should use when you manually create the database
NativeDBName	smkeydatabase	Name of the Database. Specify the name you want smkeytool to use when you manually create one.
XMLDocumentOpsImplementation	com.netegrity.smkeydatabase.api.XMLDocumentOpsImpl	Java class that implements XML signing and validation. Do not change this value; it is static and preconfigured.
EncryptedPassword	<code><encrypted_password_string></code>	Database password. (Encrypted using the policy store key at database creation.) If you have not already created a database, this entry will contain a dummy value.

Sample smkeydatabase.properties file:

```
#SMKeyDatabase properties file
#Fri Dec 10 14:03:07 EDT 2004
DBLocation= C:/Program Files/Netegrity/SiteMinder/Db/smkeydatabase
NativeDBName=smkeydatabase
XMLDocumentOpsImplementation=com.netegrity.smkeydatabase.api.XMLDocumentOpsImpl
EncryptedPassword={RC2}HduPZ7Vluuc/j9p6PVz0XaYs/6CbYru6
```

Create and Manage the Key Database Using Smkeytool

The smkeytool command-line utility allows you to populate and manage the key database. This tool is installed with the Policy Server Option Pack.

Use smkeytool to:

- Create and delete a key database

You can only have one key database per Policy Server. After the database is created, you can add keys and certificates.
- Add and delete the private key and certificate

- Add and delete an issuer certificate
- List all certificates stored in the key database

Note: smkeytool relies on values in the smkeydatabase.properties file. Ensure that this file is properly configured before running smkeytool.

smkeytool is located in the following directory:

- `<policy_server_home>/bin` (UNIX)
- `<policy_server_home>\bin` (Windows)

Run the smkeytool utility from a command line, using the following syntax:

- UNIX: `smkeytool.sh option [argument(s)]`
- Windows: `smkeytool.bat option [argument(s)]`

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

The options and arguments are described in the following table.

Option	Arguments	Function
-createDB or -cdb	<code><password></code>	Creates an empty key database to store keys and certificates. The specified password is encrypted using the policy store key and added to the smkeydatabase.properties file.
-deleteDB or -ddb	None	Deletes the key database specified in the smkeydatabase.properties file.
-addPrivKey or -apk	<code><private_key_filepath></code> <code><x.509_certificate_filepath></code> <code><password></code>	Adds the specified private key and corresponding certificate file to the key database. Note that <code><password></code> is the password used to encrypt the private key file being loaded, not the one associated with the key database.
-deletePrivKey or -dpk	<code><x.509_certificate_filepath></code>	Deletes the private key entry from the key database based on the specified certificate.

Option	Arguments	Function
-addCert or -ac	<x.509_certificate_filepath>	Adds a certificate to the key database.
-deleteCert or -dc	<x.509_certificate_filepath>	Deletes a certificate from the key database based on the specified certificate.
-listCerts or -lc	None	Lists the issuer/subject name (DN) and serial number of all the certificates stored in key database.
-help or -h	None	Lists smkeytool usage information.

Smkeytool Examples

- Create a key database

UNIX:

```
smkeytool.sh -cdb password
```

Windows:

```
smkeytool.bat -cdb password
```

- Add a private key and certificate:

UNIX:

```
smkeytool.sh -apk /opt/netegrity/webagent/certs/samplePrivKey.pkcs8  
/opt/netegrity/webagent/certs/sampleRobm.cer passphrase
```

Windows:

```
smkeytool.bat -apk "c:\program  
files\netegrity\webagent\certs\samplePrivKey.pkcs8" "C:\program  
files\netegrity\webagent\certs\sampleRobm.cer" passphrase
```

- Add an issuer certificate:

UNIX:

```
smkeytool.sh -ac /opt/netegrity/webagent/certs/sampleCARoot.cer
```

Windows:

```
smkeytool.bat -ac "c:\program files\netegrity\webagent\certs\sampleCARoot.cer"
```

More information:

[The smkeydatabase.properties File](#) (see page 416)

Chapter 18: Variables

This section contains the following topics:

[eTelligent Rules](#) (see page 421)

[Variables Overview](#) (see page 425)

[Web Service Variables](#) (see page 430)

[Create a Variable](#) (see page 433)

eTelligent Rules

eTelligent Rules let you to define variables. These variables enable a SiteMinder administrator to define fine-grained access-control criteria known as policy expressions. These policy expressions:

- Are SiteMinder policy attributes
- Include operators and customer-defined variables that are evaluated at runtime, when a user actually needs to access a protected resource on a web site

Variables can use local information, stored within the enterprise, as well as remote information that is provided by various Web Services.

You define variable objects and incorporate them into policy logic through policy expressions. You can also include variables in SiteMinder response objects.

SiteMinder eTelligent Rules Benefits

- Reduce complexity and eliminate the need for custom code—Authorization access is defined by the SiteMinder administrator in policy expressions, using graphical tools rather than application code. There is no need to integrate and reconcile backend business applications' access control information, because that information is centralized in the SiteMinder Policy Server.
- Use business data dynamically in security policies—Defining access control to secure resources is based on local user information and incoming information, such as the amount of a purchase order placed by the user.
- Combine various types of information for authorization decisions—Web browser forms data, user-context data (stored locally in the Policy Server), and remote data (obtained through a service bureau) can be flexibly combined in policy expressions.
- Make transactional decisions online—There is no need to go back to a backend business application each time authorization is needed to access a protected resource.

- Rely on XML-based third-party security data—eTelligent Rules use a standard XML protocol to communicate with trusted service bureaus, thus increasing the choice of web services providers.
- Use Boolean logic—Policy expressions are defined by SiteMinder security administrators, using variables together with logical operators.
- Minimize the number of policies required—Due to the use of policy expressions based on logic, fewer policies are necessary, thus keeping policy administration to a minimum.

CA Directory Variable Requirements

If you are using CA Directory as a policy store, open the .dxc file for the policy store DSA and define the following flag at the bottom of the file:

ignore-name-bindings = true

This flag is required to add add variables to the Policy Server domain objects.

eTelligent Rules Configuration

There are two aspects of eTelligent Rules configuration:

- Configuring variables and policy expressions
You configure variables and policy expressions using the Policy Server User Interface.
- Modifying properties files.

More information:

[Variables Overview](#) (see page 425)

[Policies](#) (see page 375)

[eTelligent Rules Properties Files](#) (see page 422)

eTelligent Rules Properties Files

The following properties files are for eTelligent Rules:

- JVMOptions.txt—required for eTelligent Rules. The installed location of this file is:
<policy_server_home>/config/
- LoggerConfig.properties—required to configure logging for eTelligent Rules. The installed location of this file is:
<policy_server_home>/config/properties

More information:

[JVMOptions.txt File](#) (see page 423)

[Modify the LoggerConfig.properties File](#) (see page 424)

JVMOptions.txt File

The JVMOptions.txt file contains the settings that the Policy Server uses when creating the Java Virtual Machine that is used to support eTelligent Rules.

If you encounter errors related to missing classes, you may need to modify the classpath directive in the JVMOptions.txt file. For complete information about the settings contained in the JVMOptions.txt file, see your Java documentation.

Note: You should not need to modify this file.

The following shows an example of this file.

```
-server
-Xbootclasspath/p:C:/Program Files/Netegrity/siteminder/bin/thirdparty/xerces.jar
-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBuilderFactoryImpl
-Djava.class.path=C:/Program
Files/Netegrity/siteminder/config/properties;C:/Program
Files/Netegrity/siteminder/bin/jars/SmJavaApi.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/SmActiveExpr.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/variableapi.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/util.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/smkeydatabase.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/jcel_2_1.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/js.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/xerces.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/xalan.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/certj.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/certjWithNative.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/jsafe.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/jsafeWithNative.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/sslj.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/codebase.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/asn1.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/soap.jar;C:/Program
Files/Netegrity/siteminder/bin/thirdparty/javax.jar;C:/Program
Files/Java/j2re1.4.1_02/lib/sunrsasign.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/affwebservices.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/assertiongenerator.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/saml.jar;C:/Program
Files/Netegrity/siteminder/bin/jars/smauthsaml.jar
```

Modify the LoggerConfig.properties File

On the Policy Server, the LoggerConfig.properties file allows you to specify logging features that are used when you start the SiteMinder service from a command line. The properties contained in this file are not used when the service is started from the Policy Server Management Console. The settings in this file are generally only used for debugging purposes.

You may want to modify this file to obtain more output for debugging purposes.

The following shows an example of a LoggerConfig.properties file.

```
// LoggingOn can be Y, N
LoggingOn=Y

// LogLevel can be one of LOG_LEVEL_NONE, LOG_LEVEL_ERROR,
LOG_LEVEL_INFO, LOG_LEVEL_TRACE
LogLevel=LOG_LEVEL_TRACE

// If LogFileName is set Log output will go to the file named
LogFileName=affwebserv.log

// AppendLog can be Y, N. Y means append output to LogFileName if
specified
AppendLog=Y

// AlwaysWriteToSystemStreams can be Y, N.
// Y means log messages are written to System.out
// or System.err regardless of what the logger streams are
// set to. If the logger streams are set to System.out
// or System.err log messages will be written multiple times.
// This facilitates logging messages to System.out/System.err
// and a file simultaneously.
AlwaysWriteToSystemStreams=N

// DateFormatPattern can be any valid input to java.text.DateFormat
constructor.
// See the Java documentation for java.text.DateFormat for details
// If not specified, the default format for the default locale is used
DateFormatPattern=MMM d, yyyy h:mm:ss.S a
```

The settings in this file are:

LoggingOn

Enables or disables logging. Set this parameter to Y to enable logging. Set this parameter to N to disable logging.

LogLevel

Indicates the level of detail contained in logs. The LogLevel can be one of the following:

LOG_LEVEL_NONE

No messages will be logged.

LOG_LEVEL_ERROR

Only records error messages.

LOG_LEVEL_INFO

Records error messages and warnings.

LOG_LEVEL_TRACE

Records error messages, warnings, and general processing information that may be useful for tracking problems.

LogFileName

If LogFileName is set, all log output will go to the file named in this parameter.

AppendLog

Indicates whether log information should be appended to an existing file at startup or a new file should be created at startup. Set this parameter to Y to append output to the file specified in the LogFileName parameter. Set this file to N if a new file should be created at startup.

AlwaysWriteToSystemStreams

Set this parameter to Y to log messages to System.out or System.err regardless of what the logger streams are set to. If the logger streams are set to System.out or System.err, log messages will be written multiple times. This facilitates logging messages to System.out/System.err and a file simultaneously.

DateFormatPattern

DateFormatPattern can be any valid input to java.text.DateFormat constructor. See the Java documentation for java.text.DateFormat for details.

If not specified, the default format for the default locale is used.

Variables Overview

In the context of Policy Server, variables are objects that can be resolved to a value which you can incorporate into the authorization phase of a request. The value of a variable object is the result of dynamic data and is evaluated at runtime. Variables provide a flexible tool for expanding the capabilities of policies and responses.

Variable Types

The following types of variables are available:

- [Static Variables](#) (see page 426)
- [Request Context Variables](#) (see page 426)
- [User Context Variables](#) (see page 426)
- [Form Post Variables](#) (see page 427)
- [Web Services Variables](#) (see page 427)

Static Variables

Static variables consist of a simple name/value pair of a particular type, such as string, boolean, and others. The key benefit of a static variable is to implement good programming practices. Instead of repeating the value of a constant each time it's used in a policy, a static variable provides a single piece of data that can be used throughout multiple policies.

Request Context Variables

Each request processed by SiteMinder establishes a request context. This context identifies:

Action

Type of action specified in the request, such as GET or POST.

Resource

Requested resource, such as /directory_name/.

Server

Full server name specified in the request, such as server.company.com.

A request context variable may capture any of this information and make it available for inclusion in a policy expression or response. The key benefit of this type of variable is to provide fine-grained request context information without any programming logic.

User Context Variables

When the Policy Server authenticates a user against an entry in a directory, a user context is created. The user context consists of information about the user directory and the contents of the directory that pertain to the authenticated user.

User context variables can be based on an attribute of a directory connection, or based on the contents of the directory. The key benefit of this type of variable is to provide flexibility in defining rules based on particular user context without any programming logic.

Form Post Variables

HTML forms are often used to collect information required by back-end applications. Form Post variables can be used to capture any information entered in an HTML form and POSTed. For example, if the business logic associated with an application requires a purchase order amount specified on a HTML form used for logging into the application, you can create a Form Post variable object that collects the value of the purchase order supplied by a user. The variable can then be used in policies.

Note: Form Post variables are not supported by EJB or Servlet Agents. Do not use Form Post variables in policies enforced by EJB or Servlet Agents.

The key benefit of this type of variable is that it allows the Policy Server to use POST data as a part of a policy expression rather than forcing enterprises to build security logic into back end server applications. Using HTTP POST variables results in efficient network usage between Agents and Policy Servers. The Agent only needs to extract the HTTP variable information from the HTTP stream so that the information can be used during authorization processing by the Policy Server.

Web Services Variables

Web Services variables can be used to capture information retrieved from a Web Service for use in policies or responses. The key benefit of this type of variable is to allow a Policy Server administrator to define a policy based on the dynamic customer information provided in real time by a Web Service.

More information:

[Web Service Variables](#) (see page 430)

Variable Use in Policies

Variables allow you to include business logic in policies by capturing a wide range of dynamic data that can be built into policy expressions. When you define variable objects in the Policy Server User Interface, you may use those variables in expressions in the Policy dialog on the Expression tab. You can build expressions that use multiple variable objects and boolean operators to capture very complex business logic in your policies.

For example, a policy may contain an expression that requires the value of a user's account type and a credit score in order to allow access to an application. An expression can be defined in the policy so that only users whose account type is "gold", and whose credit score is greater than a specific value may have access to a resource. This example requires two variables, which must be combined in an expression on the Expression tab of the Policy dialog.

More information:

[Expressions in Policies](#) (see page 413)

Variable Use in Responses

Variables may be used in responses. When you define variable objects in the Policy Server User Interface, you can use those variables in responses. The value of the response is created at runtime by the Policy Server as it resolves the value of a variable object.

How the Policy Server Processes Variables

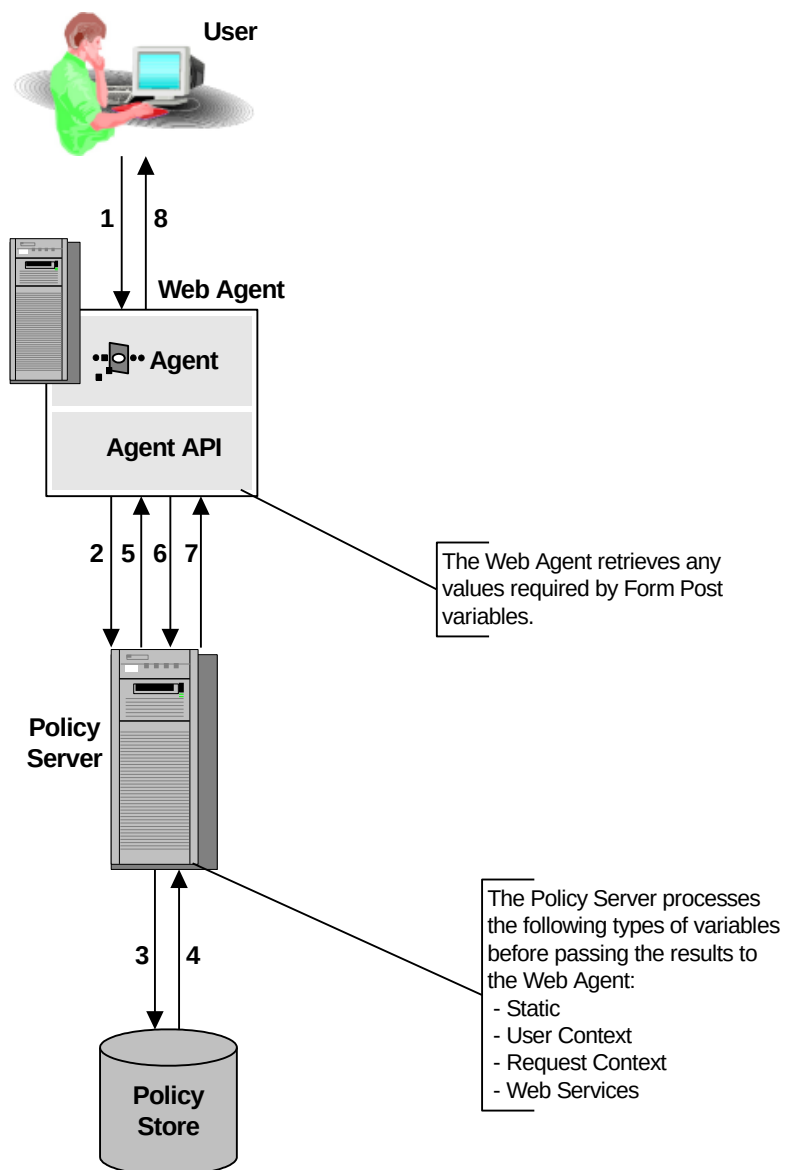
Variables are evaluated when the Policy Server processes an authorization for a request and determines that a user is authorized for the requested resource. The details of variable processing are slightly different based on whether the variable objects are contained in a response or in a policy expression.

How the Policy Server Processes Variables Contained in Policy Expressions

As part of policy evaluation, the variables contained in a policy expression are placed in an unresolved variable list during the authorization of a request. As the Policy Server resolves variables, they are moved to a resolved variables list. When all variables in a policy expression have been resolved, the Policy Server grants or denies access based on the entire policy.

The following figure illustrates how the authorization of a user's request is processed by a Web Agent and the Policy Server when the policy for the requested resource contains variables. This diagram does not include Web Service variables.

Note: The process in the following figure assumes that the user has already been authenticated by SiteMinder. For unauthenticated users, the authentication process must occur before the authorization.



1. The user requests a resource from a server that is protected by a SiteMinder Web Agent.
2. The Agent verifies that the resource is protected and the Policy Server begins authorization processing.
3. The Policy Server retrieves policy information from the Policy Store about the requested resource.

4. The Policy Server receives a list of unresolved variables contained in the policy expression associated with the requested resource. The Policy Server evaluates static, user context, and request context variables contained in the unresolved variables list.
5. If all variables and variable expressions have been resolved, the Policy Server indicates to the Web Agent whether or not the user may access the requested resource.
6. If the unresolved variables list still contains unresolved variables, the list is passed to the Agent API layer with a Not Resolved indicator. The values of any Form Post variables are resolved by the Web Agent and passed to the Policy Server in a new request that includes the Form Post variable value in the resolved variables list.
7. If the policy contained Form Post variables, the Policy Server processes the policy with the newly resolved values extracted from the POST data.
8. The user is either allowed or denied access to the requested resource.

More information:

[Web Service Variables](#) (see page 430)

How the Policy Server Processes Variables contained in Responses

SiteMinder processes variables contained in responses as described in the previous section. Since Form Post variables cannot be used for responses, all variables are resolved by the time a response fires at the Policy Server.

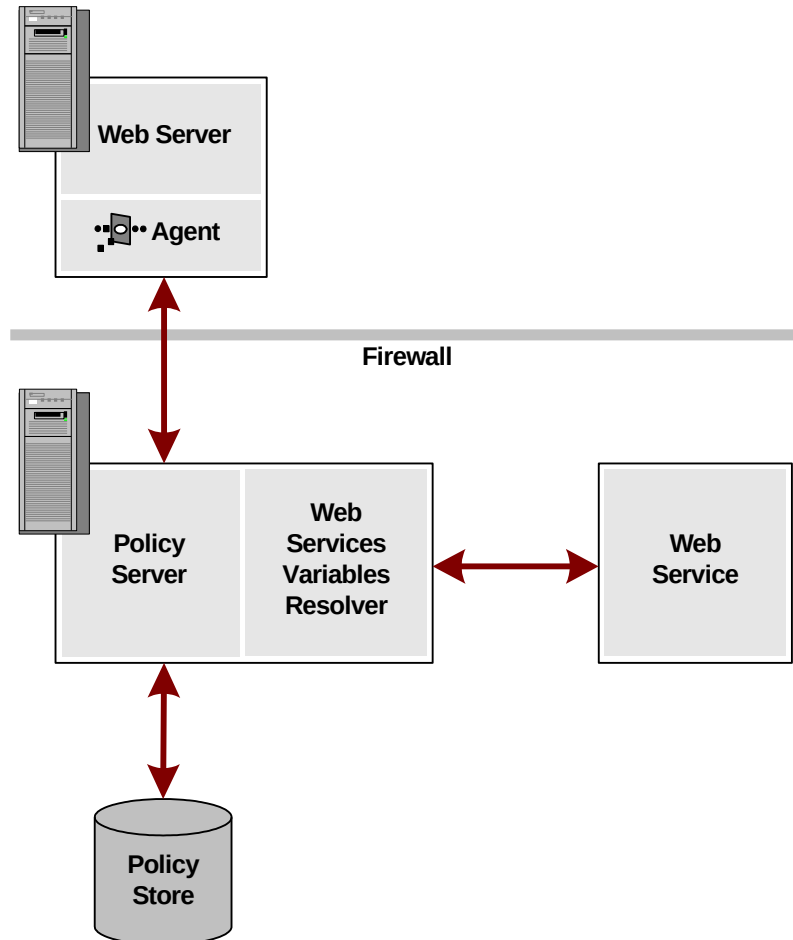
Web Service Variables

Web Service Variables provide a method for including dynamic data retrieved from a Web Service in a Policy Server policy. Web Service Variables are resolved by calling a Web Service. The Policy Server sends a SOAP request document, as specified in the Web Service Variable's definition, and receives a SOAP response document as a reply. The Policy Server extracts the value of the Web Services Variable from the SOAP response document.

The Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol that consists of three parts:

- An envelope that defines a framework for describing what is in a message and how to process it
- A set of encoding rules for expressing instances of application-defined data types
- A convention for representing remote procedure calls and responses.

The following figure shows how a SiteMinder deployment resolves a Web Services Variable for a Web Service inside an intranet, and thus on the same side of the firewall as the Policy Server.



In this scenario, if a Web Service variable is associated with an authorization request, it is resolved on the Policy Server side by calling the Web Service Variables Resolver, which runs in the same process space.

The user, when defining the Web Service variable, specifies the SOAP document to be sent to the Web Service, along with the authentication credentials and other parameters as defined in the rest of this document.

The resolver sends the specified SOAP document to the Web Service, extracts the value of the variable from the response and forwards it to the Policy Server to complete the authorization request.

Even if there is a firewall between the Policy Server and the Web Service, it can be configured to allow communication between the two. The Policy Server issues the request and reads the response, so the firewall is only required to allow outbound requests from the Policy Server to the Web Service.

A secure SSL connection can be configured between the Policy Server and the Web Service to allow for the inbound responses to come from the Web Service to the Policy Server, using the server-side certificates on the Web Service and a list of trusted CAs configured on the Policy Server side.

Security Requirements When Resolving Web Services Variables

Security for Web Services Variables requires an SSL connection between the Policy Server and the Web Service. You can also include a WS-Security header with a username token that the Web Service has been configured to recognize. WS-Security is a standard set of SOAP extensions that provides security token propagation, message integrity and confidentiality through signing and encryption.

For a secure resolution of a Web Services Variable:

- The Policy Server must make a SOAP request to a Web Service on behalf of a known identity (a Web Service account). This is similar to the model used by SiteMinder to access attributes in a User Directory.
- The WS-Security header containing the Web Services credentials can be configured to include a base-64-encoded SHA-1 digest of the password.
- A Web Service variable can be configured to use an SSL connection with a server-side certificate. This means that the Policy Server needs to be configured with the list of trusted CAs.

Note: For SSL connections, server-side certificates should be configured for the Web Service. A list of trusted CAs should be configured on the Policy Server. To configure trusted CAs, use the `smkeydatabase` tool described in [Certificate Authorities and Web Services Variables](#) (see page 415).

Configure the Web Service Variable Resolver

In order for the Policy Server to resolve a Web Service variable, you must configure the Web Service Variable Resolver to properly connect to the Web Service. The connection to the Web Service will fall into one of two categories:

- Both the Policy Server and the Web Service are on the same side of a firewall. There is no firewall between the two.
- There is a firewall between the Policy Server and the Web Service, but it is configured to allow one-way communication between the two (outbound requests from the Policy Server to the Web Service).

Before being able to use the Web Service Variables functionality, the Policy Server must be configured with a list of trusted CAs, using the SmKeyTool command line utility. If several Policy Servers are used in a load balancing or failover configuration, each of them must be configured with the same list of trusted CAs.

Default configuration settings are provided in the WebServiceConfig.properties file in the SiteMinder/Config/properties directory, and can be modified by the user.

Sample WebServiceConfig.properties Configuration File

```
# Netegrity Web Service Variable Resolver properties configuration file:
# This file must be in the classpath that is used when the policy server runs.
# ResolutionTimeout is the amount of time the resolver will at most wait to resolve
all Web Service variables related to a given request.
#
# This setting is intended to end sessions that are waiting on a web service that is
not responding. The time that the Web Agent will typically wait before responding is
typically 60 sec (but may be changed # in the future), which means this setting should
be 60000 or greater to cancel transactions that cannot be returned.
ResolutionTimeout=75000
# MaxThreadCount is the maximal number of active threads running within the Web Service
variables resolver.
MaxThreadCount=10
```

Create a Variable

In order to make a variable object available for use in policies or responses, you must define the variable in the Policy Server User Interface. Variables are domain objects. They must be created within a specific policy domain, or imported into a domain using the smobjimport tool. For information about the import tool, see the *Policy Server Management* guide.

Note: The value of a resolved variable of any type can be no larger than 1K.

For information about importing objects into policy domains, see the *Policy Server Management* guide.

To create a variable

1. Open the Policy Server User Interface.
2. If you have the Manage System and Domain Objects privilege:
 - a. In the Object pane, click on the Domains tab.
 - b. Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

3. If you have the Manage Domain Objects privilege:

In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

4. Click on the + symbol to the left of the Variables to expand the list of variable types.

Note: If you have purchased TransactionMinder, additional variable types are available.

5. Select the type of variable you want to create.
6. From the menu bar, select Edit, Create Variable.

The corresponding Variable Editor dialog opens. For more information, see:

- [Static Variable Editor Dialog](#) (see page 717) in the *Policy Design* guide
- [Request Context Variable Editor Dialog](#) (see page 719) in the *Policy Design* guide
- [User Context Variable Editor Dialog](#) (see page 721) in the *Policy Design* guide
- [POST Variable Editor Dialog](#) (see page 725) in the *Policy Design* guide
- [Web Services Variable Editor Dialog](#) (see page 727) in the *Policy Design* guide

If you have purchased TransactionMinder, also see:

- [TransactionMinder-related User Interface Elements](#) (see page 919) in the *Policy Design* guide

7. Click OK to save the variable.

Note: The value of a resolved variable of any type may be no larger than 1K.

Your variables can then be used in policy expressions or in responses:

- See [Expressions in Policies](#) (see page 413)
- See [Use Variable Objects in Responses](#) (see page 364)

More information:

[Domains](#) (see page 307)

Chapter 19: Global Policies, Rules, and Responses

This section contains the following topics:

[Global Policies](#) (see page 435)

[Configure Global Policies](#) (see page 440)

Global Policies

Standard SiteMinder policies are created in the context of a single policy domain. However, large production environments may contain thousands of domains. In this type of environment it can be useful to define types of behavior (represented by policies) that are common for many domains. Using standard policies, the same policy must be recreated for each domain that requires the same behavior. Global policies allow you to configure policies (and their associated rules and responses) as system level objects, that are applied across all domains.

The following terms are used for discussing global policies:

Access Rule

An access rule allows or denies access to a resource. Global policies do not include access rules. Only event rules may be added to global policies.

Event Rule

An event rule is invoked when an authentication or authorization event occurs. Behaviors that are commonly implemented across all domains are associated with event rules, and may be included in global policies.

Global Policy

A policy which is defined as a system object.

Global Rule

A rule which is defined as a system object.

Global Response

A response which is defined as a system object.

Policy Link

A logical entity used for policy definition. It consists of a rule- response pair. A policy may contain one or more policy links.

More information:

[Policies](#) (see page 375)

[Allow/Deny and Enable/Disable Group Box](#) (see page 697)

[Authentication Events](#) (see page 331)

[Authorization Events](#) (see page 333)

Global Policy Object Characteristics

The following sections discuss the characteristics of global policy objects, outlining the basic similarities and differences when compared to their standard (non-global) counterparts.

Global response vs. standard response

Differences:

- Defined at the system level. Only system level administrators can define a global response.
- Cannot use variables-based attributes.
- Cannot be added to response groups. There are no global response groups.
- Associated with the specific agent type.

Similarities:

- Can use active expressions.
- Is not returned unless it is specified in a particular policy.

Global rule vs. standard rule

Differences:

- Defined at the system level. Only system level administrators can define a global rule.
- The filter for the global rule is not bound to a specific realm. The filter for the global rule is an absolute filter (that may or may not use a regular expression).
- Bound to specific agent or agent group. The agent is explicitly specified when the rule is created.
- Available only for SiteMinder (not RADIUS) agents.
- Only defined for an authentication or authorization event.
- Only used in global policies.
- Cannot be associated with Agents that do not support authentication and authorization events. For example, you cannot associate a global rule with a RADIUS Agent.
- Cannot be added to rule groups. There are no global rule groups.

Similarities:

- Can be a member of a global rule group. Global rules cannot participate in domain-specific rule groups.

Global policy vs. standard policy

Differences:

- Defined at the system level. Only system level administrators can define a global policy.
- Bound to all users. Specific users cannot be included or excluded from a global policy.
Note: Individual domains can be explicitly enabled or disabled for global policy processing.
- Defined using only global rules, global responses and groups of these global objects.
- Cannot use variable-based attributes or variable expressions.
- Cannot contain allow/deny access rules. Only event rules may be included in a global policy.
- Cannot include or exclude a particular resource/realm from the global policy. The global policy is applicable to all resources that match at least one of the rule filters defined for the policy, on the domain for which global policy processing is enabled.
- Are not supported through the Java Policy Management API.

Similarities:

- Can use active expressions.
- Associated with the specific Agent. However, it's possible to create a group containing all the Agents of the same type and bind a global rule to such group.
- When the global policy is being processed, the responses defined for the fired global rules, are added to the list of other responses. A global rule fires when the following is true:
 - The resource being accessed matches the absolute resource filter defined for the global rule.
 - The event that occurs is as defined for the global rule.
 - The resource being requested is protected by the same agent/agent group, which was specified for the rule.
 - The resource/realm being accessed belongs to a domain for which global policies processing is enabled.

More information:

[Disable Global Policy Processing for a Domain](#) (see page 310)

SiteMinder Global Policy Concept

SiteMinder uses a policy-based access control model. A SiteMinder policy defines the type of access a user has to a particular resource and what happens when the user accesses the resource. Each standard SiteMinder policy is a linkage between a set of users and a set of resources, and is designed to protect resources by binding together users, rules and responses. Every policy must specify the users or groups of users to which the policy applies. Users can be either included or excluded from the policy.

In addition, a standard policy must contain at least one rule or rule group. Rules are the parts of a policy that determine precisely which resources are protected and what type of action should cause a rule to fire. A rule identifies a resource or resources that are included in the policy using a combination of a string-based *resource filter* and *action*. The filter in turn consists of *realm filter* and *rule filter*. For information about realms, rules, and responses in standard SiteMinder policies, see the following:

- [Grouping Resources in Realms](#) (see page 315)
- [Rule Groups](#) (see page 349)
- [Rules](#) (see page 327)
- [Responses and Response Groups](#) (see page 353)
- [Policies](#) (see page 375)

SiteMinder objects can be of two types: system level and domain level. In a standard (non-global) SiteMinder policy, all policy objects must be created in the context of a specific domain. However, global policies are system level policies that may be applied across all domains in a SiteMinder deployment. An administrator with system level privileges can define global policies, that include global rules and global responses. These global policies may be applied to any resource in any domain.

Global objects are similar to their standard, domain-specific counterparts. The roles of global objects in a global policy definition are different from domain-specific policy objects in the way they are created and linked to form policies. However, there are no global domain or global realm objects.

The following summarizes the differences between domain-specific and global policy objects.

Policy Object	Domain Specific	Global
All	Created by domain administrators in the context of the specific domain.	Created by the system administrator at the system level.
Response	Can use variables-based attributes.	Cannot use variables-based attributes.
	Can be used only in domain specific policy.	Can be used in any global or domain-specific policy.
	Can be a member of a domain specific response group only.	Can be a member of any global or domain specific response group.
Rule	The resource filter is bound to a specific realm (realm filter + rule filter).	The resource filter is absolute (not bound to a specific realm).
	Associated with an agent through the realm.	Associated with specific agent or agent group. Agent is explicitly specified when defining a rule
	Can be defined as access rule or event rule.	Can be defined for authentication and authorization events only (event rule).
	Can be used in domain specific policies only.	Can be used in global policies only.
	Can be a member of a domain specific rule group only.	Can be a member of a global rule group only.
	Fires only for resources defined within a specific domain.	Can fire for resources on any domain for which global policy processing is enable.
Policy	Bound to specific users or groups of users.	Bound to all the users.
	Users can be included or excluded from the policy.	User cannot be included or excluded from the policy.
	Can be defined using domain-specific rules/rule groups, domain-specific responses/response groups and global responses/response groups.	Can be defined only using global rules, global responses and groups of these objects.
	Can use variable expressions	Cannot use variable expressions

Global Policy Processing

Policies are evaluated as described in [Policy Processing](#) (see page 405). In addition, any global rules contained in global policies will fire if the following conditions are met:

- The requested resource belongs to a domain on which global policy processing was not disabled
- The requested resource matches the absolute resource filter defined for the global rule. It is important, that in the case of global rule the filter will be obtained from the rule (not from the realm).
- The event that occurs is the same as that defined for the global rule.
- The requested resource is protected by the same agent or agent group, which was specified for the rule.

Whenever an authentication or an authorization event happens the responses defined for the fired global rules are added to the list of other responses.

Configure Global Policies

Important! You can configure both global policies and domain-specific policies that affect the same resources. For example, if you can configure domain-specific policies for access control, and global policies that provide a standard set of responses. However, in order for global policies to function, the realms included in the domain-specific policies be configured to allow event processing.

For administrators with system-level privilege, Manage System and Domain Objects, the tab pane contains a tab for Global Policies. This tab allows an administrator to configure the global objects required for Global Policies. The process for creating a global policy requires the following steps:

1. [Global Rule Objects](#) (see page 440)
2. [Global Response Objects](#) (see page 445)
3. [Configure Global Policy Objects](#) (see page 449)

More information:

[Configure a Realm](#) (see page 320)

Global Rule Objects

The part of a global policy that defines a resource and event that will trigger the processing of a global policy is a global rule object. Global rules are similar to domain-specific rules. However, a global rule must be associated with an authentication or authorization event. There are no global allow/deny access rules.

More information:

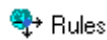
[Rules](#) (see page 327)

Enable and Disable Global Rules

The Policy Server User Interface allows you to enable and disable global rules. If a global rule is enabled, and a user accesses the resource specified by the rule, the rule fires when the authentication or authorization event specified in the rule occurs. If a global rule is disabled, the rule does not fire when a user triggers the authentication or authorization event.

To enable or disable a global rule

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the Global Policies tab.
3. Click on the Global Rules icon.



4. In the List pane, select the global rule you want to disable or enable.
5. From the menu bar, select Edit, Properties of Rule.

The Global Rule Properties dialog opens.

6. In the Allow/Deny and Enable/Disable group box, select the Enabled check box. Selecting this check box toggles the check mark. If the box contains a check mark, the rule is enabled. If it does not, the rule is disabled.
7. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Start the Policy Server User Interface](#) (see page 64)

Global Rules for Authentication Event Actions

Global rules that include SiteMinder authentication events allow you to control actions that occur when users authenticate to gain access to a resource (On-Auth event).

The following is a list of possible On-Auth events:

On-Auth-Accept

Occurs if authentication was successful. This event may be used to redirect a user after a successful authentication.

On-Auth-Reject

Occurs if authentication failed for a user that is bound to a policy containing an On-Auth-Reject rule. This event may be used to redirect the user after a failed authentication.

OnAuthAccept and OnAuthReject events fire both at authentication time (when the user enters his / her username and password) and at validation time (when the user's cookie is read for user information). However, there are certain special actions that only occur at authentication time:

Realm timeout override (unless EnforceRealmTimeouts is used).

Unless you have a version of the Web Agent that supports the EnforceRealmTimeouts option and that option is enabled, the Idle and Max Timeouts for the user will stay at the values for the realm in which the user last authenticated (only changes if the user has to reenter credentials). See section 3.3 of the *SiteMinder 4.x Web and Affiliate Agent Quarterly Maintenance Release 4 Release Notes* for more information on EnforceRealmTimeouts.

Redirects.

Redirects are only allowed at authentication time for a number of reasons, but one of the most practical is that it would be very easy to configure an infinite loop of redirection if OnAuth redirection were allowed at validation time as well.

Access to the user's password.

The password is not stored in the SMSESSION cookie, so the only time it is available is when the user actually enters it (authentication time).

Note: OnAuth event results are per realm, so for example, if a user goes from realm A to realm B and had an OnAuthAccept header in realm A, it will not be available in realm B. When the user goes back to realm A, the header will be set again.

On-Auth-Attempt

Occurs if the user was rejected because SiteMinder does not know this user (an unregistered user, for example, can be redirected to register first).

On-Auth-Challenge

Occurs when custom challenge-response authentication schemes are activated (for example, a token code).

When a user is authenticated (or rejected), the Policy Server passes any global responses associated with the applicable On-Auth rule back to the requesting Agent.

More information:

[Global Response Objects](#) (see page 445)

Create a Global Rule for Authorization Event Actions

Global rules that include SiteMinder authorization events allow SiteMinder to call responses based on whether a user is or is not authorized for the resource the user requested. Authorization events occur after a user is authenticated, if a rule that protects a resource contains an On-Access event. When the user has been granted or denied access based on their privileges, the appropriate event is triggered.

The following is a list of possible On-Access events:

On-Access-Accept

Occurs as the result of successful authorization. This event may be used to redirect users who are authorized to access a resource.

On-Access-Reject

Occurs as the result of failed authorization. This event may be used to redirect users who are not authorized to access a resource.

When a user is authorized (or rejected), the Policy Server passes any responses associated with the applicable On-Access rule back to the requesting Agent.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a global rule for authorization event actions

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the Global Policies tab.
3. Click on the Global Rules icon.



4. From the Edit menu, select Create Rule.
The Global Rule dialog opens.
5. In the Name field, enter the name of the new global rule.

6. Optionally, in the Description field, enter a brief description of the new global rule.
7. In the Agent field, enter the name of the Agent or Agent Group for which the global rule should apply.

To search for an Agent or Agent Group name, you can click the Lookup button.

If you specify an Agent Group in a global rule and you have also configured domain-specific rules associated with the same resource, you can adversely affect system performance by effectively duplicating processing steps. When configuring global rules and global policies, be sure to consider domain-specific rules that may duplicate the responses generated by global rules. Note that in such cases, only one response is returned to the Agent, since the Policy Server automatically deletes duplicate responses before passing information back to a requesting Agent.

8. In the Resource Filter field, enter the filter that will determine the resources affected by the global rule.

A resource can be a specific file or an expression that uses resource matching.

9. In the Action group box, select the Authorization Events radio button.
10. In the Action group box, select an OnAccess action from the drop-down list to the right of the radio buttons.
11. Click Apply to save the rule, or click OK to save the rule and return to the SiteMinder Administration window.

More information:

[Responses and Response Groups](#) (see page 353)

[Start the Policy Server User Interface](#) (see page 64)

[Global Rule Dialog](#) (see page 751)

[Resource Matching and Regular Expressions](#) (see page 341)

Policy Considerations When Configuring Global OnAccessReject Rules

When you create global rules that include OnAccessReject events, sometimes referred to as OnAccessReject rules, you must take into account how the Policy Server processes global policies. This section outlines the special circumstances created by OnAccessReject rules.

Configure Time Restrictions for Global Rules

The Advanced group box of the Global Rules Properties dialog contains a Time Restrictions tab. This region contains two buttons and a message that indicates whether or not time restrictions have been applied to the current global rule. The process for configuring time restrictions for global rules is identical to the process for configuring time restrictions for domain-specific rules.

More information:

[Add Time Restrictions for Rules](#) (see page 344)

Configure an Active Rule as part of a Global Rule

An Active Rule is used for dynamic authorization based on external business logic. The Policy Server invokes a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

The process for configuring active rules for global rules is identical to the process for configuring active rules for domain-specific rules.

More information:

[Configure an Active Rule](#) (see page 346)

Global Rule Deletion

If you delete a global rule, the rule is automatically removed from any global policies that include the global rule. The global policies remain on your system. You should verify that the global policies function without the deleted rule.

Note: Global policies must contain at least one global rule.

Global Response Objects

The part of a global policy that defines the attributes to be returned after the authentication or authorization event in an associated global rule is the global response.

Note: You may use global responses in domain policies. In order to be returned, a global response *must* be added to a domain-specific or global policy. Within policies, the global response will be processed like a domain-specific response.

Configure a Global Response

When you configure a global response, you must create the response in the Global Response Properties dialog, then add response attributes to the response.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a global response

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the Global Policies tab.
3. Click on the Global Responses icon.
 Responses
4. From the Edit menu, select Create Response.
The Global Response dialog opens.
5. In the Name field, enter the name of the new global response.
6. Optionally, in the Description field, enter a description for the new global response.
7. Select a SiteMinder Agent type from the drop-down list in the Agent Type group box.
Your choices are Web Agent, Affiliate Agent, EJB Agent, or Servlet Agent. Select the Agent type to which the response must pass information.
8. Click Apply to save your changes.
You can now add response attributes to your response. See the next section for details about adding response attributes for each Agent type.

More information:

[Start the Policy Server User Interface](#) (see page 64)
[Global Response Dialog](#) (see page 755)

Response Attributes for Global Responses

Each SiteMinder global response may contain one or more response attributes. Response attributes identify the pieces of information that the Policy Server passes to a SiteMinder Agent. Each SiteMinder Agent type can accept different response attributes.

Global Response Attribute Types

SiteMinder supports different types of response attributes. The types of response attributes determine where the Policy Server finds the proper values for the response attributes. The types of response attributes that you can configure for a global response are identical the types of response attributes you can configure for a domain-specific response.

More information:

[Response Attribute Types](#) (see page 359)

Configure a Global Web Agent Response Attribute

Web Agent responses can have a wide range of response attributes. These attributes include HTTP header variables, cookie variables, redirections to other resources, text, and timeout values. For details about most specific Web Agent response attribute types, see the *SiteMinder Web Agent Guide*.

Note: If you have purchased TransactionMinder, see the *TransactionMinder Operations Guide* for details about the WebAgent-SAML-Session-Ticket-Variable response attribute.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a global response attribute

1. In the Global Response Properties dialog, select the appropriate Agent type from the drop-down list in the Agent Type group box.

2. Click Create.

The SiteMinder Response Attribute Editor dialog opens.

3. From the Attribute drop down list, select the type of response attribute you want to add to the global response. For complete descriptions of response attributes, see the *SiteMinder Web Agent Guide*.
4. In the Attribute Setup tab, select a radio button for the type of response attribute in the Attribute Kind group box.

You can select one of the following radio buttons: Static, User Attribute, DN Attribute, or Active Response.

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return a response with an attribute contained in a group that is nested in a group specified by a policy. Nested groups often occur in complex LDAP deployments.

Your selection from the Attribute drop-down list and the response attribute type radio button you select determine the available fields in the Attribute Fields group box.

5. If you selected the Static radio button, in the Attribute Fields group box, complete the Variable Name and Variable Value fields (for header variables) or the Cookie Name and Cookie Value fields (for cookie variables).

For example, to return the name/value pair show_content=yes to the Web Agent, enter show_content in the Variable Name field, and yes in the Variable Value field.

6. If you selected the User Attribute radio button, in the Attribute Fields group box, complete the Variable Name (or Cookie Name for cookie variables) and Attribute Name fields.

For example, to return the value of a user's email attribute to the Web Agent, enter email_address in the Variable Name field, and email (where email is the name of the attribute in the user directory) in the Attribute Name field.

Note: For a list of automatically generated SiteMinder user attributes that can be used in responses, see [SiteMinder Generated User Attributes](#) (see page 366).

7. If you selected the DN Attribute radio button, in the Attribute Fields group box, complete the Variable Name (or Cookie Name for cookie variables), DN Spec, and Attribute Name fields.

If the response attribute will be retrieved from an LDAP user directory that contains nested groups (groups that contain other groups), and you want the Policy Server to retrieve DN attributes from the nested groups, select the Allow Nested Groups check box in the Attribute Kind group box.

8. If you selected the Active Response radio button, in the Attribute Fields group box, complete the Variable Name field (or Cookie Name for cookie variables), Library Name and Function Name fields. Optionally, complete the Parameters field.
9. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

[Global Response Attribute Types](#) (see page 446)

Advanced Tab

The Advanced tab of the Global Response Attribute dialog contains the Script field. This field initially displays the script that the Policy Server generates based on your entries in the Attribute Setup tab.

You can manually edit the contents of the Script field. For example, you may be constructing an Active Response that returns the name of the attribute. So, instead of the standard:

```
Name=<@lib="MyLib.lib" func="Func" param="Param"@>
```

You may want to remove the Name= so that Script contains:

```
<@lib="MyLib.lib" func="Func" param="Param"@>
```

You can also copy the contents of the Script field and paste the contents into the Script field for another response. However, once you type or paste into the Script field, you cannot return to the Attribute Setup tab.

Important! If you copy the contents of the Script field, when creating the destination attribute, select the appropriate radio button in the Attribute Kind group box of the Attribute Setup *tab* before moving to the Advanced tab. Once you paste into the Script field, you can't return to the Attribute Setup tab.

Configure Global Policy Objects

You can configure Global Policy object using the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure a Global Policy object

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the Global Policies tab.
3. Click on the Global Policies icon.



4. From the Edit menu, select Create Policy.

The Global Policy dialog opens.

5. In the Name field, enter the name of the new policy.
6. Optionally, in the Description field, enter a brief description of the new policy.

More information:

[Start the Policy Server User Interface](#) (see page 64)

[Global Policy Dialog](#) (see page 761)

Add Global Rules and Global Responses to a Global Policy

Global rules indicate the specific resources included in a global policy. Global responses indicate the actions that should take place when the rule fires.

You must add at least one global rule to a global policy.

To add global rules to a global policy

1. In the Global Policy Properties dialog, click on the Rules tab.
2. Click the Add/Remove Rules button.

A list of available rules opens.

3. To add a global rule to the global policy, select a global rule from the Available Members list and click on the Left Arrow button, which points to the Current Members list.

The reverse procedure removes rules from the Current Members list.

You can select multiple rules by holding the CTRL or SHIFT key and clicking on rules in one of the Members lists. When you select multiple rules and click one of the Arrow buttons, all of the selected rules are moved to the other list.

4. Click OK to save your changes and return to the Global Policy Properties dialog.
5. If you want to associate a global response with a global rule, in the Rules tab, select the rule, and click the Set Response button.

The Available Global Responses dialog opens.

This dialog contains a list of the existing global responses.

6. Click on a response and click OK.

The Policy Server associates the global response with the global rule you selected in the Rules tab, and the Available Global Responses dialog closes.

More information:

[Rule Items Dialog](#) (see page 742)

Enable and Disable Global Policies

The Policy Server User Interface allows you to enable and disable global policies. By default, when you create a global policy, the policy is enabled. When a global policy is enabled, global rules contained in the global policy fire when users attempt to access the resources specified in the global rules.

If you disable a global policy, the rules contained in the policy will not fire.

To enable or disable a global policy

1. Log into the Policy Server User Interface.
2. In the Object pane, click on the Global Policies tab.
3. Click on the Global Policies icon.
4. In the List pane, select the policy you want to enable or disable.
5. From the menu bar, select Edit, Properties of Policy.

The Global Policy Properties dialog opens.

6. Select the Enabled check box.

If the check box is marked with a check, the policy is enabled. If the check box is not marked with a check, the policy is disabled. If you disable a policy, the policy does not fire.

7. Click Apply to save your changes, or click OK to save your changes and return to the SiteMinder Administration window.

More information:

[Start the Policy Server User Interface](#) (see page 64)

Specify IP Addresses for Global Policies

You can specify an IP address, range of IP addresses, or a subnet mask that users must use in order for a global policy to fire. For example, if you specify a range of IP addresses, only those users who login in from one of the specified IP addresses will trigger the global policy.

The process for specifying IP address for global policies is identical to the process for specifying IP addresses for domain-specific policies.

To specify IP addresses:

1. In the Global Policy Properties dialog box, click on the IP Addresses tab.
2. Click on the Add... button.
The Add an IP Address dialog box opens.
3. Select one of the radio buttons to indicate whether you want to add a Single Host, Host Name, Subnet Mask, or Range of IP addresses to the global policy.
4. Fill in the appropriate information based on your selection. (Refer to the following sections for details.)
5. Click OK to save IP address information and return to the SiteMinder Policy dialog box.

More information:

[Specify Allowable IP Addresses for Policies](#) (see page 391)

Specify a Single IP Address for a Global Policy

You can specify a single IP address for a global policy. If you specify a single IP address, the global policy only fires for users who access the policy's resources from the specified IP address.

To specify a single IP address for a global policy

1. In the Add an IP Address dialog box, click on the Single Host radio button.
The dialog box changes to display the fields associated with your selection.
2. Enter an IP Address in the fields provided.
If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog box.
Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog box closes and the IP address for the host name you entered appears in the Global Policy Properties dialog box.
3. Click OK to save the IP address and return to the Global Policy Properties dialog box.
The IP address appears in the list of addresses in the Global Policy Properties dialog box.

Add a Host Name for a Global Policy

You can specify a host name for a global policy. If you specify a host name, the policy only fires for users who access the policy's resources from the specified host.

To add a host name to a global policy

1. In the Add an IP Address dialog box, click on the Host Name radio button.
The dialog box changes to display the fields associated with your selection.
2. In the Host Name field, enter the host name.
3. Click OK to save your settings and return to the Global Policy Properties dialog box.
The host name appears in the list of addresses in the Global Policy Properties dialog box.

Add a Subnet Mask for a Global Policy

You can specify a subnet mask for a global policy. If you specify a subnet mask, the global policy only fires for users who access the policy's resources from the specified subnet mask.

To add a subnet mask to a global policy

1. In the Add an IP Address dialog box, click on the Subnet Mask radio button.
The dialog box changes to display the fields associated with your selection.
2. Enter an IP Address in the fields provided.
If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog box.
Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog box closes and the IP address for the host name you entered appears in the Add an IP Address dialog box.
3. Use the Left and Right arrow buttons, or click and drag the slider bar to select a subnet mask.
4. Click OK to save your settings and return to the Global Policy Properties dialog box.
The subnet mask appears in the list of addresses in the Global Policy Properties dialog box.

Add a Range of IP Addresses for a Global Policy

You can specify a range of IP addresses for a global policy. If you specify a range of IP addresses, the global policy only fires for users who access the policy's resources from one of the IP addresses included in the range of addresses.

To add a range of IP addresses to a global policy

1. In the Add an IP Address dialog box, click on the Range radio button.
The dialog box changes to display the fields associated with your selection.
2. Enter a starting IP Address in the From fields.
If you do not know the IP address, but you have a domain name for the address, you can click on the DNS Lookup button to open the DNS Lookup dialog box.
3. Enter a fully qualified host name in the Host Name field and click OK. The DNS Lookup dialog box closes and the IP address for the host name you entered appears in the Add an IP Address dialog box.
4. Enter an ending IP address in the To fields.
5. Click OK to save the IP address and return to the Global Policy Properties dialog box.
The range of IP addresses appears in the list of addresses in the Global Policy Properties dialog box.

Add and Remove Global Policy Time Restrictions

You can add time restrictions to a global policy. When you add a time restriction, the global policy only fires during the period specified in the time restriction. If a user attempts to access a resource associated with the policy outside of the period specified by the time restriction, the global policy does not fire.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed.

The process for configuring time restrictions for global policies is identical to the process for configuring time restrictions for domain-specific policies.

To add a time restriction to a global policy

1. In the Global Policy Properties dialog box, click on the Time tab.
2. In the Time Restrictions group box, click Set.

The Time dialog box opens.

3. When you finish entering time restrictions, click OK to save your changes and return to the SiteMinder Policy dialog box.

More information:

[How the Web Agent and Policy Server Calculate Time](#) (see page 83)

[Add and Remove Time Restrictions for Policies](#) (see page 394)

How Rule and Policy Time Restrictions Interact

If you specify a time restriction for a global policy, and that policy contains a global rule with a time restriction, the policy fires during the times that are an intersection of the two restrictions.

For example, if a policy has a time restriction of 9AM to 5PM, and a rule has a time restriction of Monday through Friday, then the policy only fires between 9AM and 5PM, Monday through Friday.

Configure a Global Active Policy

An active policy is used for dynamic authorization based on external business logic. An active policy is included in the authorization decision by having the Policy Server invoke a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

The process for configuring active policies for global policies is identical to the process for configuring active policies for domain-specific policies.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure an Active Policy

1. In the Global Policy Properties dialog box, click on the Advanced tab.
2. In the Library Name field, enter the name of the shared library that supports the Active Policy.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default path for your system to search for the shared library at run-time. You should not include a file extension for the shared library.

For example, if you are using a Windows NT platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory which is added to your system path when you install the Policy Server, you simply need to enter lib in the Library Name field.

3. In the Function Name field, enter the name of the actual function in the shared library that implements the Active Policy.
4. In the Parameter(s) field, enter the (optional) list of parameters to be passed to the function in the shared library.

As you enter information in the fields, the system displays the generated string in the Generated Script field.

5. Click OK to save the Active Policy and return to the Global Policy Properties dialog box.

More information:

[Configure an Active Policy](#) (see page 395)

Chapter 20: Impersonation

This section contains the following topics:

[Impersonation Overview](#) (see page 457)

[Impersonation Process](#) (see page 458)

[Security Considerations for Impersonation](#) (see page 459)

[Policy Server Objects for Impersonation](#) (see page 461)

[Sample Implementation of Impersonation](#) (see page 473)

Impersonation Overview

Impersonation provides a method for a privileged user to assume the role of another user without ending the privileged user's session. This feature facilitates the following:

- Customer service representatives (CSRs) impersonate customers to investigate access problems.
- Helpdesk representatives impersonate employees to investigate access problems.
- Employees impersonate co-workers who are on vacation or out of the office.
- Any other situation in which one user must temporarily assume the identity of another user.

Impersonation provides a secure solution in the above situations. In all cases, passwords are not disclosed in order to allow one user to impersonate another user.

The following terms will be used when describing impersonation:

Impersonated session

A user session created for the purpose of assuming another user's identity.

Impersonatee

The user whose identity can be assumed by a privileged user.

Impersonator

The privileged user who has the ability to assume the identities of other users.

Impersonation authentication scheme

A method for authenticating a user that allows a privileged user to assume the identity of another user while preserving the identity of the impersonator.

Session

Also known as user session. This is the time between authenticating and logging out.

Session Specification

Also known as the Session Ticket or Session Spec, it is the information held in a proprietary format on the Policy Server that describes a user and the characteristics of the current session.

SMSESSION

The name of the Web Agent cookie that contains the Policy Server's Session Specification.

Note: For information about sessions, see the *SiteMinder Policy Server Management Guide*. For information about Web Agents, see the SiteMinder Web Agent Guide.

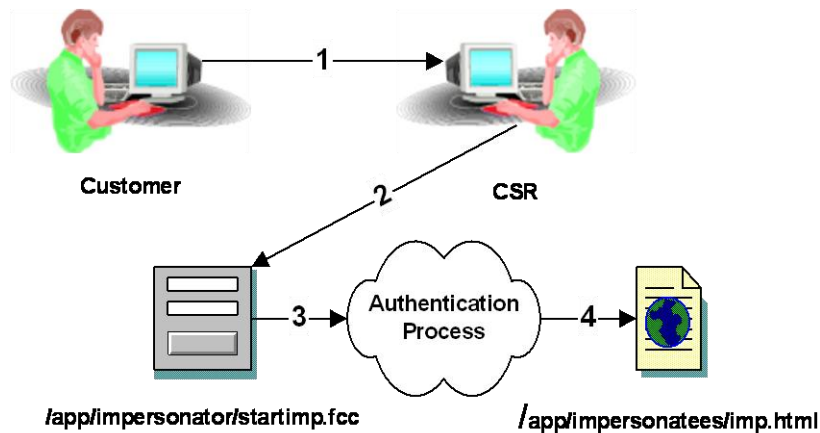
Impersonation Process

The process of impersonating another user consists of the following:

- A privileged user (impersonator) establishes a SiteMinder session by accessing a resource protected by SiteMinder. This can be the Policy Server User Interface or some other application or resource protected by a SiteMinder policy.
- The impersonator, using the currently established identity, accesses a form which allows the impersonator to specify the user to be impersonated (impersonatee). The impersonator does not present the credentials of the impersonatee. The form is an .fcc file with special logic to enable the impersonated session.
- The impersonator submits information about the impersonatee.
- The Policy Server determines whether or not the impersonator can impersonate using a series of policies.
 - Determines whether or not the impersonator has sufficient rights to impersonate the impersonatee.
 - Uses the impersonator's established session when granting the impersonator access as the impersonatee.
 - Audits all impersonator activity.

To begin an impersonated session, an impersonator accesses an .fcc file directly that has a target that points to a resource in the realm protected by the impersonation authentication scheme.

The following diagram shows an example of impersonation where a customer service representative (CSR) accesses an .fcc file directly to begin an impersonation session.



1. A customer with a problem calls his CSR. The CSR determines that the best way to solve the customer's problem is to impersonate the customer.
2. The CSR accesses the impersonation start page which in this case is named `"/app/impersonator/startimp.fcc"` to begin the impersonation process. The Forms Credential Collector (FCC) presents a form to the CSR. The CSR provides the name of the customer to be impersonated, and possibly other attributes for disambiguation. Within the .fcc file are directives (not shown) that collect the session specification of the CSR for use by the authentication scheme. The impersonation authentication scheme is invoked on the Policy Server because the target of the FCC resides in a SiteMinder realm that is protected by the impersonation authentication scheme.
3. The Policy Server takes the information gathered by the FCC and uses it to determine if the customer to be impersonated can be found in any of the configured user directories. If found, the Policy Server determines if the CSR may impersonate the customer, and if the customer may be impersonated. If both are allowed, the Policy Server authenticates the impersonator and the CSR acts as the customer for the duration of the impersonated session.
4. Finally, the Policy Server directs the CSR (impersonating the Customer) to the target of the FCC.

Security Considerations for Impersonation

While impersonating a customer, an impersonator's session specification will look to SiteMinder much like the session specification of any customer. The major difference is that the impersonator's distinguished name and the user directory in which the impersonator originally authenticated will be present as additional fields. This allows all impersonated access to resources to be passed through additional checks. It also allows the Policy Server to record impersonated activities for auditing.

The impersonated session specification is also used to prevent impersonation chaining. When the Policy Server determines that the fields for the impersonator DN and user directory are in use, it will not allow further impersonation and will reject the login attempt. This stops impersonators from stacking one impersonation on top of another to gain access to otherwise restricted resources.

Effects of Authentication Scheme Protection Levels

While impersonating a user, the protection level at which an impersonator originally authenticated will not be checked. Normally, when accessing resources in a new realm protected by an authentication scheme at a higher level, the user would be challenged for new credentials. However, since an impersonator should be a privileged user, these types of challenges will not occur during an impersonation session. Protection levels are meant to indicate the strength of credentials used to access resources in a realm. In the case of impersonation, there are no credentials specific to the user being impersonated, therefore protection levels are not considered.

Note: Once the impersonated session ends, protection levels are once again enforced normally.

Session Idle Timeouts

During an impersonated session, an impersonator's original session can possibly idle out. This depends on the idle timeout value for the realm in which the impersonator originally authenticated. If the impersonator impersonates a user for a longer period than their original idle timeout, the impersonation session ends with the impersonator's original session. To avoid this situation, increase the session idle timeout for realms in which impersonators commonly authenticate.

Restricting Impersonation

For an impersonation to take place, the impersonator and impersonatee must be bound to policies in an impersonation realm. The impersonator policy must have a rule that fires for the `ImpersonateStart` event and the impersonatee policy must fire for the `ImpersoanteStartUser` event. The processes for configuring these objects are described in [Policy Server Objects for Impersonation](#) (see page 461).

Impersonation Realms and Events

Impersonation originates in realm specifically configured to begin the Impersonation process. When an impersonator requests a resource in the impersonation realm, the impersonator is challenged by an impersonation authentication scheme. The scheme prompts the impersonator for credentials via a form. Instead of prompting for a username and a password as is usual, the form prompts the impersonator to supply the username of an impersonatee. Within the .fcc file for the form, there is logic that sets the password to the impersonator's current Session Specification.

The Policy Server uses the impersonation authentication scheme to verify that the impersonator's current session is valid, and then the Policy Server attempts to find the user to be impersonated in the directories included in the policy domain associated with the impersonation realm. If the Policy Server finds the user, authentication proceeds.

Once the Policy Server locates the impersonatee in a directory, it must verify that the impersonator has the right to impersonate, and that the potential impersonatee can be impersonated. These rights are configured and enforced using SiteMinder Policies and Rules, as well as two impersonation events.

Impersonation events are similar to authentication and authorization events, but are not passively invoked. Policies must specifically bind both the impersonator and the impersonatee to Impersonator event rules. If policies do not exist, or do not include impersonator event rules for the impersonator and for the impersonatee, impersonation will not be allowed.

Impersonation can be allowed or disallowed in any Realm using SiteMinder policies and rules with impersonation events. Realms can be configured to disallow all impersonation, or to restrict possible impersonators and impersonatees.

Policy Server Objects for Impersonation

In order to implement impersonation in an enterprise, a number of Policy Server objects must be configured. The combination of objects provides the authentication and policy entitlements that are required to enable one user to impersonate another. The following objects are required for impersonation:

System Objects:

Agent

Impersonation requires a Web Agent and its associated Policy Server Agent object.

Note: To implement impersonation, you must have at least one SiteMinder Web Agent installed in your deployment. For information, see the *Web Agent Installation Guide*.

Authentication Scheme

Impersonation requires an authentication scheme object based on the Impersonation Template.

User Directory

Impersonation requires one or more user directory objects that point to user stores which contain impersonators and impersonatees. The two populations of users should be distinguishable due to an attribute value or group membership.

Domain

Impersonation requires a policy domain object that includes the user directory object(s) mentioned in the previous bullet.

Domain Objects:

Realms

Impersonation requires a minimum of two configured realm objects. One realm contains the resources accessible by the impersonatees. The other realm is the impersonation realm, and contains the resources and rules required to initialize an impersonation session.

Rules

Impersonation requires access control rules to be in place. In addition, a rule with the ImpersonateStart event must exist for impersonators to begin an impersonation session. A rule with the ImpersonateStartUser event must exist in to allow a user to be impersonated.

Policies

Besides the policies that must be in place to protect a set of resources, impersonation requires additional policies to allow access to resources in the impersonation realm, to qualify users as impersonators, and to limit the set of impersonatees.

More information:

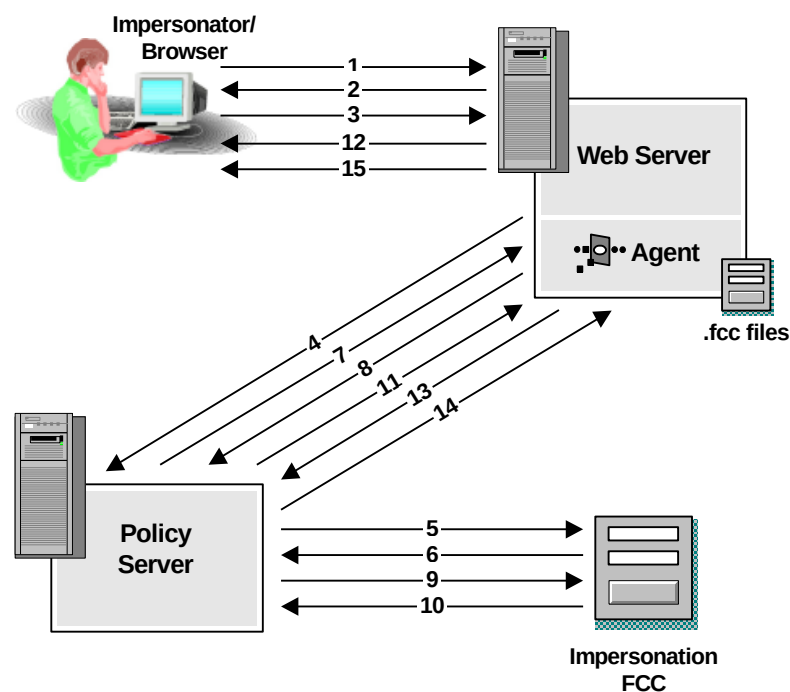
[Impersonation Authentication Schemes](#) (see page 292)

[Rules](#) (see page 327)

[Rule Groups](#) (see page 349)

How an Impersonation Session is Initiated

The following figure is a detailed process flow for how a privileged user initiates an impersonation session.



1. A previously authenticated CSR attempts to impersonate a user by accessing an Impersonation .fcc file.
2. The CSR is allowed to access the file, and the Web Agent presents the .fcc form that was requested in Step 1. The FCC has a hidden form field configured to indicate that the target is in a realm protected by the Impersonation authentication scheme.
3. The impersonator indicates the name of the user to be impersonated.
4. The Web Agent uses the target field of the .fcc to determine the realm.
5. The Policy Server determines the credentials required by the Impersonation authentication scheme.
6. The required credentials are returned.
7. The Policy Server indicates the realm that is being used to protect the .fcc target.
8. The Web Agent attempts to authenticate the impersonated user using the impersonator's session spec as a password.
9. The Impersonation authentication scheme's functionality is invoked to authenticate the user given the supplied credentials (the impersonatee name and the impersonator's session spec). The impersonator's session spec is validated. The Policy Server disambiguates the user and authenticates the impersonation session.

10. The Policy Server verifies that there is an event policy tied to the ImpersonateStart event. If there is a policy that applies to the impersonator, a similar check is performed to determine if there is an ImpersonateStartUser event bound to a policy that applies to the impersonatee. Responses can be tied to rules in either policy.

11. Once the impersonator has been authenticated using the impersonation authentication scheme, he can now attempt to impersonate the user in any application.

Note: Because the impersonator has not accessed any resources, no impersonation has taken place.

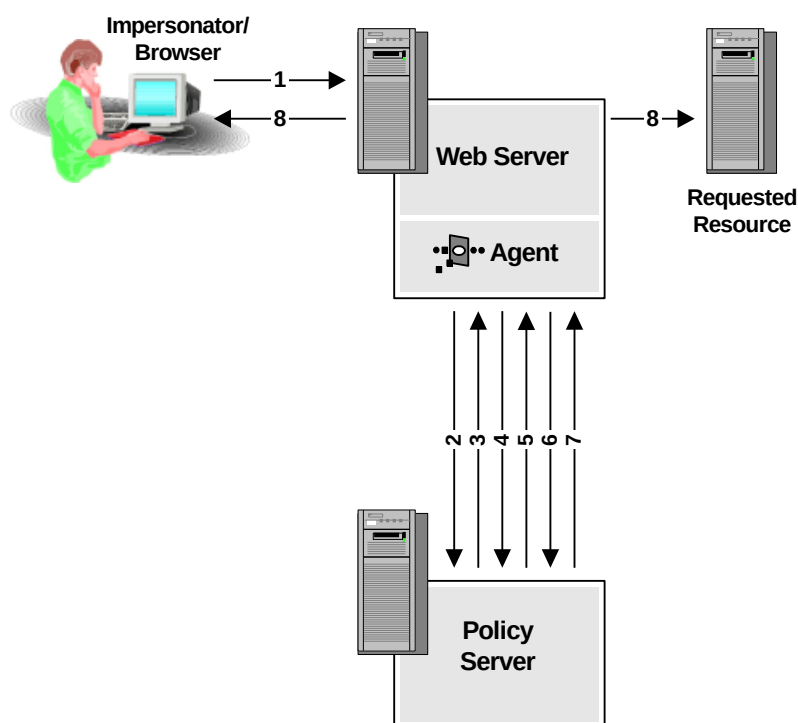
The session specification is returned to the Web Agent. It contains the DN of the impersonatee, the DN of the impersonator, and the directories where both the impersonator and the impersonatee were located.

The Web Agent moves the existing SMSESSION to SMSAVEDSESSION and sets a new SMSESSION cookie equal to the new session spec due to a @pushsession directive in the FCC.

12. The Web Agent attempts to determine whether the impersonatee is authorized for the .fcc target resource.

13. The Policy Server indicates that the impersonatee is authorized for the .fcc target resource.

14. The impersonator accesses the resource that is indicated in the impersonation .fcc. For example, this resource can be a .jsp page that specifies the DN of the impersonator and of the impersonatee.



1. The impersonator (now impersonating a user) navigates to a new application.
2. The Web Agent calls the Policy Server to determine whether the resource is protected.
3. The impersonator accesses the requested resource.
4. Because the resource is protected and an SMSESSION cookie exists in the impersonator's Web Browser for this cookie domain, the Web Agent calls the Policy Server to validate the session spec.
5. The session spec is validated according to current authentication logic except that additional checks are made against the impersonator and the user because the validation logic can determine that impersonation is occurring due to the contents of the session spec. The impersonator must be bound to an event policy in the current realm that has a rule for an ImpersonateStart event. The user must be bound to a similar policy for the ImpersonateStartUser event that applies to users that can be impersonated.
6. The Web Agent asks the Policy Server if the user is authorized for the resource.

7. The Policy Server responds with an authorization response. The authorization response can indicate that the access is allowed or that the access is denied for a multitude of reasons, such as no rule fired for this resource, access explicitly forbidden for this user, authentication level too low, or session timeout. All access reject reasons mirror the user experience. However, Protection Levels are not enforced for impersonated sessions.
8. The requested resource is returned.

More information:

[Enable Impersonation through an .fcc File](#) (see page 466)

Configure an Agent to Use startimp.fcc

Before the Web agent can process the .fcc file (such as startimp.fcc) and initiate an impersonation session, the .fcc filename must be specified in the impersonation authentication scheme and the Web agent must be configured to recognize the .fcc file extension. By default, Web agents ignore the .fcc extension.

Configure an Impersonation Authentication Scheme

Impersonation requires an Impersonation authentication scheme. This scheme is used as a method for an impersonator to initiate the impersonation process. To begin an impersonation session, an Impersonator directly accesses an .fcc file which is a resource protected by the Impersonation authentication scheme.

More information:

[Impersonation Authentication Schemes](#) (see page 292)

Enable Impersonation through an .fcc File

A critical part of impersonation is the authentication process using an appropriate authentication scheme and .fcc files. The .fcc files that enable the impersonation authentication scheme are files installed on a SiteMinder-enabled Web server. The files must end in the extension: .fcc. Files with this extension can take advantage of special processing by the Web Agent. The impersonation authentication scheme is similar to an HTML forms authentication scheme.

An impersonator can invoke the impersonation authentication scheme by directly accessing the .fcc file for the scheme. The proper authentication scheme is invoked through the .fcc files's target parameter. This value can be hard-coded in the .fcc file using the @target directive, or by setting the target (hidden form post variable) to the appropriate Web page.

Note: The target resource must reside in a realm that is protected by the impersonation authentication scheme.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

Basic .fcc Requirements for Impersonation

To configure an impersonation authentication scheme, you need more than one .fcc file and more than one form. For example, you need one file and form to begin impersonation and another file and form to end impersonation. In addition, you can have a .fcc file that configures a form for successful impersonation and another .fcc file that configures a form for failed impersonation. Below are the directives that you need in the .fcc files for impersonation to succeed.

Include the following directives in a .fcc file configured to begin impersonation:

@smpushsession=true

The @smpushsession=true directive instructs the Web Agent to save the impersonator's session specification and to set the session cookie to the impersonated user's session specification.

@username

Set @username to the name of the impersonatee.

Example: @username=%USER%

@password

Set @password to the password of the impersonator.

Example: @password=%SMSERVERSESSIONSPEC%

Note: As shown in the @username and @password examples, you can use the FCC's ability to substitute the contents of headers or cookies in directives at the time the form is posted.

Include the following directives in a .fcc file configured to end impersonation:

@smpopsession=true

The @smpopsession=true directive instructs the Web Agent to end the impersonated user's session and restore the impersonator's session specification.

@target

Set @target to the resource where the impersonator initiated impersonation.

Example: @target=/impersonators/end.htm

@smredirect

Set @smredirect to the resource where the impersonator initiated impersonation.

Example: @smredirect=/impersonators/end.htm

Note: Set both @target and @smredirect to the same resource. Both directives are needed.

For more information about .fcc files, see [HTML Forms Authentication Schemes](#) (see page 244).

FCC Directives for Impersonation

When constructing an .fcc file for impersonation, the following directives should be used in the file:

@logout

This directive logs the user out of SiteMinder and removes the SMSESSION cookie.

@smheaders

This directive adds HTTP request headers to the FCC namespace. For impersonation, this directive provides the contents of the session specification header, SMSERVERSESSIONSPEC (or SM_SERVERSESSIONSPEC; see Note about SMSERVERSESSIONSPEC and LegacyVariables), to the FCC namespace so that it is available for use as a password.

@smpushsession

This directive allows a user to “impersonate” another user and then return to the original session. This directive must be set to "true".

@smpopsession

This directive returns to the original session after @smpushsession has been used. This setting must be set to "true".

@smredirect

This directive redirects requests to the specified target.

@smaltcreds

Allows custom authentication schemes to send credentials larger than 4KB. This may be used in the same manner that the @password directive is used. When credentials are posted to an FCC using @smaltcreds, its value is sent to the Policy server during login as a byte buffer avoiding the password field which is restricted to 4k bytes. The @smaltcreds directive may *not* be used with existing out-of-the-box authentication schemes, but it may be used for custom authentication. Developers of custom authentication schemes must code their authentication scheme libraries to look for the @smaltcreds credentials in the IpszCertBinary field of the user credential struct passed through the Agent API during login.

@target

This directive tells the FCC where to redirect to after processing a URL.

@password

This directive specifies what the contents of the password to be passed to the Policy Server.

@username

This directive specifies the username to be passed to the Policy Server.

% and \$ replacement functionality

The "%" and "\$\$" functionality is used for data replacement similar to scalar variables in perl. "%NAME%" is used to replace "NAME" with the data associated with "NAME" on a post. "\$\$NAME\$\$" is used to replace "NAME" with the data associated with "NAME" on a get.

More information:

[Include Header Values in the FCC Namespace](#) (see page 255)

Obtain the Session Specification using an FCC

The Web Agent includes the session specification in the set of headers that it includes with every request. The session specification header can be submitted as a password using several FCC directives. The @smheaders directive is used to include the SMSERVERSESSIONSPEC header in the FCC's namespace, and the @password directive is used to set the password to the contents of the SMSERVERSESSIONSPEC header.

The SM_SERVERSESSIONSPEC/SMSERVERSESSION header is only available if DisableSessionVars is set to its default value of false in the Web Agent configuration file.

If the LegacyVariables Web Agent parameter is set to Yes, then the header should be SM_SERVERSESSIONSPEC.

If the LegacyVariables Web Agent parameter is set to No, then the header should be SMSERVERSESSIONSPEC.

Note: The LegacyVariables parameter is not supported for Web Agents on IIS 6.0 web servers. For Web Agents on IIS 6.0, SMSERVERSESSIONSPEC is always the correct header.

Example .fcc Files for Impersonation

The following .fcc file is invoked directly by an impersonator or called by the impersonation authentication scheme to begin the impersonation process:

```
@username=%USER%
@smheaders=SMSEVERSESSIONSPEC
@password=%SMSEVERSESSIONSPEC%
@smpushsession=true
<html>
<head><title>Sample Impersonation Form</title></head>
<body>
<h3>Please enter your Impersonation Information</h3>
<form method=post><table>
<tr>
<td>User Name:</td>
<td><input type=text name=USER></td>
</tr>
<input type=hidden name=target
value="server.company.com/app/impersonatee/successimp.htm">
<tr><td><input type=submit></td></tr>
</table></form>
</body>
</html>
```

The following .fcc file is invoked by impersonators to return to their original sessions:

```
@smpopsession=true
@target=/impersonators/end.htm
@smredirect=/impersonators/end.htm
```

Effects of FCCCompatMode

By default, a SiteMinder Web Agent is configured with the attribute FCCCompatMode=true. In this default state, an Agent can process impersonation requests. However, if you configure a Web Agent so that FCCCompatMode=false, then you must ensure the following in order for the Agent to process impersonation requests:

- The EncryptAgentName setting for the Web Agent must be set to false.
- The name of the Agent that will process impersonation requests must be hard-coded in the .fcc file that initiates impersonation or in the query data that is sent to the Agent that processes the initiation of an impersonation session.

Impersonation Event Configuration

Impersonation events are rule events that must be configured and included in policies in order to allow a privileged user to impersonate another user.

You must configure at least one of each of the following impersonation events in order to enable impersonation:

ImpersonateStart

When included in an appropriate policy, a rule that includes this event allows an impersonation session to begin.

ImpersonateStartUser

When included in an appropriate policy, a rule that includes this event allows a set of users to be impersonated.

More information:

[Configure a Rule for Impersonation Event Actions](#) (see page 339)

Configure Policies for Impersonation

In order for impersonation to function correctly, multiple policies must be configured. They include:

For the resource that initiates impersonation:

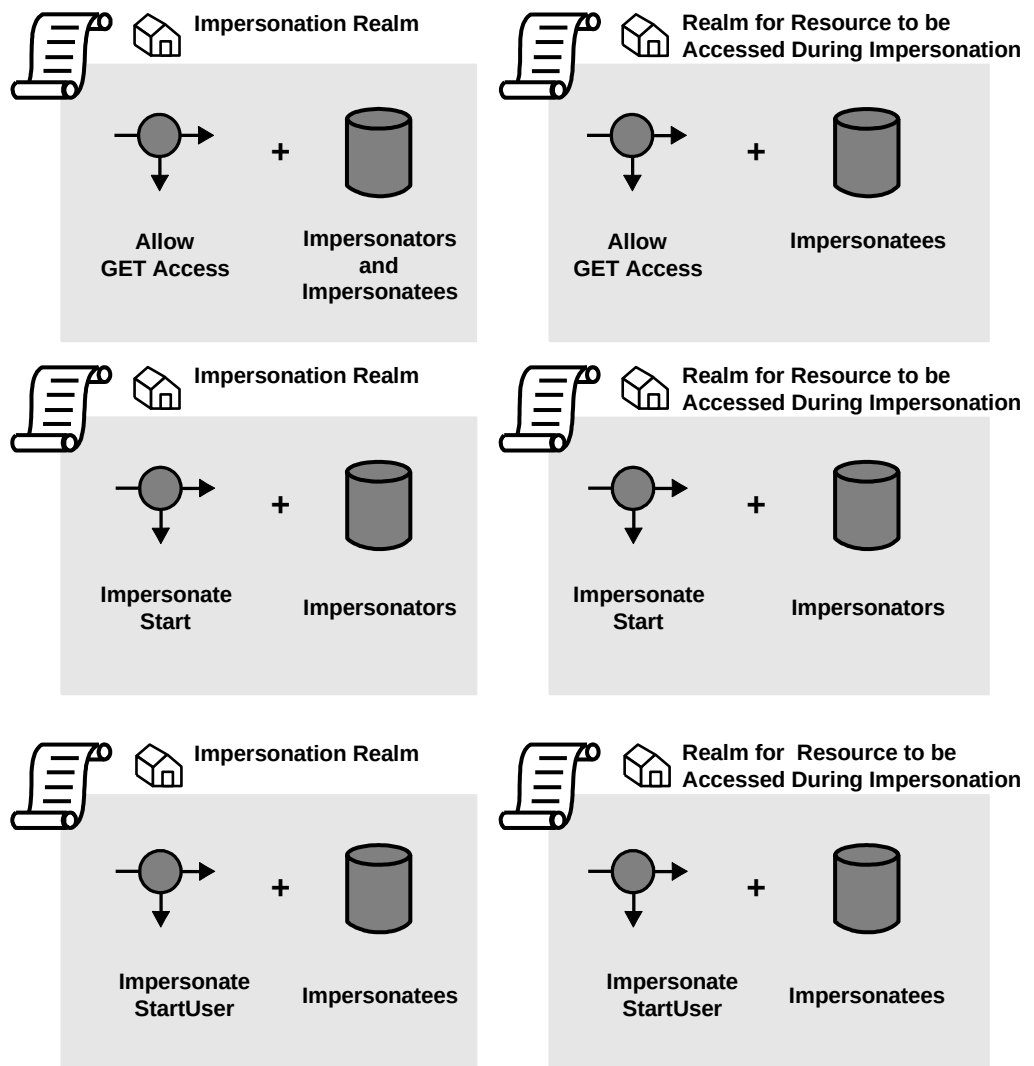
- Access control policy for both impersonators and impersonatees. For example, a policy that includes a GET access rule for both impersonators and impersonatees.
- Impersonation policy that includes an ImpersonationStart rule and impersonators.
- Impersonation policy that includes an ImpersonationStartUser rule and impersonatees.

For the resource to be accessed by the impersonator:

- Access control policy for impersonatees. For example, a policy that includes a GET access rule for impersonatees.
- Impersonation policy that includes an ImpersonationStart rule and impersonators.
- Impersonation policy that includes an ImpersonationStartUser rule and impersonatees.

Note: Policies must be in place for each resource that may be accessed by an impersonator.

The following figure shows the minimum policies required for an impersonator to access a resource.



More information:

[Policies](#) (see page 375)

Multiple Cookie Domain Support

If a site is composed of multiple cookie domains, an Impersonator's identity could be confused while moving between resources in separate cookie domains. To avoid this problem the SMSESSION cookies in all of the cookie domains other than the current one must be cleared. This must be accomplished by modifying the forms that are used to begin and to end Impersonation. These forms should be augmented by HTML or script that will call server-side code to clear the SMSESSION cookies in all of the cookie domains other than the current one.

For example, consider Web Agents installed at yourcompany.com, subsidiaryA.com, and subsidiaryB.com. A Web Agent that carries out Impersonation could be located in the yourcompany.com domain. The .fcc files to start and end impersonation would need to call server-side functionality (JSP pages possibly) in subsidiaryA.com and subsidiaryB.com to clear out the SMSESSION cookies in those cookie domains for Impersonation to function correctly.

Sample Implementation of Impersonation

This section contains a description of a simple implementation of impersonation. The minimum Policy Server objects required to implement impersonation are:

System Objects:

Agent

Impersonation requires a Web Agent and its associated Policy Server Agent object.

Note: To implement impersonation, you must have at least one SiteMinder Web Agent installed in your deployment. For information, see the *Web Agent Installation Guide*.

Authentication Scheme

An impersonation authentication scheme based on the Impersonation Authentication Scheme Template is required. For the sample defined in this section, the authentication scheme is named "Impersonation Auth".

User Directory

Impersonation requires one or more user directory objects that point to user stores which contain impersonators and impersonatees. The two populations of users should be distinguishable due to an attribute value or group membership.

Domain

A policy domain is required. For the sample defined in this section, the policy domain is named "Impersonation Domain".

Domain Objects:

Realms

For the sample described in this section, two realms are required: "Impersonation" and "App1". The "Impersonation" realm should use the "Impersonation Auth" authentication scheme. The "App1" realm can use any authentication scheme.

Rules

For the sample described in this section, you must configure a rule under the "Impersonation" realm that allows access to all resources for the "Get" action. In other words, an asterisk should be entered in the Resource field for the rule. You must also configure the rules for the impersonation events. One rule allows impersonation if the impersonator is included in an applicable policy, and the other rule allows an impersonatee to be impersonated if included in a different, applicable policy.

Rules

A similar set of rules to those in the "Impersonation" Realm should be created under the "App1" realm.

Policies

For the sample described in this section, six policies are needed. One policy must be defined for each rule in the "Impersonation" realm, and one policy must be defined for each rule in the "App1" realm.

The previous figure shows a group of domain objects similar to the objects required to implement impersonation.

Sample Impersonation Implementation Assessment

Once all of the required Policy Server objects are in place, an administrator initiates an impersonation session by doing the following:

1. The administrator who will become the impersonator logs into a SiteMinder protected network.
2. The administrator provides credentials and is authenticated and authorized by SiteMinder.
3. The administrator accesses the imp.fcc file using a Web browser.
4. The administrator is prompted to enter a user ID for the person to be impersonated. The administrator may also be required to provide additional information about the user to be impersonated.
5. The administrator submits the information.

6. The Policy Server uses the policies defined for impersonation to determine the following:

- Is the administrator allowed to act as an impersonator?
- Is the user allowed to be impersonated?

7. If both are true, the impersonator impersonates the impersonatee.

Note: Without custom development (.jsp pages, servlets, etc.) the impersonation session can be tracked using the Policy Server's audit logging. However, it may be beneficial for an enterprise to create some custom Web applications to monitor and track impersonation sessions.

Sample Forms for Impersonation

Impersonation requires a number of Web-based forms that must be customized for an appropriate interface for an enterprise. This section outlines the basic requirements of these forms and makes several suggestions about the contents of the forms.

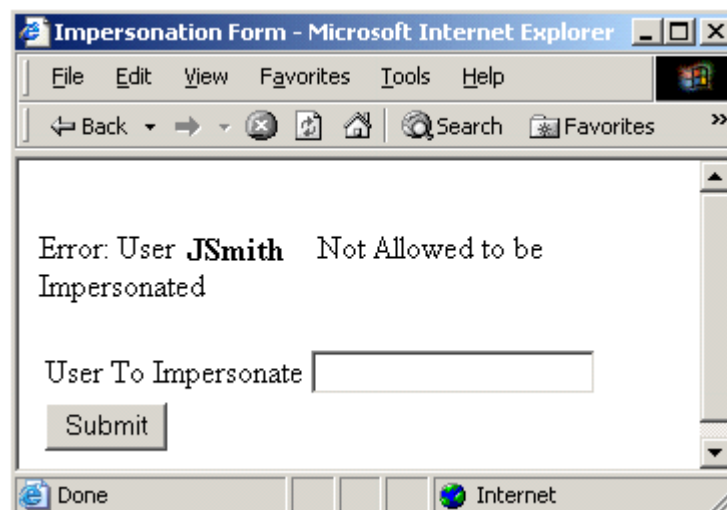
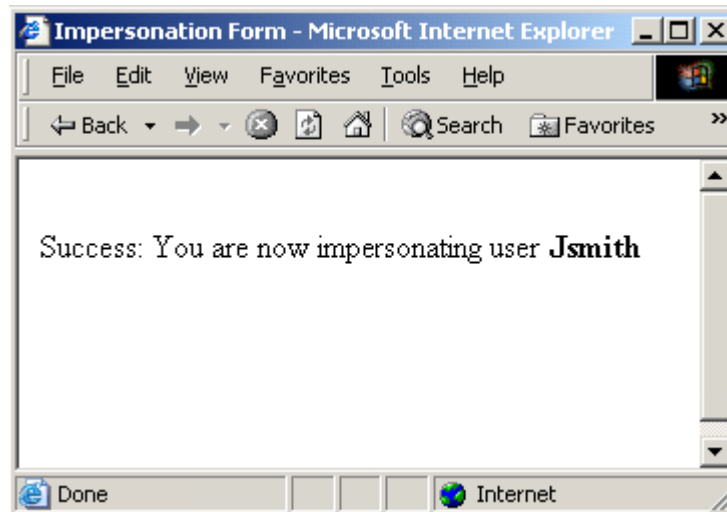
Forms for Initiating the Impersonation Process

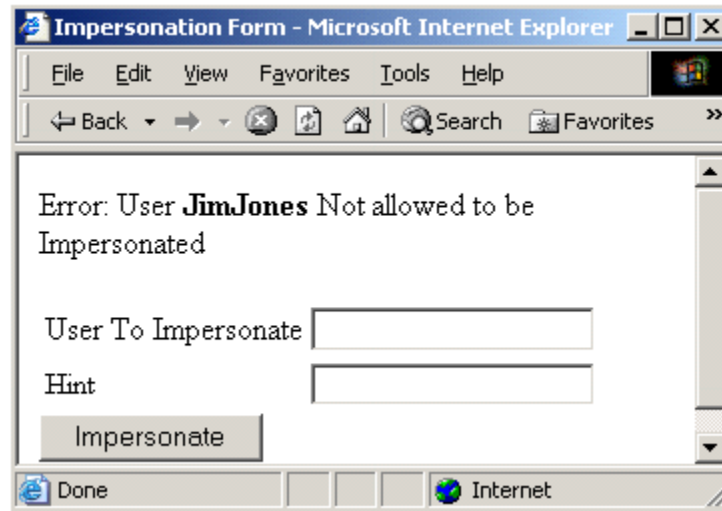
The directives in the .fcc files define the forms that the impersonator sees.

Note: The hint field configures impersonation so that additional attributes can be used to disambiguate an impersonatee. The hint is configurable in the impersonation authentication scheme as a list of additional attributes, which can include information such as an employee number or any other user attribute that uniquely defines a user in the user directory.

Impersonation Result Forms

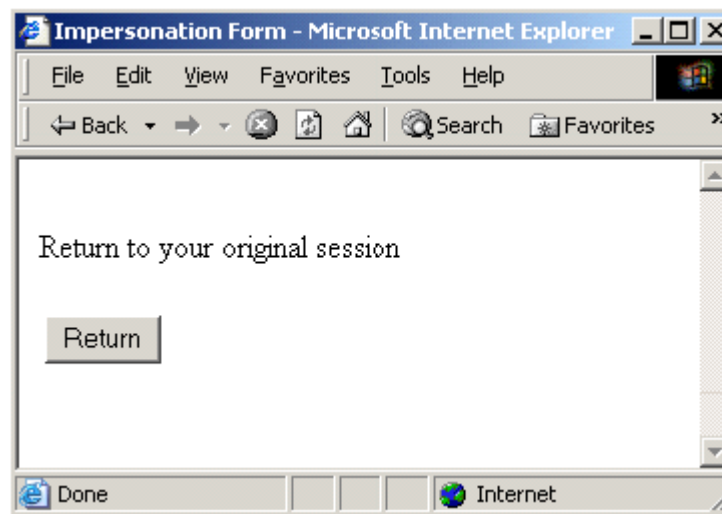
The results of an impersonation attempt must be communicated in a form. The following are examples of forms to represent both successful and failed impersonation attempts.





Impersonation Logoff Forms

A form for ending an impersonation session must be configured.



Chapter 21: Password Policies

This section contains the following topics:

[Password Services Overview](#) (see page 479)
[Create Password Policies](#) (see page 483)
[Configure Password Expiration](#) (see page 485)
[Configure Password Composition](#) (see page 487)
[Password Regular Expressions](#) (see page 488)
[Configure Password Restrictions](#) (see page 490)
[Configure Advanced Password Options](#) (see page 491)
[User-initiated Password Changes](#) (see page 492)
[Remove the Login ID When Redirecting for Password Services](#) (see page 496)
[CGI-based Password Services HTML Form Customization](#) (see page 497)
[Servlet-based Password Services JSP Form Customization](#) (see page 502)
[Authentication Schemes Requiring Additional Attribute Information](#) (see page 507)
[Enhanced Active Directory Integration with Password Services](#) (see page 510)
[Password Policy Troubleshooting](#) (see page 512)

Password Services Overview

Password Services provide an additional layer of security to protected resources by allowing you to manage user passwords in LDAP user directories or relational databases. To manage user passwords, you create password policies that define rules and restrictions governing password expiration, composition, and usage. In addition to managing passwords, Password Services enables users to select their own passwords:

- At the times you specify, such as when a password expires
- When users feel it is necessary to change their passwords
- When new users register using Registration Services

The password policy ensures that the user will select a valid password without additional administrative involvement.

When Password Services are active, SiteMinder invokes a password policy whenever a user attempts to access a protected resource. Password Services then evaluates the user's credentials. If the user's password has expired based on criteria defined in the password policy, the user's account can be disabled to prevent unauthorized access to the resource or the user can be forced to change his password. If disabled, the user's account must be re-activated by an administrator.

Password policies can be associated with an entire user directory or database, or a subset of the directory or database, called a namespace. Multiple password policies can be configured for the same user directory or namespace, in which case they are applied according to priorities you can specify for them.

Password Services provides two mechanisms for implementing password services:

CGI-based

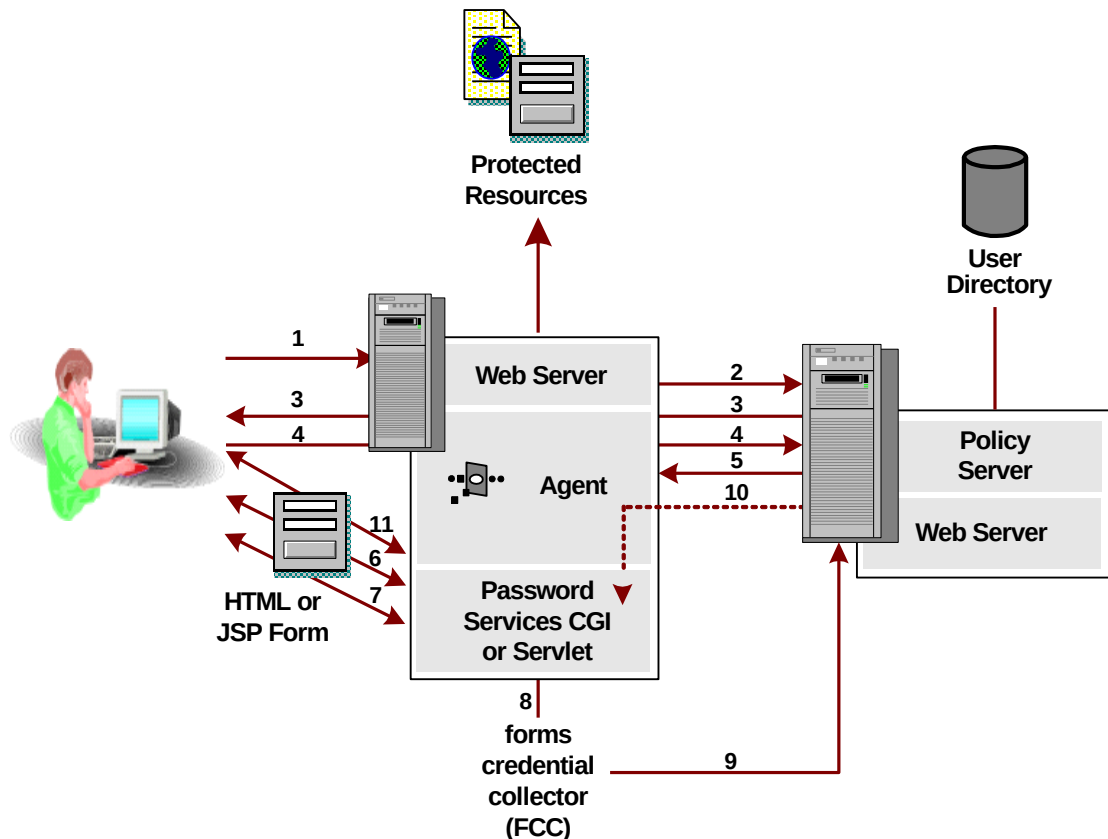
Password Services CGI with customizable HTML forms (the default mechanism).

Servlet-based

Password Services servlet with customizable JSP forms.

How Password Services Work

The following illustration depicts an example of how Password Services work in a SiteMinder environment configured to protect web resources. In the example, a user's password has expired and must be changed.



When a user attempts to access a protected resource:

1. The user attempts to access a Web page by sending a request to the Web Server.
2. The SiteMinder Web Agent intercepts the request and checks with the Policy Server to see if the requested resource is protected.
3. If the resource is protected, the Policy Server requests user credentials from the browser.
4. The user sends credentials to the Policy Server.
5. The Policy Server authenticates the user.

Once the user is authenticated, the Policy Server checks to see if user information is stored in a directory or database associated with a password policy. If it is, the Policy Server makes sure that the user's password is valid based on the password policy criteria.

6. If the password has expired, the Policy Server sends a redirection URL pointing to the password services servlet to the Agent.

The Agent redirects the browser to request the password services servlet using the redirect URL.

7. The password services CGI or servlet determines which HTML/JSP (as appropriate) form to present to the user from a set of HTML/JSP templates and displays that form in the browser.

The form displays a message explaining why the user has been redirected. It prompts users to enter their old password and new password, then confirm the new password by re-entering it.

8. Once the user has completed the form, the Password Services CGI or servlet passes the information received from the Agent and the encrypted information the user entered to the Forms Credential Collector (FCC).
9. The FCC passes the information received from the Password Services CGI or servlet to the Policy Server.

The Policy Server checks the new password against the password policy to ensure that it is valid, then changes the password.

10. The Policy Server again sends a request to the Agent to redirect the browser to the Password Services CGI/servlet.
11. The Password Services servlet displays a message informing the user that the password has been successfully changed.

Once the user has read the message, she is redirected to the page she originally requested.

Apache Web Server Prerequisites

Depending on the operating system on which your environment is running, you must manually edit the Apache httpd.conf configuration file to ensure that an implementation of CGI-based Password Services works.

To modify the httpd.conf configuration file for Solaris, add the PassEnv line after the ServerRoot line.

Example:

```
ServerRoot "/export/home/smuser/apache/"  
PassEnv LD_LIBRARY_PATH "/export/home/smuser/netegrity/webagent/bin"
```

To modify the httpd.conf configuration file for Windows, add the following lines:

```
# To use CGI scripts:  
AddHandler cgi-script .exe  
AddHandler smformsauth-handler .fcc  
AddHandler smcookieprovider-handler .ccc
```

Important Considerations

When setting up Password Services, it is important that you consider the following:

Time Differential Effect on Password Services on Multiple Policy Servers

If you use Password Services with multiple Policy Servers, be sure the time settings on the Policy Server systems are synchronized, to prevent inadvertent disabling of user accounts or pre-mature forced password changes.

Working With Directory Server Password Policies

Some Directory servers let you create simple password policies which are evaluated by the directory before SiteMinder policies. If the Directory Server policies are more restrictive than SiteMinder policies, the Directory server will accept or reject passwords without notifying SiteMinder. SiteMinder will not know that a user is attempting to login and will not provide any of the password management features.

Important! To use Password Policies, you must disable any Directory server policies or make them less restrictive than the Password Policies.

Change the Default Redirect URL

By default, a SiteMinder Web Agent installation includes the following:

`/web_agent_install/samples/forms/smpwservices.fcc`

`web_agent_install`

Specifies the Web Agent installation path.

When a user cannot be properly disambiguated by the authentication server, the user is redirected to this location.

To customize the redirection destination, you must set the `NETE_PWSERVICES_REDIRECT` environment variable to the relative URL of the redirect destination, and restart the Web server associated with the Web Agent.

Create Password Policies

You use the Password Policy Properties dialog in the Policy Server User Interface to configure password policy objects.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a password policy object

1. In the Policy Server User Interface, create a password policy object to open the Password Policy Properties dialog.
2. In the Name field, enter the name of the new password policy.
3. Optionally, in the Description field, enter a brief description.
4. In the User Directory Information group box, select the User Directory to which the password policy will apply.
5. If you want to apply the password policy you are creating to the entire directory, select the Password Policy applies to the whole directory radio button.
6. If you want to apply the password policy to part of the directory you specified, elect the Password Policy applies to part of the Directory radio button and then use the Lookup button to select the namespace to which you want to apply the password policy.

Note: For more information about LDAP User Directories, see [LDAP Overview](#) (see page 142). For information about relational database directories, see [ODBC Database Overview](#) (see page 153).

7. Select the Allow Nested Groups check box if your password policy is associated with nested groups within your LDAP directory.
8. In the Password Policy Enabled State group box, enable the password policy by selecting the Enabled check box.

If you want to create and configure a password policy but not apply it to a user directory, leave the Enabled check box cleared.

9. Enter the URL to which users should be redirected when the password policy determines that an invalid password has been entered in the Redirection URL field. To enable redirection, you must specify the virtual path to the Password Services CGI or servlet. By default, the Policy Server fills in the path to the Password Services CGI:

`http://myserver.mycompany.org/siteminderagent/pwcgi/smpwservicescgi.exe`

If you want to use the Password Services servlet, you should specify its path:

`http://myserver.mycompany.org/siteminderagent/pwservlet/PSWDChangeServlet`

If you choose to set up a custom Password Services directory on a non-default web server, be sure to do the following:

- Copy the smpwservicescgi.exe to a location accessible by the non-default web server.
- The CGI extracts an agent_path from the Redirect URL to replace the default /siteminderagent/pw path. When you setup a custom Password Services directory, create two virtual directories: pw and pwcgi. The latter is a CGI directory. For example:

`<custom directory>/pwcgi`

`<custom directory>/pw`

When a user is redirected to the Password Services CGI or servlet, it takes the information from the Policy Server, determines why the password is invalid, and displays a form that provides information or requests additional credentials from the user.

Make sure that the Password Services CGI or servlet is not protected. If SiteMinder is protecting directories above the servlet, create a realm that specifies the following:

- Resource Filter if using CGI: siteminderagent/pwcgi/smpwservicescgi.exe
- Resource Filter if using Servlet: siteminderagent/pwservlet/PSWDChangeServlet
- Default Resource Protection: Unprotected

Do not create a policy for this realm.

Note: If a user who is accessing resources through an Agent that is not using an SSL connection must change passwords, the user's new password information will be received over the non-secure connection. To provide a secure change of passwords, set up a password policy that redirects the user over an SSL connection using the Redirection URL field.

10. Click Apply to save the new password policy or click OK to save the new password policy and return to the Policy Server User Interface.

Once you have created a basic password policy and bound it to an LDAP Directory by following the steps in this procedure, you can configure the policy to reflect your desired password logic. For more information, see the following sections:

- For information about configuring expiration settings, see [Configure Password Expiration](#) (see page 485).
- For information about configuring composition settings, see [Configure Password Composition](#) (see page 487).
- For more information about configuring regular expression settings, see [Password Regular Expressions](#) (see page 488).
- For information about configuring password restrictions, see [Configure Password Restrictions](#) (see page 490).
- For information about configuring advanced password settings, see [Configure Advanced Password Options](#) (see page 491).

More information:

[Password Policy Dialog](#) (see page 769)

Configure Password Expiration

To help manage user access, administrators can define events such as multiple failed login attempts or account inactivity. When those events are triggered, the Policy Server disables the user account that triggered the event and optionally redirects the user to a new Web page.

You use the Expiration tab of the Password Policies Properties dialog to configure events that disable user accounts.

Note: All password expiration settings are optional — specify a value for each setting that you want to enable; if you do not want to enable a particular setting, leave the field blank.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure password expiration

1. In the Policy Server User Interface Password Policy Properties dialog, select the Expiration tab.
2. Enable the Track successful logins option to log information about successful user logins, including the time of the last login, to the user directory. This option must be set if you want to configure passwords to expire due to inactivity.

Note: If you do not need to configure passwords to expire from inactivity, it is recommended that you do not set this option for performance reasons.

3. Enable the Track Failed Logins check box to log information about failed user login attempts to the user directory. This option must be set if you want to configure accounts to be disabled because of unsuccessful logins.
4. Enable or disable the Authenticate on Login Tracking Failure check box:
 - If you enable this check box, users can login even if password data cannot be written to the user directory.
 - If you leave this check box disabled, users cannot login if their activity cannot be written to the user directory.
5. In the Password expires if not changed group box:
 - a. Specify the amount of time that may elapse between password changes in the After *<number>* Days field.
 - b. Specify the action that Password Services takes if a password is not changed within the specified time by selecting the disable user or force password change radio button.
 - c. Specify how many days in advance users should be notified that their password will expire in the Issue expiration warnings for *<number>* days field.
6. If you set the Track Failed Logins option, set options in the Incorrect Password group box:
 - a. Specify the number of incorrect passwords a user can try before the user's account is disabled in the Account will be disabled after *<number>* successive incorrect passwords field.

- b. In the After *<number>* minutes field, specify the number of minutes that will apply to the condition indicated by the radio buttons described Select one of the following radio buttons to determine the behavior of the password policy if a user fails to login after the number of attempts specified in step 5a:
 - allow 1 login attempt—A user may attempt a single login after the specified number of minutes (Step 5b). For each further failed attempt, the user must wait the specified number of minutes before making another login attempt. For example, consider a value of 3 attempts in step 5a, and a value of 10 minutes in step 5b. If a user fails to login three times, the user may attempt a single login every ten minutes until successful.
 - re-enable account—A user's account will be re-enabled after the specified number of minutes (Step 5b). The user may attempt as many logins as specified in Step 5a before the user's account is disabled. For example, consider a value of 3 attempts in step 5a, and a value of 10 minutes in step 5b. If a user fails to login three times, the user may attempt three logins every ten minutes until successful.
7. If you set the Track successful logins option, set options in the Password expires from inactivity group box:
 - a. Specify the amount of time that can elapse between user logins in the After *<number>* Days field.
 - b. Specify the action that Password Services takes if users do not login within the specified time by selecting the disable user or force password change radio button.
8. Click Apply to save the changes or click OK to save the changes and return to the Policy Server User Interface.

More information:

[Password Policy Dialog—Expiration Tab](#) (see page 774)

Configure Password Composition

You can specify rules that determine the character composition of newly created passwords. You use the Composition tab of the Password Policy Properties dialog in the Policy Server User Interface to configure password composition settings.

Note: Password composition rules are optional. Enter a value for the rules you want to enable. If you do not want to enable a rule, leave the field blank.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure password composition restrictions

1. In the Password Policy Properties dialog, select the Composition tab.
2. In the Password Length group box, specify the minimum and maximum password lengths in the Minimum Length and Maximum Length fields.
3. In the Maximum field in the Repeating Characters group box, enter the maximum number of characters that can appear consecutively in a password.
4. Specify whether to allow different character types and if so, the minimum requirements for passwords in the Content Minimums group box.
5. Click Apply to save the changes or click OK to save the changes and return to the Policy Server User Interface.

More information:

[Password Policy Dialog—Composition Tab](#) (see page 777)

Password Regular Expressions

Password regular expression matching allows you to specify regular expressions (text patterns used for string matching) that each password must match or not match in order to be valid. This can be useful, for example, if you want to require that the first character in the password is a digit, but that a digit not be the last character.

You use the Regular Expressions tab of the Password Policy Properties dialog to configure password regular expression matching.

Regular Expressions Syntax

This section describes the syntax you should use to construct regular expressions for password matching. This syntax is consistent with the regular expression syntax supported for resource matching when specifying realms.

Characters	Results
\	Used to quote a meta-character (like '*')
\\	Matches a single '\' character
(A)	Groups subexpressions (affects order of pattern evaluation)
[abc]	Simple character class (any character within brackets matches the target character)

Characters	Results
[a-zA-Z]	Character class with ranges (any character range within the brackets matches the target character)
[^abc]	Negated character class
.	Matches any character other than newline
^	Matches only at the beginning of a line
\$	Matches only at the end of a line
A*	Matches A 0 or more times (greedy)
A+	Matches A 1 or more times (greedy)
A?	Matches A 1 or 0 times (greedy)
A*?	Matches A 0 or more times (reluctant)
A+?	Matches A 1 or more times (reluctant)
A??	Matches A 0 or 1 times (reluctant)
AB	Matches A followed by B
A B	Matches either A or B
\1	Backreference to 1st parenthesized subexpression
\n	Backreference to <i>n</i> th parenthesized subexpression

All closure operators (+, *, ?) are greedy by default, meaning that they match as many elements of the string as possible without causing the overall match to fail. If you want a closure to be reluctant (non-greedy), you can simply follow it with a '?'. A reluctant closure will match as few elements of the string as possible when finding matches.

Limit: Each regular expression can contain no more than 10 subexpressions, including the expression itself. The number of subexpressions equals the number of left or opening parentheses in the regular expression plus one more left parenthesis for the expression itself.

Configure Regular Expression Matching

Regular expression matching for passwords is optional. You need only specify entries for expressions that passwords must match or must not match; if you have no expression matching requirements, do not create any regular expression entries.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure regular expression matching restrictions for passwords

1. In the Password Policy Properties dialog, select the Regular Expressions tab.
2. In the Passwords MUST Match These Expressions group box, add entries for each regular expression that passwords must match. Each regular expression entry is a name/value pair consisting of the expression definition and a descriptive tag.

You can also edit or remove existing required regular expression entries.

3. In the Passwords MUST NOT Match These Expressions group box, add entries for each regular expression that passwords must *not* match. Each regular expression entry is a name/value pair consisting of the expression definition and a descriptive tag.

You can also edit or remove existing forbidden regular expression entries.

More information:

[Password Policy Dialog—Regular Expressions Tab](#) (see page 779)

Configure Password Restrictions

Using password policies, you can place restrictions on password usage. The restrictions include how long a user must wait before reusing a password and how different the password must be from ones previously selected. You can also prevent users from specifying words that you determine are a security risk or contain users' personal information.

You use the Restrictions tab in the Password Policy Properties dialog to configure password restrictions.

Note: Password restrictions are optional. Enter a value for restrictions you want to enable. If you do not want to enable a restriction, leave the field blank.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure password restrictions

1. In the Password Policy Properties dialog, select the Restrictions tab.
2. In the Reuse group box, specify requirements for reusing passwords:
 - a. In the Minimum number of days before reuse field, specify the number of days a user must wait before reusing a password.
 - b. In the Minimum number of passwords before reuse field, specify the number of new passwords a user must use before reusing an old one.

3. In the Change Required group box, specify how different new passwords must be from old ones by completing the following steps:
 - a. In the Percent different from last password field, specify the percentage of characters within a new password that must be different from the last password.
 - b. To ignore the position of the characters in the password when determining the percentage, select the Ignore sequence when checking for differences check box.
4. In the Match Length field in the Profile Attributes group box, specify the number of consecutive characters the password policy compares to personal information stored in user profiles.
5. In the Dictionary group box, complete the following steps to have Password Services check passwords against a user-defined dictionary:
 - a. In the Path field, specify the location of the user-defined dictionary file.
 - b. In the Match Length field, specify the minimum length of words checked against the dictionary.
6. Click Apply to save the changes or click OK to save the changes and return to the Policy Server User Interface.

More information:

[Password Policy Dialog—Restrictions Tab](#) (see page 782)

Configure Advanced Password Options

Advanced password policy options allow you to configure pre-processing of submitted passwords prior to validation and storage (for example, forcing all characters to upper case) and to assign the policy a priority to allow predictable evaluation of multiple password policies that apply to the same user directory or namespace.

You use the Advanced tab in the Password Policy Properties dialog to configure advanced password options.

Note: Password pre-processing options are optional. You *should* specify a unique password policy evaluation priority for each password policy that may be assigned to the user directory or namespace.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To configure advanced password options

1. In the Password Policy Properties dialog, select the Advanced tab.
2. In the Password Pre-Processing group box, specify options to process submitted passwords prior to evaluation and storage.
Note: You should specify *identical* pre-processing options for each password policy that will be applied to the same user directory or namespace.
3. If the password policy is one of multiple policies that will apply to the same user directory or namespace, specify an Evaluation Priority (0-999; 999 is highest) for the password policy and whether to Apply Lower Priority Password Policies in the Password Policy Priority group box.

For detailed information about each setting on the Advanced tab, see [Password Policy Dialog—Advanced Tab](#) (see page 785).

User-initiated Password Changes

User-initiated password changes allow end users to change their passwords without any intervention from an administrator. Users can elect to change their passwords by clicking a link to access the Password Change Request form.

How a Change Password Link is Added

To enable user-initiated password changes, the Policy Server administrator must add a Change Password link to an HTML page. For example, administrators might add this link to a login page so users can opt to change their password at login.

Add a Change Password Link to the Password Services CGI

In most situations, use the hyperlink shown in the following example for the Password Change Request form:

```
<a href="/siteminderagent/pwcgi/smpwservicescgi.exe?SMAUTHREASON=34&TARGET=URL_of_Protected_Web_Page_After_Password_Change">Change Password</font></a>
```

where *URL_of_Protected_Web_Page_After_Password_Change* is the URL of a *protected* web page to which users are directed after they change their password. For example, <http://server.myorg.org/AfterPasswordChange.html>.

Note: Adding this link to an unprotected page allows all users to change their own passwords. The URL to which users are directed after they change their password *must* be protected, otherwise the password change will fail.

Certain Web Agent configuration parameter settings require slightly different URLs for the password change request form. See the following examples:

- If you set the value of the FCCCompatMode parameter to no, then specify the SMAGENTNAME in your URL, as shown in the following example:

```
"<a
href="/siteminderagent/pwcgi/smpwservicescgi.exe?SMAUTHREASON=34&SMAGENTNAME=
$$SMAGENTNAME$$&TARGET=$TARGET$">Change Password</font></a>"
```

- If you set the value of the SecureURLs parameter to yes, then add the SMQUERYDATA to the *end* of your URL, as shown in the following example:

```
<a
href="https://server.example.com/siteminderagent/forms/smpwservices.fcc?SMAUT
HREASON=34&TARGET=URL_of_Protected_Web_Page_After_Password_Change&SMQUERYDATA
=value">Change Password</font></a>
```

Allow Specific Users to Change Their Passwords in CGI

If you want to allow only certain users to change their own passwords, complete the following procedure.

To allow specific users to change their passwords

1. Modify the permissions for the PWLogin.template file:
 - a. Navigate to the following location:
`<web agent installation>/pw`
where `<web agent installation>` is the installed location of the Web Agent.
 - b. In File, Properties, deselect the read-only attribute.
2. Edit the text in the PWLogin.template file:
 - a. Open PWLogin.template in a text editor.
 - b. Add the following line to the template at an appropriate location:
`<a href="/siteminderagent/pwcgi/smpwservicescgi.exe?SMAUTHREASON=34&TARGET=$$TARGET$$">Change Password</a>`
 - c. Save the file.
3. Access the Policy Server User Interface.
4. Create an Authentication Scheme with the following settings:
 - Server Name: The name of the server where the Password Services CGI resides.
For example:
`myserver.mycompany.org`
 - Target: `/siteminderagent/pwcgi/smpwservicescgi.exe`

5. Optionally, create a new policy domain.

The policy domain should include the user directory that contains the users that are allowed to change their own passwords.

If you do not create a new policy domain, select an existing policy domain.

6. Create a realm that specifies the directory that you are protecting. In the Authentication Scheme list box, select the authentication scheme you created in step 4.
7. Create a rule under the realm that specifies the resource(s) that you are protecting.
Note: If you create a rule that specifies all of the resources (*) in the directory that you are protecting, you do not have to create separate rules for localized Password Services.
8. Create a policy that binds the rule you created and the users/groups who are allowed to change their passwords.

More information:

[Create and Use a Localized Password Services Properties Files](#) (see page 501)

Add a Change Password Link to the Password Services Servlet

The link to the Password Change Request form is:

```
<a href="/siteminderagent/pwservlet/PSWDChangeServlet?
SMAUTHREASON= 34&TARGET=<URL_of_Web_Page_After_Password_Change>">
Change Password</font></a>
```

where *<URL_of_Web_Page_After_Password_Change>* is the URL of the Web page where users are directed after they change their password. For example, <http://server.myorg.org/AfterPasswordChange.html>.

Adding this link to an unprotected page will allow all users to change their own passwords.

Allow Specific Users to Change Their Passwords in Servlet

If you want to allow only certain users to change their own passwords, complete the following procedure.

To allow specific users to change their passwords

1. Modify the permissions for the PWLogin.jsp file:
 - a. Navigate to the following location:
<web agent installation>/pwservlet/default
where *<web agent installation>* is the installed location of the Web Agent.
 - b. In File, Properties, deselect the read-only attribute.

2. Edit the text in the PWLogin.jsp file:
 - a. Open PWLogin.jsp in a text editor.
 - b. Add the following:

```
<p align="center"><font size="1">  
<a href="/siteminderagent/pwservlet/  
PSWDChangeServlet?SMAUTHREASON=34&TARGET  
=<%=TARGET%>&SMAGENTNAME=<%=agentname%>">Change Password</font></a>
```
 - c. Save the file.
3. Access the Policy Server User Interface.
4. Create an Authentication Scheme with the following settings:
 - Server Name: The name of the server where the Password Services servlet resides. For example:

myserver.mycompany.org
 - Target: /siteminderagent/pwservlet/PSWDChangeServlet
5. Optionally, create a new policy domain.

The policy domain should include the user directory that contains the users that are allowed to change their own passwords.

If you do not create a new policy domain, select an existing policy domain.
6. Create a realm that specifies the directory that you are protecting. In the Authentication Scheme list box, select the authentication scheme you created in step 4.
7. Create a rule under the realm that specifies the resource(s) that you are protecting.

If you create a rule that specifies all of the resources (*) in the directory that you are protecting, you do not have to create separate rules for localized Password Services.
8. Create a policy that binds the rule you created and the users/groups who are allowed to change their passwords.

More information:

[Localize Servlet-based Password Services](#) (see page 505)

Password Self-Changes

When users want to change their passwords they must:

1. Click Change Password.

The Policy Server User Interface displays the Password Change Request form.

2. Enter the requested information, then click the Change Password button.

The Policy Server User Interface displays another Password Change Information page, indicating that the user's password has been changed.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Remove the Login ID When Redirecting for Password Services

During password services processing, a user's request is redirected multiple times. When the request is redirected, the login ID (typically the username) which was entered by the user, is appended to the request URL by default. To modify the default behavior so that the login ID (username) is not appended to redirects, you can do one of the following procedures.

To remove the login ID when redirecting for password services in Windows

1. Add the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\PolicyServer\
DisallowUsernameInURL
2. Set the DWORD value to one of the following:
 - 0 — Applies the default behavior of appending the UID to the request URL.
 - 1 — Changes the default behavior so that the UID is not appended to the request URL.

To remove the login ID when redirecting for password services in UNIX

1. Navigate to:
`<policy-server-install-dir>/registry/`
2. In a text editor, open the following file:
`sm.registry`

3. Add the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Authentication\DisallowUsernameInURL
4. Set the DWORD value to one of the following:
 - 0 — Applies the default behavior of appending the UID to the request URL.
 - 1 — Changes the default behavior so that the UID is not appended to the request URL.

Note: If you choose to modify the registry setting so that the login ID is not appended to password services redirects, password services automatically uses the templates with the PWnn prefix. These templates are described in CGI-based Password Services HTML Form Templates (see page 497).

More information:

[How Password Services Work](#) (see page 480)

CGI-based Password Services HTML Form Customization

The Password Services CGI uses HTML forms to allow users to change their passwords. You can customize these forms to create a branded look and modify the information provided on the forms.

CGI-based Password Services HTML Form Templates

The Password Services CGI uses HTML forms to collect password information from users and send it to the Policy Server. The Policy Server checks the information against the password policy and then sends a message back to the Password Services using another form.

A SiteMinder Web Agent installation provides form templates that you can customize. The sample forms are installed in the following locations:

- Windows: <installdir>\SiteMinder Web Agent\pw
- UNIX: <installdir>/siteminder/webagent/pw

The templates include the following:

Template	Description
PWAccountInactive.template	Informs users that their account has been disabled due to inactivity.

Template	Description
PWnnAccountInactive.template	Identical to PWAccountInactive.template, except that the login ID is not appended to password services redirects.
PWChange.template	Requires users to change their password before accessing the protected resource. Users must enter their old password, new password, then confirm the new password.
PWnnChange.template	Identical to PWChange.template, except that the login ID is not appended to password services redirects.
PWChangeAccept.template	Informs users that their password has been changed successfully.
PWChangeErr.template	Informs users that their password has not been changed because of an internal problem, such as a server overload, which caused the password change request to fail.
PWnnChangeErr.template	Identical to PWChangeErr.template, except that the login ID is not appended to password services redirects.
PWExcessLogin.template	Informs users that they have exceeded the permissible number of login attempts. Users must wait a specified amount of time before attempting to log in again or contact the administrator to reactivate their password.
PWnnExcessLogin.template	Identical to PWExcessLogin.template, except that the login ID is not appended to password services redirects.
PWChangeMismatch.template	Informs users that the password they selected is not valid and prompts them to select a new one.
PWnnChangeMismatch.template	Identical to PWChangeMismatch.template, except that the login ID is not appended to password services redirects.
PWChangeOption.template	Allows users whose passwords are about to expire to change their passwords now or be reminded later. Users can enter their old password, new password, and confirm a new password.

Template	Description
PWnnChangeOption.template	Identical to PWChangeOption.template, except that the login ID is not appended to password services redirects.
PWExpired.template	Informs users that their password has expired.
PWnnExpired.template	Identical to PWExpired.template, except that the login ID is not appended to password services redirects.
PWAccountDisabled.template	Informs users to contact an Administrator because their account has been disabled.
PWnnAccountDisabled.template	Identical to PWAccountDisabled.template, except that the login ID is not appended to password services redirects.
PWSelfChangeLogin.template	Allows users to change their passwords from a login page.

Modify CGI-based Password Services HTML Templates

You can customize the layout and formatting of Password Services HTML templates just as you would customize any other HTML document.

However, when you customize a Password Services template be careful *not* to alter:

- JavaScript functions at the beginning of the template file.
- <form> HTML tags. Password Services CGI uses the <form> output to pass information to the appropriate scripts for processing.
- Text that uses the *\$text\$* notation. The Password Services CGI uses the information held in variables denoted by \$ symbols.

Changes to any of these elements may cause Password Services to malfunction.

Password CGI-based Services Properties Files

Password Services properties files determine the text displayed on the HTML pages used to collect information from users and display status messages. This text includes:

- Field names
- Dialog titles
- Status messages and instructions for users
- Links and button text

To generate HTML pages, the Password Services CGI uses the Password Services templates. When the CGI encounters a parameter in the templates, it searches for a corresponding identifier defined in a Properties file. The Properties file pairs the identifier with a text string, which is added to the generated HTML page.

The following is a sample of the contents of a Password Services Properties file:

```
szPWNewPWChange = New Password
```

You can modify the text string to the right of the equal sign (=) to change the label that appears in the generated HTML pages; however, you should not modify the identifier on the left.

Note: Password Services requires that all characters in the properties files are UTF-8 encoded. Any characters that are not UTF-8 encoded (such as umlauts or tildes in the Windows or the 8-bit ASCII character sets) will not be processed correctly.

Password Services installs Properties files for the following languages:

- English (US-EN): PasswordServicesUS-EN.properties
- French (FR-FR): PasswordServicesFR-FR.properties
- Japanese (JP-JP): PasswordServicesJP-JP.properties

These files are located in the following directory:

- Windows: <web agent installation>\pw
- UNIX: <web agent installation>/pw

where <web agent installation> is the installed location of the SiteMinder Web Agent.

Localize CGI-based Password Services

The Password Services template files provide Unicode support for multiple languages. The default encoding is UTF-8, which enables users to enter a password in any UTF-8 supported language.

To localize the Password Services templates for French or Japanese, modify the sample French and Japanese Properties files described in [Password CGI-based Services Properties Files](#) (see page 500).

To configure your site to use localized Password Services, see [Overview of Servlet-based Password Services JSP Forms](#) (see page 502).

Create and Use a Localized Password Services Properties Files

Prior to 5.x QMR 2, the Web Agent passed on the value of the SMLOCALE variable so that the proper language encoding was set for Password Services change forms. This required you to append the variable SMLOCALE=<country>-<language> to the Target URL for the protected resource—for example:

`http://server.mysite.org/dir1/targetpage.html?SMLOCALE=FR-FR`

Beginning at 5.x QMR 2, the language encoding is read from the Web browser so no manual intervention is required. The Password Services CGI reads the ACCEPT_LANGUAGE variable from the Web browser to determine the language in which to display the password change forms.

For each language, the change forms included with the SiteMinder Web Agent use a different properties file. The properties file defines certain aspects of the text displayed on the HTML pages.

The following properties files are automatically installed with the SiteMinder Web Agent:

- English: PasswordServicesUS-EN.properties
- French: PasswordServicesFR-FR.properties
- Japanese: PasswordServicesJP-JP.properties

If the ACCEPT_LANGUAGE indicates a language for which no properties file is present the English properties file is used.

To create and use a properties file for another language

1. Translate one of the properties files provided by SiteMinder into the language of your choice.

For example, copy the PasswordServicesUS-EN.properties file and translate anything to the right of the equals sign (=).

2. Rename the file according to the convention for Password Services, which is:

PasswordServices<country>-<language>.properties

Note: RFC 3066 defines the language tags that you should use as part of the naming convention (<http://www.ietf.org/rfc/rfc3066.txt>).

Note: Be sure the browser is set for the correct language.

If you set multiple languages for the browser, Password Services will only use the first language in the list. If the Password Services CGI cannot find a properties file for that language, the English properties file is used. The other languages specified for the browser are ignored.

Servlet-based Password Services JSP Form Customization

The Password Services Servlet uses *JavaServer Page (JSP)* forms that allow users to change their passwords. You can customize these forms to create a branded look and modify the information provided on the forms.

Overview of Servlet-based Password Services JSP Forms

The Password Services servlet uses JSP forms to collect password information from users and send it to the Policy Server. The Policy Server checks the information against the password policy and then sends a message back to the Password Services servlet using another form.

The SiteMinder Web Agent installation provides form templates that you can customize. The sample JSP forms are installed in the following locations:

- Windows: <Web-Agent-Install-Dir>\jpw
- UNIX: <Web-Agent-Install-Dir>/jpw

Where <Web-Agent-Install-Dir> is the SiteMinder Web Agent installation directory.

The templates include the following:

Template	Description
PWAccountInactive.jsp	Informs users that their account has been disabled due to inactivity.
PWChange.jsp	Requires users to change their password before accessing the protected resource. Users must enter their old password, new password, then confirm the new password.

Template	Description
PWChangeAccept.jsp	Informs users that their password has been changed successfully.
PWChangeErr.jsp	Informs users that their password has not been changed because of an internal problem, such as a server overload, which caused the password change request to fail.
PWExcessLogin.jsp	Informs users that they have exceeded the permissible number of login attempts. Users must wait a specified amount of time before attempting to log in again or contact the administrator to reactivate their password.
PWChangeMismatch.jsp	Informs users that the password they selected is not valid and prompts them to select a new one.
PWChangeOption.jsp	Allows users whose passwords are about to expire to change their passwords now or be reminded later. Users can enter their old password, new password, and confirm a new password.
PWExpired.jsp	Informs users that their password has expired.
PWAccountDisabled.jsp	Informs users to contact an Administrator because their account has been disabled.
PWSelfChangeLogin.jsp	Allows users to change their passwords from a login page.
PWLogin.jsp	Allows a user to enter credentials and login. This .jsp file generates the form shown to the user for login if the redirect path in the HTML Forms Based authentication scheme to the servlet (i.e. /siteminderagent/pwservlet/PSWDChangeServlet).
UserMsg.jsp	UserMsg.jsp returns the messages depending on the reason of user rejection.
PWExpiredLogin.jsp	When a user's session expires, this .jsp displays a login screen to allow a user to establish a new session.

Template	Description
PSWDChangeError.jsp	This jsp informs users that their password has not been changed because of an internal problem, which caused the password change request to fail.
PWAuthLevelLogin.jsp	If a user with an established session attempts to access a resource that is protected at a higher level than the original resource requested at login, this .jsp presents a form that allows the user to authenticate again, gaining access to the more secure resource.
Resource.jsp	Specifies the content type as UTF-8 and indicates the resource bundle that contains localized strings for all password services forms.

Modify Servlet-based Password Services JSP Forms

You can customize the layout and formatting of Password Services JSP forms just as you would customize any other HTML document.

Do *not* alter the following:

JavaScript functions at the beginning of the template file.

<form> HTML tags

Policy Server uses <form> output to pass information to the Password Services servlet for processing

Parameters enclosed in <% %> notation

These <%text%> parameters are used by the Password Services servlet. Although you cannot change the existing ones, you can add your own <%text%> parameters or use Password Services-specific <%text%> parameters for your own purposes, if desired.

Different <%text%> parameters are used in different JSP template files.

Localize Servlet-based Password Services

Servlet-based password services uses a single set of template .jsp files that are populated at runtime using the strings/messages from an externalized set of properties files. The language used by the .jsp files is determined by the locale setting of a users' browser. If a browser is configured for multiple locales, the specified locales are compared, in order, and the first locale that matches a supported language for password services forms determines the language used for the forms displayed to users.

Note: If no language preference is specified in a user's browser, the default properties file, PasswordServices.properties will be used.

Modify Strings Displayed in Forms

In the .jsp files, the strings and messages represented by a tag in the following format:

```
<%=getI18NParameter (resourcebundle, TitleName)%>
```

For example, in the PWChange.jsp, the title "SiteMinder Password Services" is now replaced with the following tag:

```
<%=getI18Nparameter (resourcebundle, "PWChange.Title")%>
```

where resourcebundle is the object that contains the name-value pair strings that are loaded from the properties file and "PWChange.Title" is the key whose value will be retrieved from the properties file.

When a tag occurs in the .jsp file, it is replaced with the value of the corresponding identifier defined in a properties file.

The following is a sample of the format for contents of a Password Services properties file:

```
PWChange.Title = SiteMinder Password Services
```

The text string to the right of the equal sign can be modified to change the label that appears in forms displayed to users.

Add New Strings

To add a new string or message to a form, you must add a new tag to the .jsp file and corresponding entry in the associated properties file. For example, to add a new tag called NewTag to a .jsp file called PWScreen.jsp, you must do the following:

1. Add a new property to the properties file:
PWScreen.NewTag=NewValue
2. Add a new tag to the .jsp file:
<%=getI18NParameter (resourcebundle, "PWScreen.NewTag")%>

At runtime, the tag in the .jsp file is replaced with the string: NewValue.

Properties Files

SiteMinder includes the following properties files by default:

PasswordServices.properties

The default properties file, this file contains the key-value pairs for English (EN).

PasswordServicesJA.properties

This file contains the key-value pairs for Japanese (JA).

All characters in the properties files are UTF-8 encoded. If you create your own properties files or add to the existing files, the files must be UTF-8 encoded.

Properties must be named according to one the following conventions:

- PasswordServices<LANGUAGE>-<COUNTRY>.properties
- or
- PasswordServices<LANGUAGE>.properties

where <LANGUAGE> is the two letter code for a language (e.g. EN for English, FR for French, JA for Japanese), and <COUNTRY> is the two letter code for country (e.g. US for United States, JP for Japan). For example, the properties file for French in Canada would be named PasswordServicesFR-CA.properties.

Support a New Language

Using properties files, you can add additional language support.

To support a new language

1. Create a UTF-8 encoded properties file.
2. Name it in the required format.
3. Place it in the <Web-Agent-Install-Dir>\jpw directory, along with the .jsp files.

Set a Language Other Than English as the Default Language

If no language preference is specified in a user's browser, the default properties file, PasswordServices.properties will be used.

To set a language other than English as the default language

1. Rename the following file to some other name:
`<Web-Agent-Install-Dir>\jpw\PasswordServices.properties`
2. Create a new default properties file and name it:
`PasswordServices.properties`
3. Save the new PasswordServices.properties file in the following directory:
`<Web-Agent-Install-Dir>\jpw\`

Authentication Schemes Requiring Additional Attribute Information

To configure Password Services to work with Authentication Schemes requiring additional attributes you must add the following to Password Services' HTML forms .template files and the sample .jsp forms files before users submit any data to the Policy Server:

```
document.PWChange.OLDPASSWORD.value="PASSWORD="+escape(
document.PWChange.OLDPASSWORD.value)
+"&<attribute>="+escape(document.PWChange.<attribute>.value)
```

Password Services' .template files and the sample .jsp files are installed in the following locations:

- `<web agent installation>/pw`
- `<web agent installation>/jpw/default`
- `<web agent installation>/pwservlet/default`

As an example, the following shows how to modify the sample PWChange.jsp file (the text highlighted in bold are the new lines to add):

```
<html>
<head>
<title>SiteMinder Password Services</title>
...
...
function CheckForm(form)
{
...
...
else {
document.PWChange.OLDPASSWORD.value="PASSWORD="+escape(document.PWChange.OLDP
ASSWORD.value)+"&<attribute>="+escape(document.PWChange.<attribute>.value)
document.PWChange.submit()
return true
}
}
```

In another example, the following shows how to modify the sample PWChange.template file (the text highlighted in bold are the new lines to add):

```
<!-- SiteMinder Encoding=UTF-8; -->
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>$TITLE$</title>
<script LANGUAGE="JavaScript">
<!--
// Function for Validating Forms
function CheckForm(form)
{
...
...
else {
document.PWChange.OLDPASSWORD.value="PASSWORD="+escape(document.PWChange.OLDP
ASSWORD.value)+"&<attribute>="+escape(document.PWChange.<attribute>.value)
document.PWChange.submit()
return true
}
}
```

The Password Services Servlet provides Unicode support for multiple languages. The default encoding is UTF-8, which enables users to enter a password in any UTF-8 supported language.

The Password Services Servlet uses the HTTP protocol standard ACCEPT-LANGUAGE request header set by the browser to determine user language preferences. If the servlet can obtain the value of ACCEPT-LANGUAGE (which represents a standard language encoding, such as EN-US, or FR), it looks for appropriately localized JSP forms in a correspondingly named directory:

- NT: <Web-Agent-Install-Dir>\jpw\<encoding>
- UNIX: <Web-Agent-Install-Dir>/jpw/<encoding>

Where <Web-Agent-Install-Dir> is the root Netegrity installation directory and <encoding> is a standard language encoding.

As installed, Password Services provides a set of sample American English JSP files in the jpw directory. Password Services uses the contents of this folder whenever it cannot find a JSP directory corresponding to the ACCEPT-LANGUAGE value or it cannot obtain an ACCEPT-LANGUAGE value.

To localize servlet-based Password Services for another language, you must therefore create a new, appropriately named directory each containing a *complete* set of translated JSP forms.

Example localized JSP forms directory names:

Language	Folder Location
English (American)	EN
English (British)	EN-GB
French	FR
German	DE
Japanese	JP

If you want to localize for a different default language, you should rename the installed default directory to EN and create a new default directory containing a complete set of JSP forms translated into the new default language.

Enhanced Active Directory Integration with Password Services

Enhanced integration of Active Directory (AD) user stores with Password Services is a SiteMinder feature that synchronizes user attributes specific to Active Directory with the standard LDAP user attributes used by SiteMinder. When you configure a connection from the Policy Server to an Active Directory user store, enable enhanced integration by selecting Global Settings... in the Tools menu and the Enhance Active Directory Integration check box on the Global Settings dialog. For detailed information on how to enable enhanced integration, see the *Policy Server Management Guide*.

Enhanced Active Directory Integration Considerations

This topic covers factors to consider when enabling enhanced Active Directory integration.

When to Use Enhanced Active Directory Integration

Use Enhanced Active Directory Integration in the following case:

- The Policy Server needs access to an Active Directory user store that stores domain users for a Windows domain.
- You want Policy Server authentication and Windows domain authentication to operate simultaneously.
- You want Windows security policies to apply to the Active Directory user store.

User Attribute Mapping

To integrate Windows Password Policy with SiteMinder Password Services, map Active Directory user attributes to SiteMinder user attributes on the User Attributes tab on the User Directory Properties pane in the Policy Server User Interface. Without user attribute mapping, however, you can still integrate Active Directory with SiteMinder user account management.

Windows Security Policies

When Enhanced Active Directory Integration is selected, the Policy Server can read and enforce the Windows domain `lockoutDuration` and `maxPwdAge` settings. While the Policy Server cannot read the Windows domain controller's local security settings, the settings do affect the Policy Server.

How Windows Password Policy and SiteMinder Password Services Work Together

When users log in through Windows, only Windows Password Policy takes effect. When users log in through the Policy Server, Windows works together with the Policy Server to provide Password Services. In this case, the Policy Server enforces the maxPwdAge setting. When users change passwords, the Policy Server validates the new password, and the Windows domain controller authorizes or denies the new password.

When the Policy Server Does Not Recognize Users Locked Out by Windows

If the Policy Server does not recognize users locked out by Windows, check the following settings:

1. Verify that the Enhance Active Directory Integration check box is selected on the Global Settings dialog, which can be accessed through the Tools menu in the Policy Server User Interface.
2. Verify that the default Windows domain DN is typed in the Root field on the User Directory Properties pane corresponding to the Active Directory.

Example: dc=WindowsDomain,dc=com

3. If account lockout duration is defined on the Policy Server, verify that the setting is the same value as the lockoutDuration setting in the Windows domain.

User Account Management with Active Directory Integration

When enhanced Active Directory integration is enabled, the Policy Server manages Active Directory user accounts as follows:

- The Policy Server disables the accounts of users who are locked out or disabled by the Windows Network Operating System (NOS) and lists disabled accounts as "Disabled - administrative" in the Current Settings group box on the User Management Dialog in the Policy Server User Interface.
- You can unlock these disabled accounts in Windows by enabling them in the Current Settings group box on the User Management Dialog in the Policy Server User Interface.
- User accounts that expire from inactivity enter a native disabled state in the Policy Server that can only be changed by the Windows domain controller.
- The accounts of users whose passwords have expired enter a password-must-change-at-next-login state in the Policy Server.
- You can clear the password-must-change-at-next-login state for users in the Password group box on the User Management Dialog in the Policy Server User Interface, or users can change their own passwords. In both cases, the Windows NOS does not prompt the user for a password change at the next login.

- When the Windows NOS enables the Account Policies/Password Policy defined in the Default Domain Policy, the accounts of users whose passwords have expired enter a password-must-change-at-next-login state in the Policy Server.

Note: For more information about managing user accounts, see the *Policy Server Management Guide*.

Password Services with Active Directory Integration

When enhanced Active Directory integration is enabled, the Policy Server manages Password Services as follows:

- Users who are locked out or disabled by the Windows Network Operating System (NOS) or whose accounts are expired in the Windows NOS are redirected by the Policy Server to the "User Is Disabled" page at login.
- Users who must change their passwords at the next login or whose passwords are expired by the Account Policies/Password Policy defined in the Default Domain Policy of the Windows NOS are redirected by the Policy Server to the password-change page after a successful login.
- User account activities, such as login successes and failures, password changes, and user enabling and disabling, in the Windows domain are recognized by Password Services and used by the Policy Server to evaluate Password Services policies.
- User account activities cause the Policy Server to update user attributes in the Active Directory user store. The Windows domain, in turn, recognizes these attribute changes.
- If the Password Services policies configured in the Policy Server determine that the user must change the password, both Windows NOS and the Policy Server prompt the user for a password change at the next login.
- If the Password Services policies configured in the Policy Server determine that the user must be disabled, the Policy Server redirects the user to the "User Is Disabled" page at the next login.

Password Policy Troubleshooting

The following sections describe and provide solutions to problems that may occur when implementing password policies.

New User Passwords are Rejected

Symptom:

User-specified passwords are always rejected.

Solution:

The password policy may be too strict or improperly configured. Check the content minimums and the password length composition settings for consistency.

User Accounts are Mistakenly Disabled

Symptom:

Users accounts that have not exceeded the number of permitted failed login attempts are becoming disabled.

Solution:

Check the incorrect password settings. The setting for disabling an account after a specific number of consecutive incorrect password attempts may be too low.

Setting this value too low causes a problem when two or more users, which are located in different user directories, have the same user name. When the Policy Server attempts to authorize a user, it checks all user names that correspond to the login and then attempts to match the password. If the Policy Server finds a user name that the password does not match, it records a failed attempt for that user. If this happens more than the number of times specified by the in the incorrect password settings, the account is disabled.

User Accounts are Prematurely Disabled

Symptom:

User accounts are prematurely disabled in a multi-Policy Server environment.

Solution

Check that there is no time differential between the Policy servers.

Password Changes are Forced

Symptom:

User accounts are forced to changed passwords too soon in a multi-Policy Server environment.

Solution:

Check that there is no time differential between the Policy servers.

LDAP Users Do Not Disable

Symptom:

Password policies do not disable LDAP users.

Solution:

Check the following:

- The LDAP directory to which the policy is bound is writable. You must be able to save passwords in each user's record in the LDAP directory.
- The user directory setting is valid for password attribute, password data, and disable flag user profile attributes. More information on configuring the password attribute, password data, and disabled flag user profile attributes exists in Specify Directory Attributes.

Active Directory Users Cannot Change Passwords

Symptom:

Users stored in Active Directory user directories cannot change their passwords.

Solution:

Check the following:

- The Active Directory user directory to which the policy is bound is configured with a secure (SSL) connection.
- The Active Directory user directory to which the policy is bound is configured to use the unicodePWD Password Attribute.

Incorrect Password Message Does Not Appear

Symptom:

When a user submits a password change request that contains an invalid current password, the Password Change Information screen does not open with a message stating that the current password is incorrect. Rather, the Policy Server redirects the user to:

- The login screen without the message if an On-Auth-Reject-Redirect response is not bound to the policy configured with the user directory
- The URL associated with the On-Auth-Reject-Redirect response bound to the policy configured with the user directory

Solution:

Enable the DisallowForceLogin registry key, which is located at `HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\PolicyServer`.

DisallowForceLogin

Redirects users to the Password Change Information screen to re-enter the current password when the change request contains an invalid current password.

KeyType: REG_DWORD

Value: 0 (disabled) or 1 (enabled)

Default: 0 (disabled)

Note: If you specify a value other than 0 or 1, the key is disabled and the Policy Server will redirect users to the login screen without the invalid current password message.

Chapter 22: SiteMinder Test Tool

This section contains the following topics:

[Test Tool Overview](#) (see page 517)

[Configure Your Test Environment Agent](#) (see page 518)

[Run a Functionality Test](#) (see page 523)

[Perform a Regression Test](#) (see page 527)

[Stress Tests](#) (see page 528)

[Certificate-based Authentication Tests](#) (see page 531)

Test Tool Overview

The SiteMinder Test Tool is a utility that simulates the interaction between Agents and Policy Servers. It tests the functionality of the Policy Server. During testing, the Test Tool acts as the Agent, making the same requests to the Policy Server as a real Agent. This allows you to test your SiteMinder configuration before deploying it.

The SiteMinder Test Tool is only available on Windows systems. It can be used to emulate Windows-based Web Agents and RADIUS Agents without limitations. The SiteMinder Test Tool can create connections with any Policy Server on all platforms; however, post 4.x Web Agents on Solaris cannot be tested with the SiteMinder Test Tool. To test policies for SiteMinder 5.x and later Web Agents, do one of the following:

- Use the perl scripting interface for Web Agent registration and testing.
- Install and configure the SiteMinder Web Agent and test it manually.

The SiteMinder Test Tool performs the following types of tests:

Functionality

Tests policies to ensure they are configured correctly.

Regression

Tests whether or not changes, such as migrating a policy store or implementing a new feature, affect SiteMinder.

Stress

Tests the performance of the Policy Server as it receives multiple requests.

Note: You can also test policies using the Command Line Interface. See the *SiteMinder Scripting Guide for Perl* for more information.

Configure Your Test Environment Agent

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

The Agent that the Test Tool simulates must be configured in the Policy Server User Interface.

To configure Agent information, specify the following options:

Agent Type

Specify one of the following Agent Types:

Version 4

Simulates SiteMinder 4.x Agents.

Version 5

Simulates SiteMinder 5.x Agents.

Note: If you want to use the Test Tool on a system to simulate a SiteMinder 5.x Web Agent, you must run the smreghost.exe application on the system where you will run the Test Tool. The smreghost.exe file is included with your Web Agent, and described in the *SiteMinder Web Agent Installation Guide*. The file is also located in <Policy Server install dir>/siteminder/bin.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

RADIUS

Simulates RADIUS devices.

Agent Name

Enter the name of the Agent as it appears in the Policy Server User Interface. This field is required for both Version 4 and Version 5 Agent simulations.

Secret

Enter the Agent's shared secret. This must match the shared secret entered when the Agent was created. A Secret is required for Version 4 and RADIUS Agent simulations.

(Optional) Server

Enter the full name of the server on which the Agent resides. For example, to test the Policy Server for <http://www.myorg.org>, enter www.myorg.org in this field. This field may be used for Version 4 Agent simulations.

SmHost.conf Path

Enter the path to the SmHost.conf file that contains the settings for the Version 5 Agent you want to simulate. You can use the Browse button to search for the SmHost.conf file.

Policy Server Identification

The test tool requires information about the Policy Server that will be used when simulating the interaction with the Agent described in the SiteMinder Agent group box. The required information differs slightly depending on the type of Agent you selected.

Set Up the Policy Server for Version 4 Agents and RADIUS Agent Simulations

For Version 4 Agents and RADIUS Agent simulations, you must specify the IP address and port information of the Policy Server(s) used in the test. If you want to simulate a multiple Policy Server environment, you can specify how those Policy Servers operate.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

To set up Policy Server(s) for Version 4 Agent and RADIUS Agent simulations

1. Specify the following Policy Server options, as necessary:

Policy Server

Indicates whether you are specifying the primary or secondary Policy Server.

IP Address

Specifies the IP address of the Policy Server. By default, this field contains the IP address of the local system.

Authorization, Authentication, and Accounting Ports

Specifies the TCP ports used for authorization, authentication, and accounting requests. These fields are populated with the Policy Server's default port numbers.

Timeout

Displays the time (in seconds) that the Test Tool should wait for a response from the Policy Server.

2. Select one of the following operation modes:

Failover

Enables failover. During failover, the Test Tool directs requests to the initial Policy Server. If the initial Policy Server fails, the Test Tool redirects requests to the secondary Policy Server.

Round Robin

Enables round robin load balancing. Round robin load balancing divides requests between the primary and secondary Policy Servers. For each connection, the Test Tool alternates between Policy Servers.

3. Click Connect to make sure that the Test Tool can connect to the Policy Server.

If the Test Tool makes a connection, the IsProtected and DoManagement stop lights turn green.

Note: You must specify an Agent before testing the Policy Server connection.

Policy Server Information for Version 5 Agents

For Version 5 Agents simulations, you may specify the IP address and port information of the Policy Server(s) used in the test, or you may use the Policy Server information contained in the Host Configuration Object contained in the policy store.

By default, the Policy Server information will be retrieved from the policy store when the Test Tool uses the SmHost.conf file to establish an initial connection to the Policy Server. To specify Policy Server information manually, select the Override check box and fill in Policy Server information as described in [Set Up the Policy Server for Version 4 Agents and RADIUS Agent Simulations](#) (see page 519).

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

Select a Test Mode

You can use one of the test modes in the following table to determine how tests are run and results are displayed. Depending on the test mode that you select, you may also have to specify script information.

Mode	Description
Interactive	Allows you to enter data, run tests, and see the results displayed immediately in the Server Response section.

Mode	Description
Record	Combines Interactive operation with a script generation feature that writes test results to a plain-text script file. This file can subsequently be used as an input file to repeat the test in playback mode. You can run multiple tests and record them to the same script file. The Test Tool appends the test results to the end of the file. You can then use the script file for regression and stress testing. Optionally, you can add a comment to the output script file by entering it in the Comment field before running a test. To stop recording, specify a new mode. Note: When you select Record, you must enter the path and filename for the text file where the test results will be stored in Output Script field.
Basic Playback	Uses script files created in the Record mode to automate sequential tests for regression testing. When you select Basic Playback, the only active fields in the Test Tool are the Input Script and Output Script fields.
Advanced Playback	Runs multi-threaded stress tests. When you select Advanced Playback, the only active field in the Test Tool is the Control Script field.

Specify Resource Information

You can specify the resource against which you want to conduct tests. Providing a resource simulates a user entering a URL in a browser.

To specify resource information, provide values for the following options

Resource

Enter the relative path of the resource that SiteMinder is protecting as it is configured in the realm. The path is relative to the Web server's publishing directory. For example, /protected/.

Action

Enter the Agent action, Authentication event, or Authorization event specified in the rule that you are testing.

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

Specify User Credentials

The Test Tool requires user credentials to test whether or not a policy can authenticate or authorize a user.

To specify user credentials, complete the following fields:

User Name

Enter the user name you want to use to access the resource.

Password

Enter the password for the user entered in User Name.

CHAP Password

If you are using a RADIUS CHAP authentication scheme, select this check box.

Certificate File

If the protected resource requires certificates to authenticate users, you must provide a certificate file so that the Test Tool can simulate certificate authentication.

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

Set the Encoding Spec

The encoding spec field allows you to specify a language encoding parameter. The Test Tool uses this parameter to encode headers in the same manner as a Web Agent. It then displays the encoded response attribute data in the Attributes field.

For more information about language encoding, see the *SiteMinder Web Agent Guide*.

To set the encoding spec, enter a value for the encoding spec as follows:

encoding_spec, wrapping_spec

where:

- *encoding_spec* is a text string that represents one of the following encoding types: UTF-8, Shift-JIS, EUC-J, or ISO-2022 JP
- *wrapping_spec* is the wrapping specification, which must be RFC-2047.

Note: If you leave this field blank, the default is UTF-8 with no wrapping.

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

Save and Load Configurations

To avoid re-entering user-supplied information, such as Agent, resource, and user information, you can save and load settings.

Use the Save Settings button to save the current settings to a plain text .ini file. When you click Save Settings, the Test Tool prompts you to enter a file name.

To retrieve saved information, click Load Settings, select the file, and click Open. The user-configured fields are automatically filled in with the values defined in the file.

You can configure the Agent that the Test Tool simulates during a test in the SiteMinder Agent group box.

Run a Functionality Test

The SiteMinder Test Tool allows you to test the functionality of policies in a simulated real-world environment. To perform a functionality test, you must have the following:

- A SiteMinder Policy Server that is configured and running
 - A SiteMinder Agent that is configured in the Policy Server User Interface
- Note:** If the Test Tool is simulating a SiteMinder Agent v5.x, that Agent must have 4.x support enabled.
- A policy domain configured with any type of user directory
 - A policy that pairs a rule with a user(s)

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

SiteMinder allows you to perform the following functionality tests:

IsProtected

Indicates whether or not a policy is protecting the resource you specified.

IsAuthenticated

Indicates whether or not the Policy Server can authenticate a set of user credentials against a user directory.

When user credentials are authenticated, the Policy Server compares the credentials to entries in a user directory. If the credentials match an entry, the Policy Server creates a session ticket and authenticates the user.

In a "real" SiteMinder deployment, SiteMinder confirms that a user's session ticket is valid instead of rechecking the user's credentials against a directory when an authenticated user makes additional requests. By default, the Test Tool authenticates the user each time the `IsAuthenticated` test is run, regardless of whether or not the user has a session ticket.

You can configure the Test Tool to validate a user's session ticket by entering `Validate` in the Comment field in the Test Tool before running an `IsAuthenticated` test; however, SiteMinder must authenticate the user before validating the session ticket.

Note: You can specify `Validate` when you run multiple tests in Interactive mode (using the Repeat count field), and in Playback mode.

IsAuthorized

Indicates whether or not the Policy Server can authorize a user based on a policy.

These tests must be run in the order they appear above. For example, you must run `IsProtected` before running `IsAuthenticated`. The order reflects the steps that SiteMinder uses to determine a user's access rights.

While running functionality tests, you can also use the Test Tool to perform the following tasks:

DoAccounting

Logs the most recent accounting server transactions.

DoManagement

Requests Agent commands, such as cache flush commands that clear the Agent cache. Running `DoManagement` ensures that the Test Tool receives current information from the Policy Server.

To run a functionality test

1. Configure a test environment.

Note: You can also test policies using the Scripting Interface. See the *SiteMinder Scripting Guide for Perl*.

2. (Optional) Specify the number of times you want the Test Tool to run your test in the Repeat Count field in the Command group box.
3. In the Command group box, select one of the following tests to run:
 - `IsProtected`
 - `IsAuthenticated`
 - `IsAuthorized`
4. If you are running an `IsAuthenticated` test and you want the Test Tool to validate an authenticated user's session ticket instead of authenticating the user's credentials against a user directory, enter `Validate` in the Comment field.

Note: Before a user's session ticket is validated, the user must be authenticated. Once the user is authenticated, SiteMinder creates a session ticket for the user.

More information:

[Configure Your Test Environment Agent](#) (see page 518)

[Calculate an Average Elapsed Time](#) (see page 527)

Functionality Test Results

The tables in this section describe the results of each type of functionality test.

If isProtected...	Then...
Succeeds	The Test Tool displays Protected in the Message field. This means that the Test Tool made a successful connection to the Policy Server and a policy is protecting the resource. The Test Tool also populates the following fields with values returned by the Policy Server: Realm Name Name of the realm that contains the resource Realm OID The realm object identifier Credentials The authentication scheme used to protect the resource Redirect The redirect string used by the authentication scheme, if one is specified. All certificate and HTML forms-based schemes return this string, which typically instructs the Agent where to display a form.
Fails	The Test Tool displays Error or Not Protected in the Message field. Error indicates that the Test Tool could not connect to the Policy Server; Not Protected indicates that the specified resource is not protected by a policy. If the test fails: Make sure that the policy is configured correctly. Check the Authentication server log for debugging information.

If isAuthenticated...	Then...								
Succeeds	The Test Tool displays Authenticated in the Message field and populates the following fields with values returned by the Policy Server: Session ID A unique SiteMinder-assigned session ID. The Policy Server uses this ID to identify the cookie where session information is stored. Attributes The attributes the Policy Server sends back in the response. For example: <table><tr><td>ID</td><td>Length</td><td>ASCII Format</td><td>Hexidecimal format</td></tr><tr><td>4> id</td><td>215</td><td>len 005</td><td>'LDAP:' - '4c 44 41 50 3a'</td></tr></table> The response indicates the name of the user directory where the user was authenticated. Note: Click Reset to clear responses displayed in the Attributes field without removing user-supplied information. Reason The reason code associated with the outcome of the test. This field is used to supply information to developers using the SiteMinder SDK. Reason codes are listed in SmApi.h.	ID	Length	ASCII Format	Hexidecimal format	4> id	215	len 005	'LDAP:' - '4c 44 41 50 3a'
ID	Length	ASCII Format	Hexidecimal format						
4> id	215	len 005	'LDAP:' - '4c 44 41 50 3a'						
Fails	The Test Tool displays Not Authenticated in the Message field. If the test fails: Make sure that you are using valid user credentials. Check the Authentication server log for debugging information.								
If IsAuthorized...	Then...								
Succeeds	The Test Tool displays Authorized in the Message field and and the SiteMinder-assigned Session ID in the Session ID field. This ID identifies the cookie where session information is stored.								
Fails	The Test Tool displays Not Authorized in the Message field. If the test fails: Make sure that the policy is configured correctly. Check the Authorization server log for debugging information.								

Calculate an Average Elapsed Time

After performing a test, the Test Tool displays the amount of time the test took to run in the Elapsed Time field of the Command group box. Because of fluctuations in the system, averaging the elapsed time of multiple tests provides more accurate results.

To get an average elapsed time

1. In the Repeat Count field, specify the number of times you want to run the test.
The Test Tool runs the test the specified number of times and then displays the total elapsed time.
2. Divide the elapsed time by the number of times the test was run to determine the average elapsed time.

Perform a Regression Test

Regression tests allow you to test whether or not changes made to SiteMinder, such as upgrading the policy store or implementing a new feature, affect policies. To run a regression test, you run one test with the current environment, make changes, then run the test again. By comparing the results of the tests, you can determine if the changes affect SiteMinder.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

To perform a regression test

1. Run functionality tests in Record mode. Optionally, enter a comment in the Comment field. The comment appears in the output file.
2. When the test is complete, select Basic Playback in the Mode group box.
3. In the Script Information group box, enter the name of the text file in the Input Script field.
This file name should match the name of the Output Script file you created in Record mode.
4. In the Output Script field, specify an output file name.
5. In the Command group box, click Run Script.
The Test Tool runs the input script and creates the output script file.
6. Compare the input and output script files.

Stress Tests

The Test Tool allows you to test SiteMinder's performance when the Policy Server receives more than one request at a time. Using stress tests, you can simulate multiple Agents talking to the Policy Server simultaneously or a single Agent communicating with the Policy Server on multiple threads.

Stress tests are run in Advanced Playback mode. The Test Tool receives instructions from a thread control file that specifies which tests to run and how many times to run them. After executing the instructions in the thread control file, the results of the test are written to an output file.

Important! Before running a SiteMinder utility or executable on Windows Server 2008, open the command line window with Administrator permissions. Open the command line window this way, even if your account has Administrator privileges. For more information, see the release notes for your SiteMinder component.

More information:

[Thread Control Files](#) (see page 528)

Thread Control Files

A thread control file determines the number of repetitions and threads the Test Tool runs in the Advanced Playback mode. It contains multiple instruction lines in the Test Tool's own scripting language and comments, indicated by the # symbol at the beginning of a line.

The basic instructions are in the format: *<script file name>, <number of script repetitions>, <number of threads>.*

For example:

```
# c:\temp\test_data.txt, 8, 6
```

This line indicates that:

- The input script is c:\temp\test_data.txt
- The Test Tool will run the script eight times
- There will be six simultaneous threads running the script

The Test Tool writes the output of the test to a file. The output file name is the name of the input file with _out# appended to it, where # is the incremented thread number. For example, the output files for the test above are c:\temp\test_data.txt_out1 to c:\temp\test_data.txt_out6.

The Test Tool scripting language includes the commands in the following table to control the script file output. See the following figure for a sample thread control file.

Command	Description
.report	Generates a final report (as an output file) summarizing the test results. The report does not include the status of each server request. This is the default.
.output	Generates a final report (as an output file) summarizing the overall results including the status of each server request.
.viewstats	Displays final statistics in a text editor.
.verbose	Generates output files containing the details of each server request.
.brief	Generates output files containing the brief results of each server request. This option is only valid when used with the .output command.
.sleep	Lets the Test Tool pause for a specified amount of time (in milliseconds). This simulates intermittent server requests.
.connect <setting_file>	Initializes the Test Tool using information from the settings file to set up a multi-threaded test with one simulated Agent. The default multi-threaded test comprises multiple simulated Agents with one simulated Agent per thread. You can also set up a test with one simulated Agent and multiple threads by using this option.
.disconnect	Un-initializes the Test Tool to indicate the end of one simulated Agent multi-threaded test.

```
.output
.brief
c:\temp\test_data1.txt, 2, 3
.verbose
.sleep 5000
c:\temp\test_data1.txt, 2, 2
.brief
c:\temp\test_data1.txt, 3, 4
.connect smtest.ini
c:\temp\test_data1.txt, 5, 6
.disconnect
```

Report Viewing

When you run a stress test, the Test Tool generates a report summarizing the results. This report contains the following information:

- Time the test started and finished
- Total elapsed time
- Minimum, maximum, and average request time
- Total number of requests
- Throughput
- The number of tests run and their results

The report is saved in the directory where the thread control file is located. The name of the report is the name of the thread control file with `_stats` appended to it. For example, the thread control file, `thread.txt`, yields a report named `thread.txt_stats`.

Control File: C:\temp\control.txt

Started at: 0:02:05.481

Finished at: 0:03:22.672

Total Elapsed: 0:00:01.396

Minimum Request Time: 0:00:00.400

Maximum Request Time: 0:00:05.498

Average Request Time: 0:00:01.396

Total Requests 234

Throughput (Req/Sec): 3.241

Request	Count	Yes	No	Timeout	Error
IsProtected	78	72	0	0	6
IsAuthenticated	78	78	0	0	0
IsAuthorized	78	72	0	6	0
Total:	234	222	0	6	6

Certificate-based Authentication Tests

The SiteMinder Test Tool simulates user authentication and authorization. Certificate-based authentication schemes require additional configuration.

Different certificate-generation tools sometimes affect the format of the issuer DN and other attributes of a certificate.

For example, the Issuer DN for a certificate generated with certutil.exe on an IIS web server could use ST= to represent the state. However, the Issuer DN for a certificate generated with OpenSSL tools on an Oracle iPlanet web server could possibly use S= to represent the state.

Note: For more information about the actual values used by a specific certificate-generation tool, see the documentation provided by the vendor of your certificate-generation tool.

To test certificate-based authentication schemes, configure certificate mappings in the Policy Server to accommodate certificates created with different certificate-generation tools.

Certificate Attributes that Require Custom Mappings

Some common certificate attributes differ slightly according to the third-party tool (such as certutil.exe or OpenSSL) used to generate the certificate. Differences between the following attributes could possibly cause errors in the Test Tool:

Email Address

Represented by E or Email depending on the vendor of the certificate-generation tool.

US State

Represented by S or ST depending on the vendor of the certificate-generation tool.

User ID Number

Represented by UID or UserID depending on the vendor of the certificate-generation tool.

Note: For more information about the actual values used by a specific certificate-generation tool, see the documentation provided by the vendor of your certificate-generation tool.

Custom Attribute Mappings for Testing

Using the test tool for a certificate authentication scheme sometimes fails, even if it works typically (through a browser and the web server). The authentication log shows that the test tool expects an attribute of the issuer DN appears from the Issuer DN in the certificate.

This situation occurs when the issuer DN and other attributes differ according to the type of certificate-generation tool used. For example, The certutil.exe program on an IIS web server could possibly use ST= to abbreviate the name of the state in the issuer DN. The OpenSSL tools on an Oracle iPlanet web server, however, could possibly use S= to abbreviate the name of the state.

Note: For more information about the actual values used by a specific certificate-generation tool, see the documentation provided by the vendor of your certificate-generation tool.

The situation is similar for the other attributes listed in [Certificate Attributes that Require Custom Mappings](#) (see page 531).

To resolve this problem, have an administrator create mappings for each Issuer DN format in the Policy Sever. Then, the Policy Sever can accept the Issuer DN formats created by different certificate-generation tools.

Issuer DN Mapping

Different certificate-generation tools (such as certutil.exe and OpenSSL) create the Issuer DN in slightly different ways. For example, one tool could possibly create an Issuer DN like the following:

```
CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, S=Western Cape, C=ZA
```

Another tool could possibly create an Issuer DN like the following:

```
CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, ST=Western Cape, C=ZA
```

To support both possibilities, have your administrator create mappings in the Policy Server for both Issuer DN formats.

Note: For more information about the actual values used by a specific certificate-generation tool, see the documentation provided by the vendor of your certificate-generation tool.

Create Custom Certificate Mappings

You can use the certificate mapping feature of the SiteMinder Policy Server to create custom mappings for certificates.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create and use a custom attribute in a certificate mapping

1. Open the Policy Server User Interface.
2. From the menu bar, select Advanced, Certificate Mapping.
The Certificate Mappings dialog box opens.
3. To add a new mapping, click Add. To modify an existing mapping, select the mapping and click Edit.

The Certificate Mapping Properties dialog opens.

4. In the Issuer DN field, enter the full issuer DN.
5. In the Mapping group box, select the custom Custom radio button.
6. Enter an expression similar to the following for the attributes mentioned in [Certificate Attributes that Require Custom Mappings](#) (see page 531).

Note: This notation specifies two different attributes that are acceptable for a certificate mapping.

- For Email:
 %{E/Email}
- For ST:
 %{S/ST}
- For User Id:
 %{UID/UserID}

7. Click OK to save your changes and close the Certificate Mapping Properties dialog.

The Policy Server should now handle requests from different certificate-generation tools (such as certutil.exe and OpenSSL) and the SiteMinder Test tool where the Email attribute is represented differently in the Issuer DN. You can use this process for any of the other attributes mentioned in [Certificate Attributes that Require Custom Mappings](#) (see page 531).

More Information:

[Certificate Mapping](#) (see page 293)

Chapter 23: Agent and Related Dialogs Reference

This section contains the following topics:

[Agent Dialog](#) (see page 535)
[Host Configuration Object Dialog](#) (see page 538)
[Agent Configuration Object Dialog](#) (see page 540)
[Agent Group Dialog](#) (see page 541)
[Edit Parameter Dialog](#) (see page 543)
[Cluster Setup Dialog](#) (see page 544)
[Agent Group Items Dialog](#) (see page 544)

Agent Dialog

The SiteMinder Agent Dialog is where you create an Agent object to establish a Web Agent identity.

Agent Object Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.

Decide how you want to configure a Web Agent using one of the following two methods:

Centrally

Using an Agent Configuration Object at the Policy Server to define the Agent's configuration.

Locally

Using a file that is installed on the system with the web server to define the Agent's configuration.

For additional prerequisites, see the *SiteMinder Web Agent Guide*.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Agent Dialog

To navigate to the Agent Dialog, from the System tab of the Policy Server User Interface, select Edit, System Configuration, Create Agent.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Agent Dialog Fields and Controls

Name field

Name for a Web Agent.

Description field

Description for a Web Agent.

Support 4.x agents check box

Provide an Agent object for a 4.x Web Agent identity. If you check this box, see the next section.

Agent Type group box

Provides the options to choose Web, Radius, Affiliate, EJB, or Servlet Agents.

Agent Dialog—Support 4.x Agents Checked

If you are configuring an Agent object for a 4.x Web Agent identity, check Support 4.x agents, and the following additional items appear in the Agent Dialog:

IP Address or Host Name field

IP address or the host name of the server on which the Agent resides.

Note: For virtual servers, define virtual server names and IP addresses as you would define a single server. Each Agent on a virtual server must have a unique Agent Name.

DNS Lookup button

Allows you to enter a host name to find an IP address.

Secret field

Specify an alphanumeric shared secret to the Agent when the Agent was installed on the Web server. The shared secret is used for mutual authentication when a 4.x Agent attempts to connect to a Policy Server. The secret must be between 1 and 255 characters in length and contain no embedded spaces.

Confirm Secret field

Re-enter the shared secret.

Agent Dialog—RADIUS Radio Button Selected

Name field

Name for a RADIUS Agent.

Description field

Description for a RADIUS Agent.

Agent Type group box

The drop-down list shows the vendor type of the RADIUS device the Agent will protect.

Note: Generic RADIUS can be used to protect any type of RADIUS device, however, Generic RADIUS Agent types do not provide access to vendor-specific response attributes.

IP Address field

IP address or the host name of the RADIUS client (NAS device).

Shared Secret field

Alphanumeric shared secret used by the NAS device.

The shared secret is used for mutual authentication between the NAS device and the Policy Server. The secret must be between 1 and 255 characters in length and contain no embedded spaces.

Confirm Secret field

The shared secret re-entered.

Realm Hint Attribute field

Needed if a RADIUS Agent protects a NAS device that must authenticate users in different domains, such as mydomain.com and yourdomain.com. A realm hint is a RADIUS attribute that provides SiteMinder with information about the domain in which to authenticate the user.

For more information about realm hints, see the *SiteMinder Policy Server Management Guide*.

Tasks Related to the Agent Dialog

The following tasks are related to the Agent Dialog:

- [Create an Agent Object to Establish a Web Agent Identity](#) (see page 123)
- [Configure an Agent Object for a 4.x Web Agent Identity](#) (see page 126)
- [Set the Configuration Parameters in the Agent Configuration Object](#) (see page 126)

Host Configuration Object Dialog

The SiteMinder Host Configuration Object Dialog is where you configure a Host Configuration Object for the Web Agent.

Host Configuration Object Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.

For additional prerequisites, see the *SiteMinder Web Agent Guide*.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Host Configuration Object Dialog

To navigate to the Host Configuration Object Dialog, from the System tab of the Policy Server User Interface, select Edit, System Configuration, Create Host Conf Object.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Host Configuration Object Dialog Fields and Controls

Name field

Name for a Host Configuration Object.

Description field

Description for a Host Configuration Object.

General tab

Specifies the configures values for the Host Configuration Object.

Clusters tab

Specifies the available clusters for the Host Configuration Object.

More information:

[Host Configuration Object Dialog—General Tab](#) (see page 539)

[Host Configuration Object Dialog—Clusters Tab](#) (see page 539)

Host Configuration Object Dialog—General Tab

Configuration Values list

List configuration values for the Host Configuration Object.

Add button

Opens the Edit Parameter Dialog from where you can create a new configuration value to add to the Configuration Values list.

Delete button

Removes a selected entry from the Configuration Values list.

Edit button

Opens an entry selected entry from the Configuration Values list for editing in the Edit Parameter Dialog.

More information:

[Edit Parameter Dialog](#) (see page 543)

Host Configuration Object Dialog—Clusters Tab

Failover Threshold Percent field

Specifies a percentage of Policy Servers that must be active and click Apply.

If the percentage of active servers in the cluster falls below the percentage you specify, the cluster fails over to the next available cluster in the list of clusters.

The Policy Server User Interface automatically calculates the Failover Threshold value displayed in the column to the right of the lists of servers in each cluster from the Failover Threshold Percent you specify by taking the specified percentage of servers in the cluster and rounding up to the nearest integer. For example, if the number of servers is 5 and the Failover Threshold Percent is in the range between [0 – 20], the Failover Threshold is 1.

When you set the Failover Threshold Percentage, it applies to all clusters that use the Host Configuration Object.

Add button

Opens the Cluster Setup Dialog.

More information:

[Cluster Setup Dialog](#) (see page 544)

Tasks Related to the Host Configuration Object Dialog

The following tasks are related to the Host Configuration Object Dialog:

- [Configure a Host Configuration Object for a Trusted Host](#) (see page 111)
- [Create a Host Configuration Object](#) (see page 112)
- [Add Multiple Policy Servers to the Host Configuration Object](#) (see page 114)
- [Edit a Host Configuration Object](#) (see page 115)
- [Configure Policy Server Clusters for a Host Configuration Object](#) (see page 117)
- [Delete a Host Configuration Object](#) (see page 119)

Agent Configuration Object Dialog

The SiteMinder Agent Configuration Object Dialog is where you configure an Agent Configuration object for the Web Agent.

Agent Configuration Object Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.

For additional prerequisites, see the *SiteMinder Web Agent Guide*.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Agent Configuration Object Dialog

To navigate to the Agent Configuration Object Dialog, from the System tab of the Policy Server User Interface, select Edit, System Configuration, Create Agent Conf Object.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Agent Configuration Dialog Fields and Controls

Name field

Name for an Agent Configuration Object.

Description field

Description for an Agent Configuration Object.

Configuration Values group box

Lists the configuration values for an Agent Configuration Object.

Add Button

Opens the Edit Parameter Dialog from where you can.

More information:

[Edit Parameter Dialog](#) (see page 543)

Tasks Related to the Agent Configuration Object Dialog

The following tasks are related to the Agent Configuration Object Dialog:

- [Configure an Agent Configuration Object for a Web Agent](#) (see page 126)
- [Create an Agent Object to Establish a Web Agent Identity](#) (see page 123)
- [Create an Agent Configuration Object](#) (see page 128)

Agent Group Dialog

The SiteMinder Agent Group dialog is where you add agents to and remove agents from an agent group.

Agent Group Object Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Agent Group Dialog

There are multiple ways to access the Agent Group Dialog. The following procedure is the method recommended by CA.

To navigate to the Agent Group Dialog

1. From the System tab of the Policy Server User Interface, select View, Agent Groups.

The Policy Server places a check mark in the View menu to the left of the menu selection and *Agent Groups* appear in the System tab of the SiteMinder Administration window.

Note: If there is already a check next to *Agent Groups* in the View menu, selecting the option again will remove *Agent Groups* from the System tab.

2. Select Edit, System Configuration, Create Agent Group.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Agent Group Dialog Fields and Controls

Name field

Name for an Agent Configuration Object.

Description field

Description for an Agent Configuration Object.

Agent Type Group Box

The Agent Type group box is where you specify the type of Agents in the Agent group:

SiteMinder radio button

If selected, the Agent group contains SiteMinder agents of the type specified in the associated drop down list.

RADIUS radio button

If selected, the Agent group contains SiteMinder agents of the type specified in the associated drop down list.

Group Members list

Lists the Agents that belong to the Agent group.

Add/Remove button

Opens the Agent Group Items Dialog from where you can add Agents to and remove Agents from the Agent group.

More information:

[Agent Group Items Dialog](#) (see page 544)

Tasks Related to the Agent Group Dialog

The following tasks are related to the Agent Group Dialog:

- [Configure an Agent Group](#) (see page 139)
- [Add Agents to an Agent Group](#) (see page 140)

Edit Parameter Dialog

The Edit Parameter Dialog is where you edit Agent and Host Configuration object parameters.

Edit Parameter Dialog—Plain Radio Button Selected

Parameter Name field

Specifies a parameter name.

Value field

Specifies a value for the parameter.

Edit Parameter Dialog—Encrypted Radio Button Selected

Parameter Name field

Specifies a parameter name.

Encrypted Value field

Specifies a value for the parameter.

Encrypted Value field

Re-entered encrypted value.

Edit Parameter Dialog—Multi-value Radio Button Selected

Parameter Name field

Specifies the name of each parameter.

Enter Single Value field

Specifies the value of each parameter.

Add button

Adds the specified parameter and value to the Multiline Display field.

Multiline Display list

Lists the entries you specified in the Parameter Name and Enter Single Value fields.

Cluster Setup Dialog

The Cluster Setup Dialog is where you configure Policy Server clusters.

Current Setup list

Lists the IP addresses and port numbers for the Policy Servers in a cluster.

IP Address radio button

If selected, specify the IP address of the Policy Server in the cluster in the associated fields.

Note: If you do not know the IP address of the Policy Server, but you know the host name, click the DNS Lookup button to search for the IP address of the Policy Server.

Domain Name radio button

If selected, specify the domain name of the system where the Policy Server is installed in the associated fields. For example, server.company.com.

Server Port field

Specifies the port for Policy Server processing.

Agent Group Items Dialog

The Agent Group Items Dialog is where you add members to or remove members from an Agent group.

Current Members list

Lists the current members of an Agent group.

Available Members list

Lists the members available to an Agent group.

Chapter 24: User Directory and Search Dialog Reference

This section contains the following topics:

[User Directory Dialog](#) (see page 545)
[LDAP Setup Dialog](#) (see page 571)
[ODBC Setup Dialog](#) (see page 574)
[SQL Query Scheme Dialog](#) (see page 576)
[Directory List Dialog](#) (see page 582)
[Search LDAP/AD Dialog](#) (see page 584)
[Search Database Dialog](#) (see page 586)
[Search WinNT Directory Dialog](#) (see page 587)
[User Lookup Dialog](#) (see page 589)
[Directory Mappings Dialog](#) (see page 590)
[Add Directory Mapping Dialog](#) (see page 592)
[Edit Directory Mapping Dialog](#) (see page 593)
[Delete Dialog](#) (see page 594)
[AuthValidate Directory Mappings Dialog](#) (see page 595)
[Add AuthValidate Directory Mapping Dialog](#) (see page 596)
[Edit AuthValidate Directory Mapping Dialog](#) (see page 597)

User Directory Dialog

The SiteMinder User Directory Dialog allows you to configure a connection between the Policy Server and a user store. The Dialog contains four tabs:

Directory Setup tab

Identifies the namespace (type of user store), specifies the directory or database that contains user data, and, if necessary, provides additional information on how to locate user data in the directory or database.

Credentials and Connection tab

Specifies administrator credentials that the Policy Server uses to access the user store. For LDAP namespaces, you can also specify whether or not you are connecting to the LDAP server using a secure connection (SSL). For Active Directory, WinNT directories and custom directories, you can specify whether or not to use an authenticated user's security context when connecting to SiteMinder.

User Attributes tab

Designates user profile attributes that will be used by the Policy Server to store data for SiteMinder features, including Password Services and Registration Services.

View Contents Button tab

Opens the Directory List Dialog which you can use to the contents of the directory specified in the User Directory Dialog.

The type of user directory connection you choose determines the fields of information you must complete for a directory connection.

More information:

[Directory List Dialog](#) (see page 582)

User Directory Object Prerequisites

The following prerequisites must be met in order to successfully create a directory connection using the User Directory Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- Network Connectivity

To verify a connection to a user store defined in the User Directory Dialog, you must ensure that the system where the user store is installed can be accessed from the system where the Policy Server is installed. From the Policy Server system, ping the system that contains the user store. If the systems are connected, you can use the View Contents button in the User Directory Dialog to open the Directory List Dialog and view the users and groups contained in the user store.

More information:

[Policy Server Administrators](#) (see page 223)

[Directory List Dialog](#) (see page 582)

Navigate to the User Directory Dialog

There are multiple ways to access the User Directory Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA. To access the User Directory Dialog in the object pane menu bar, select Edit, System Configuration, Create User Directory.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Navigate to the User Directory Dialog for an Existing User Directory Connection

There are multiple ways to access the User Directory Dialog in the Policy Server User Interface for an existing user directory connection. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the User Directory Dialog for an existing user directory connection

1. In the object pane, click on User Directories.
2. In the list pane, click the existing directory connection.
3. From the menu bar, select Edit, Properties of User Directory.

Tasks Related to the User Directory Dialog

For information on these related tasks, see the following topics:

- [Configure User Directory Connections](#) (see page 156)
- [Define the Same User Directory Connection in Multiple Policy Stores](#) (see page 157)
- [Specify Directory Attributes](#) (see page 158)

User Directory Dialog—LDAP Namespace

When you select the LDAP option from the NameSpace drop-down list, the Directory Setup tab of the User Directory Dialog provides fields for configuring a connection to an existing LDAP-compliant user store.

At the top of the dialog, complete the following fields:

Name field

Name of the new user directory connection object. This is the name that will be used in the Policy Server User Interface to see the user store.

Note: To maintain single sign-on in a SiteMinder environment that includes multiple policy stores, use the same user directory name when you define a directory connection. For example, if you have an LDAP user directory connection called LDAP1 which you configured in Policy Store A, and you have a second policy store called Policy Store B, to maintain single sign-on, you must configure the connection in Policy Store B using the name LDAP1.

Description field

(Optional) A brief description of the user directory connection.

For details about the fields and controls in the User Directory Dialog, see:

- [User Directory Dialog—LDAP Namespace—Directory Setup Tab](#) (see page 548)
- [User Directory Dialog—LDAP Namespace—Credentials and Connection Tab](#) (see page 550)
- [User Directory Dialog—LDAP Namespace—User Attributes Tab](#) (see page 551)

User Directory Dialog—LDAP Namespace—Directory Setup Tab

The Directory Setup tab is where you specify LDAP servers, search roots for the LDAP servers, and parameters for locating users in an LDAP user store.

Directory Setup Group Box

NameSpace drop down list

Namespace of the user store. For LDAP namespaces using the Netscape LDAP SDK, select LDAP.

For information about choosing between the LDAP namespace (Netscape SDK) and AD namespace (Microsoft SDK).

Server field

IP address(es) and port number(s) of the LDAP server(s).

If you do not enter a port number, the default port (389) is used.

If you list multiple IP addresses or hostnames, spaces indicate failover and commas indicate load balancing groups. For example, consider the following entry:

domain1 domain2, domain3 domain4

Requests from the Policy Server will be load balanced between domain1 and domain3. If domain1 fails, its requests will be fulfilled by domain2. If domain3 fails, its requests will be fulfilled by domain4.

Note: For secure connections from the Policy Server to an LDAP user store over SSL, you must specify the port number in this field.

Configure button

Opens the LDAP Setup Dialog where you can specify multiple servers for load balancing and failover. You can use the LDAP Setup Dialog search for an IP address by providing a host name.

More information:

[Pros and Cons of Choosing an AD Namespace for an Active Directory Connection](#) (see page 173)

[LDAP Setup Dialog](#) (see page 571)

LDAP Search Group Box

Root field

Location in the LDAP tree that the Policy Server uses as the starting point for the directory connection—typically, an organization (o) or organizational unit (ou). The Policy Server begins searching at the root when locating a user.

Note: Depending on the LDAP vendor, the root for an LDAP user store is represented in a number of ways.

Scope drop down list

Determines how far down the LDAP directory tree the Policy Server searches for users:

One Level

Indicates one level below the root.

Subtree

Specifies all levels below the root.

Max Time field

Defines the time in seconds passed by the Policy Server to the user directory after which the user directory stops searching for results.

After the specified amount of time, the user directory stops searching and returns matching records or an error.

Default: 30

Note: The timeout can also be configured through the user interface of your LDAP directory. If the timeout periods are different, the shorter timeout takes precedence. A zero (0) value is not supported. A zero value can lead to threads hanging indefinitely in the Policy Server. Consider your system response time when setting a maximum value.

Max Results field

Maximum number of records that can be returned for a single search of the LDAP user directory.

The default value for this field is 0, which indicates there is no maximum limit on the number of records that can be returned for a single search of the LDAP user directory.

More information:

[Configure User Directory Connections](#) (see page 156)

LDAP User DN Lookup Group Box

Start field

Text string that will act as the beginning of an LDAP search expression or user DN. When a user attempts to login, the Policy Server prepends this string to the beginning of the username.

Search expressions depend on the setup of your user directory.

End field

Text string that will act as the end of an LDAP search expression or user DN. When a user attempts to login, the Policy Server appends the End string to the end of the username. The combination of the Start string, username, and End string is used to search the LDAP user directory.

An example of the search expression or user DN that the Policy Server will use to query the LDAP user directory based on the Start and End fields is displayed in the LDAP User DN Lookup Group Box.

Note: Whenever you use the Policy Server User Interface to update parameters relating to an LDAP connection, restart the Policy Server to make the new parameters effective.

More information:

[User Disambiguation in an LDAP Directory](#) (see page 143)

User Directory Dialog—LDAP Namespace—Credentials and Connection Tab

The Credentials and Connection tab of the User Directory Dialog is where you specify administrative credentials that the Policy Server uses to access the LDAP user directory, and enables you to indicate whether or not the Policy Server uses an SSL connection to access the user directory.

Administrator Credentials Group Box

Require Credentials check box

If selected, the credentials specified in the tab will be used by the Policy Server when connecting to the user directory.

Username

Distinguished name (DN) of a user who has sufficient privileges.

If...	Then...
a <i>read</i> Access Control List (ACL) has been defined to control access to the LDAP branch containing the user directory,	you must provide credentials for an administrative account with read access to that branch of the user directory.

If...	Then...
you want to use SiteMinder features that require read/write access to the directory	the administrator account that you specify must have read/write privileges.

For example, enter cn=Administrator.

Password and Confirm Password

Password associated with the account specified in the Username field.

The Policy Server uses the DN from the Username field and the password from the Password field to access the user directory.

Run in Authenticated User's Security Context check box

For LDAP-compliant Active Directory user stores for which you want to use the LDAP namespace, select this check box to provide a Windows user security context to Web applications that run on IIS Web servers and are protected by SiteMinder.

Secure Connection check box

Optional check box to force a Secure Sockets Layer (SSL) connection when the Policy Server and LDAP user directory communicate with each other.

Note: Be sure to specify the port for SSL in the Server field of the Directory Setup tab. Also, for SSL connections, make sure that you have specified the location of the certificate database file in the Policy Server Management Console (see the *SiteMinder Policy Server Management Guide*).

More information:

[Specify Directory Attributes](#) (see page 158)

[Pros and Cons of Choosing an AD Namespace for an Active Directory Connection](#) (see page 173)

User Directory Dialog—LDAP Namespace—User Attributes Tab

Some SiteMinder features require read or read/write access to directory attributes. This information must be stored in a user directory to which SiteMinder has access. As you configure user directory connections, you must provide the names of certain user profile attributes in the directories. You specify attributes in the User Attributes tab of the User Directory Dialog.

The attributes you specify must have specific data types. Password Attribute and Password Data require binary attributes, and all others require string attributes.

The attributes in the preceding figure are sample attributes. The attributes that you select for these fields may differ based on your directory type, schema, or preferences.

The attribute(s) you specify must:

- Exist in your user directory
- Be the correct type (string, binary)
- Not be used to store other types of data (only for read/write attributes)

The following attributes may be configured for LDAP directories:

Universal ID field

LDAP directory attribute SiteMinder uses as the Universal ID—for example, uid.

Disabled Flag field

LDAP directory attribute that SiteMinder uses to track disabled users—for example, carLicense.

The Disabled Flag attribute must have the type *string*.

Password Attribute field

LDAP directory attribute that SiteMinder uses to authenticate a user's password—for example, userPassword.

The attribute name you enter in this field must correspond to the location in the LDAP directory that contains user passwords. The attribute you choose must be of the type *binary*.

Password Data field

LDAP directory attribute that SiteMinder uses for Password Services data, such as old passwords—for example, displayName.

The attribute you choose must be of the type *binary*.

Anonymous ID field

LDAP directory attribute that contains the user's anonymous ID—for example, audio.

Email field

This attribute is not currently used by a SiteMinder feature.

Challenge/Response field

LDAP directory attribute containing a response that SiteMinder can send back to a user—for example, jpegphoto.

More information:

[Universal IDs](#) (see page 212)

[Anonymous Authentication Schemes](#) (see page 288)

Tasks Related to the User Directory Dialog—LDAP Namespace

The following tasks are related to the User Directory Dialog - LDAP Namespace:

- [Configure User Directory Connections](#) (see page 156)
- [Configure CA, iPlanet, and IBM Directory Server LDAP Directory Connections](#) (see page 160)
- [Domino Directory Connections Prerequisites](#) (see page 163)
- [Configuration of Novell eDirectory LDAP Directory Connections](#) (see page 166)
- [Configure Active Directory Connections](#) (see page 171)

User Directory Dialog—ODBC Namespace

When you select the ODBC option from the NameSpace drop-down list, the Directory Setup tab of the User Directory Dialog provides fields for configuring a connection to an existing ODBC-compliant user store.

At the top of the dialog, complete the following fields:

Name field

Name of the new user directory connection object. This is the name that will be used in the Policy Server User Interface to see the user store.

Note: To maintain single sign-on in a SiteMinder environment that includes multiple policy stores, use the same user directory name when you define a directory connection. For example, if you have an ODBC user directory connection called ODBC1 which you configured in Policy Store A, and you have a second policy store called Policy Store B, to maintain single sign-on, you must configure the connection in Policy Store B using the name ODBC1.

Description field

Optionally, a brief description of the user directory connection.

For details about the fields and controls in the User Directory Dialog, see:

- [User Directory Dialog—ODBC Namespace—Directory Setup Tab](#) (see page 553)
- [User Directory Dialog—ODBC Namespace—Credentials and Connection Tab](#) (see page 555)
- [User Directory Dialog—ODBC Namespace—User Attributes Tab](#) (see page 556)

User Directory Dialog—ODBC Namespace—Directory Setup Tab

Use the Directory Setup tab to specify ODBC data sources and to identify the SQL Query Scheme for an ODBC user directory connection.

Directory Setup Group Box

The Directory Setup group box is where you select the ODBC namespace and identify the data source that points to the database containing the user directory data.

NameSpace drop down list

Namespace of the user store. For ODBC namespaces (relational databases), select ODBC.

Data Source field

Name(s) of the ODBC data source(s).

This data source must be configured in order for the Policy Server to access the ODBC database. On Windows platforms, the data source must be defined as a system data source.

To enter multiple data source names for failover, separate names with commas or use the Configure button.

Configure button

Opens the ODBC Setup Dialog where you can specify multiple ODBC databases for failover.

More information:

[ODBC Setup Dialog](#) (see page 574)

SQL Query Scheme Group Box

The SQL Query Scheme group box is where you create or select a query scheme that the Policy Server will use for your ODBC user directory connection. A SQL Query Scheme is what the Policy Server uses to build queries for finding user data in a relational database. Without a SQL Query Scheme, the Policy Server cannot match attributes in your ODBC database with the information it needs to authenticate and authorize users.

SQL Query Scheme drop down list

Specifies an SQL query scheme from a list of all existing SQL query scheme objects.

Note: Don't use the same data source with different SQL query schemes. Create a unique data source for each SQL Query scheme.

Configure button

Opens the SQL Query Scheme Dialog for the SQL query scheme specified in the SQL Query Scheme field.

New button

Opens the SQL Query Scheme Dialog so that you can configure a new SQL query scheme.

More information:

[SQL Query Scheme Dialog](#) (see page 576)

[Configure SQL Query Schemes](#) (see page 197)

User Directory Dialog—ODBC Namespace—Credentials and Connection Tab

The Credentials and Connection tab of the User Directory Dialog allows you to specify administrative credentials that the Policy Server uses to access the ODBC database.

Administrator Credentials Group Box

Require Credentials check box

If selected, the credentials specified in the tab will be used by the Policy Server when connecting to the database.

Username field

Username for an account that has administrative database privileges.

The name you enter in the Username field must be the name of the user who owns the tables containing user directory data.

Note: This administrator account must have read or read/write privileges for the user directory based on the features of SiteMinder included in your implementation.

Password and Confirm Password

Password for the account specified in Username the field.

The Policy Server uses the credentials in the Username and Password fields to connect to the database. The Policy Server has access to any information available to the database user specified in the Username field.

More information:

[Specify Directory Attributes](#) (see page 158)

User Directory Dialog—ODBC Namespace—User Attributes Tab

Some SiteMinder features require read or read/write access to directory attributes. This information must be stored in a user directory to which SiteMinder has access. As you configure user directory connections, you must provide the names of certain user profile attributes in the directories. You can specify attributes in the User Attributes tab of the User Directory Dialog.

The attributes you specify must have specific data types. Password Attribute and Password Data require binary attributes, and all others require string attributes.

The attributes in the preceding figure are sample attributes. The attributes that you select for these fields may differ based on your directory type, schema, or preferences.

The attribute(s) you specify must:

- Exist in your database
- Be the correct type (string, binary)
- Not be used to store other types of data (only for read/write attributes)

The following attributes may be configured:

Universal ID field

(Optional) Name of the ODBC user directory attribute SiteMinder uses as the Universal ID.

The Universal ID can be passed to other applications that communicate with SiteMinder in order to maintain the identity of the user. This feature is mainly used as a bridge between SiteMinder and legacy applications which often use attributes other than a user name to identify the user. The attribute you choose must be of the type, string.

Disabled Flag field

(Optional) Name of the ODBC user directory attribute that SiteMinder uses to track disabled users. The attribute you choose must be of the type string.

For information about enabling and disabling user accounts, see the *SiteMinder Policy Server Management Guide*.

Password Attribute field

(Optional) Name of the ODBC user directory attribute that SiteMinder should use to authenticate a user's password. The attribute you choose must be of the type binary.

The attribute name you enter in this field must correspond to the location in the ODBC directory that contains user passwords.

Password Data field

(Optional) Name of the user directory attribute that SiteMinder can use for Password Services data.

SiteMinder's Password Services feature requires a location in the user directory to store a history of user password information. The attribute you choose must be of the type varchar.

More information:

[Universal IDs](#) (see page 212)

[Password Policies](#) (see page 479)

Tasks Related to the User Directory Dialog—ODBC Namespace

The following tasks are related to the User Directory Dialog - ODBC Namespace:

- [Configure User Directory Connections](#) (see page 156)
- [Configure ODBC Directory Connections](#) (see page 183)

User Directory Dialog—WinNT Namespace

You can use the User Directory Dialog to configure a connection to your WinNT user directory.

At the top of the dialog complete the following fields:

Name field

Specifies the name of the new user directory connection.

Note: To maintain single sign-on in a SiteMinder environment that includes multiple policy stores, use the same user directory name when you define a directory connection. For example, if you have an WinNT user directory connection called NT1 which you configured in Policy Store A, and you have a second policy store called Policy Store B, to maintain single sign-on, you must configure the connection and store it in the Policy Store B using the name NT1.

Description field

Optionally, specifies a brief description of the user directory connection.

More information:

[User Directory Dialog—WinNT Namespace—Directory Setup Tab](#) (see page 558)

[User Directory Dialog—WinNT Namespace—Directory Credentials and Connection Tab](#)
(see page 559)

[User Directory Dialog—WinNT Namespace—User Attributes Tab](#) (see page 561)

User Directory Dialog—WinNT Namespace—Directory Setup Tab

You can specify information that the Policy Server uses to connect to a WinNT directory in the Directory Setup tab.

Namespace drop down list

Namespace of the user directory. For WinNT namespaces, select WinNT.

The contents of the User Directory Dialog change to support a WinNT user directory connection.

NT Domain Name field

Name of the WinNT domain, computer within the WinNT domain, or stand-alone WinNT computer:

WinNT domains

To define a namespace that represents global users and groups in a WinNT domain, specify the name of the domain in the NT Domain Name field. For user accounts in domains, SiteMinder supports both domain authentication as well as trusted domain authentication. In order for domain authentication to work, the system on which the Policy Server is installed must have a computer account in the appropriate domain. If this system does not have a computer account in all domains in which users need to be authenticated, the appropriate trust relationships must be established between domains.

You can change the NT account under which the Policy Server is running. To do so, open the Settings, Control Panel, Services Dialog, and change the account under which the Policy Server is running to an account that has the necessary privileges to access the specified domain. This account must have the Act As Part of Operating System system privilege.

Individual WinNT computer in a domain

To define a Namespace that represents local users and groups in a computer that is a member of a domain, specify the name of the domain, followed by the name of the computer in the NT Domain Name field. The domain name and computer name must be separated by a forward slash (/). If you do not specify the name of the computer, performance during searches may suffer.

For example, if a domain called SampleDomain contains a computer called Comp1 which contains user information, you can enter the following in the NT Domain Name field:

SampleDomain/Comp1

Stand-alone WinNT computer

To define a namespace that represents local users and groups in a stand-alone computer, specify the name of the computer in the NT Domain Name field. The stand-alone computer must have a Policy Server installed on it in order for this namespace to be accessible. There may be an initial delay when the Policy Server accesses this namespace for the first time

User Directory Dialog—WinNT Namespace—Directory Credentials and Connection Tab

The Credentials and Connection tab of the User Directory Dialog contains the Administrator Credentials group box. This group box is where you specify whether or not credentials are required for user directory access. If credentials are required for access to the WinNT directory, you must provide SiteMinder with the appropriate credentials in the Administrator Credentials group box in order for the Policy Server to access the user directory.

A SiteMinder Web Agent can provide a Windows user security context for accessing Web resources in IIS Web servers. In the Credentials and Connection tab, you can specify whether or not SiteMinder should provide a Windows user security context. This enables the IIS server to enforce Windows access control mechanisms.

Note: Before SiteMinder can provide the Windows user security context, you must configure a session store and enable persistent sessions on a per realm basis.

More information:

[Agents and Agent Groups](#) (see page 99)

Administrator Credentials Group Box

Require Credentials check box

Indicates whether or not credentials are required to access the WinNT user directory.

Most WinNT user directories do not require credentials. The Policy Server runs as a system service on WinNT. By default a system service will not require credentials unless your network setup has been configured for greater security.

Note: If you do specify an account, it must have read or read/write privileges for the user directory based on the SiteMinder features included in your implementation.

Username field

Administrative username.

If your network requires credentials, enter the username for an account that has administrative privileges for the WinNT user directory. This account is used by SiteMinder to enable or disable users, and to access trusted domains. This account does not need to be the default administrator account, but it must be an account with the appropriate privileges.

If you specified a computer name in the NT Domain Name field, you must enter the computer name, followed by a backslash (\), followed by the name of the user account that has appropriate privileges.

For example, if you are entering credentials for a user called jsmith on a computer called Comp1, you enter the following:

Comp1\jsmith

Password field

Password for the privileged account.

Confirm Password

Repeat password for confirmation.

If you cannot access the user directory after supplying credentials, your network may require that you run the Policy Server from a privileged account.

Note: For information about WinNT account privileges, see your Windows documentation.

Run in Authenticated User's Security Context check box

(Optional) Enables SiteMinder to provide a Windows user security context to Web applications that run on IIS Web servers and are protected by SiteMinder.

More information:

[Specify Directory Attributes](#) (see page 158)

User Directory Dialog—WinNT Namespace—User Attributes Tab

The User Attributes tab of the User Directory Dialog is where you specify attributes in the WinNT user directory that the Policy Server uses to store data for certain SiteMinder features.

Universal ID field

(Optional) Name of the WinNT domain attribute SiteMinder uses as the Universal ID.

The Universal ID can be passed to other applications that communicate with SiteMinder in order to maintain the identity of the user. This feature is mainly used as a bridge between SiteMinder and legacy applications which often use attributes other than a user name to identify the user.

More information:

[Universal IDs](#) (see page 212)

Tasks Related to the User Directory Dialog—WinNT Namespace

The following tasks are related to the User Directory Dialog - WinNT Namespace:

- [Configure User Directory Connections](#) (see page 156)
- [Windows Directory Connections](#) (see page 204)

User Directory Dialog—Custom Namespace

To use a custom directory connection, you must configure the connection using the Policy Server User Interface.

At the top of the dialog, complete the following fields:

Name field

Name of the custom user directory connection.

Description field

Optionally, a brief description of the user directory connection.

More information:

[User Directory Dialog—Custom Namespace—Directory Setup Tab](#) (see page 562)

[User Directory Dialog—Custom Namespace—Credentials and Connection Tab](#) (see page 563)

[User Directory Dialog—Custom Namespace—User Attributes Tab](#) (see page 564)

User Directory Dialog—Custom Namespace—Directory Setup Tab

You can specify information that the Policy Server uses to connect to your custom directory in the Directory Setup tab. This tab contains the Directory Setup and Custom Namespace group boxes. The Directory Setup group box is where you identify the user directory as a custom namespace and enter the path and name of the custom library created for the directory connection. The Custom Namespace group box is where you enter parameters and timeouts for the custom user directory connection.

Directory Setup Group Box

Namespace drop down list

Namespace of the user directory. For custom namespaces, select Custom.

The contents of the User Directory Dialog change to support a custom user directory connection.

Library field

Shared library for the custom directory connection. You can specify the shared library using one of the following methods:

If the library is located...	Then specify...
Under one of the following Policy Server directories (recommended): ■ Windows—Bin ■ UNIX—lib	Only the library name, without a file extension.
Elsewhere	The full path to the library, including the library name, without a file extension

For example, to try the sample code that is shipped with the Directory API:

1. Compile the sample for the Directory API.
2. Place the library in one of the following directories of the Policy Server:
 - Windows—Bin
 - UNIX—lib
3. In the Library field, enter smdirapi.

If your custom library is not in the default Policy Server directory (Bin for Windows, lib for UNIX), you need to enter the full path to the library. For example, for a Policy Server on Windows, suppose that you built a DLL named MyLib.dll and placed it in D:\MyStuff. In the Library field, enter D:\MyStuff\MyLib.

Custom Namespace Group Box

Parameter field

Required parameters.

The requirements of the shared library created with the SiteMinder APIs determine the parameters required by the Policy Server for the custom user directory connection.

For example, using third-party APIs, you may have built a shared library that the SiteMinder APIs need to call. You can use the Parameter field to specify that third-party library.

Max Time field

Maximum search time for the user directory connection.

This is the time, in seconds, the Policy Server will search for results in the custom user directory. If the appropriate results cannot be found in the time specified in the field, the Policy Server returns an error message.

Max Results field

Maximum number of results that the Policy Server can return when searching the custom user directory.

When the Policy Server searches the custom user directory, the results of the search are limited to the amount you enter in this field.

User Directory Dialog—Custom Namespace—Credentials and Connection Tab

In the Credentials and Connection tab for a custom user directory, you can:

- Specify whether or not credentials are required for user directory access. If credentials are required for access to the custom directory, you must provide the Policy Server with the appropriate credentials in the Administrator Credentials group box.
- Indicate whether or not SiteMinder should provide a Windows user security context for accessing Web applications on IIS Web servers. Providing a windows users security context enables the IIS server to enforce Windows access mechanisms.
- Specify whether or not the Policy Server uses a secure connection (SSL) to connect to the user directory.

You can specify the following:

Require Credentials check box

If selected, administrator credentials will be used to connect to the custom user directory.

Username field

If Require Credentials is selected, defines the username of an administrator for the custom user directory. This is the username the Policy Server uses when connecting to the custom directory.

Password and Confirm Password field

Define the password for the privileged account specified in the Username field.

Run in Authenticated User's Security Context check box

(Optional) Enables SiteMinder to provide a Windows user security context to Web applications that run on IIS Web servers and are protected by SiteMinder.

Secure Connection check box

(Optional) Forces a Secure Sockets Layer (SSL) connection when the Policy Server and LDAP user directory communicate with each other.

Note: For SSL connections, make sure that you have specified the location of the certificate database file in the Policy Server Management Console (see the *SiteMinder Policy Server Management Guide*).

More information:

[How SiteMinder Is Configured to Provide a Windows User Security Context](#) (see page 96)

User Directory Dialog—Custom Namespace—User Attributes Tab

The User Attributes tab of the User Directory Dialog allows you to specify attributes in the WinNT user directory that the Policy Server uses to store data for certain SiteMinder features.

You must use the Apply button in the User Directory Dialog to save your custom user directory before configuring user attributes. When you apply your changes, the Policy Server uses the shared library you specified in the Library field of the Directory Setup tab to determine which fields in the User Attributes tab are available for the custom directory.

The following fields may be available based on the library for your custom user directory connection:

Universal ID field

Name of the attribute SiteMinder uses as the Universal ID. To try the sample for the Directory API, enter login.

The Universal ID can be passed to other applications that communicate with SiteMinder to maintain the identity of the user. This feature is mainly used as a bridge between SiteMinder and legacy applications that often use attributes other than a user name to identify the user.

Disabled Flag field

Name of the user directory attribute that holds the disabled state of the user. To try the sample for the Directory API, enter disabled.

The attribute name you enter in this field must correspond to the location in the directory that contains the disabled flag.

Password Attribute field

Name of the user directory attribute that SiteMinder should use to authenticate a user's password. To try the sample for the Directory API, enter password.

The attribute name you enter in this field must correspond to the location in the directory that contains user passwords. The attribute you choose must be of the type binary.

Password Data field

Name of the user directory attribute that SiteMinder can use for Password Services data.

SiteMinder's Password Services feature requires a location in the user directory to store a history of user password information.

The attribute you choose for Password Data in the user directory must be of the type binary.

Anonymous ID field

Name of the attribute in the user directory that contains the user's anonymous ID.

For SiteMinder sites that allow anonymous login, users may access resources by authenticating with an anonymous ID established by an Anonymous authentication scheme. When a user logs in using the anonymous scheme, SiteMinder stores the specified DN as the value of the attribute specified in the Anonymous ID field.

Challenge/Response field

Name of the attribute in the user directory that contains a response that SiteMinder can send back to a user.

This user directory attribute contains challenge and response information used in Password Services. For the example in step 6, this field indicates a user directory attribute that contains the hint users enter when they register a new user account. If users forget their passwords, SiteMinder uses the attribute specified in this field to determine the password hint that should be returned to the users.

More information:

[Universal IDs](#) (see page 212)

[Password Policies](#) (see page 479)

[Anonymous Authentication Schemes](#) (see page 288)

Tasks Related to the User Directory Dialog—Custom Namespace

The following tasks are related to the User Directory Dialog - Custom Namespace:

- [Configure User Directory Connections](#) (see page 156)
- [Configure Custom Directory Connections](#) (see page 209)

User Directory Dialog—AD Namespace

Configuring an Active Directory connection using the AD namespace is very similar to configuring a connection using the LDAP namespace.

Note: The AD namespace does not support multi-byte character sets. To use a multi-byte character set with Active Directory, configure your directory connection using the LDAP namespace.

At the top of the dialog, complete the following fields:

Name field

Name of the new Active Directory connection.

Note: To maintain single sign-on in a SiteMinder environment that includes multiple policy stores, use the same user directory name when you define a directory connection. For example, if you have an AD user directory connection called AD1 which you configured in Policy Store A, and you have a second policy store called Policy Store B, to maintain single sign-on, you must configure the connection in Policy Store B using the name AD1.

Description field

(Optional) A brief description of the user directory connection.

More information:

[Configure User Directory Connections](#) (see page 156)

[Pros and Cons of Choosing an AD Namespace for an Active Directory Connection](#) (see page 173)

[User Directory Dialog—AD Namespace—Directory Setup Tab](#) (see page 567)

[User Directory Dialog—AD Namespace—Credentials and Connection Tab](#) (see page 569)

[User Directory Dialog—AD Namespace—User Attributes Tab](#) (see page 570)

User Directory Dialog—AD Namespace—Directory Setup Tab

You can specify information that the Policy Server uses to connect to your Active Directory directory in the Directory Setup tab. This tab contains the Directory Setup, LDAP Search, and LDAP User DN Lookup group boxes.

Directory Setup Group Box

Namespace drop-down list

Specifies the namespace of the user store. For directories that use the Microsoft SDK, select AD.

Note: For information about choosing between the LDAP namespace (Netscape SDK) and AD namespace (Microsoft SDK), see [Pros and Cons of Choosing an AD Namespace for an Active Directory Connection](#) (see page 173).

Server field

Specifies the domain name of the AD server.

Configure button

Opens the LDAP Setup Dialog where you can specify multiple servers for load balancing and failover. You can use the LDAP Setup Dialog search for an IP address by providing a host name.

More information:

[Pros and Cons of Choosing an AD Namespace for an Active Directory Connection](#) (see page 173)
[LDAP Setup Dialog](#) (see page 571)

LDAP Search Group Box

Root field

Active Directory root that the Policy Server uses as the starting point for the directory connection.

For Active Directory, the Root will typically have the following format:

`dc=<server>,dc=<domain>,dc=<extension>`

For example, `dc=server,dc=myorg,dc=org`

Scope drop down list

Specifies how far down the directory tree the Policy Server searches for users:

One Level

Indicates one level below the root

Subtree

Specifies all levels below the root

Max Time field

Time in seconds after which the Policy Server will stop searching the user directory for results.

After the specified amount of time, the Policy Server stops searching the directory and returns matching records or an error.

Default: 30

Note: The timeout can also be configured through the user interface of your Active Directory. If the timeout periods are different, the shorter timeout takes precedence. A zero (0) value is not supported. A zero value can lead to threads hanging indefinitely in the Policy Server. Consider your system response time when setting a maximum value.

Max Results field

Maximum number of records that can be returned for a single search of the LDAP user directory.

The default value for this field is 0, which indicates there is no maximum limit on the number of records that can be returned for a single search of the LDAP user directory.

LDAP User DN Lookup Group Box

Start field

Text string that will act as the beginning of an Active Directory (LDAP) search expression or user DN.

When a user attempts to login, the Policy Server prepends this string to the beginning of the username.

Search expressions depend on the setup of your user directory.

End field

Text string that will act as the end of an Active Directory (LDAP) search expression or user DN.

When a user attempts to login, the Policy Server appends the End string to the end of the username. The combination of the Start string, username, and End string is used to search the LDAP user directory.

Note: Whenever you use the Policy Server User Interface to update parameters relating to an LDAP connection, restart the Policy Server to make the new parameters effective.

More information:

[User Disambiguation in an LDAP Directory](#) (see page 143)

User Directory Dialog—AD Namespace—Credentials and Connection Tab

The Credentials and Connection tab of the User Directory Dialog allows you to specify administrative credentials that the Policy Server uses to access Active Directory, and enables you to indicate whether or not the Policy Server uses an SSL connection to access the directory.

Administrator Credentials Group Box

Require Credentials check box

If selected, the credentials specified in the tab will be used by the Policy Server when connecting to the user directory.

Username field

Distinguished name (DN) of a user who has sufficient privileges.

Note: For Active Directory you must enter a full DN. For example:

For example:

`cn=<administrator>,cn=<administrator's group>,dc=<server>,dc=<organization>,dc=<com, net, etc.>`

Password and Confirm Password fields

Password associated with the account specified in the Username field.

The Policy Server uses the DN from the Connect Username field and the password from the Connect Password field to access the user directory.

Secure Connection check box

If selected, forces a Secure Sockets Layer (SSL) connection when the Policy Server and Active Directory communicate with each other.

Note: Be sure to specify the port for SSL in the Server field of the Directory Setup tab.

Note: For SSL connections, make sure that you have specified the location of the certificate database file in the Policy Server Management Console (see the *SiteMinder Policy Server Management Guide*).

Run in Authenticated User's Security Context check box

If selected, provides a Windows user security context to Web applications that run on IIS Web servers and are protected by SiteMinder.

More information:

[How SiteMinder Is Configured to Provide a Windows User Security Context](#) (see page 96)

User Directory Dialog—AD Namespace—User Attributes Tab

Some SiteMinder features require read or read/write access to directory attributes. This information must be stored in a user directory to which SiteMinder has access. As you configure user directory connections, you must provide the names of certain user profile attributes in the directories. You can specify attributes in the User Attributes tab of the User Directory Dialog.

The attributes you specify must have specific data types. Password Attribute and Password Data require binary attributes, and all others require string attributes.

The attributes in the preceding figure are sample attributes. The attributes that you select for these fields may differ based on your directory type, schema, or preferences.

The attribute(s) you specify must:

- Exist in your user directory
- Be the correct type (string, binary)
- Not be used to store other types of data (only for read/write attributes)

The following attributes may be configured for LDAP directories:

Universal ID field

Active Directory attribute SiteMinder uses as the Universal ID—for example, uid.

Disabled Flag field

Active Directory attribute that SiteMinder uses to track disabled users—for example, carLicense.

The Disabled Flag attribute must have the type *string*.

Password Attribute field

Active Directory attribute that SiteMinder uses to authenticate a user's password—for example, userPassword.

The attribute name you enter in this field must correspond to the location in the LDAP directory that contains user passwords. The attribute you choose must be of the type *binary*.

Password Data field

Active Directory attribute that SiteMinder uses for Password Services data, such as old passwords—for example, displayName.

The attribute you choose must be of the type *binary*.

Anonymous ID field

Active Directory attribute that contains the user's anonymous ID—for example, audio.

Email field

This attribute is not currently used by a SiteMinder feature.

Challenge/Response field

Active Directory attribute containing a response that SiteMinder can send back to a user—for example, jpegphoto.

More information:

[Universal IDs](#) (see page 212)

[Anonymous Authentication Schemes](#) (see page 288)

Tasks Related to the User Directory Dialog—AD Namespace

The following tasks are related to the User Directory Dialog - AD Namespace:

- [Configure User Directory Connections](#) (see page 156)
- [Configure Active Directory Connections](#) (see page 171)

LDAP Setup Dialog

The LDAP Setup Dialog is where you configure a series of LDAP directory servers in a user directory object to provide a load balancing and failover configuration for redundant LDAP directory servers.

Note: You can add the same server multiple times for load balancing to force more requests to be serviced by a specific system. For example, consider two servers in a group (Server1 and Server2). Server1 is a high-performance server and Server2 is a lesser system. You can add Server1 to the load balancing list twice so that it will process two requests for each request processed by Server2.

LDAP Setup Dialog Prerequisites

The following prerequisites must be met in order to successfully configure failover and load balancing of LDAP user directories in the LDAP Setup Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- Network Connectivity

All systems that contain directories for failover and load balancing must be accessible from the system where the Policy Server is installed. From the Policy Server system, ping the systems that contain the user directories.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the LDAP Setup Dialog

You can configure your LDAP from the User Directory Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the LDAP Setup Dialog

1. Open the User Directory Dialog.
2. Ensure that the Namespace drop-down list in the Directory Setup tab specifies an LDAP namespace.
3. In the Directory Setup tab, click the Configure button.

More information:

[User Directory Dialog](#) (see page 545)

LDAP Setup Dialog Fields and Controls

The LDAP Setup Dialog contains two separate group boxes:

- [Add an IP Address or Name Group Box](#) (see page 572)
- [Current Setup Group Box](#) (see page 573)

Add an IP Address or Name Group Box

IP Address radio button

Indicates that you will enter the IP address of an LDAP user directory in the fields to the right of the radio button.

Domain Name radio button

Indicates that you will enter the domain name of an LDAP user directory server in the field to the right of the radio button.

Port field

Specifies the port number for the LDAP user directory specified in the Add an IP Address or Name group box.

If you do not specify a port number, the Policy Server uses the default port (636 for SSL and 389 for non-SSL).

Note: If you assign a port number to the last server in a failover group, that port number is used for every server in the failover group.

DNS Lookup button

Opens the DNS Lookup Dialog. Enter a host name in the DNS Lookup Dialog and click OK. The DNS Lookup Dialog closes and the IP address for the server appears in the fields to the right of the IP Address Radio Button in the LDAP Server Dialog.

Add Load Balancing button

Adds the IP address or domain name specified in the Add an IP Address or Name group box to a new line in the Load Balancing and Failover list in the Current Setup group box.

Add Failover button

Adds the IP address or domain name specified in the Add an IP Address or Name group box to the end of the selected line in the Load Balancing and Failover list in the Current Setup group box.

Note: If you are adding a server for failover, the failover directory must use the same type of communication (SSL or non-SSL) as the primary directory, since both directories share the same port number.

Current Setup Group Box

The Policy Server User Interface displays the IP address or domain name in the Current Setup group box list below the existing server(s) for load balancing or beside the existing server(s) for failover.

Load Balancing and Failover list

Displays the current configuration of directory servers for load balancing and failover associated with the directory connection in the User Directory Dialog.

Each entry along the vertical axis of the table indicates a server or group of servers configured for load balancing. Each set of entries along the horizontal axis indicates a group of servers configured for failover.

Remove button

Remove the selected group of failover directory servers from the list of servers.

Note: When removing servers in a failover group, all servers in the group are removed. If you want to remove a single server from a failover group, you must remove all of the servers in the group, then add the remaining servers to the list.

More information:

[User Directory Dialog](#) (see page 545)

Tasks Related to the LDAP Setup Dialog

The following tasks are related to the LDAP Setup Dialog:

- [View User Directory Contents](#) (see page 210)
- [Configure CA, iPlanet, and IBM Directory Server LDAP Directory Connections](#) (see page 160)
- [Domino Directory Connections Prerequisites](#) (see page 163)
- [Configuration of Novell eDirectory LDAP Directory Connections](#) (see page 166)
- [Configure Active Directory Connections](#) (see page 171)

ODBC Setup Dialog

You can configure multiple ODBC data sources for failover. Once you configure the data sources for failover, if the primary data source fails to respond to a request, the next data source defined in the failover order responds to the request.

ODBC Setup Dialog Prerequisites

The following prerequisites must be met in order to successfully configure failover of ODBC databases in the ODBC Setup Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege. See [Policy Server Administrators](#) (see page 223) for more information.
- Network Connectivity

All systems that contain directories for failover must be accessible from the system where the Policy Server is installed. From the Policy Server system, ping the systems that contain the databases.

Navigate to the ODBC Setup Dialog

You can configure the ODBC from the User Directory Dialog.

To access the ODBC Setup Dialog

1. Open the User Directory Dialog.
2. Ensure that the Namespace drop-down list in the Directory Setup tab specifies an ODBC namespace.
3. In the Directory Setup tab, click the Configure button.

More information:

[User Directory Dialog](#) (see page 545)

ODBC Setup Dialog Fields and Controls

The ODBC Setup Dialog contains two separate group boxes:

- [Add a Data Source Group Box](#) (see page 575)
- [Current Setup Group Box](#) (see page 575)

Current Setup Group Box

Failover list

Each entry along the vertical axis indicates an ODBC data source configured for failover. The failover order is sequential from the top of the list to the bottom of the list.

Remove button

Removes the selected ODBC data source from the Failover List.

Add a Data Source Group Box

ADBC Data Source field

Name of the ODBC data source you want to add to the Failover list in the Current Setup group box.

Add Failover button

Adds the data source in the ODBC Data Source field to the Failover list in the Current Setup group box.

Tasks Related to the ODBC Setup Dialog

The following tasks are related to the ODBC Setup Dialog:

- [Configure ODBC Data Sources for Failover](#) (see page 183)
- [Configure User Directory Connections](#) (see page 156)

SQL Query Scheme Dialog

The Policy Server uses SQL Query Schemes to build queries that find user data in a relational database. You create and edit SQL Query Schemes using the SiteMinder SQL Query Scheme Dialog.

Note: The “SM_— prefix in column names is reserved for additional special names required by SiteMinder. Column names in your user directory should not begin with the “SM_— prefix. Policy Server errors will occur during user lookups if this prefix appears in column names.

SQL Query Scheme Dialog Prerequisites

The following prerequisites must be met in order to successfully a query scheme in the SQL Query Scheme Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the SQL Query Scheme Dialog

To access the SQL Query Scheme Dialog and create a new query scheme in the Policy Server User Interface, from the menu bar, select Edit, System Configuration, Create SQL Query Scheme.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Navigate to the SQL Query Dialog for an Existing Query Scheme

There are multiple ways to access the SQL Query Scheme Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the SQL Query Scheme Dialog for an existing query scheme

1. In the list pane, click on an existing directory connection
2. From the menu bar, select Edit, Properties of SQL Query Scheme

SQL Query Scheme Dialog Fields and Controls

The SQL Query Scheme Dialog contains fields that describe SQL queries that that Policy Server uses to access user and group information. The default values for each query field correspond to the SiteMinder sample relational database schema called SmSampleUsers. You must change the table and column names if you are using a different database schema.

You can use * for multi-character SQL searches and & for single character searches. When the Policy Server performs the search, it replaces:

- * with the SQL wild character % in a multi-character SQL query
- & with the SQL wild character _ in a single character SQL query

The “%s— expression in a query is a place-holder for a parameter to be supplied by the Policy Server when the query is executed.

Name field

Unique name of the SQL query scheme.

Description field

Brief description of the SQL query scheme.

Enumerate field

Defines a query that fetches a list of the names of group objects in the database. The Policy Server User Interface lists these, along with the class name (User or Group) if desired. It is advisable to enumerate only the groups and not the users (omit the union in this query). Individual users can still be entered by typing them into the Users/Groups Dialog.

Default value:

```
select Name, 'Group' as Class from SmGroup order by Class
```

Lookup field

Defines a query that is used when a SiteMinder administrator specifies a value without an attribute name in a User Directory search query.

The query looks in the default user name and group name columns of the database and tries to match the value specified in the query. The result of the query is the combination of all of the values that match the search criteria from the user and group columns.

For example, if an administrator searches for J*, this query returns a list of all users and groups that begin with the letter J.

Default value:

```
select Name, 'User' as Class from SmUser where Name %s Union select Name, 'Group' as Class from SmGroup where Name %s order by Class
```

Lookup Users field

Defines a query that

Default value:

```
select Name, 'User' as Class from SmUser where %s
```

This query is used when a SiteMinder administrator specifies an attribute from the user table in the database in a User Directory query. It is also used by the Policy Server during directory mappings, and when authorizing users against an ODBC database.

For example, if an administrator specifies an attribute of name=J*, this query returns a list of all users whose Name attribute begins with J.

SiteMinder executes this query along with Lookup Groups to provide a combined list of results that includes users and groups. In the example above, the Policy Server would return a list of users and groups whose Name attributes begin with the letter J.

Lookup Groups field

Defines a query that

Default value:

```
select Name, 'Group' as Class from SmGroup where %s
```

This query is used when a SiteMinder administrator specifies an attribute from the group table in the database in a User Directory query.

For example, if an administrator specifies an attribute of name=J*, this query returns a list of all groups whose Name attribute begins with J.

The Policy Server executes this query along with Lookup Users to provide a combined list of results that includes users and groups. In the example above, the Policy Server would return a list of users and groups whose Name attributes begin with the letter J.

Get User/Group Info field

Defines a query that

Default value:

```
select Name, 'User' from SmUser where Name = '%s' Union select Name, 'Group' from  
SmGroup where Name = '%s'
```

The purpose of this query is to discover the class of a given object. The %s parameter is the name of the object that can be a User or a Group.

Init User field

Defines a query that

Default value:

```
select Name from SmUser where Name = '%s'
```

This query is used when a SiteMinder administrator specifies a User Directory query to find out if a user with a given name exists in the database. The %s parameter represents the user name.

If the manual entry query does not contain a WHERE statement, the WHERE statement from Init User is appended. For example:

Init User: select Name from SmUser where Name = '%s'

Manual Entry: Select Name from customers

Result: select Name from customers where Name = '%s'

If the manual entry query contains a WHERE statement, the portion of the query following the Init User WHERE statement is appended. For example:

Init User: select Name from SmUser where Name = '%s'

Manual Entry: Select Name from customers where balance >1000

Result: select Name from customers where balance >1000 and Name = '%s'

Authenticate User field

Defines a query that is executed to get a password for a user. The %s parameter represents the user name.

If you are configuring a query scheme for an Oracle database and you are using Oracle's encrypted password feature, replace the entire query string with the word connect. Using the word connect for this query indicates to the Policy Server that a user's name and password should be evaluated by the Oracle encrypted password feature.

Default value:

```
select Name from SmUser where Name = '%s' and Password = '%s'
```

Is Group Member field

Defines a query that should answer the question of group membership for a particular user. The first %s parameter is the user name and the second is the group name.

Default value:

```
select Id from SmUserGroup where UserId = (select UserId from SmUser where Name = '%s') and GroupId = (select GroupId from SmGroup where Name = '%s')
```

Get User Groups field

Defines a query that fetches the names of the groups of which the user is a member. The %s parameter represents the user name.

```
select SmGroup.Name from SmGroup, SmUser, SmUserGroup where SmUser.Name = '%s' and SmUser.UserId = SmUserGroup.UserId and SmGroup.GroupId = SmUserGroup.GroupId
```

Get User Properties field

Defines a comma-separated list of user attributes. These are expected to reside in the same table as the user name.

Default value:

```
Name, UserId, FirstName, LastName, TelephoneNumber, EmailAddress, PIN, Mileage, Disabled
```

Get User Property field

Defines a query that fetches the value of a user property. The first %s parameter is the property name, the second is the user name. All properties are assumed to have string values.

The property must be one of the properties specified in the Get User Properties field.

Default value:

```
select %s from SmUser where Name = '%s'
```

Set User Property field

Defines a query that sets the value of a user property. The first %s parameter is the property value, the second is the property name, and the third is the user name. Note that this is a SQL 'update' rather than a SQL 'select'. Database write access is required. All properties are assumed to have string values.

The property must be one of the properties specified in the Get User Properties field.

Default value:

```
update SmUser set %s = '%s' where Name = '%s'
```


Set User Password field

Defines a query that sets the value of a user password is set. Note that this is a SQL 'update' rather than a SQL 'select'. Database write access is required. All properties are assumed to have string values.

Default value:

```
update SmUser set Password = '%s' where Name = '%s'
```

Get Group Properties field

Defines a comma separated list of group attributes. These attributes are used to search the contents of a group, or to bind policies to group attributes. The attributes are expected to reside in the same table as the group name.

Default value:

```
Name, GroupId
```

Get Group Property field

Defines a query that is similar to that in the Get User Property field, except that it returns a property of the group to which a user belongs. The first %s parameter is the property name, the second is the group name. All properties are assumed to have string values.

The property must be one of the properties specified in the Get Group Properties field.

Default value:

```
select %s from SmGroup where Name = '%s'
```

Set Group Property field

Defines a query that sets the value of a property. The first %s parameter is the property value, the second is the property name, and the third is the group name. Note that this is a SQL 'update' rather than a SQL 'select'. Database write access is required. All properties are assumed to have string values.

The property must be one of the properties specified in the Get Group Properties field.

Default value:

```
update SmGroup set %s = '%s' where Name = '%s'
```

Use Literal Query

Select to use the literal query type.

Default: This is the default query type.

Use Bind Query

Select to use bind-based parameters in the query. Replace each instance of '%' with a question mark (?) to use bind-based parameters in the query. Any instance not changed should remain as '%s'.

Restrictions:

- Advanced Password Services (APS) does not support bind-based parameters.
- A query uses bind-based parameters if it contains a question mark (?) and literal parameters if it contains '%s'.
- A query cannot contain '%s' and question marks (?) together.

More information:

[Users/Groups Dialog](#) (see page 912)

Tasks Related to the SQL Query Scheme Dialog

The following tasks are related to the SQL Query Scheme Dialog:

- [Configure SQL Query Schemes](#) (see page 197)
- [Configure ODBC Directory Connections](#) (see page 183)

Directory List Dialog

The Policy Server User Interface allows you to view the contents of a user directory. By default you can only view user groups, not individual users. To view individuals, use a user directory search.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[User Directories](#) (see page 141)

Directory List Dialog Prerequisites

The following prerequisites must be met in order to successfully access the Directory List Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- User directory object must exist. If you are configuring a new user directory connection, the View Contents button you use to open the Directory List Dialog is not available until you click Apply to save the settings for your new directory connection.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Directory List Dialog

To access the User Directory Dialog, in the User Directory Dialog, click the View Contents button.

More information:

[User Directory Dialog](#) (see page 545)

Directory List Dialog Fields and Controls

The Directory List Dialog displays a list of user groups. This information is only for viewing. Use the Search icon to search in the directory. Click OK to close the Directory List Dialog.

More information:

[Search Directories](#) (see page 211)

Tasks Related to the Directory List Dialog

The following tasks are related to the Directory List Dialog:

- [View User Directory Contents](#) (see page 210)

Search LDAP/AD Dialog

When searching for a user or user group in an LDAP-compliant user directory, you can search based on an attribute-value pair, or you can search using an LDAP search expression.

Search LDAP/AD Dialog Prerequisites

The following prerequisites must be met in order to access the Search LDAP/AD Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- User directory object must exist. If you are configuring a new user directory connection, you cannot open the Search LDAP/AD Dialog until you click Apply to save the settings for your new directory connection.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Search LDAP/AD Dialog

To access the Search LDAP/AD Dialog in the Policy Server User Interface, from the Directory List Dialog for an LDAP or Active Directory connection, click the search icon.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Directory List Dialog](#) (see page 582)

Search LDAP/AD Dialog Fields and Controls

The Search LDAP/AD Dialog allows you to search an LDAP-compliant user directory using an attribute/value pair or an LDAP query. Select the appropriate radio button associated with your search type.

LDAP Directory Attribute Search Group Box

Attribute-Value Pair radio button

If selected, the user directory is searched using the attribute/value pair specified in the associated fields:

Attribute field—User directory attribute

For example, to search for a user by telephone number, you might enter telephone in the Attribute field.

Value field—Search string

The search string in this field can contain wild cards. When you click OK, the Policy Server searches for users or groups that contain the attribute you specified in the Attribute field that match the value you entered in the Value field.

Note: The following characters must be "escaped" as part of attribute values when specifying a search expression or an attribute- value pair: close parenthesis ')', open parenthesis '(', and backslash '\'. For example, the following search expression "&(uid=*)(cn=*)" must include escape characters (marked in **bold**) as follows: "&(uid=**)(cn=**)"".

LDAP Query radio button

If selected, the user directory is searched using the LDAP query specified in the associated field.

Search Expression field—LDAP search expression

The Policy Server User Interface automatically adds encapsulating parentheses to the expression. For example, to specify the search filter (&(uid=*)(cn=*)), enter the following Search Expression:

&(uid=*)(cn=*)

OK button

Click to close the Search LDAP Directory Dialog and displays the results in the Directory List Dialog.

More information:

[User Disambiguation in an LDAP Directory](#) (see page 143)

[Configure User Directory Connections](#) (see page 156)

[Directory List Dialog](#) (see page 582)

Tasks Related to the Search LDAP/AD Dialog

The following tasks are related to the Search LDAP/AD Dialog:

- [Search Directories](#) (see page 211)

Search Database Dialog

When searching a Microsoft SQL Server or Oracle user directory, you can search for users, groups, or both. Searches are based on attribute-value pairs.

Search Database Dialog Prerequisites

The following prerequisites must be met in order to access the Search Database Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- User directory object must exist. If you are configuring a new user directory connection, you cannot open the Search Database Dialog until you click Apply to save the settings for your new database connection.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Search Database Dialog

To access the Search Database Dialog in the Policy Server User Interface, from the Directory List Dialog for an ODBC connection, click the search icon.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Directory List Dialog](#) (see page 582)

Search Database Dialog Fields and Controls

Search drop down list

Indicates the scope for the search. Selections include Users, Groups, and All (users and groups).

Attribute Value Pair

Attribute field

User directory attribute (the name of a column in the database table), such as Name or UserID.

Value field

Search string. For example, L*.

The search string in the Value field can contain wild cards, as shown above. In this example, the user is searching for all users whose names begin with the letter L.

OK Button

Closes the Search Database Directory Dialog and displays the results Directory List Dialog.

Tasks Related to the Search Database Dialog

The following tasks are related to the Search Database Dialog:

- [Search Directories](#) (see page 211)

Search WinNT Directory Dialog

When searching a WinNT user directory, you can search for users, groups, or both. Searches are based search expressions.

Search WinNT Dialog Prerequisites

The following prerequisites must be met in order to access the Search WinNT Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- User directory object must exist. If you are configuring a new user directory connection, you cannot open the Search WinNT Dialog until you click Apply to save the settings for your new database connection.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Search WinNT Dialog

To access the Search WinNT Dialog in the Policy Server User Interface, use any of the following methods, from the Directory List Dialog for a WinNT connection, click the search icon.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More information:

[Directory List Dialog](#) (see page 582)

Search WinNT Dialog Fields and Controls

Search drop down list

Indicates the scope for the search. Selections include Users, Groups, and All (users and groups).

Search Expression field

WinNT search expressions are strings that the Policy Server matches to the group or user name. You can use wild cards in a WinNT user directory search. For example, if you wanted to search for all users whose names begin with the letter S, you would select Groups from the Search drop-down list, then enter a Search Expression of S*.

OK Button

Closes the Search WinNT Directory Dialog and displays the results in the SiteMinder Directory List Dialog.

Tasks Related to the Search WinNT Dialog

The following tasks are related to the Search WinNT Dialog:

- [Search Directories](#) (see page 211)

User Lookup Dialog

The User Lookup Dialog combines the features of the following:

- [Directory List Dialog](#) (see page 582)
- Search Dialogs:
 - [Search LDAP/AD Dialog](#) (see page 584)
 - [Search Database Dialog](#) (see page 586)
 - [Search WinNT Directory Dialog](#) (see page 587)

You can use the User Lookup Dialog to locate a specific user or group of users.

User Lookup Dialog Prerequisites

The following prerequisites must be met in order to access the User Lookup Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or the Manage Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the User Lookup Dialog

You can access the User Lookup Dialog in the Policy Server User Interface from the Response Attribute Editor Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the User Lookup Dialog

1. From the Response Attribute Editor Dialog, select the Attribute Setup tab.
2. In the Attribute Kind group box, select the DN Attribute radio button.
3. In the Attribute Fields group box, click the Lookup button.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

User Lookup Dialog Fields and Controls

Directory drop down list

Specifies the directory in the associated policy domain in which you want to search.

View Contents button

Displays the groups contained in the directory specified in the Directory field in the Contents group box.

To search for a specific user, click the Search icon.

For information about searching for individual users, see:

- [Search LDAP/AD Dialog](#) (see page 584)
- [Search Database Dialog](#) (see page 586)
- [Search WinNT Directory Dialog](#) (see page 587)

OK Button

When you select a group from the list in the Contents group box and click this button, the Policy Server returns the appropriate user or group DN (or Name, for relational databases) of the selected user or group to the Response Attribute Editor Dialog.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

Tasks Related to the User Lookup Dialog

The following tasks are related to the User Lookup Dialog:

- [User Disambiguation in an LDAP Directory](#) (see page 143)

Directory Mappings Dialog

The Directory Mappings Dialog gives SiteMinder the ability to authenticate users against one directory and authorize users against another. Using this dialog, you can add, delete, or edit Directory Mappings configurations.

Directory Mappings Dialog Prerequisites

Before you can create a directory mapping, you must create one or more of the following Policy Server objects:

- User Directory Object(s)
- Policy Domain Objects(s)
- Realm Objects(s)
- Rule Object(s)
- Policy objects may also see any of the following Policy Server Objects:
 - Response Object(s)
 - Response Group Object(s)

Navigate to the Directory Mappings Dialog

To access the Directory Mappings Dialog, from the Policy Server User Interface menu bar, select Advanced, Directory Mapping.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Directory Mappings Dialog Fields and Controls

Current Mappings list

Lists all existing directory mappings.

Add button

Opens the Add Directory Mapping Dialog from where you can create a new directory mapping.

Delete button

Deletes the selected item from the Current Mappings list.

Edit button

Opens the item selected in the Current Mappings list for editing in the Edit Directory Mapping Dialog.

More information:

[Edit Directory Mapping Dialog](#) (see page 593)

Tasks Related to the Directory Mappings Dialog

The following tasks are related to the Directory Mappings Dialog:

- [Configure a Directory Mapping](#) (see page 215)
- [Assign an Authorization Directory to a Realm](#) (see page 216)

Add Directory Mapping Dialog

The Add Directory Mapping Dialog is where you create directory mappings that allow you to authenticate users against one directory and authorize users against another.

Navigate to the Add Directory Mapping Dialog

To access the Add Directory Mapping Dialog in the Policy Server User Interface, from the Directory Mappings Dialog, click the Add button.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Add Directory Mapping Fields and Controls

Authentication Directory drop down list

Specifies the directory against which users will be authenticated.

Authorization Directory drop down list

Specifies the directory against which users will be authorized.

Mapped DNs Group Box

The Mapped DNs group box contains radio buttons that determines how DNs are mapped.

Identical DN radio button

Maps the user's distinguished name (DN) exactly from the authentication directory to the authorization directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the authorization directory. Therefore, the authorization directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID radio button

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the authorization directory to identify the user.

More information:

[Universal IDs](#) (see page 212)

Tasks Related to the Add Directory Mapping Dialog

The following tasks are related to the Add Directory Mapping Dialog:

- [Configure a Directory Mapping](#) (see page 215)

Edit Directory Mapping Dialog

The Edit Directory Mapping Dialog is where you edit existing directory mappings that allow you to authenticate users against one directory and authorize users against another.

Navigate to the Edit Directory Mapping Dialog

To access the Edit Directory Mapping Dialog in the Policy Server User Interface, from the Directory Mappings Dialog, select an entry in the Current Mappings list and click the Edit button.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Edit Directory Mapping Fields and Controls

Authentication Directory drop down list

Specifies the directory against which users will be authenticated.

Authorization Directory drop down list

Specifies the directory against which users will be authorized.

Mapped DNs Group Box

The Mapped DNs group box contains radio buttons that determines how DNs are mapped.

Identical DN radio button

Maps the user's distinguished name (DN) exactly from the authentication directory to the authorization directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the authorization directory. Therefore, the authorization directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID radio button

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the authorization directory to identify the user.

More information:

[Universal IDs](#) (see page 212)

Tasks Related to the Edit Directory Mapping Dialog

The following tasks are related to the Edit Directory Mapping Dialog:

- [Configure a Directory Mapping](#) (see page 215)

Delete Dialog

The Delete Dialog asks you to confirm the deletion of the object. You can delete the object by clicking OK.

Tasks Related to the Delete Dialog

The following task is related to the Delete Dialog:

- [Delete a Policy Server Object](#) (see page 81)

AuthValidate Directory Mappings Dialog

The AuthValidate Directory Mappings Dialog lets you authenticate users against one user directory and validate users against another user directory. Using this dialog, you can add, delete, or edit mapping configurations.

AuthValidate Directory Mappings Dialog Prerequisites

Before you can create a directory mapping, you must create one or more of the following Policy Server objects:

- User Directory Objects
- Policy Domain Objects
- Realm Objects
- Rule Objects

Policy objects may also see any of the following Policy Server Objects:

- Response Objects
- Response Group Objects

Navigate to the AuthValidate Directory Mappings Dialog

To access the AuthValidate Directory Mappings Dialog, from the Policy Server User Interface menu bar, select Advanced, AuthValidate Dir Mapping.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

AuthValidate Directory Mappings Dialog Fields and Controls

Current Mappings list

Lists all existing directory mappings.

Add button

Opens the Add AuthValidate Directory Mapping Dialog from which you can create a new directory mapping.

Delete button

Deletes the selected item from the Current Mappings list.

Edit button

Opens the item selected in the Current Mappings list for editing in the Edit AuthValidate Directory Mapping Dialog.

Tasks Related to the AuthValidate Directory Mappings Dialog

The following task is related to the AuthValidate Directory Mappings Dialog:

- [Configure Authentication and Validation Directory Mapping](#) (see page 217)

Add AuthValidate Directory Mapping Dialog

The Add AuthValidate Directory Mapping Dialog is where you create directory mappings that let you authenticate users against one directory and validate users against another.

Navigate to the Add AuthValidate Directory Mapping Dialog

To access the Add AuthValidate Directory Mapping Dialog in the Policy Server User Interface, from the AuthValidate Directory Mappings Dialog, click the Add.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Add AuthValidate Directory Mapping Dialog Fields and Controls

Authentication Directory field

Specifies the directory against which users will be authenticated.

Validation Directory drop down list

Specifies the directory against which users will be validated.

Mapped DNs group box

Contains radio buttons that determine how DNs are mapped:

Identical DN radio button

Maps the user's distinguished name (DN) exactly from the authentication directory to the validation directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the validation directory. Therefore, the validation directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID radio button

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the validation directory to identify the user.

Edit AuthValidate Directory Mapping Dialog

The Edit AuthValidate Directory Mapping dialog is where you edit existing directory mappings that let you authenticate users against one directory and validate users against another.

Navigate to the Edit AuthValidate Directory Mapping Dialog

To access the Edit AuthValidate Directory Mapping Dialog in the Policy Server User Interface, from the AuthValidate Mappings Dialog, select an entry in the Current Mappings list and click Edit.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Edit AuthValidate Mapping Directory Dialog Fields and Controls

Authentication Directory field

Specifies the directory against which users will be authenticated.

Validation Directory drop down list

Specifies the directory against which users will be validated.

Mapped DNs group box

Contains radio buttons that determine how DNs are mapped:

Identical DN radio button

Maps the user's distinguished name (DN) exactly from the authentication directory to the validation directory. For example, if a user is located in an LDAP authentication directory and has a DN of uid=jsmith, ou=people, o=security.com, the exact string is used to locate the user in the validation directory. Therefore, the validation directory must contain the DN: uid=jsmith, ou=people, o=security.com.

Universal ID radio button

Matches the value of the Universal ID attribute from the authentication directory with the value of the Universal ID field from the validation directory to identify the user. For example, if a company has different authentication and validation directories, a user who is authenticated can be identified by the validation directory because the Universal ID is identical for both user directories.

Chapter 25: Administrator and Related Dialogs Reference

This section contains the following topics:

[Administrator Properties Dialog](#) (see page 599)

[Domain Manager For These Domains Dialog](#) (see page 604)

Administrator Properties Dialog

The Administrator Properties Dialog allows you to create and edit SiteMinder administrators.

SiteMinder Administrator Properties Prerequisites

The following prerequisites must be met in order to successfully create an administrator using the Create Administrator Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Administrator Properties Dialog

There are multiple ways to access the Administrator Properties Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Administrator Properties Dialog to create or edit an administrator

1. In the object pane Click System Configuration.
2. From the menu bar, select Edit, System Configuration, Create Administrator

Administrator Properties Dialog General Group Box

The Administrator Properties dialog lets you define a SiteMinder administrator.

The first two fields on the dialog are as follows:

Name

Identifies an administrator.

Description

(Optional) Describes the administrator.

The following sections discuss how to setup authentication for an administrator, as well as how to assign a scope and tasks for the administrator's privileges.

Administrator Lookup Group Box

The Administrator Lookup group box is where you specify whether an administrator authenticates against the policy store or against an external user directory by selecting the corresponding radio button:

SiteMinder Database

Indicates that the administrator's username and password are stored in the policy store. When the administrator attempts to login to the Policy Server User Interface, the administrator's credentials are verified against the information in the policy store. If you select SiteMinder Database, you must enter password information in the Password group box.

External Directory

Indicates that the administrator's credentials are maintained in an external user directory. A user directory connection must be configured for this directory.

Note: This includes LDAP user directory connections with appropriate Lookup Start and Lookup End values.

The group box below the Administrative User Lookup group box is context-sensitive, changing from the Password group box to the Authentication group box according to your selection.

More information:

[User Directories](#) (see page 141)

Password Group Box

If you selected the SiteMinder Database radio button in the Administrator Lookup group box, the Password group box appears:

User Password field

Password the administrator will use to login to the Policy Server User Interface.

Note: The password is case-insensitive, except in cases where the password is stored in an Oracle policy store.

Confirm Password field

Administrator password a second time to confirm its accuracy.

The configured administrator will use the name that appears in the Name field and the password entered in the Password group box to authenticate when logging into SiteMinder.

Authenticate Group Box

If you selected the External Directory radio button in the Administrator Lookup group box, the Authentication group box appears:

User Directory drop-down list

The user directory that contains the administrator's authentication information.

The configured administrator will use the login name stored in the user directory to log into the Policy Server User Interface. The Policy Server uses this directory to locate the administrator.

Authentication Scheme drop-down list

Authentication method the Policy Server uses when the administrator attempts to login to the Policy Server User Interface.

The list contains authentication schemes that have been configured and saved in the policy store. The Policy Server installation process only configures the Basic authentication scheme. You can configure other authentication schemes with user name and password attributes that may be used to authenticate administrators. Procedures for configuring different authentication schemes indicate whether the scheme may be used to authenticate administrators.

Note: The case-sensitivity of passwords is determined by the type of directory in which the administrator passwords are stored.

More information:

[Authentication Schemes](#) (see page 231)

Scope Group Box

The Scope group box is where you specify whether an administrator can create, edit, and delete Policy Server objects for all policy domains, or only for selected policy domains by selecting the appropriate radio button.

System

Indicates that the administrator can create, edit, and delete objects in all policy domains.

Domains

Indicates that the administrator can create, edit, and delete objects in selected policy domains.

When the Domains radio button is selected, two additional user interface elements appear in the Scope group box:

Domains list

A list of policy domains for which the administrator can create, edit, and delete objects.

Add/Remove

Opens the Domain Manager For These Domains Dialog, which allows you to add or remove policy domains for which the administrator can create, edit, and delete objects

More information:

[Administrator Scope and Tasks](#) (see page 227)

[Domain Manager For These Domains Dialog](#) (see page 604)

Tasks Group Box

The Tasks group box is where you specify the tasks that the administrator has privileges to complete. The options presented in the tasks region changes according to the option you selected in the Scope group box.

The following option is always presented:

Manage Users

Allows the administrator to use the Manage User Accounts feature. See the *SiteMinder Policy Server Management Guide*.

If you selected System, the following additional options are presented:

Manage System and Domain Objects

Allows the administrator to create, edit, and delete all objects, which includes all objects in all policy domains, as well as system objects.

Manage Keys and Password Policies

Allows the administrator to manage password policies and use the global settings and key management features. For information about global settings and key management, see the *SiteMinder Policy Server Management Guide*.

View Reports

Allows the administrator to view all SiteMinder reports (see the *SiteMinder Policy Server Management Guide*).

Register Trusted Hosts

Allows the administrator to register a Trusted Host. For instructions on registering a Trusted Host, see the *SiteMinder Web Agent Installation Guide*.

If you selected Domains, the following options are presented in addition to Manage Users:

Manage Domain Objects

Allows the administrator to create, edit, and delete objects within the policy domains specified in the Scope group box.

Manage Password Policies

Allows the administrator to manage password policies and use the global settings feature. For information about global settings, see the *SiteMinder Policy Server Management Guide*.

View Reports

Allows the administrator to view SiteMinder reports for the policy domains specified in the Scope group box (see the *SiteMinder Policy Server Management Guide*).

More information:

[Administrator Scope and Tasks](#) (see page 227)

[Password Policies](#) (see page 479)

Tasks Related to the Administrator Properties Dialog

The following tasks are related to the Administrator Properties Dialog:

- [Configure an Administrator](#) (see page 226)
- [Precautions on Deleting or Editing an Administrator](#) (see page 229)

Domain Manager For These Domains Dialog

The Domain Manager For These Domains Dialog is where you assign policy domains.

Domain Manager For These Domains Dialog Prerequisites

The following prerequisites must be met in order to successfully assign policy domains for which an administrator can create, edit, and delete objects using the Domain Manager For These Domains Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- You must be in the process of creating or editing an administrator for which the Scope option set to Domains using the Administrator Properties Dialog.

Navigate to the Domain Manager For These Domains Dialog

Using the Domain Manager for These Domains Dialog in the Policy Server User Interface, you can assign policy domains for which an administrator can create, edit, and delete objects.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Domain Manager For These Domains Dialog

1. Open the Administrator Properties Dialog.
2. Set the Scope option to Domains.
3. In the Scope Group Box, Click the Add/Remove... button.

Domain Manager For These Domains Dialog Fields and Controls

Use the Domain Manager For These Domains Dialog to select the policy domains that an administrator can manage.

Note: If you do not select at least one policy domain in this dialog, the administrator cannot create, edit, or delete objects in the policy domain. In addition, an administrator may only view reports, and manage users and their passwords in directories associated with policy domains that are added in this dialog.

Current Members list

Lists policy domains that the administrator can manage.

Available Members list

Lists all available policy domains. Move policy domains from this list to the Current Members list to configure the administrator as a domain manager for those domains.

Arrow buttons

Move selected entries from one list to the other

Left Arrow button

Moves selected domains from the Current Members list to the Available Members list.

Right Arrow button

Moves selected domains from the Available Members list to the Current Members list

Tasks Related to the Domain Manager For These Domains Dialog

The following tasks are related to the Domain Manager for These Domains Dialog:

- [Configure a Domain Administrator](#) (see page 228)
- [Administrator Scope and Tasks](#) (see page 225)

Chapter 26: Authentication Scheme and Related Dialogs Reference

This section contains the following topics:

[Authentication Scheme Dialog](#) (see page 607)

[Certificate Mappings Dialog](#) (see page 665)

[Certificate Mapping Properties Dialog](#) (see page 667)

[Certificate Map Test Dialog](#) (see page 672)

Authentication Scheme Dialog

Authentication schemes determine the credentials and method required by the Policy Server to establish the identity of a user.

The contents of the Authentication Scheme Dialog change when you select a scheme from the Authentication Scheme Type drop-down list. The area below the Scheme Common Setup group box contains two tabs for each type of scheme. The Scheme Setup tab includes fields for information that you must supply in order for the scheme to function properly. The Advanced tab usually contains one or more read-only fields that display information that will be used to resolve authentication. The advanced tab also contains a check box that indicates whether or not an authentication scheme may be used to authenticate an Administrator.

Note: Only schemes that require a user name and password can be used to authenticate an administrator.

To configure an authentication scheme object, in the Policy Server User Interface, use the Authentication Scheme Dialog. Once you configure a scheme, you can assign it to realms using the SiteMinder Realms Dialog.

More information:

[Authentication Schemes](#) (see page 231)

[Policy Server Administrators](#) (see page 223)

[Grouping Resources in Realms](#) (see page 315)

Authentication Scheme Object Prerequisites

The following prerequisites must be met in order to successfully create an authentication scheme using the Authentication Scheme Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Authentication Scheme Dialog

There are multiple ways to access the Authentication Scheme Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog for a new user directory connection

1. In the object pane click System Configuration.
2. From the menu bar, select Edit, System Configuration, Create Authentication Scheme.

Navigate to the Authentication Scheme Dialog for an Existing Authentication Scheme

There are multiple ways to access the Authentication Scheme Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog for an existing authentication scheme

1. In the object pane, click on Authentication Schemes.
2. In the list pane, click on an existing directory connection
3. From the menu bar, select Edit, Properties of Authentication Scheme.

Authentication Scheme Dialog Fields and Controls

The contents of the Authentication Scheme Dialog change depending on the type of scheme you choose to configure. However, the following fields and controls are common to all authentication schemes:

Name

Name of the authentication scheme.

Description

(Optional) Brief description of the authentication scheme.

Scheme Common Setup Group Box

Authentication Scheme Type

Drop-down list of authentication scheme templates.

The list contains all authentication scheme templates predefined for use with the Policy Server's authentication libraries.

Protection Level

Level between 1 and 1000.

Protection levels provide an additional measure of flexibility in access control.

Note: Authentication schemes have a default protection level that can require change. Be sure to note protection levels as you define your schemes and assign them to realms. High protection levels are recommended for critical resources, while lower-level schemes are recommended for commonly accessible resources.

Password Policies Enabled for this Authentication Scheme Check Box

(Optional) Select this check box to enable password policies.

Note: Not all authentication schemes support password policies.

For information about the fields and controls for a specific type of authentication scheme, see the following:

- [Authentication Scheme Dialog—Basic Template](#) (see page 610)
- [Authentication Scheme Dialog—Basic over SSLTemplate](#) (see page 611)
- [Authentication Scheme Dialog—HTML Form Template](#) (see page 612)
- [Authentication Scheme Dialog—Windows Template](#) (see page 615)
- [Authentication Scheme Dialog—MS Passport Template](#) (see page 619)
- [Authentication Scheme Dialog—RADIUS CHAP/PAP Template](#) (see page 622)
- [Authentication Scheme Dialog—RADIUS Server Template](#) (see page 623)
- [Authentication Scheme Dialog—SafeWord Template](#) (see page 624)

- [Authentication Scheme Dialog—SafeWord HTML Form Template](#) (see page 625)
- [Authentication Scheme Dialog—SafeWord Template](#) (see page 624)
- [Authentication Scheme Dialog—SecurID Template](#) (see page 627)
- [Authentication Scheme Dialog—SecurID HTML Form Template](#) (see page 628)
- [Authentication Scheme Dialog—X509 Client Cert Template](#) (see page 631)
- [Authentication Scheme Dialog—X509 Client Cert and Basic Template](#) (see page 632)
- [Authentication Scheme Dialog—X509 Client Cert or Basic Template](#) (see page 634)
- [Authentication Scheme Dialog—X509 Client Cert and Form Template](#) (see page 637)
- [Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template](#) (see page 639)
- [Authentication Scheme Dialog—Anonymous Template](#) (see page 642)
- [Authentication Scheme Dialog—Custom Template](#) (see page 644)
- Authentication Scheme Dialog—SAML Artifact Template
- Authentication Scheme Dialog—SAML POST Template
- Authentication Scheme Dialog—Impersonation Template
- SAML 2.0 Auth Scheme Properties Dialog

More information:

[Password Policies](#) (see page 479)

Authentication Scheme Dialog—Basic Template

When you install the Policy Server, a Basic authentication scheme is created automatically. You can create additional instances of Basic authentication schemes with different protection levels.

Authentication Scheme Dialog—Basic Template—Scheme Setup Tab

The Scheme Setup tab for a Basic scheme is blank. You do not need to configure anything except the information in the Scheme Common Setup group box.

Authentication Scheme Dialog—Basic Template—Advanced Tab

Library

This field displays the library that the Policy Server uses to resolve Basic authentication.

Enable this scheme for SiteMinder Administrators

This check box is enabled and cannot be modified. Basic is the default authentication scheme for an administrator.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—Basic over SSLTemplate

The Basic over SSL authentication scheme collects username and password credentials from a user over a Secure Sockets Layer connection.

Authentication Scheme Dialog—Basic over SSLTemplate—Scheme Setup Tab

The Scheme Setup tab for Basic over SSL authentication is where you enter server and target information for your Basic over SSL scheme.

Server

Specifies the fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the Web server that is responsible for establishing an SSL connection. Although it is possible, this is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that the Policy Server uses to redirect a user's credentials over an SSL connection.

Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and name for the SSL Credentials Collector (SCC).

The target tells the SiteMinder Web Agent what to use to invoke the SCC. It completes the URL that the Policy Server uses to redirect the user's credentials over an SSL connection. The target can be customized in circumstances where proxy servers require specific URLs in order to support Basic over SSL authentication.

The default value for the Target field is:
/siteminder/nocert/smgetcred.scc?basic

Authentication Scheme Dialog—Basic over SSL Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a Basic over SSL scheme contains the following:

Library

Contains the name of the library used to process the authentication scheme.

The default authentication library value is smauthcert. This field contains the name of the library the Policy Server needs in order to process Basic over SSL authentication.

Note: Typically, you do not need to edit this field.

Parameter

Displays the string created from your entries in the Scheme Setup tab, which will be processed by the authentication library you entered in the Authentication Library field. The parameter consists of the Server and Target values you entered, with the addition of the ?basic string, which indicates the use of Basic user name and password authentication.

Enable this scheme for SiteMinder Administrators

This check box is disabled. Basic over SSL cannot be used to authenticate a Policy Server Administrator. The Policy Server communicates directly with a user directory or the Policy Store to authenticate an administrator, so SSL communication is not applicable.

Authentication Scheme Dialog—HTML Form Template

HTML Forms authentication schemes provide a method for authentication based on credentials gathered in a custom HTML form.

Authentication Scheme Dialog—HTML Form Template—Scheme Setup Tab

The Scheme Setup tab for HTML Forms authentication is where you enter the server, target and attribute list for the HTML Forms authentication scheme.

Server Name

Specifies the fully qualified domain name of the web server on which the FCC was installed.

Note: IP addresses are not supported.

The server does not have to be the same server on which the Agent is installed.

Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com.

This name is case-sensitive. The port is only required for communication over a non-default port.

For information about cookie domains, see the *SiteMinder Web Agent Guide*.

Note: If your network includes multiple cookie domains, you must configure a separate HTML Forms authentication scheme in each cookie domain in which you want to implement HTML Forms authentication.

Use SSL Connection check box

Select this check box to use an SSL connection to process HTML Forms authentication.

Target

Path and .fcc file used by the scheme.

The default path points to a virtual directory on the Web server specified in the Server Name field that was created during installation. The default target specifies the login.fcc file, a sample file that can be customized.

Allow Form Authentication Scheme to Save Credentials check box

Select this check box to allow user credentials to be saved.

You can use the special smsavecreds name/value pair in your .fcc file to allow a user to save their login credentials. If you select this check box and the .fcc file has an appropriate input (such as a check box on your HTML form), users may choose to save their login credentials so they can be used automatically the next time they log into the Web site.

When a user chooses to have credentials saved, the Policy Server instructs the Web Agent to create a persistent cookie with the user's credentials. The cookie remains in place for the duration specified in the SaveCredsTimeout configuration parameter for the Agent. The default is 30 days. The cookie allows Web Agents to authenticate a user based on the credentials saved in the cookie, rather than challenging the user to authenticate.

Additional Attribute List

(Optional) Attributes other than user name that will be collected from the user.

When listing attributes, begin with AL= and use commas to separate the user attribute names.

Example: AL=PASSWORD,age,zipcode

The AL= is a SiteMinder notation that indicates the list of attributes that should be considered. By default, the list of attributes is considered an AND-style query. The Policy Server compares all of the attribute values collected from the user to the corresponding attribute values in the user directory. If all of the attribute values match exactly, the user will authenticate successfully.

Note: You can authenticate users with attributes that contain multiple values. To specify that an attribute has multiple values, prefix the attribute name with a carat (^).

Example: If you are using a multi-valued "mail" attribute to authenticate users, you would specify "AL=^mail" to indicate that "mail" is multi-valued. A user can provide one of the valid values to successfully authenticate.

Limit: The values of a multi-valued attribute should not contain a carat. A value that contains a carat introduces the possibility of users being improperly authenticated. For example, if a value is 123^456, a user would be able to authenticate with 123 and 456, in addition to 123^456.

In order for SiteMinder to collect additional attributes, the .fcc file used by SiteMinder to generate a form for HTML Forms authentication must be modified to include the attributes.

When using additional attributes in an HTML Forms scheme, consider the following:

- The password value supplied by the form is limited to 255 characters. This limitation means that the ampersand-delimited list of AttributeName=URLEncodedAttributeValue elements must be less than 255 characters long.
- The PASSWORD attribute must always be specified in the form created from the .fcc file.
- The attributes are only used to match attribute values associated with the user who has been located in a directory by user name.
- The attribute names are case-sensitive. They must match exactly both the attribute names implemented by the directory and the attribute names specified in the authentication scheme.

Note: If you have installed the CA Software Development Kit, you can use the SiteMinder authentication API (see the *Developer's Guide for C*) to define additional notations.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

Authentication Scheme Dialog—HTML Form Template—Advanced Tab

Library

This field contains the name of the shared library that processes HTML Forms authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for HTML Forms authentication is `smauthhtml`.

Parameter

This field displays the Server and Target you entered in the Scheme Setup tab. It also displays any additional attributes you entered for the authentication scheme. The server and target make up the location of the `.fcc` file on the Web Server.

Enable this scheme for SiteMinder Administrators

This check box is disabled. HTML Forms authentication cannot be used to authenticate a Policy Server administrator.

Authentication Scheme Dialog—Windows Template

Integrated Windows Authentication (IWA) is a proprietary mechanism developed by Microsoft to validate users in pure Windows environments. IWA enforces Single Sign-On by allowing Windows to gather user credentials during the initial interactive desktop login process and subsequently transmitting that information to the security layer. SiteMinder, using the Windows Authentication scheme, secures resources by processing user credentials obtained by the Microsoft Integrated Windows Authentication infrastructure.

Previous versions of SiteMinder supported Windows authentication through the NTLM authentication scheme. However, this support was limited to environments with NT Domains or where the Active Directory service is configured to support legacy NT Domains in mixed mode.

The Windows authentication scheme allows SiteMinder to provide access control in deployments with Active Directories running in native mode, as well as Active Directories configured to support NTLM authentication. The Windows Authentication scheme replaces SiteMinder's previous NTLM authentication scheme. Existing NTLM authentication schemes continue to be supported and can be configured using the new Windows Authentication scheme.

Authentication Scheme—Windows Template—Scheme Setup Tab

The Scheme Setup tab for a Windows authentication scheme is where you specify a server and target for the scheme.

This scheme supports:

Select either Active Directory/LDAP or WinNT.

Your selection depends on where user data is stored and the namespace you selected when configuring the user directory connection in the Policy Server User Interface. If you select Active Directory/LDAP, user data must be stored in a directory configured using an AD or LDAP namespace. If you select AD, the Active Directory must be running in native mode. If you select WinNT, user data must be stored in a directory configured using the WinNT namespace.

Use Relative Target

Select this checkbox when you want to specify a relative path name for the Target or resource that this Windows authentication scheme protects. When this checkbox is selected, the Web Server Name field is dimmed.

Web Server Name

Enter the fully qualified domain name of the IIS web server that contains the virtual directory to which you want to redirect for Windows authentication.

Note: IP addresses are not supported.

The server specified in this field is where SiteMinder redirects in order to resolve authentication based on the user's current Windows login username and password.

Target

Enter the name of virtual directory and .ntc file used by SiteMinder to invoke the authentication scheme.

The Target field should point to a .ntc file which SiteMinder Agents interpret in order to authenticate users based on their current login user names and passwords. The Policy Server uses the following value by default:

/siteminderagent/ntlm/creds.ntc

User DN Lookup

If you selected the Active Directory/LDAP radio button, you must specify a lookup string for the user DN in this field. The formats for the User DN Lookup are described in the next section.

User DN Lookup Formats for Windows Authentication Schemes

When an authentication scheme is created using the Windows Authentication Template, authentication occurs at the site of the IIS Web server instead of the Policy Server. The Policy Server, in turn, depends on the IIS Web server to securely authenticate the user.

The IIS Web server passes to the Policy Server the user name and domain name, but not the user password, which is stored in a SAM account. SAM is an acronym for Security Accounts Manager, a Windows database that stores user passwords in a hashed format. The user name and domain name are passed from the IIS Web server to the Policy Server in this format:

`domain_name\user_name`

When you create a Windows authentication scheme for an Active Directory (AD) or LDAP user directory, you specify the User DN Lookup string that the Policy Server uses to locate the user in the user directory. The Policy Server searches any or all AD/LDAP user directories bound to the SiteMinder policy domain associated with the SiteMinder realm that contains the domain protected by the Windows authentication scheme.

There are two ways to specify the User DN Lookup string:

AD/LDAP Lookup

Specifies a fully qualified Distinguished Name (DN) that the Policy Server uses to look up the specified user in the user directories bound to the SiteMinder policy domain.

AD/LDAP Search

Specifies a Distinguished Name (DN) that the Policy Server uses to search for the specified user in the user directories bound to the SiteMinder policy domain.

Note: The Policy Server searches the user directories in the directory order specified by the policy domain object.

Note: The value specified in the User DN Lookup field supersedes the User Lookup Start and User Lookup End values specified in the user directory object associated with the AD/LDAP user directory.

AD/LDAP Lookup

In this scenario, there is one domain named ACompany, one user directory, and a user named John Smith who has the following attributes:

```
cn: jsmith
distinguishedName: CN=jsmith,CN=Users,DC=ACOMPANY,DC=com
sAMAccountName: jsmith
```

User credentials are passed from the IIS Web server to the Policy Server as follows:

`ACOMPANY\jsmith`

To map these credentials to a User DN in the user directory, configure the User DN Lookup string in the Windows authentication scheme as follows:

```
CN=%{UID},CN=Users,DC=%{DOMAIN},DC=com
```

When the authentication scheme is invoked, the %{UID} and %{DOMAIN} variables in the Lookup string are replaced by the user name and domain name passed to the Policy Server, as follows:

```
CN=jsmith,CN=Users,DC=ACOMPANY,DC=com
```

The user jsmith exists in the user directory and is located by the Policy Server.

AD/LDAP Search

In this scenario, there are two domains named ACompany and BCompany. The SiteMinder policy domain has two user directories associated with it: one is an AD directory and the other can be an AD or LDAP directory. The AD user directory contains a user named jsmith who has the following attributes:

```
cn: jsmith
distinguishedName: CN=jsmith,CN=Users,DC=ACOMPANY,DC=com
sAMAccountName: jsmith
```

The AD/LDAP user directory also contains a user named jsmith who has these attributes:

```
cn: jsmith
distinguishedName: CN=jsmith,CN=Users,DC=BCOMPANY,DC=com
sAMAccountName: jsmith
```

User credentials are passed from the IIS Web server to the Policy Server in this format:

```
domain_name\jsmith
```

To map these credentials to a User DN in one of the two user directories, configure the User DN Lookup string in the Windows authentication scheme as follows:

```
(sAMAccountName=%{UID})
```

When the authentication scheme is invoked, the %{UID} variable in the Lookup string is replaced by the user name passed to the Policy Server, as follows:

```
(sAMAccountName=jsmith)
```

The Policy Server searches for the specified user in the user directories in the directory order specified by the SiteMinder policy domain object. Depending on the directory order, the Policy Server may locate jsmith in the AD user directory or in the AD/LDAP user directory first. Once the Policy Server locates the first user, it stops searching.

Note: Because the Policy Server does not have access to passwords in a Windows environment, it cannot use them to disambiguate users with identical user names.

Important! There is a security risk associated with multiple domains. For example, consider a trust relationship between two domains: ACompany and BCompany. When the IIS Web server authenticates a user in a trusted domain such as BCompany, the Policy Server can grant the user access to the domain ACompany, even though the user may not be authenticated in this domain. To prevent this outcome, always include the domain name in the User DN Lookup string.

Authentication Scheme—Windows Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a Windows authentication scheme contains the following:

Library

Displays the library that the Policy Server uses to resolve Windows authentication. The default library is smauthntlm.

Enable this scheme for SiteMinder Administrators check box

Disabled and cannot be modified. Windows authentication may not be used as an authentication scheme for an administrator.

More information:

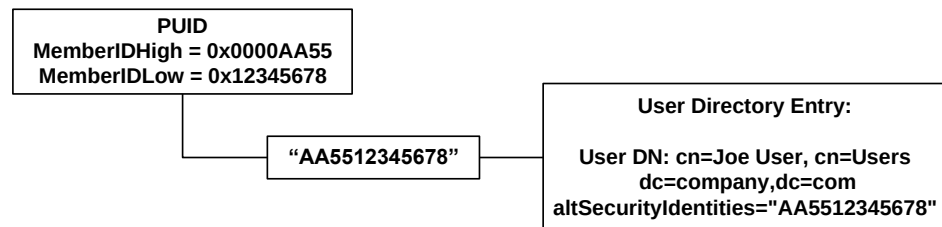
[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—MS Passport Template

Microsoft® .NET Passport is an online service that provides common Internet authentication across participating Web sites. Using a .NET Passport account, a user can move among participating sites without the need to authenticate with each of them. These sites become participating .NET Passport sites by implementing support for the .NET Passport authentication service through the .NET Passport single sign-in (SSI). This implementation includes a link to login through Passport using a common interface that supports co-branding. Users that are already logged into Passport will automatically be authenticated with the site, by means of a browser redirect to acquire Passport identity, and creation of site cookies containing the Passport identity. However, Passport does not authorize or deny a specific user's access to participating sites or resources.

The Passport identity does not contain any data for authorization. The only field that can be used to derive identity and subsequently permissions is the Passport Unique ID (PUID). SiteMinder uses this PUID to map to a local identity that is used to personalize the site and to authorize access to resources protected by SiteMinder policies.

The PUID is converted to a string and is mapped to a SiteMinder identity through a user directory attribute. In the following diagram, the user DN for "Joe User" is mapped to a Passport PUID through the directory attribute "altSecurityIdentities".



Since Passport authentication in SiteMinder is based on mapping the Passport identity to a SiteMinder user, registration is a required component for using Passport for authentication or personalization.

Registration is controlled through a registration URL that is configured in the authentication scheme. The configuration parameter is ?registrationurl=? followed by a registration page or the keyword "FORM=" and SiteMinder form URL. This model provides two methods for registration of Passport users.

The first method uses a site-provided web application to link the Passport identity with a SiteMinder user account through the attribute configured in the authentication scheme. This requires the site to develop the web pages to accept registration data and to provide a service for setting the user directory attribute. SiteMinder can be used to provide the interfaces to the user directory and the secure tunnel behind the firewall using a tunnel service.

The second method of registration leverages the SiteMinder FCC and the forms authentication model. When the registrationurl includes the keyword FORM=, the subsequent URL is treated as a forms redirect. SiteMinder provides the passport.fcc file to use as a template for developing a Passport registration page using forms.

SiteMinder provides a Passport Authentication Scheme template. The library name is smauthmspp. This authentication scheme can be used on both Windows and UNIX Policy Servers.

Authentication Scheme Dialog—MS Passport Template—Scheme Setup Tab

This tab is where you specify the configuration information for your implementation of the Passport authentication scheme.

Map Search Specification to Namespace group box

Select a namespace and define the appropriate mapping.

Registration URL

Registration redirect URL. This can specify a custom registration URL or a SiteMinder form.

Note: To include a SiteMinder form you must select the check box as described in Step.

The following is an example of a custom registration URL:

`http://passport.xanadu.local/registration/passportreg.asp`

The following is an example of a forms registration URL:

`FORM=http://passport.xanadu.local/forms/passport.fcc`

Use SiteMinder Forms for Registration check box

Select this check box if user registration will be completed through a SiteMinder form.

Allow anonymous login when no Passport identity exists check box

Select this check box if anonymous logins will be supported for users without Passport identities.

User DN

User Distinguished Name (DN) to be used to authorize access for a user who does not have a Passport. Specify a DN in this field if the Allow anonymous login when no Passport identity exists is selected.

This option allows Passport and Anonymous access to be combined into a single authentication scheme and thereby allows using Passport identities for personalization. This option should only be used when the authentication scheme is set to protection level 0, which is consistent with the Anonymous authentication scheme protection level.

More information:

[MS Passport Authentication Schemes](#) (see page 261)

Authentication Scheme Dialog—MS Passport Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a Passport scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a Passport authentication.

Parameter

This field displays the attribute name you entered on the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for Passport schemes. You cannot use Passport to authenticate Policy Server administrators.

Authentication Scheme Dialog—RADIUS CHAP/PAP Template

Before you can assign a RADIUS CHAP/PAP authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

Authentication Scheme Dialog—RADIUS CHAP/PAP—Scheme Setup Tab

This tab is where you specify the clear text password for CHAP/PAP authentication.

RADIUS CHAP/PAP Clear Text Password Attribute in Directory

Name of a user directory attribute.

The contents of this attribute will be used as the clear text password for authentication.

Authentication Scheme Dialog—RADIUS CHAP/PAP—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a RADIUS CHAP/PAP scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a RADIUS CHAP/PAP authentication.

Parameter

This field displays the attribute name you entered on the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for RADIUS CHAP/PAP schemes. You cannot use RADIUS CHAP/PAP to authenticate Policy Server administrators.

Authentication Scheme Dialog—RADIUS Server Template

Before you can assign a RADIUS Server authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

SiteMinder supports the RADIUS protocol by using the Policy Server as the RADIUS server and an NAS client as the RADIUS client. RADIUS Agents allow the Policy Server to communicate with the NAS client devices. In the RADIUS Server authentication scheme the Policy Server acts as a RADIUS server attached to the SiteMinder protected network.

This scheme accepts user name and password as credentials. Multiple instances of this scheme can be defined. This scheme does not interpret RADIUS attributes that may be returned by the RADIUS server in the authentication response.

Authentication Scheme Dialog—RADIUS Server Template—Scheme Setup Tab

This is where you enter the IP address and port for the RADIUS server, as well as the secret attribute required for authentication by the RADIUS server.

IP Address

IP address of the RADIUS Server.

Be sure to separate the IP address across the four fields. For example, to enter an IP Address of 12.34.56.78, type 12 in the first field, 34 in the second field, etc.

If you do not know the IP address of the RADIUS server, click the DNS Lookup button. Enter a host name in the DNS Lookup Dialog and click OK. The DNS Lookup Dialog closes and the IP address for the server appears in the appropriate fields of the RADIUS Server group box.

Port

Port number for the server.

The Port is used for communication with the RADIUS Server. If you do not specify a port number, the Policy Server uses 1645 as the default UDP port.

Secret the Confirm Secret

Text string to be used as the secret.

The secret is the user attribute that will be used by the RADIUS server as the clear text password.

Authentication Scheme Dialog—RADIUS Server Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a RADIUS Server scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a RADIUS Server authentication.

Parameter

This field displays the RADIUS server IP address and port number you entered in the Scheme Setup.

Enable this scheme for SiteMinder Administrators

This check box is enabled by default for RADIUS Server schemes and indicates that the scheme can be used by the Policy Server to authenticate administrators. This check box cannot be disabled.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—SafeWord Template

Before you can assign a SafeWord authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

This Authentication Scheme authenticates users against a SafeWord Server, including users who are logging in via the SafeWord hardware tokens. You can define multiple instances of this scheme. The exact configuration parameters of the SafeWord Server are specified by the SafeWord configuration file.

Authentication Scheme Dialog—SafeWord Template—Scheme Setup Tab

This is where you enter the location of the SafeWord Server configuration file. This file is required by the Policy Server in order to properly process a SafeWord authentication.

Secure Computing SafeWord Configuration File

Name of the SafeWord configuration file.

You must enter the full path and file name for the SafeWord configuration file. The Policy Server installation includes a sample configuration file:

SmSWEC.cfg located in *<policy server install dir>/config*

See SafeWord's documentation for information about the contents of this configuration file.

Authentication Scheme Dialog—SafeWord Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SafeWord scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a SafeWord authentication.

Parameter

This field displays the configuration file name and path you specified in the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is enabled by default for SafeWord schemes and indicates that the SafeWord scheme can be used by the Policy Server to authenticate administrators. This check box cannot be disabled.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—SafeWord HTML Form Template

Before you can assign a SafeWord authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

This Authentication Scheme authenticates users against a SafeWord Server in conjunction with a custom HTML form, including users who are logging in via the SafeWord hardware tokens. You can define multiple instances of this scheme. The exact configuration parameters of the SafeWord Server are specified by the SafeWord configuration file.

Authentication Scheme Dialog—SafeWord HTML Form Template—Scheme Setup Tab

This is where you enter the location of the SafeWord Server configuration file and specify details about the forms portion of the authentication process. This SafeWord Server configuration file is required by the Policy Server in order to properly process a SafeWord authentication.

Server Name

Fully qualified domain name of the web server on which the FCC was installed.

Note: IP addresses are not supported.

The server does not have to be the same server on which the Agent is installed. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-standard port.

This name is case-sensitive. For information about cookie domains, see the *SiteMinder Web Agent Guide*.

Note: If your network includes multiple cookie domains, you must configure a separate HTML Forms authentication scheme in each cookie domain in which you want to implement HTML Forms authentication.

Use SSL Connection check box

Select this check box if you want SiteMinder to use an SSL connection to process HTML Forms authentication.

Target

Path and .fcc file used by the scheme.

The default path points to a virtual directory on the Web server specified in the Server Name field that was created during installation. The default target specifies the login.fcc file, a sample file that can be customized.

Secure Computing SafeWord Configuration File

Name of the SafeWord configuration file.

You must enter the full path and file name for the SafeWord configuration file. The Policy Server installation includes a sample configuration file:

SmSWEC.cfg located in *<policy server install dir>/config*

See SafeWord's documentation for information about the contents of this configuration file.

More information:

[SafeWord Server and HTML Forms Authentication Schemes](#) (see page 269)

Authentication Scheme Dialog—SafeWord HTML Form Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SafeWord scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a SafeWord authentication.

Parameter

This field displays the configuration file name and path you specified in the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is enabled by default for SafeWord schemes and indicates that the SafeWord scheme can be used by the Policy Server to authenticate administrators. This check box cannot be disabled.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—SecurID Template

Before you can assign a SecurID authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

The RSA Ace/SecureID authentication schemes authenticate users logging in with ACE credentials, which include the user names, PINs, and TOKENCODEs. ACE usernames and passwords reside in the ACE/Server user store and can be changed by ACE/Server administrator. Single-use TOKENCODEs are generated by SecureID tokens.

Authentication Scheme Dialog—SecurID Template—Scheme Setup Tab

This is where you specify the attribute in the authentication user directory that contains the ACE Server user ID for the user.

ACE User ID Attribute Name in Directory field

Attribute name.

If user distinguished names (DNs) are different from the ACE server's user IDs, enter the name of the attribute in the user directory that contains the user ID in the ACE server. For example, if the user's ACE ID is jsmith, and her DN in an LDAP directory is cn=Jane Smith, ou=sales, o=security.com, then user profile attributes can be used to map back to the ACE user ID. For example, you can create a new attribute called "aceid," or use an existing unused attribute. The Policy Server will retrieve the value of this attribute and use that as the ACE user ID.

Authentication Scheme Dialog—SecurID Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SecurID scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve a SecurID authentication.

Parameter

This field displays the attribute name that you entered in the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is enabled by default for SecurID schemes and indicates that the SecurID scheme can be used by the Policy Server to authenticate administrators. This check box cannot be disabled.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—SecurID HTML Form Template

You can configure SecurID HTML Forms authentication using the SecurID HTML Form template in the Authentication Scheme Dialog.

The RSA Ace/SecureID authentication schemes authenticate users logging in with ACE credentials, which include the user names, PINs, and TOKENCODEs. ACE usernames and passwords reside in the ACE/Server user store and can be changed by ACE/Server administrator. Single-use TOKENCODEs are generated by SecureID tokens.

Authentication Scheme Dialog—SecurID HTML Form Template—Scheme Setup Tab

The Scheme Setup tab for SecurID HTML Form authentication is where you enter the server, target, and ACE attribute for SecurID authentication with HTML forms support.

Web Server Name

Specifies the fully qualified domain name of the web server on which the FCC was installed.

Note: IP addresses are not supported.

The server does not have to be the same server on which the Agent is installed. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Note: If your network includes multiple cookie domains, configure a separate HTML Forms authentication scheme in each cookie domain in which you want to implement HTML Forms authentication.

Use SSL Connection

Select this check box if you want SiteMinder to use an SSL connection to process HTML Forms authentication.

Target

Specifies the path SiteMinder uses to process SecurID authentication with HTML forms support. This Target value assumes that the Web Agent receiving the request processes the password services. If password services templates reside on another Web Agent, the full path to that target URL is required.

Default: /siteminderagent/forms/smpwservices.fcc

ACE User ID Attribute Name in Directory

Specifies the attribute that contains ACE IDs.

If user distinguished names (DNs) are different from the ACE server's user IDs, enter the name of the attribute in the user directory that contains the user ID in the ACE server. For example, if the ACE user ID is jsmith and the DN in an LDAP directory is cn=Jane Smith, ou=sales, o=security.com, you can map the user profile attributes back to the ACE user ID. In this example, you can create an attribute named aceid or use an existing unused attribute. The Policy Server retrieves the value of this attribute and uses it as the ACE user ID.

Authentication Scheme Dialog—SecurID HTML Form Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an HTML Forms scheme for SecurID authentication contains the following:

Library

This field contains the name of the shared library that processes SecurID and HTML Forms authentication.

Parameter

This field displays the Server and Target you entered in the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is disabled. SecurID and HTML Forms authentication cannot be used to authenticate a Policy Server administrator.

Configure the Scheme Setup Tab

You can use the Scheme Setup tab for SecurID authentication and HTML forms support.

To configure the Scheme Setup tab

1. In the Server Name field, enter the name of the Web server on which the FCC was installed.

The server does not have to be the same server on which the Agent is installed. This server name must be a fully qualified name with at least two dots. For example, server.myorg.org

2. If you want SiteMinder to use a secure sockets layer connection for this authentication scheme, select the Use SSL Connection check box.

3. In the Target field, enter the following:

/siteminderagent/pwcgi/smpwservicescgi.exe

This is the default path SiteMinder uses in order to process SecurID authentication with HTML forms support. This Target value assumes that password services will be processed by the Web Agent that receives a request. If password services templates reside on another Web Agent, the full path to that target URL should be specified in the Target field.

4. In the Ace User Attribute Name in Directory field, enter the name of the attribute that contains ACE IDs.

If user distinguished names (DNs) are different from the ACE server's user IDs, enter the name of the attribute in the user directory that contains the user ID in the ACE server. For example, if the user's ACE ID is jsmith, and her DN in an LDAP directory is cn=Jane Smith, ou=sales, o=security.com, then user profile attributes can be used to map back to the ACE user ID. For example, you can create a new attribute called "aceid,— or use an existing unused attribute. The Policy Server will retrieve the value of this attribute and use that as the ACE user ID.

5. Click Apply to save the authentication scheme.

Authentication Scheme Dialog—X509 Client Cert Template

In order for SiteMinder to authenticate users based on X.509 client certificates, you must configure an authentication scheme in the Policy Server User Interface and set up certificate mapping for the authentication scheme. The following procedures discuss the steps for configuring the authentication scheme.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert Template—Scheme Setup Tab

This is where you enter information about the SSL server and the .scc file for your X.509 client certificate authentication scheme.

Server Name

Fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the server that is responsible for establishing an SSL connection. Although it is possible, this server is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's X.509 certificate over an SSL connection. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and name for the SSL Credentials Collector (SCC).

The target tells the SiteMinder Web Agent what to use to invoke the SCC. It completes the URL that SiteMinder uses to redirect the user's credentials over an SSL connection. The target can be customized in circumstances where proxy servers require specific URLs in order to support Basic over SSL authentication.

SiteMinder provides a default path when you select the X.509 Client Certificate authentication scheme.

Authentication Scheme Dialog—X509 Client Cert Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a X.509 Client Certificate scheme contains the following:

Library

This field contains the name of the Policy Server library used to process the authentication scheme.

The default authentication library value is smauthcert. This field contains the name of the library the Policy Server needs in order to redirect user credentials over an SSL connection and verify the client's certificate.

Note: Typically, you do not need to edit this field.

Parameter

This field displays the string created from your entries in the Scheme Setup tab. This parameter is the string that Policy Server processes using the Authentication Library specified in the Scheme Setup tab. The Parameter consists of the Server and Target from the Scheme Setup tab, with the addition of the ?cert string, which indicates the use of client certificate authentication.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for X.509 Client Certificate schemes. You cannot use this scheme to authenticate Policy Server administrators.

Note: You must set up Certificate Mapping for this authentication scheme. Certificate Mapping links certificate information to a user entry in a directory.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert and Basic Template

In order for SiteMinder to authenticate users based on X.509 client certificates, you must configure an authentication scheme in the Policy Server User Interface and set up certificate mapping for the authentication scheme.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert and Basic Template—Scheme Setup Tab

The Scheme Setup tab for X.509 client certificate authentication is where you enter server and target information for your certificate authentication scheme.

Server Name

Fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the server that is responsible for establishing an SSL connection. Although it is possible, this is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's X.509 certificate over an SSL connection. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and name for the SSL Credentials Collector (SCC).

The target tells the SiteMinder Web Agent what to use to invoke the SCC. It completes the URL that SiteMinder uses to redirect the user's credentials over an SSL connection. The target can be customized in circumstances where proxy servers require specific URLs in order to support Basic over SSL authentication.

The Policy Server User Interface provides a default path when you select the X.509 Client Certificate authentication scheme.

Note: When this authentication scheme is associated with a realm, users who attempt to access a resource in the protected realm must have a valid X.509 client certificate that can be verified by SiteMinder using a server certificate. In addition, the user must supply a user name and password that can be located and verified in a user directory associated with the realm's policy domain.

More information:

[Grouping Resources in Realms](#) (see page 315)

Authentication Scheme Dialog—X509 Client Cert and Basic Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a X.509 Client Certificate and Basic scheme contains the following:

Library

This field contains the name of the Policy Server library used to process the authentication scheme.

The default authentication library value is smauthcert. This field contains the name of the library that the Policy Server needs to redirect user credentials over an SSL connection and verify the client's certificate.

Note: Typically, you do not need to edit this field.

Parameter

This field displays the string created from your entries in the Scheme Setup tab. This string is the parameter that the Policy Server processes using the Authentication Library specified in the Scheme Setup tab. The Parameter consists of the Server and Target from the Scheme Setup tab, with the addition of the ?cert+basic string, which indicates the use of client certificate and basic authentication.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for X.509 Client Certificate schemes. You cannot use this scheme to authenticate Policy Server administrators.

Note: You must set up Certificate Mapping for this authentication scheme. Certificate Mapping links certificate information to a user entry in a directory.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert or Basic Template

In order for SiteMinder to authenticate users based on X.509 client certificates, you must configure an authentication scheme in the Policy Server User Interface and set up certificate mapping for the authentication scheme.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X.509 Client Cert or Basic Template—Scheme Setup Tab

The Scheme Setup tab for X.509 Client Certificate or Basic authentication is where you enter server and target information for your certificate authentication scheme.

Server Name

Fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the server that is responsible for establishing an SSL connection. Although it is possible, this is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's X.509 certificate over an SSL connection. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and name for the SSL Credentials Collector (SCC).

The target tells the SiteMinder Web Agent what to use to invoke the SCC. It completes the URL that SiteMinder uses to redirect the user's credentials over an SSL connection and process certificate authentication. The target can be customized in circumstances where proxy servers require specific URLs in order to support Basic over SSL authentication.

SiteMinder provides a default path when you select the X.509 Client Certificate authentication scheme.

Basic Credentials Over SSL check box

Select this check box if basic user name/password credentials should be delivered over an SSL connection.

Basic Server Name

Fully qualified name of the SSL server. Enter a value in this field if the Basic Credentials Over SSL check box is selected.

This server is responsible for establishing an SSL connection for basic authentication. Although it is possible, this server is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's credentials over an SSL connection. The syntax for the server differs slightly based on the type of Web server on which the Web Agent is installed. The following definitions describe the proper syntax for the Server field based on the Web server type:

IIS or Oracle iPlanet web servers— *servername.domain:port*

Apache—*servername.domain:port*

Domain names must contain at least 2 periods. For example:
.security.com

Basic Target

Path and name for the SSL Credentials Collector (SCC).

The target tells the SiteMinder Web Agent what to use to invoke the SCC. It completes the URL that SiteMinder uses to redirect the user's credentials over an SSL connection for basic authentication. The target can be customized in circumstances where proxy servers require specific URLs in order to support Basic over SSL authentication.

SiteMinder provides a default path when you select the X.509 Client Certificate authentication scheme.

Note: When this authentication scheme is associated with a realm, users who attempt to access a resource in the protected realm may be authenticated with a valid X.509 client certificate or by a user name and password that can be located and verified in a user directory associated with the realm's policy domain.

More information:

[Domains](#) (see page 307)

Authentication Scheme Dialog—X509 Client Cert or Basic Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a X.509 Client Certificate and Basic scheme contains the following:

Library

This field contains the name of the Policy Server library used to process the authentication scheme.

The default authentication library value is `smauthcert`. This field contains the name of the library that the Policy Server needs in order to redirect user credentials over an SSL connection and verify the client's certificate.

Note: Typically, you do not need to edit this field.

Parameter

This field displays the string created from your entries in the Scheme Setup tab. This string is the parameter that the Policy Server processes using the Authentication Library specified in the Scheme Setup tab. The Parameter consists of the Server and Target from the Scheme Setup tab, with the addition of the ?certorbasic string, which indicates the use of both certificate and basic authentication.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for X.509 Client Certificate schemes. You cannot use this scheme to authenticate Policy Server administrators.

Note: You must set up Certificate Mapping for this authentication scheme. Certificate Mapping links certificate information to a user entry in a directory.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert and Form Template

In order for SiteMinder to authenticate users based on X.509 client certificates, you must configure an authentication scheme in the Policy Server User Interface and set up certificate mapping for the authentication scheme.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert and Form Template—Scheme Setup Tab

The Scheme Setup tab is where you enter server and target information for your X.509 Client Certificate and HTML Forms authentication scheme.

Note: When this authentication scheme is associated with a realm, users who attempt to access a resource in the protected realm must have a valid X.509 client certificate that can be verified by SiteMinder using a server certificate. In addition, the user must supply forms credentials that can be located and verified in a user directory associated with the realm's policy domain.

Server Name

Fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the server that is responsible for establishing an SSL connection. Although it is possible, this is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's X.509 certificate over an SSL connection. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and .scc file used by the scheme.

The default path points to a virtual directory on the Web server specified in the Server Name field that was created by the SiteMinder Agent installation. The default target specifies the smgetcred.scc file, a sample file that can be customized.

Additional Attribute List

(Optional) Specifies the attributes other than user name that will be collected from the user. When listing attributes, begin with AL= and use commas to separate the user attribute names.

Example: AL=PASSWORD,age,zipcode

The AL= is a SiteMinder notation that indicates the list of attributes that should be considered. By default, the list of attributes is considered an AND-style query. The Policy Server compares all of the attribute values collected from the user to the corresponding attribute values in the user directory. If all of the attribute values match exactly, the user will authenticate successfully.

Note: You can authenticate users with attributes that contain multiple values. To specify that an attribute has multiple values, prefix the attribute name with a carat (^).

Example: If you are using a multi-valued "mail" attribute to authenticate users, you would specify "AL=^mail" to indicate that "mail" is multi-valued. A user can provide one of the valid values to successfully authenticate.

Limit: The values of a multi-valued attribute should not contain a carat. A value that contains a carat introduces the possibility of users being improperly authenticated. For example, if a value is 123^456, a user would be able to authenticate with 123 and 456, in addition to 123^456.

In order for SiteMinder to collect additional attributes, the .fcc file used by SiteMinder to generate a form for HTML Forms authentication must be modified to include the attributes.

More information:

[Domains](#) (see page 307)

Authentication Scheme Dialog—X509 Client Cert and Form Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a X.509 Client Certificate and HTML Forms authentication scheme contains the following:

Library

This field contains the name of the Policy Server library used to process the authentication scheme.

The default authentication library value is smauthcert. This field contains the name of the library that the Policy Server uses to verify the client's certificate.

Note: Typically, you do not need to edit this field.

Parameter

This field displays the string created from your entries in the Scheme Setup tab. This string is the parameter that the Policy Server processes using the Authentication Library specified in the Scheme Setup tab. The Parameter consists of the Server and Target from the Scheme Setup tab, with the addition of the ?cert+forms string, which indicates the use of X.509 Client Certificate and HTML Forms authentication scheme.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for X.509 Client Certificate schemes. You cannot use this scheme to authenticate Policy Server administrators.

Note: You must set up Certificate Mapping for this authentication scheme. Certificate Mapping links certificate information to a user entry in a directory.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template

In order for SiteMinder to authenticate users based on X.509 client certificates, you must configure an authentication scheme in the Policy Server User Interface and set up certificate mapping for the authentication scheme.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—X.509 Client Cert or HTML Forms Template—Scheme Setup Tab

The Scheme Setup tab for X.509 Client Certificate or HTML Forms authentication is where you enter server and target information for your certificate authentication scheme.

Note: When this authentication scheme is associated with a realm, users who attempt to access a resource in the protected realm may be authenticated with a valid X.509 client certificate or by forms credentials that can be located and verified in a user directory associated with the realm's policy domain.

Server Name

Fully qualified domain name of the SSL server.

Note: IP addresses are not supported.

This is the server that is responsible for establishing an SSL connection. Although it is possible, this is usually not the same server where the Web Agent is installed.

The server acts as the beginning of the URL that SiteMinder uses to redirect a user's X.509 certificate over an SSL connection. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

Target

Path and name for the SSL Credentials Collector (SFCC).

The target is `<URI/name>.sfcc`

The target tells the SiteMinder Web Agent what to use to invoke the SFCC. It completes the URL that SiteMinder uses to redirect the user's credentials over an SSL connection and process certificate authentication.

SiteMinder provides a default path when you select the X.509 Client Certificate or HTML Forms authentication scheme.

Additional Attribute List

(Optional) Specifies the attributes other than user name that will be collected from the user. When listing attributes, begin with AL= and use commas to separate the user attribute names.

Example: AL=PASSWORD,age,zipcode

The AL= is a SiteMinder notation that indicates the list of attributes that should be considered. By default, the list of attributes is considered an AND-style query. The Policy Server compares all of the attribute values collected from the user to the corresponding attribute values in the user directory. If all of the attribute values match exactly, the user will authenticate successfully.

Note: You can authenticate users with attributes that contain multiple values. To specify that an attribute has multiple values, prefix the attribute name with a carat (^).

Example: If you are using a multi-valued "mail" attribute to authenticate users, you would specify "AL=^mail" to indicate that "mail" is multi-valued. A user can provide one of the valid values to successfully authenticate.

Limit: The values of a multi-valued attribute should not contain a carat. A value that contains a carat introduces the possibility of users being improperly authenticated. For example, if a value is 123^456, a user would be able to authenticate with 123 and 456, in addition to 123^456.

In order for SiteMinder to collect additional attributes, the .fcc file used by SiteMinder to generate a form for HTML Forms authentication must be modified to include the attributes.

Use Alternate FCC Location check box

Select this check box if you want to specify a target to which forms collection is redirected. A possible use of such redirection is to avoid multiple SSL-based challenges.

Alternate Server Name

Name of the Web server on which the alternate FCC is installed. Enter a value in this field if the Use Alternate FCC Location is selected.

Alternate Target

Path and .fcc file. Enter a value in this field if the Use Alternate FCC Location is selected.

The default path points to a virtual directory on the Web server specified in the Alternate Server Name field that was created by the SiteMinder Agent installation. The default target specifies the login.fcc file, which is a sample file that can be customized.

Use SSL Connection check box

Select this check box if HTML Forms credentials should be delivered over an SSL connection. You must check the Use Alternate FCC Location check box to access the Use SSL Connection checkbox.

More information:

[Domains](#) (see page 307)

Authentication Scheme Dialog—X509 Client Cert or HTML Form Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a X.509 Client Certificate or HTML Forms authentication scheme contains the following:

Library

This field contains the name of the Policy Server library used to process the authentication scheme.

The default authentication library value is `smauthcert`. This field contains the name of the library that the Policy Server needs in order to redirect user credentials over an SSL connection and verify the client's certificate.

Note: Typically, you do not need to edit this field.

Parameter

This field displays the string created from your entries in the Scheme Setup tab. This string is the parameter that the Policy Server processes using the Authentication Library specified in the Scheme Setup tab. The Parameter consists of the Server and Target from the Scheme Setup tab, with the addition of the `?certorform` string, which indicates the use of either certificate or HTML form.

Enable this scheme for SiteMinder Administrators

This check box is disabled by default for X.509 Client Certificate schemes. You cannot use this scheme to authenticate Policy Server administrators.

Note: You must set up Certificate Mapping for this authentication scheme. Certificate Mapping links certificate information to a user entry in a directory.

More information:

[Certificate Mapping](#) (see page 293)

Authentication Scheme Dialog—Anonymous Template

Before you can assign an Anonymous authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

An anonymous authentication scheme allows non-registered users to access specific Web content. When a user accesses a resource that has anonymous authentication, SiteMinder assigns the user a Global User Identification (GUID). SiteMinder places this GUID in a persistent cookie on the user's browser so that the user can access specific resources without being challenged to authenticate.

Authentication Scheme Dialog—Anonymous Template—Scheme Setup

This is where you specify the distinguished name (DN) of a user. When an anonymous user attempts to access any resources on the network, the DN is what SiteMinder uses to determine the user's privileges.

DN

Distinguished name for guests that will be used by the Anonymous scheme.

This DN determines the privileges of an anonymous user. When you assign the Anonymous authentication scheme to a realm, then bind policies to the guest DN, an anonymous user has access privileges specified for the guest DN.

Note: You must specify a DN for a user, not a group. The Anonymous scheme will only function if you supply the DN of a user.

Authentication schemes are associated with realms. Realms are located under policy domains. Policy domains allow connections to specific user directories. When you specify a DN in this field, the Policy Server checks for the DN in the user directories specified in the realm's policy domain. The Policy Server checks the user directories in the search order defined for the policy domain.

To search for the DN you want to assign as the guest DN, click Lookup. The User Lookup Dialog appears. When you select a DN from the search results in the User Lookup Dialog and click OK, that DN appears in the DN field.

More information:

[User Directories](#) (see page 141)

[User Directory Dialog](#) (see page 545)

Authentication Scheme Dialog—Anonymous Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an Anonymous scheme contains the following:

Library

This field displays the name of the library that the Policy Server uses to resolve an Anonymous authentication.

Parameter

This field displays the guest DN you entered in the DN field of the Scheme Setup tab.

Enable this scheme for SiteMinder Administrators

This check box is disabled. The Anonymous scheme does not require a user name and password, and therefore it cannot be enabled as a method for Policy Server administrator authentication.

Authentication Scheme Dialog—Custom Template

Before you can assign a Custom authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog.

If you want to use an authentication method that is not provided by SiteMinder, you can create a custom authentication scheme. Once you create a Custom scheme, you must configure the scheme in the SiteMinder Authentication Dialog.

Note: For information on configuring an `smauthetsso` custom authentication scheme, which is needed for enabling single sign-on between SiteMinder and CA Single Sign-On, see [Configure an `smauthetsso` Custom Authentication Scheme](#) (see page 992).

Authentication Scheme Dialog—Custom Template—Scheme Setup Tab

This is where you specify a library that process credentials for the Custom scheme, as well as the parameter that must be passed to the library.

Library

Name of the library the Policy Server uses to process the custom authentication scheme.

The library is the shared library you created using the APIs.

Note: If you have installed the Software Development Kit, you can see the *Developer's Guide for C* for more information about creating a Custom authentication scheme.

Secret and Confirm Secret Shared secret.

If the Custom authentication scheme requires a shared secret for encryption of credentials, you can enter the secret in this field.

Parameter

String that will be passed from the Policy Server to the library to process the Custom authentication.

Enable this scheme for SiteMinder Administrators check box

Select this check box if you want the Custom scheme to be used to authenticate Policy Server administrators.

If this check box is selected, the scheme can be used for authenticating administrators.

More information:

[Policy Server Administrators](#) (see page 223)

Authentication Scheme Dialog—Custom Template—Advanced Tab

The Advanced tab is not used by a Custom authentication scheme.

Authentication Scheme Dialog--SAML Artifact Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML artifact authentication scheme for SAML 1.x. After the scheme is configured, you can assign it to a realm.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

More Information:

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 645)

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 652)

Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x artifact authentication is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML Artifact Template are as follows:

Affiliate Name

Names of the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name of a consumer in the Affiliate domain at the producer site.

Important! For the SAML artifact profile, the producer sends the assertion over a protected back channel to the consumer. If you are using basic authentication to protect the back channel, the value of this field is the name of the consumer. If you are using client certificate authentication for the back channel, the value of this field is the alias of the client certificate stored in the smkeydatabase.

Redirect Mode drop-down list

Indicates the method by which the SAML credential collector redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data (default)

Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:

`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application—that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects—one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

—`Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

—`Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

PersistAttributes

Redirects the user through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

Password

Defines the password that the consumer uses to identify itself to the producer.

This password must match the password entered for the consumer at the producer site.

Verify Password

Confirms the password that you specified in the Password field.

Company Source ID

Defines the source ID of the producer.

The SAML specification standard defines a source ID as a 20-byte binary, hex-encoded number that identifies the producer. The consumer uses this ID to identify an assertion issuer.

Enter a number that matches the Source ID that the producer uses. At the producer, this ID is located in the SAML assertion generator properties file, `AMAssertionGenerator.properties`, which is located in the following directory:

policy_server_home/config/properties

The default Company Source ID value is:

`b818452610a0ea431bff69dd346aeff83128b6a`

SAML Version

Specifies the version of the SAML specification with which the assertion is generated. The choices are 1.0 or 1.1. SAML 1.1 is the default.

There must be a match between the SAML protocol and assertion versions used by the producer or consumer. For example, if a producer using SAML 1.1 receives a SAML 1.0 request, it returns an error. If a consumer using SAML 1.1 receives a SAML 1.0 assertion embedded into a SAML 1.1 protocol element, it returns an error.

Assertion Retrieval URL

Defines the URL of the service at the producer where the consumer retrieves the SAML assertion. Specifically, this URL identifies the location of the Assertion Retrieval Service, which must be a URL accessed over an SSL connection.

If the assertion retrieval service at the producer is part of a realm using the Basic over SSL authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/assertionretriever`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

If the assertion retrieval service at the producer is part of a realm using the X.509 client certificate authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/certassertionretriever`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Authentication

Specifies the authentication method for the realm at the producer that contains the Assertion Retrieval Service. The value of this field determines the type of credentials that the SAML credential collector at the consumer must provide to access the Assertion Retrieval Service and retrieve the SAML assertion.

The Assertion Retrieval Service retrieves the assertion from the Policy Server at the producer, and then sends it to the consumer across an SSL back channel. We recommend protecting the Assertion Retrieval Service.

Select one of the following:

- Basic Auth if the Assertion Retrieval Service at the producer is part of a realm protected by a Basic over SSL authentication scheme.
- Client Cert if the Assertion Retrieval Service at the producer is part of a realm protected by an X.509 client certificate authentication scheme.

When you select this option, configure access to the Assertion Retrieval Service with a client certificate.

- No Auth if the Assertion Retrieval Service is unprotected.

Depending on the authentication method you select, there can be some additional configuration required.

- When you select Basic Auth, no additional configuration is required at the producer or consumer, unless the certificate of the Certificate Authority that established the SSL connection is not in the consumer's smkeydatabase. If the appropriate certificate is not in the key database, you must import it.
- If you select Client Cert, there are several configuration steps at the consumer and the producer that you must complete to access the Assertion Retrieval Service with the client certificate.

For client certificate authentication, you can use non-FIPS 140 encrypted certificates to secure the back channel even if the Policy Server is operating in FIPS-only mode. However, for FIPS-only installations use certificates only encrypted with FIPS 140-compatible algorithms.

Audience

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.companya.com/SampleAudience`

Issuer

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, producerA.ca.com

The consumer accepts assertions from only this issuer. The administrator at the producer determines the issuer.

Note: The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:
siteminder_home/Config/properties/AMAssertionGenerator.properties

Search Data XPATH

Defines an Xpath query that tells the authentication scheme where to look in the SAML assertion for data used to look up a user in the user directory. The Xpath query and the Search Specification value map data from a SAML assertion to a user store entry.

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

Select the namespace and click Edit to the right of the search specification scroll box. The SiteMinder Authentication Scheme Namespace Mapping dialog box opens. In this dialog, you map the data collected by the Xpath query to a namespace attribute.

Example:

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Use  
rname/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/  
SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query string extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site.

Using the following abbreviated syntax extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax:

```
"substring-after(//SMprofile/NVpair[1]/text(),"header:uid=")"
```

D-Sig Alias

Specifies the alias of the public key in the smkeydatabase that is used to verify the digital signature which signed the assertion. This alias is associated with the certificate in the smkeydatabase. When you list the certificates in the smkeydatabase, the type of certificate used for verification is a CertificateEntry or KeyEntry certificate.

Namespace

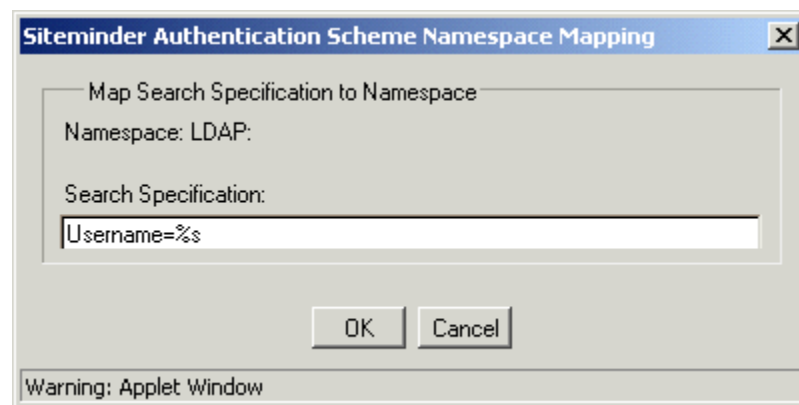
Selectable list of namespace types and defined search specifications.

Search Specification

Enter the attribute that the authentication scheme uses to search a namespace.

Edit button

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.



For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. Compare this string against the user directory to find the correct record for authentication.

Additional Configuration

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an SAML artifact scheme is where you define information about the shared library that processes SAML 1.x artifact authentication.

The fields on the Advanced tab are as follows:

Library

This field contains the name of the shared library that processes SAML artifact authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for HTML Forms authentication is `smauthsaml`.

Parameter

This field displays the information you entered in the Scheme Setup tab.

Authentication Scheme Dialog--SAML POST Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML POST Template authentication scheme for the SAML 1.x POST profile. After configuring the authentication scheme and assign it to a realm.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

More Information:

[Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab](#) (see page 653)

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 658)

Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x POST is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML POST Template are as follows:

Affiliate Name

Names the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name for a consumer in the Affiliate domain at the producer site.

SAML Version

Specifies the SAML version (inactive; the value defaults to 1.1, indicating that POST profile assertions are compliant with SAML version 1.1).

The SAML producer and consumer must be generating and consuming assertions and responses that are the same version.

Audience

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer site.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.ca.com/SampleAudience`

Redirect Mode

Specifies the method by which the SAML credential collector servlet redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data

(Default). Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:

`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects—one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

- `Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

- `Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

- `Netegrity.smSessionID` attribute represents the SiteMinder session ID

- `Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

PersistAttributes

User is redirected through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

Assertion Consumer URL

Specifies the URL of the assertion consumer (synonymous with the SAML credential collector). This URL is where the user's browser must POST the generated assertion.

The default URL is:

`http://consumer_server:port/affwebservices/public/samlcc`

consumer_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway. For example:

`http://www.discounts.com:85/affwebservices/public/samlcc`

Issuer

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, producerA.ca.com

The consumer accepts assertions from only this issuer. The administrator determines the issuer at the producer.

Note: The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:
policy_server_home/Config/properties/AMAssertionGenerator.properties

Dsig Issuer DN

Specifies the distinguished name of the certificate issuer that signs the SAML POST response. The SAML POST response must be digitally signed by the producer. When the consumer receives the response, it must verify the signature using the data in this parameter and the Serial Number parameter. These two parameters list the issuer of the certificate who signed the response.

The public key used to verify the signature must be in the smkeydatabase. More information about the smkeydatabase can be found in Using Key Databases for Federation Security Services.

Serial Number

Specifies the serial number (a hexadecimal string) of the certificate of the consumer in the smkeydatabase key store. This value is used with the Dsig Issuer DN to locate the certificate for digitally signing the SAML POST response.

Search Data XPATH

Defines an XPATH query that the authentication scheme applies to the SAML assertion. The query tells the authentication scheme where to look in the assertion document. The data obtained by the query is used to look up a user in the user directory.

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

Example

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Username/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site. The string "substring-after(/SMprofile/NVpair[1]/text(),"header:uid=")" extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax.

Namespace

Selectable List of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

Edit button

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification, which maps data from a SAML assertion to a user store entry, which enables the SAML authentication scheme to locate the correct user record for authentication.

Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.

For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user directory to find the correct record for authentication.

Additional Configuration

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

Authentication Scheme Dialog--SAML POST Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SAML 1.x POST scheme is where you define information about the shared library that processes SAML 1.x POST authentication.

The fields on the Advanced tab are as follows:

Library

This field contains the name of the shared library that processes SAML POST authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML POST authentication is `smauthsaml`.

Parameter

This field displays the information you entered in the Scheme Setup tab.

Authentication Scheme Dialog -- SAML 2.0 Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML 2.0 Template authentication scheme.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

SAML 2.0 Authentication Properties--Scheme Setup Tab

The Scheme Setup tab for SAML 2.0 authentication lets you specify how the Service Provider communicates with the Identity Provider to retrieve the assertion.

The fields on this tab are:

SP ID

Specifies a URI that uniquely identifies the Service Provider.

Note: The value that you enter must match the value of the ID specified for the corresponding Service Provider object that you configure at the Identity Provider.

IdP ID

Specifies a URI that uniquely identifies the Identity Provider from which assertions for this Service Provider are issued.

The Service Provider accepts assertions from only this IdP.

Note: The value that you enter for the issuer must match the value of the IdP ID configured at the Identity Provider site.

SAML Version

Specifies the SAML version (disabled; the value defaults to 2.0, indicating that assertions sent to this IdP ID must be compliant with SAML version 2.0).

D-Sig Info Group Box

Contains fields and controls that allow you to specify digital signature information:

Disable Signature Processing

Disables all signature processing, that is, signing and verification of signatures, for this Service Provider.

Note: Signature processing *must* be enabled in a production environment. Select the Disable Signature Processing option only for debugging purposes.

Signing Options

Displays a dialog with the settings for digital signature configuration, specifically the Signing Alias and the Signature Algorithm.

Issuer DN

Specifies the distinguished name of the issuer of the certificate. This value is used with the Serial Number to locate the certificate of the Service Provider in the SMKeyDatabase file.

Note: This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

Serial Number

Specifies the serial number (a hexadecimal string) of the certificate of the Service Provider in the key store. This value is used with the IssuerDN to locate the certificate in the SMKeyDatabase key store.

Note: This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

Additional Configuration

Opens the SAML 2.0 Auth Scheme Properties dialog. From this dialog, you can specify additional configuration details for the authentication scheme.

SAML 2.0 Authentication Properties--Advanced Tab

The Advanced tab of the Authentication Scheme Properties dialog for SAML 2.0 is where you define information about the shared library that processes SAML 2.0 authentication.

Library

Contains the name of the shared library that processes SAML 2.0 authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML 2.0 authentication is `smauthsaml`.

Parameter

Displays the information you entered in the Scheme Setup tab.

Authentication Scheme Dialog—Custom Template for Federation

If you have purchased SiteMinder Federation Security Services, you can configure a custom authentication template to simplify configuration of realms for SAML and WS-Federation authentication schemes. You can create a single target realm for multiple Producers/Identity Providers/Account Partners.

Part of this feature requires the configuration of a single custom authentication scheme.

Note: You should have already configured SAML or WS-Federation authentication scheme for each Producer/Identity Provider/Account Partner prior to configuring a custom template.

Authentication Scheme Dialog—Custom Template for Federation —Scheme Setup Tab

The Scheme Setup tab for a custom template is where you name the library to be used by the custom authentication scheme.

Library field

Specifies the library name for the custom single authentication scheme. Enter `smauthsinglefed` for the library name.

Secret field and Confirm Secret field

Disregard these fields.

Parameter field

Specifies which SAML (1.x or 2.0) or WS-Federation authentication schemes SiteMinder should use for federation. The custom authentication scheme chooses the correct SAML or WS-Federation authentication scheme based on the entity that issues the assertion. The entity is identified by a query parameter it adds to the assertion and then passes to the entity where the target resource resides.

Note: For SAML 1.x, each SAML authentication scheme associated with the custom authentication scheme must be configured with a unique Issuer (SAML 1.x artifact) or Source ID (SAML 1.1 POST). If the same Issuer/Source ID value is used by more than one scheme and the SCHEMESET is set to `SAML_ALL`, `SAML_POST`, or `SAML_ARTIFACT`, Federation Security Services does not know which SAML 1.x authentication scheme in the list to use.

Specify one of the following options:

SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

Specifies the list of SAML authentication scheme names to use. If you configured an artifact scheme called `artifact_Idp1` and POST profile scheme called `samlpost_IdP2`, you will enter these schemes.

SCHEMESET=SAML_ALL;

Specifies all the schemes you have configured. The custom authentication scheme will enumerate all the SAML authentication schemes and find the one with the correct Provider Source ID for the request.

SCHEMESET=SAML_POST;

Specifies all the SAML POST Profile schemes you have configured. The custom authentication scheme will enumerate the POST Profile schemes and find the one with the correct Provider Source ID for the request.

SCHEMESET=SAML_ART;

Specifies all the SAML artifact schemes you have configured. The custom authentication scheme will enumerate the artifact schemes and find the one with the correct Provider Source ID for the request.

Enable this scheme for SiteMinder Administrators

Disregard this check box.

Authentication Scheme Dialog—Impersonation Template

Before you can assign an Impersonation authentication scheme to a realm, you must configure the scheme in the Authentication Scheme Dialog. For overview information or steps for configuring impersonation, Impersonation.

Authentication Scheme Dialog—Impersonation Template—Scheme Setup Tab

The Scheme Setup tab for Impersonation authentication is where you enter the server, target and attribute list for the Impersonation authentication scheme.

Server Name

Fully qualified domain name of the web server on which the FCC was installed.

Note: IP addresses are not supported.

The server does not have to be the same server on which the Agent is installed. Domain names must contain at least 2 periods and be specified using the following format:

servername.host.com:[port]

Example: server1.security.com. The port is only required for communication over a non-default port.

This name is case-sensitive. For information about cookie domains, see the *SiteMinder Web Agent Guide*.

Note: If your network includes multiple cookie domains, you must configure a separate Impersonation authentication scheme in each cookie domain in which you want to implement impersonation.

Use SSL Connection check box

Select this check box if you want SiteMinder to use an SSL connection to process authentication for impersonation.

Target

Path and .fcc file used by the scheme.

The default path points to a /forms subdirectory under the /siteminderagent virtual directory on the Web server specified in the Server Name field. The directory and the default .fcc file were created during Web Agent installation. The default target specifies the imp.fcc file, a sample file that can be customized. For information about .fcc file requirements for impersonation, see Impersonation.

Additional Attribute List

(Optional) Impersonatee attributes, other than user name, that will be specified by an impersonator.

When listing attributes, begin with AL= and use commas to separate the user attribute names.

Example: AL=age,zipcode

The AL= is a SiteMinder notation that indicates the list of attributes that should be considered. By default, the list of attributes is considered an AND-style query. The Policy Server compares all of the attribute values collected from the user to the corresponding attribute values in the user directory. If all of the attribute values match exactly, the user will authenticate successfully.

Note: You can authenticate users with attributes that contain multiple values. To specify that an attribute has multiple values, prefix the attribute name with a carat (^).

Example: If you are using a multi-valued "mail" attribute to authenticate users, you would specify "AL=^mail" to indicate that "mail" is multi-valued. A user can provide one of the valid values to successfully authenticate.

Limit: The values of a multi-valued attribute should not contain a carat. A value that contains a carat introduces the possibility of users being improperly authenticated. For example, if a value is 123^456, a user would be able to authenticate with 123 and 456, in addition to 123^456.

In order for SiteMinder to collect additional attributes, the .fcc file used by SiteMinder to generate a form for impersonation must be modified to include the attributes.

When using additional attributes in an impersonation scheme, consider the following:

- The attributes are only used to match attribute values associated with the user who has been located in a directory by user name.
- The attribute names are case-sensitive. They must match exactly both the attribute names implemented by the directory and the attribute names specified in the authentication scheme.

Note: If you have installed the CA Software Development Kit, you can use the SiteMinder authentication API (see the *Developer's Guide for C*) to define additional notations.

More information:

[Impersonation](#) (see page 457)

Authentication Scheme Dialog—Impersonation Template—Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an Impersonation scheme contains the following:

Library

This field contains the name of the shared library that processes impersonation. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for impersonation authentication is SmAuthImpersonate.

Parameter

This field displays the Server and Target you entered in the Scheme Setup tab. It also displays any additional attributes you entered for the authentication scheme. The server and target make up the location of the .fcc file on the Web Server.

Enable this scheme for SiteMinder Administrators

This check box is disabled. Impersonation cannot be used to authenticate a Policy Server administrator.

Tasks Related to the Authentication Scheme Dialog

The following tasks are related to the Authentication Scheme Dialog:

- [Basic Authentication Schemes](#) (see page 241)
- [Basic Over SSL Authentication Schemes](#) (see page 243)
- [HTML Forms Authentication Schemes](#) (see page 244)
- [Windows Authentication Schemes](#) (see page 258)
- [MS Passport Authentication Schemes](#) (see page 261)
- [RADIUS CHAP/PAP Authentication Schemes](#) (see page 265)
- [RADIUS Server Authentication Schemes](#) (see page 267)
- [SafeWord Server Authentication Schemes](#) (see page 268)
- [SafeWord Server and HTML Forms Authentication Schemes](#) (see page 269)
- [SecurID Authentication Schemes](#) (see page 270)

- [SecurID with HTML Forms Support Scheme Prerequisites](#) (see page 275)
- [X.509 Client Certificate Authentication Schemes](#) (see page 277)
- [X.509 Client Certificate and Basic Authentication Schemes](#) (see page 279)
- [X.509 Certificate or Basic Authentication Schemes](#) (see page 281)
- [X.509 Client Certificate and HTML Forms Authentication Schemes](#) (see page 283)
- [X.509 Client Certificate or HTML Forms Authentication Schemes](#) (see page 286)
- [Anonymous Authentication Schemes](#) (see page 288)
- [Custom Authentication Schemes](#) (see page 290)
- [TransactionMinder Authentication Schemes](#) (see page 291)
- [Federation Security Services Authentication Schemes](#) (see page 291)
- [Impersonation Authentication Schemes](#) (see page 292)

Certificate Mappings Dialog

SiteMinder requires certificate mappings for X.509 Client Certificate authentication schemes. Without a certificate mapping, SiteMinder cannot establish trust between a client certificate and the certificate authority that issued the certificate. Without an established trust, X.509 client certificates cannot be used to verify a user's identity.

X.509 client certificates contain the DN of the issuing Certificate Authority (CA), the Subject DN of the client certificate, the ID of the algorithm used for the certificate, and extensions. A certificate mapping in the Policy Server User Interface describes how a user's subject DN can be mapped to a user DN in an authentication directory.

The user DN is a unique way of identifying a particular user. If you want to use an X.509 Client Certificate authentication scheme to provide security for a realm, you must create a certificate mapping in order for the authentication scheme to function.

Certificate Mappings Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Certificate Mappings Dialog

To access the Certificate Mappings Dialog in the Policy Server User Interface, from the menu bar of the SiteMinder Administration window, select Advanced, Certificate Mapping.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Certificate Mappings Dialog Fields and Controls

This dialog contains a list of all existing certificate mappings.

Certificate Mappings Group Box

Contains a list of existing certificate mappings.

Directory Type

For each existing mapping, this column displays the directory type associated with a mapping.

Issuer DN

For each existing mapping, this column displays the Issuer DN specified for the mapping.

Add button

Opens the Certificate Mapping Dialog for a new mapping.

Delete button

Deletes the selected certificate mapping.

Edit button

Opens the Certificate Mapping Dialog for the selected certificate mapping.

Tasks Related to the Certificate Mappings Dialog

The following tasks and information are related to the Certificate Mappings Dialog:

- [Test a Certificate Mapping](#) (see page 295)
- [How SiteMinder Confirms the Validity of Certificates](#) (see page 296)
- [Custom Certificate Mapping for Multiple Attributes of the Same Type](#) (see page 304)
- [Configure Online Certificate Status Protocol Checking](#) (see page 299)

- [Map to the Certificate Serial Number or IssuerDN](#) (see page 304)
- [Map to Non-Required Attributes](#) (see page 305)

Certificate Mapping Properties Dialog

The Certificate Mapping Properties Dialog is where you configure certificate mappings.

Certificate Mapping Properties Dialog Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Certificate Mapping Properties Dialog

To access the Certificate Mappings Dialog:

1. From the menu bar of the SiteMinder Administration window, select Advanced, Certificate Mapping.

The Certificate Mappings dialog opens.

2. Do one of the following:

- Select an existing mapping and click the Edit button.
- Click the Add button.

The Directory Mapping Dialog opens.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Certificate Mapping Properties Dialog Fields and Controls

The Certificate Mapping Properties dialog contains three group boxes.

- [Map from Issuer DN in Certificate to Directory Type Group Box](#) (see page 668)
- [Mapping Group Box](#) (see page 669)
- [Certificate Revocation List \(CRL\) Checking Group Box](#) (see page 670)

Map from Issuer DN in Certificate to Directory Type Group Box

Issuer DN

Distinguished Name the server certificate.

Note: Issuer DNs cannot exceed 255 characters if a relational database is used as a policy store; Issuer DNs cannot exceed 1000 characters if an LDAP directory server is used as a policy store.

In order for the certificate mapping to work, the value of the Issuer DN field must be identical to the certificate's issuer DN (all characters, spaces, and punctuation). The Policy Server compares the Issuer DN from the user's certificate with each Issuer DN specified for the certificate mapping.

Note: The issuer DN will differ according to the Web Server vendor. For example, the issuer DN for a certificate on an IIS Web server is different from the issuer DN for the same certificate on an Apache Web server.

The following is an example of an issuerDN:

```
IssuerDN=CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated,  
OU=\"www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98\",  
OU=VeriSign Trust Network, O=\"VeriSign, Inc.\"
```

For information on finding the Issuer DN, contact your Certificate Authority (CA).

Directory Type

Type of directory to which the certificate will be mapped.

The directory that you select indicates the type of directory that will be used to authenticate users. Valid directory types include LDAP/AD, WinNT, and ODBC.

A user directory connection must be configured for the authentication directory using the SiteMinder User Directory dialog.

Certificate Required in Directory check box

Select this check box for SiteMinder to verify that the certificate presented by the user matches the certificate stored in the user's entry in the authentication directory.

You may only select this check box if the authentication directory is an LDAP user directory.

More information:

[User Directories](#) (see page 141)

Mapping Group Box

The Mapping group box is where you specify how the X.509 client certificate will map to the user information in the authentication directory. The contents of the group box change depending on the radio button you select for the type of mapping. You can select one of the following:

Single Attribute

The Policy Server matches a single attribute from the subject DN of a user's certificate to a single attribute stored in the user directory to verify the user's identity.

Attribute Name

The following table lists the types of attributes from you can select for a mapping:

Attribute	Description
UID	User ID
CN	Common Name Important! Novell eDirectory cannot have CN attribute set to a value longer than 64 characters.
SN	Surname
E	Email address
Enodomain	Contents of the email address that precede the @ symbol Note: For non-English versions of SiteMinder, the Enodomains selection is not translated from English. In order for this feature to work correctly, the ASCII string "Enodomain" must be used.
OU	Organizational unit
O	Organization
L	Locality or city
S	State
C	Country

When you configure the user directory to which the certificate is mapped, the Policy Server uses the values of the Start and End fields of the LDAP User DN Lookup group box in the SiteMinder User Directory dialog to locate the attribute in the directory to which the certificate is mapped.

For example, if you create a certificate mapping to the UID attribute of an LDAP user directory with the Start and End values pictured in the following graphic, SiteMinder maps the UID from the subject DN to the user directory using the value:

`uid=<value of UID attribute from subject DN>, ou=marketing,o=myorg.org`

Custom

The Policy Server uses a custom mapping expression to verify the user's identity. Enter the expression in the field to the right of the radio buttons.

Mapping Expression

For information about mapping expressions, see [Custom Mapping Expressions](#) (see page 300).

Note: If you enter a single attribute in the Mapping Expression field, the Policy Server automatically converts the custom expression to a single attribute as described in step 1.

Exact

The Policy Server matches the user's entire DN from the certificate to the entire DN in the authentication directory.

Test Button

This button is available regardless of the radio button selected. It opens the Certificate Map Test Dialog.

Certificate Revocation List (CRL) Checking Group Box

Perform CRL Checks check box

Select this check box to enable Certificate Revocation List (CRL) Checking.

CRL checking is an additional level of security that involves retrieving a CRL from an LDAP directory, verifying the CRL, and finally validating that the user's certificate has not been revoked. If the Perform CRL Checks check box is marked, the Policy Server processes CRL checking.

CRL Directory

Names of the available user directories.

For CRL checking, an LDAP user directory connection must be defined using the User Directory Dialog. If the user directory connection has not been defined, click the Create button.

DN in CRL Directory

Distinguished Name (DN) of the CA if the DN is different from the DN you entered in the Issuer DN field.

It is possible to have a CA whose DN in the LDAP directory for CRL checking is different from the Issuer DN in the client certificate. In such a case, you must specify the DN of the CA for the LDAP CRL directory in this field.

Verify Signature check box

Select this check box to enable signature verification.

When you enable signature verification, the Policy Server checks the CA's public certificate against a signature stored in the policy database.

Use Distribution Points check box

Select this check box if your CRL uses distribution points.

Large CRLs may contain multiple distribution points that can be used to locate a revoked user. Distribution points indicate a starting point in the CRL LDAP directory. The distribution point provides a starting point for a CRL check and saves the processing time that it would take to search the entire CRL for a particular user.

When this check box is marked, SiteMinder looks at a user's certificate, and retrieves the distribution point from the certificate, then uses it to find the appropriate LDAP directory entry point for the CRL.

Cache check box

Select this check box to have SiteMinder to cache CRL entries.

When you mark this check box, SiteMinder checks the NextUpdate field in the CRL for the date when the cached CRL information should be deleted and replaced with updated CRL information. SiteMinder uses the cached CRL information until the date specified in the NextUpdate field of the CRL.

The NextUpdate field in the CRL is optional. If there is no value for NextUpdate, SiteMinder will not cache CRL information.

Create button

Open the User Directory Dialog.

More information:

[User Directory Dialog](#) (see page 545)

Tasks Related to the Certificate Mapping Properties Dialog

The following tasks and information are related to the Certificate Mapping Properties Dialog:

- [Test a Certificate Mapping](#) (see page 295)
- [How SiteMinder Confirms the Validity of Certificates](#) (see page 296)
- [Custom Certificate Mapping for Multiple Attributes of the Same Type](#) (see page 304)
- [Configure Online Certificate Status Protocol Checking](#) (see page 299)
- [Map to the Certificate Serial Number or IssuerDN](#) (see page 304)
- [Map to Non-Required Attributes](#) (see page 305)

Certificate Map Test Dialog

Once you configure a certificate mapping, you can test the mapping using the Test button located in the Mapping group box of the Certificate Mapping dialog. This opens the Certificate Map Test dialog which displays the search string that the Policy Server will use to map client certificates to user directory attributes.

Certificate Map Test Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- A certificate mapping exists.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Certificate Map Test Dialog

To access the Certificate Map Test Dialog

1. From the menu bar of the SiteMinder Administration window, select Advanced, Certificate Mapping.

The Certificate Mappings dialog opens.

2. Do one of the following:
 - Click the Add button.
 - Select an existing mapping and click the Edit button.

The Directory Mapping Dialog opens.

3. Click the Test button.

The Certificate Map Test Dialog opens.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Certificate Map Test Dialog Fields and Controls

Directory

Lists available user directory connections.

The user directory connections available in the drop-down list include all existing directory connections of the type you selected in the Map from IssuerDN in Certificate to Directory Type group box of the Certificate Mapping Dialog.

The contents of the Directory Information group box change based on the type of user directory connection you are testing. For WinNT and ODBC user directory connections, the group box displays the Directory Type you are testing. For LDAP directory connections, the group box displays the Directory Type, as well as the Lookup Start and Lookup End values used to locate a user's DN within the LDAP directory.

When you select a user directory connection, the Certificate Map Test dialog displays information about the certificate mapping.

More information:

[Certificate Map Test Results](#) (see page 296)

Tasks Related to the Certificate Map Test Dialog

The following tasks and information are related to the Certificate Map Test Dialog:

- [Test a Certificate Mapping](#) (see page 295)
- [How SiteMinder Confirms the Validity of Certificates](#) (see page 296)

Chapter 27: Domain Dialog Reference

This section contains the following topics:

[Domain Dialog](#) (see page 675)

Domain Dialog

The SiteMinder Domain Dialog is where you configure a policy or affiliate domain.

Domain Dialog Prerequisites

The following prerequisites must be met in order to successfully create a policy or affiliate domain using the Domain Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Domain Dialog

To access the Domain Dialog in the Policy Server User Interface for a new policy or affiliate domain, from the menu bar, select Edit, System Configuration, Create Domain.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Navigate to the Domain Dialog for an Existing Policy or Object

There are multiple ways to access the Domains Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Domains Dialog for an existing policy or affiliate domain object

1. In the object pane, click on Domains.
2. In the list pane, right-click the domain object you want to edit.
3. From the pop-up menu, select Properties of Domain.

Domain Dialog Fields and Controls

The Dialog contains the following fields, check box, and tabs:

Name field

Name of the policy or affiliate domain.

Description field

Optionally, a brief description of the policy or affiliate domain.

Global Policies Apply check box

(Active only for policy domains) Assigns global policies to all of the resources in a policy domain. Global policies allow you to associate responses with particular resources and events across all domains.

More information:

[Global Policies, Rules, and Responses](#) (see page 435)

Domain Type Group Box

The Domain Type group box is where you specify the domain type:

Policy Domain radio button

Specifies that the domain being configured is a Policy Domain. When selected, the Global Policies Apply check box is active and the Realms tab is present (Affiliates and SAML Service Providers tabs are absent).

Affiliate Domain radio button

Specifies that the domain being configured is an Affiliate Domain (see the *SiteMinder Federation Security Services Guide*). When selected, the Global Policies Apply check box is inactive and the Affiliates and SAML Service Providers tabs are present (the Realms tab is absent).

More information:

[Policy Domain Overview](#) (see page 307)

User Directories Tab

The User Directories tab is where you specify the user directories associated with a policy or affiliate domain. When the Policy Server attempts to authenticate users, it compares their credentials to the directories you specify in this tab.

User Directories list

Lists user directories currently associated with the policy or affiliate domain.

Search Order buttons

Move the selected user directory up or down in the User Directories list.

Create Button

Opens the SiteMinder User Directory Dialog, which allows you to configure a new connection between the Policy Server and a user directory.

All Directories drop-down list box

Specifies the user directory connection that will acted upon by the Add button. This list is between the Create and Add buttons.

Add Button

Adds the user directory selected in the All Directories drop-down list box to the policy or affiliate domain.

Remove Button

Removes a user directory from a policy or affiliate domain.

More information:

[User Directory Dialog](#) (see page 545)

Administrators Tab

The Administrators tab is where you assign administrators to the policy or affiliate domain. These administrators may create, edit, and delete Policy Server objects within the domain.

Administrators list

Lists administrators currently associated with the policy or affiliate domain.

Create Button

Opens the SiteMinder Administrator Properties Dialog, which allows you to configure a SiteMinder administrator.

All Administrators drop-down list box

Specifies the administrator that will acted upon by the Add button. This list is between the Create and Add buttons.

Add Button

Adds an administrator to a policy or affiliate domain.

Remove Button

Removes an administrator from a policy or affiliate domain.

Realms Tab

This group box is only present for policy domains (that is, domain objects for which the Domain Type setting is Policy Domain).

The Realms tab is where you assign realms to the policy or affiliate domain.

Realms list

Lists administrators currently associated with the policy domain.

Create Button

Opens the SiteMinder Realm Dialog, which allows you to configure a realm and add it to the domain.

Note: You cannot create a realm from within the Policy Domain Dialog until the policy domain has been saved.

More information:

[Realm Dialog](#) (see page 681)

Affiliates Tab

This tab is only present for affiliate domains (that is, domain objects for which the Domain Type setting is Affiliate Domain).

The Affiliates Tab is where you associate SAML 1.x affiliate objects with an affiliate domain. Affiliates are objects that represent another site within a federated enterprise, or a portion of a federated enterprise. For more information, see the *Federation Security Services Guide*.

Affiliates list

Lists affiliates currently associated with the affiliate domain.

Create Button

Opens the Affiliates Dialog, which allows you to configure an affiliate and add it to the domain.

Note: You cannot create an affiliate from within the Policy Domain Dialog until the policy domain has been saved.

SAML Service Providers Tab

This tab only present for affiliate domains (that is, domain objects for which the Domain Type setting is Affiliate Domain).

The Service Providers Tab is where you associate SAML 2.0 Service Provider objects with an affiliate domain. Service Providers are objects that represent another site within a federated enterprise, or a portion of a federated enterprise. For more information, see the *Federation Security Services Guide*.

Service Providers list

Lists SAML Service Providers currently associated with the affiliate domain.

Create Button

Opens the SiteMinder SAML Service Providers Dialog, which allows you to configure a Service Provider and add it to the domain.

Note: You cannot create a Service Provider from within the Policy Domain Dialog until the policy domain has been saved.

Tasks Related With the Domain Dialog

The following tasks and information are related to the Domain Dialog:

- [Configure a Policy Domain](#) (see page 309)
- [Domain Editing](#) (see page 313)
- The section on configuring an Affiliate Domain in the *Federation Security Services Guide*

Chapter 28: Realm Dialog Reference

This section contains the following topics:

[Realm Dialog](#) (see page 681)

Realm Dialog

The SiteMinder Realm Dialog is where you configure realms.

Realm Object Prerequisites

The following prerequisites must be met in order to successfully create a realm using the Realm Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege for the policy domain in which the realm is to be created.

Navigate to the Realm Dialog to Create a New Realm

There are multiple ways to access the Realms Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Realm Dialog to create a new realm

1. In the Object pane, click on the Domains tab.
2. In the Object pane, click the policy domain to which you want to add a realm.
Note: If creating a nested realm, you will first need to expand the parent realm by clicking on the + symbol to the left of that realm.
3. From the menu bar, select Edit, <Selected_Policy_Domain_Name>, Create Domain.

Navigate to the Realms Dialog to Edit an Existing Realm

There are multiple ways to access the Realms Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Realms Dialog to edit an existing realm

1. In the Object pane, click on the Domains tab.
2. Expand the policy domain that contains the realm in which you want to create a nested realm by clicking on the + symbol to the left of the domain.
3. Click on the Realms entry.

Note: (For nested realms, you will first need to expand the parent realm by clicking on the + symbol to the left of that realm.)

4. From the menu bar, select Edit, Properties of Domain.

Realm Dialog Fields and Controls

Name

Name of the realm.

Description

Optionally, a brief description of the realm.

Realm Dialog—Resource Tab

The Resource tab is where you configure a realm.

More information:

[Realms](#) (see page 315)

Resource Group Box

Agent field

Specifies the name of the agent or agent group that will protect the resources in the realm.

Lookup Button

Opens the Set Agent/Group Dialog from which you pick an agent or agent group from a list of all such existing objects or search for one by name.

Resource Filter field

Specifies the path for the resource filter. This field functions as a root for locating resources.

Note: Do not create a realm whose specified Agent and resource filter are effectively the same as the specified Agent and resource filter in a pre-existing realm. Typically, the UI will prevent you from doing this, but the combination of two Agent or two resource filters that look different as strings but resolve identically can circumvent Policy Server's validation algorithms.

Every resource on the server protected by the Agent that matches the Resource Filter is included in the realm to which you are connecting. For example, /marketing/ indicates that the new realm will include only those resources located in the /marketing/ directory of the server protected by the Agent or Agent group.

Note: Asterisk (*) and question mark (?) characters are treated as literal characters in resource filters (not wildcards).

Effective Resource field

(Informational) Displays the Agent and resource protected by the realm.

Authentication Scheme Group Box

The Authentication Scheme group box is where you specify the authentication scheme that SiteMinder uses to establish the identity of a user who attempts to access a resource in the realm.

Authentication Scheme drop-down list

Specifies the authentication scheme that protects the realm.

More information:

[Authentication Schemes](#) (see page 231)

Default Resource Protection Group Box

The Default Resource Protection Group Box is where you specify whether the resources in the realm are protected or unprotected by default:

Protected

Indicates the resources in the new realm are protected from access by SiteMinder when you save the new realm. SiteMinder protects a resource as soon as it is placed in this realm. To allow access to this resource, you must create a rule and bind it to a Policy that will allow access to users or groups. The Protected radio button is selected by default.

Note: There may be a short delay between the time when you create a realm and the time when SiteMinder begins protecting the realm. Web Agents poll the Policy Server for changes at a fixed interval (30 seconds by default). By default, a newly created realm may take up to one minute to be recognized by SiteMinder. You can also stop, then start the Policy Server to have SiteMinder recognize the new realm immediately.

Unprotected

Indicates the resources in the realm are not protected by default. The resources in the realm can be protected if you create a rule and bind the rule to a policy.

Realm Dialog—Session Tab

The Session tab is where you set session timeouts, choose non-persistent or persistent sessions, and enable or disable synchronous auditing for the realm.

More information:

[User Sessions](#) (see page 85)

Session Timeouts Group Box

The Session Timeouts Group Box is where you specify timeouts for each realm protected by a SiteMinder Web Agent.

Note: A user's session timeout is based on the session that is established when a user authenticates in a realm. If a user accesses a resource in another realm, SiteMinder maintains the user's session. For example, if a user authenticates in RealmA, which has a session timeout of 30 minutes, then the user accesses a resource in RealmB 15 minutes later, regardless of the session timeout for RealmB, the user's session expires in another 15 minutes. If you want to change this default behavior, you can create responses to replace session timeout values.

Maximum Timeout Enabled

If set, the values specified in the associated Hours and Minutes fields determine the maximum amount of time a user session can be active before the Agent challenges the user to re-authenticate.

This setting is enabled by default. To specify no maximum session length, unset the checkbox. The default maximum session length is two hours.

Hours field

Specifies the hours value for the maximum session length.

Minutes field

Specifies the minutes value for the maximum session length.

Note: To use this feature with the Basic authentication scheme, your Web Agent must be configured to Require Cookies. For more information, see the *Web Agent Guide*.

Note: You can override this setting by using the WebAgent-OnAuthAccept-Session-Max-Timeout response attribute.

Idle Timeout Enabled

If set, the values specified in the associated Hours and Minutes fields help determine the amount of time that an authorized user session can remain inactive before the Agent terminates the session. If you are concerned about users leaving their workstations after accessing a protected resource, set the idle timeout to a shorter period of time. If the session times out, users must re-authenticate before accessing the resources in the realm.

This setting is enabled by default. To specify no session idle timeout, unset the checkbox. The default session idle timeout is one hour.

Note that the session actually expires within a certain maintenance time period after the specified idle timeout value. The extra time period is determined by the number of seconds specified in the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Session Server\MaintenancePeriod

The default value is 60 seconds.

For example, if the you set the idle timeout at 10 minutes, and you use the default value of the MaintenancePeriod registry setting, the longest time period before a session will timeout due to inactivity is 11 minutes (specified timeout + maintenance period).

Note: For persistent sessions, the Idle Timeout must be enabled and set to a value higher than that specified for the Validation Period.

Note: To use this feature with the Basic authentication scheme, your Web Agent must be configured to Require Cookies. For more information, see the *SiteMinder Web Agent Guide*.

Note: You can override this global setting by using the WebAgent-OnAuthAccept-Session-Idle-Timeout response attribute. A value of zero indicates that the session will not end because of inactivity.

Hours field

Specifies the hours value for the idle timeout period.

Minutes field

Specifies the minutes value for the idle timeout period.

More information:

[Responses and Response Groups](#) (see page 353)

Persistent Session Group Box

The Persistent Session Group Box is where you configure a realm to use standard (non-persistent) sessions or persistent sessions:

No Persistent Session radio button

If selected (the default), sessions in this realm are non-persistent (that is, tracked using cookies).

Persistent Session radio button

If selected, sessions in this realm are persistent (that is, user sessions are tracked in the session store and optional cookies).

Note: If Persistent Session is selected, the Idle Timeout Enabled option *must* be set.

Note: If you configure one or more realms to use persistent sessions, you must ensure that the Session Server is enabled and configured. See the *SiteMinder Policy Server Management Guide*.

Validation Period Enabled (persistent sessions only)

If set, the specified value determines the maximum period between agent calls to the Policy Server to validate a session. Session validation calls perform two functions: informing the Policy Server that a user is still active *and* checking that the user's session is still valid.

This setting is disabled by default. To specify a validation period, select the check box.

To specify the validation period length, enter values in the Hours, Minutes, and Seconds fields. If you are configuring SiteMinder to provide a Windows user security context, set this value high, for example, 15-30 minutes. Also, if active sessions are lower than the Agent's maximum user session cache value, the Agent will not have to revalidate a session with the session server.

Note: The session Validation Period must be less than the specified Idle Timeout value.

Synchronous Auditing Group Box

Synchronous Auditing check box

If enabled, SiteMinder must log Policy Server and Web Agent actions before it allows access to resources. Further, SiteMinder will not allow access to the realm's resources until after the activity has been recorded in the audit logs. For information about logs, see the *SiteMinder Policy Server Management Guide*.

Realm Dialog—Advanced Tab

The Advanced tab is where you select registration schemes, authorization directory mappings, and the types of events (authentication and authorization) that are processed for the realm. Processed events enable you to associate rules with the indicated types of events.

Registration Group Box

The Registration group box is where you specify a registration scheme for the realm. Registration schemes allow new user accounts to be created in a user directory. If the authentication scheme associated with a realm requires that users have identities in a user directory, assigning a registration scheme to the realm provides a mechanism for adding new accounts in the user directory.

Registration Scheme drop-down list box

Optionally, specifies a registration scheme for the realm.

Edit button

Active only when a registration scheme is selected in the Registration Scheme drop-down list box. Opens the selected registration scheme for editing.

Create button

Opens the Registration Dialog, from where you can create a new registration scheme.

Directory Mapping Group Box

The Directory Mapping group box is where you specify an authorization directory in which users that access resources in the realm. Directory mappings allow users who are authenticated in one directory to be authorized in another directory.

Directory Mapping drop-down list box

Specifies the authorization directory for the realm. Only directory mappings that have already been configured using the Directory Mappings Dialog appear in the drop-down list.

Edit Mappings button

Opens the Directory Mappings Dialog, from where you can edit directory mappings.

Note: You can only select mappings to authorization directories that are included in the policy domain in which the realm is located.

More information:

[Directory Mappings Dialog](#) (see page 590)

[Directory Mapping](#) (see page 213)

[Domains](#) (see page 307)

Events Group Box

The Events group box is where you enable event processing for authentication and authorization events to support rules that are triggered by these events.

Process Authentication Events check box

If enabled, supports rules triggered by authentication attempts.

Process Authorization Events check box

If enabled, supports rules triggered by authorization attempts.

Important! If the resources in the realm are associated with global rules in a global policy, authentication and authorization events must be enabled. The Policy Server will not process global policies unless authentication and authorization events are enabled.

More information:

[Global Policies, Rules, and Responses](#) (see page 435)

Tasks Related to the Realm Dialog

The following tasks are related to the Realm Dialog:

- [Configure a Realm](#) (see page 320)
- [Realm Editing and Deletion](#) (see page 325)
- [Flush a Realm from the Resource Cache](#) (see page 326)

Chapter 29: Rule and Rule Group Dialogs Reference

This section contains the following topics:

[Rule Dialog](#) (see page 691)

[Rule Group Properties Dialog](#) (see page 698)

[Rule Group Items Dialog](#) (see page 702)

Rule Dialog

The Rule Dialog is where you create and edit SiteMinder rules.

Rule Dialog Prerequisites

The following prerequisites must be met in order to successfully create a rule using the Rule Dialog:

- A rule must be associated with a realm.

More information:

[Realms](#) (see page 315)

Navigate to the Rule Dialog to Create a Rule

Your method for accessing the Rule Dialog in the Policy Server User Interface varies depending on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Dialog to create a rule

1. Do one of the following:

If you have the Manage System and Domain Objects privilege:

- In the Object pane, click on the Domains tab.
- Expand the policy domain to which you want to add a rule by clicking on the + symbol to the left of the domain.

If you have the Manage Domain Objects privilege:

- In the Object pane, expand the policy domain to which you want to add a rule by clicking on the + symbol to the left of the domain.
2. Right click the realm/nested realm to which you want to add a rule.
 3. From the pop-up menu, select Create Rule under Realm.

Navigate to the Rule Dialog to Edit an Existing Rule

Your method for accessing the Rule Dialog in the Policy Server User Interface varies depending on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Dialog to edit an existing rule

1. Do one of the following:

If you have the Manage System and Domain Objects privilege:

- In the Object pane, click on the Domains tab.
- Expand the policy domain that contains the rule that you want to edit by clicking on the + symbol to the left of the domain.

If you have the Manage Domain Objects privilege:

- In the Object pane, expand the policy domain to which you want to add a rule by clicking on the + symbol to the left of the domain.
2. In the Object pane, expand the list of realms within the policy domain by clicking on the + symbol to the left of the Realms entry.
 3. In the Object pane, click on the realm that contains the rule you want to edit. A list of rules and nested realms in the selected domain appear in the List pane.
 4. Right-click the rule you want to edit.
 5. From the pop-up menu, select Properties of Rule.

Rule Dialog Fields and Controls

Name field

Name of the rule.

Description field

Optionally, a brief description of the rule.

Realm and Resources Group Box

The Realm and Resources group box is where you select the realm and resources to which this rule will apply.

Realm drop-down list box

The realm that contains the resources to which this rule will apply.

By default, the Realm drop-down list is set to the name of the realm you selected when you opened the SiteMinder Realm Dialog.

Note: In this drop-down list, the Policy Server User Interface shows all the realms that exist in this policy domain. If realms do not exist for resources you want to protect, a rule cannot be created to protect those resources.

Resource field

The Resource that this rule will protect.

A resource can be a specific file or an expression that uses resource matching or regular expressions to include multiple files.

Below the resource field, the Policy Server User Interface displays the effective resource protected by the rule. The effective resource consists of the Agent, all of the resource filters concatenated from any parent realms that exist above the realm you selected, the resource filter of the realm specified in the Realm field, and the resource you entered in the Resource field.

For example, suppose you create the following:

- Realm1 Agent = Agent1
Agent1 IP Address = 123.123.12.12
- Realm1 with Resource Filter = /dir
- Rule1 Resource = /myfile.html

Rule1 will protect 123.123.12.12/<root directory for Agent1's web server>/dir/myfile.html.

The resource you enter in the Resource field is appended to the Resource Filter of the realm that contains the rule. If the Resource Filter of the realm ends in /, then the Resource should not begin with /. For example, if the resource filter for the realm is /dir1/, and the resource is /file.html, the rule erroneously tries to protect /dir1//file.html. In this example, the Resource should be set to file.html, so that the rule protects /dir1/file.html.

If you specify a rule in a realm that does not end in /, and then you delete the / that the Policy Server User Interface automatically places in the Resource field and enter a resource of *, the rule protects any resources and any matching directories that fall below the realm in your network. For example, if you have a realm with a Resource Filter of /dir1, and you delete the / that the Policy Server User Interface automatically inserts into the rule, then enter a * in the Resource field, the protected resource is *agent/dir1**. This resource protects everything in /dir1, as well as /dir11, /dir12, etc. If you use the default (leave the / in the Resource field), the protected resource is *agent/dir1/**, which includes all files and directories contained in the dir1 directory, but does not include *agent/dir11* or *agent/dir12*.

Note: It is possible to assign a combination of slightly different resource filters to realms and resources to rules that form the same effective URL. For example, you can create a realm with a resource filter of /dir1 and add a rule to the realm that protects /index.html. You can then create a separate realm with a resource filter of /dir1/ that protects index.html. Both of these realm-rule combinations yield the following URL: *agent/dir1/index.html*.

Since /dir1/ is the longest matching realm, and the longest matching realm takes precedence in policy processing, policies that include the realm with the /dir1/ resource filter will always take precedence over policies that include the /dir1 resource filter. This may cause unexpected behavior for administrators who create policies that include the realm-rule combination with the /dir1 resource filter.

Perform Regular Expression Matching check box

If set, specifies that a resource can be a specific file or an expression that uses resource matching with regular expressions.

More information:

[Resource Matching and Regular Expressions](#) (see page 341)

[Grouping Resources in Realms](#) (see page 315)

Action Group Box

The Action group box is where you specify the type of event and the specific action that must take place for the rule to fire:

Web Agent Actions radio button

Specifies that the rule should fire for the HTTP action or actions that you select from the clickable list that appears to the right of the radio button when this entry is selected:

Get

Retrieves a resource for viewing via HTTP.

Put

Supports legacy HTTP actions.

Post

Posts information supplied by a user via HTTP.

If you have purchased TransactionMinder, two additional Web Agent rule actions are available:

ProcessSOAP

Supports incoming XML messages wrapped with a SOAP envelope.

ProcessXML

Supports incoming raw XML messages (not wrapped with a SOAP envelope).

Affiliate Agent Actions radio button

(Present in place of Web Agent Actions for realms associated with an Affiliate Agent only). Specifies that the rule should fire for the single action available in a rule for an Affiliate Agent:

Visit

A special action for Affiliate Agents that allows a SiteMinder portal site to interact with an affiliate Web site.

Authentication Events radio button

Specifies that the rule should fire for the authentication event selected from the drop-down list box that appears to the right of the radio button when this entry is selected:

OnAuthAccept

Occurs when a user successfully authenticates. This event may be used to redirect a user after a successful authentication

OnAuthAttempt

Occurs when a user fails to authenticate because no user name was supplied.

OnAuthChallenge

May be used in custom authentication schemes to trigger a response.

OnAuthReject

Occurs when a user that is bound to a policy fails to authenticate. The user must be bound to the policy for the rule to fire.

OnAuthUserNotFound

This event is only used to trigger Active Responses. This event should not be used to trigger any response other than an Active Response.

Authorization Events radio button

Specifies that the rule should fire for the authorization event selected from the drop-down list box that appears to the right of the radio button when this entry is selected

OnAccessAccept

Occurs when SiteMinder successfully authorizes a user to access the resource.

OnAccessReject

Occurs when SiteMinder rejects a user because the user is not authorized to access the resource.

Impersonation Events radio button

Specifies that the rule should fire for the impersonation event selected from the drop-down list box that appears to the right of the radio button when this entry is selected:

ImpersonationStart

When included in an appropriate policy, a rule that includes this event allows an impersonation session to begin

ImpersonationStartUser

When included in an appropriate policy, a rule that includes this event allows a set of users to be impersonated.

More information:

[Rule Actions](#) (see page 330)

Allow/Deny and Enable/Disable Group Box

The Allow/Deny and Enable/Disable group box is where you specify whether SiteMinder should allow or deny access to a protected resource when the rule fires and whether or not the rule is enabled.

Allow Access radio button

Allow properly authenticated users who are associated with the policy that contains the rule to access the resource specified by the rule

Deny Access radio button

Deny access to properly authenticated users who are specified in the policy that contains the rule

Enabled check box

If set, the rule is enabled. If not set, the rule is disabled and the rule will not fire.

Advanced Group box

The Advanced group box in the SiteMinder Rule Dialog is where you define additional rule settings. This group box contains two tabs:

- Time Restrictions
- Active Rules

Time Restrictions Tab

The Time Restrictions tab in the advanced group box is where you manage time restrictions for the rule (that is, when the rule should and should not fire).

Set

Opens the Time Dialog in which you can add, or if they already apply, edit time restrictions for the rule.

Remove

Remove all time restrictions set for the rule (grayed-out when no time restrictions are present).

More information:

[Time Dialog](#) (see page 905)

Active Rule Tab

The Active Rule tab is where you manage active rules for dynamic authorization based on external business logic.

Active Rule field

Displays the active rule, if one has been defined.

Edit button

Opens the Active Rule Editor Dialog in which you can define a new, or edit an existing active rule.

Clear button

Remove an existing active rule (inactive when none exists).

More information:

[Active Rule Editor Dialog](#) (see page 909)

Tasks Related to the Rule Dialog

The following tasks are related to the Rule Dialog:

- [Configure a Rule for Web Agent Actions](#) (see page 334)
- [Configure a Rule for Authentication Event Actions](#) (see page 336)
- [Configure a Rule for Authorization Event Actions](#) (see page 337)
- [Configure a Rule for Impersonation Event Actions](#) (see page 339)
- [Configure a Rule for an Affiliate Agent](#) (see page 340)

Rule Group Properties Dialog

The Rule Group Properties Dialog is where you configure rule groups (sets of rules that can be bound to SiteMinder policies).

Rule Group Properties Dialog Prerequisites

The following prerequisites must be met in order to successfully configure a rule group using the Rule Group Properties Dialog:

- The Policy Server User Interface must be configured to show rule groups. If it is not, select View, Rule Groups from the menu bar.
- Rules that are to be added to the rule group must exist in realms in the same policy domain in which the rule group is to be configured.
- Other Rule groups that are to be added to the rule group must exist in the same Policy domain in which the rule group is to be configured.

Navigate to the Rule Group Properties Dialog to Create a Rule Group

How you access the rule group properties Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Group Properties Dialog to create a rule group

1. Do one of the following:

If you have the Manage System and Domain Objects privilege.

- In the Object pane, click on the Domains tab.
- Expand the policy domain to which you want to add a rule group by clicking on the + symbol to the left of the domain.

If you have the Manage Domain Objects privilege.

- In the Object pane, expand the policy domain to which you want to add a rule group by clicking on the + symbol to the left of the domain.

2. In the Object pane, click Rule Groups.
3. From the menu bar, select Edit, Create Rule Group

Navigate to the Rule Group Properties Dialog to Edit an Existing Rule

How you access the rule group properties Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Group Properties Dialog to edit an existing rule

1. Do one of the following

If you have the Manage System and Domain Objects privilege.

- In the Object pane, click on the Domains tab.
- Expand the policy domain that contains the rule group that you want to edit by clicking on the + symbol to the left of the domain.

You have the Manage Domain Objects privilege.

- In the Object pane, expand the policy domain to which you want to add a rule group by clicking on the + symbol to the left of the domain.
2. In the Object pane, click Rule Groups.
 3. In the List pane, click the rule group you want to edit.
 4. From the menu bar, select Edit, Properties of Rule Group

Rule Group Properties Dialog Fields and Controls

Name field

Name of the rule group. This name will be used to identify the rule group when incorporating it into a policy.

The value of the Name field can be no more than 255 characters.

Description field

Optional description of the new rule group.

The contents of the Description field must be less than 1K.

Agent Type Group Box

The Agent Type group box is where you identify the agent type for the rule group by clicking on the appropriate radio button and selecting from the drop-down list box. The Agent Type for a rule group corresponds to the Agent Type of rules that can be added to the rule group—only rules with the same Agent type will appear in the list of available rules when adding rules to the rule group.

SiteMinder

The rule group is associated with SiteMinder Agent types. Select the specific agent type from the drop-down list box:

- Web Agent
- Affiliate Agent
- EJB Agent

- Servlet Agent
- AffiliateMinder

RADIUS

The rule group is associate with RADIUS Agent types. Select the specific agent type from the drop-down list box:

- 3Com
- Ascend
- Bay Networks
- CheckPoint
- Cisco
- Generic RADIUS
- Livingstone
- Network Associate's Sniffer
- Shiva
- TeleBit
- U.S. Robotics

Group Members Group Box

The Group Members group box is where you can view, add, and remove the members (rules and rule groups) of the rule group.

Members list

A list of rules and rule groups currently in the rule group.

Add/Remove... button

Opens the Rule Group Items Dialog, which allows you to add or remove rules and rule groups to the rule group.

More information:

[Rule Group Items Dialog](#) (see page 702)

Tasks Related to the Rule Group Properties Dialog

The following tasks are related to the Rule Group Properties Dialog:

- [Create a Rule Group](#) (see page 350)
- [Rule Group Property Editing](#) (see page 351)

Rule Group Items Dialog

The Rule Group Items Dialog is where you specify the members (rules and rule groups) of a rule group. You can add any number of rules of the same Agent type to a rule group. All of the rules must exist in the same policy domain.

Rule Group Items Dialog Prerequisites

The following prerequisites must be met in order to successfully assign rule group members using the Rule Group Items Dialog:

- You must be in the process of creating or editing a rule group using the Rule Group Properties Dialog.

Navigate to the Rule Group Items Dialog

To access the Rule Group Items Dialog in the Policy Server User Interface to assign policy domains for which an administrator can create, edit, and delete objects, click the Add/Remove... button that appears in the Group Members group box on the Rule Group Properties Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Rule Group Items Dialog Fields and Controls

Use the Rule Group Items Dialog to select the members of a rule group.

Current Members list

Lists rules and rule groups that are in the rule group.

Available Members list

Lists all available rules and rule groups. Move items from this list to the Current Members list to add them to the rule group.

Arrow buttons

Move selected entries from one list to the other.

Left Arrow button

Moves selected items from the Current Members list to the Available Members list.

Right Arrow button

Moves selected items from the Available Members list to the Current Members list

Tasks Related to the Rule Group Items Dialog

The following task is related to the Rule Group Items Dialog:

- [Add a Rule To or Remove a Rule From a Rule Group](#) (see page 350)

Chapter 30: Response and Related Dialogs Reference

This section contains the following topics:

- [Response Dialog](#) (see page 705)
- [Response Attribute Editor Dialog](#) (see page 707)
- [Response Group Properties Dialog](#) (see page 712)
- [Response Group Items Dialog](#) (see page 715)

Response Dialog

The SiteMinder Response Dialog allows you to configure a response.

Response Object Prerequisites

The following prerequisites must be met in order to successfully create a response using the Response Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege for the policy domain in which the response is to be created.

Navigate to the Response Dialog

To access the Response Dialog in the Policy Server User Interface to create a new response object, from the menu bar on the Domains tab, select Edit, *domain_name*, Create Response.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Response Dialog Fields and Controls

Name field

Name of response.

Description field

(Optional) A description of the response.

Agent Type Group Box

The Agent Type group box is where you identify agent type for the response by clicking on the appropriate radio button and selecting from the drop-down list box. The Agent Type for a rule group corresponds to the type of agent to which the response will pass information.

SiteMinder

The response will pass information to a SiteMinder Agent. Select the specific agent type from the drop-down list box:

- Web Agent
- Affiliate Agent
- EJB Agent
- Servlet Agent
- AffiliateMinder

RADIUS

The response will pass information to a RADIUS Agent. Select the specific agent type from the drop-down list box:

- 3Com
- Ascend
- Bay Networks
- CheckPoint
- Cisco
- Generic RADIUS
- Livingstone
- Network Associate's Sniffer
- Shiva
- TeleBit
- U.S. Robotics

More information:

[Types of Responses](#) (see page 355)

Attribute List Group Box

The Attribute List group box is where you assign response attributes to the response object.

Attribute List

Clickable list of the attribute names assigned to this response.

Create button

Opens the Response Attribute Editor Dialog, from where you can create a new response attribute and add it to the response.

Edit button

Opens the response attribute selected in the Attribute List for editing in the Response Attribute Editor Dialog.

Remove button

Removes the selected response attribute from the Attribute List.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

Tasks Related to the Response Dialog

The following tasks are related to the Response Dialog:

- [Configure a Response](#) (see page 357)
- [Response Attributes](#) (see page 356)
- [Configure Response Attribute Caching](#) (see page 363)
- [Response Property Editing](#) (see page 374)
- [Response Deletion](#) (see page 372)

Response Attribute Editor Dialog

The Response Attribute Editor Dialog is where you configure response attributes.

Response Attribute Editor Dialog Prerequisites

The following prerequisites must be met in order to successfully create a response attribute using the Response Attribute Editor Dialog:

- You must be in the process of creating or editing a response object in the Response Dialog.

Navigate to the Response Attribute Editor Dialog to Create an Attribute

To access the Response Attribute Editor Dialog in the Policy Server User Interface to create a new response attribute, from the Response Dialog, click the Create button.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Navigate to the Response Attribute Editor Dialog to Edit an Attribute

To access the Response Attribute Editor Dialog in the Policy Server User Interface to edit an attribute, select an existing attribute from the Attribute List in the Response Dialog and click the Edit button.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

SiteMinder Response Attribute Editor Dialog Fields and Controls

The Response Attribute Editor Dialog contains the following controls and fields:

Attribute drop-down list

Lists the type of response attribute to add to the Web Agent response.

Attribute Setup tab

Shows the attribute kind and field.

Advanced tab

Displays and allows you to edit the script generated from the values entered in the Attribute Setup tab.

More information:

[Response Attributes](#) (see page 359)

[Response Attribute Editor Dialog - Attribute Setup Tab](#) (see page 709)

[Response Attribute Editor Dialog - Advanced Tab](#) (see page 712)

Attribute Caching Group Box

The Attribute Caching group box is where you specify whether the agent should cache the attribute value or periodically recalculate the value.

Cache value radio button

Agent should cache attribute value.

Recalculate value every <integer> seconds radio button

Attribute value varies over time; agent should recalculate at the specified interval.

More information:

[Configure Response Attribute Caching](#) (see page 363)

Response Attribute Editor Dialog - Attribute Setup Tab

The Attribute Setup tab is where you specify the attribute kind and fields.

Attribute Kind Group Box

The Attribute Kind group box is where you specify the type of response attribute.

Static radio button

If selected, the attribute returns the static value defined in the Variable Value (header variable) or Cookie Value (cookie variable) field.

User Attribute radio button

If selected, the attribute returns profile information stored in the user attribute specified in the Attribute Name field from a user's entry in a user directory.

DN Attribute radio button

If selected, the attribute returns profile information from a directory object associated with a DN in an LDAP or ODBC user directory (specified in the DN Spec and Attribute Name fields).

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Active Response radio button

If selected, the attribute returns values from a customer supplied library that is based on the SiteMinder Authorization API (defined in the Library Name, Function Name, and (optional) Parameters fields).

Variable Definition radio button

If selected, the attributes returns the value determined by the variable object specified in the Variable field.

Allow Nested Groups check box

(Active only when the DN Attribute option is selected in the Attribute Kind group box). If set, this option allows SiteMinder to return a response with an attribute contained within a group that is nested in a group specified by a policy. Nested groups often occur in complex LDAP deployments.

Attribute Fields Group Box

The Attribute Fields group box contains fields that specify information about the selected attribute type. The fields in this group box are context-sensitive, being determined according to the selection made in the Attribute Kind group box.

Variable Name field

Defines the name of the attribute SiteMinder returns in a header variable.

Note: Some of the attributes you can select from the Attribute drop-down list use default names and do not require you to enter a Variable Name.

Cookie Name field

Defines the name of the attribute SiteMinder returns in a cookie.

Variable Value field

Defines static text for the value of the attribute name/value pair SiteMinder returns in a header variable.

Cookie Value field

Defines static text for the value of the attribute name/value pair SiteMinder returns in a cookie.

Attribute Name field

Defines the attribute in a user directory that provides the attribute value.

DN Spec field

Defines the distinguished name of the user group from which SiteMinder should retrieve the user attribute. The DN must be related to the users for whom you want to return values to the consumer.

Lookup button

Opens the SiteMinder User Lookup Dialog to locate the user group and select a DN to populate the DN Spec field.

Library Name field

Defines the name of the shared library that will retrieve values for the response attribute.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default PATH for your system to search for the shared library at run-time.

You should not include a file extension for the shared library.

For example, if you are using a Windows platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory, you simply need to enter lib in the Library Name field.

Function Name field

Defines the name of the function in the shared library that will be called to retrieve values for the response attribute.

Parameters field

(Optional) Defines the list of parameters the Policy Server passes to the function in the shared library.

Variable field

Defines the variable object that provides the value for the response attribute.

Lookup button

Opens the SiteMinder Variable Lookup Dialog to locate the variable to populate the Variable field.

Response Attribute Editor Dialog - Advanced Tab

The Advanced tab displays the script that SiteMinder generates based on your entries in the Attribute Setup tab and allows you to edit it.

Script field

Initially displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can manually edit the contents of the Script field.

Note: Once you make any changes in the Script field, you cannot return to the Attribute Setup tab.

For example, you may be constructing an Active Response that returns the name of the attribute. So, instead of the standard:

```
Name=<@lib="MyLib.lib" func="Func" param="Param"@>
```

You may want to remove the Name= so that the Script field contains:

```
<@lib="MyLib.lib" func="Func" param="Param"@>
```

You can also copy the contents of the Script field and paste the contents into the Script field for another response.

Note: If you copy the contents of the Script field, when creating the destination attribute, select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab *before* moving to the Advanced tab. Once you paste into the Script field, you can't return to the Attribute Setup tab.

Tasks Related to the Response Attribute Editor Dialog

The following tasks are related to the Response Attribute Editor Dialog:

- [Response Attributes](#) (see page 356)
- [Configure Response Attribute Caching](#) (see page 363)
- [Use Variable Objects in Responses](#) (see page 364)

Response Group Properties Dialog

The SiteMinder Response Group Properties Dialog lets you configure a group of responses.

Response Group Properties Dialog Prerequisites

The following prerequisites must be met in order to successfully create a response group using the Response Group Properties Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege for the policy domain in which the response is to be created.
- At least one existing response

Navigate to the Response Group Properties Dialog

To access the Response Group Properties Dialog in the Policy Server User Interface, open the Domain tab and click Edit, Create Response Group.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Response Group Properties Dialog Fields and Controls

Name field

Name of response group.

Description field

(Optional) A description of the response group.

Agent Type Group Box

The Agent Type group box is where you identify agent type for the response group by clicking on the appropriate radio button and selecting from the drop-down list box. The Agent Type for a response group corresponds to the type of agent to which the response group will pass information.

SiteMinder

The response group will pass information to a SiteMinder Agent. Select the specific agent type from the drop-down list box:

- Web Agent
- Affiliate Agent
- EJB Agent
- Servlet Agent
- AffiliateMinder

RADIUS

The response group will pass information to a RADIUS Agent. Select the specific agent type from the drop-down list box:

- 3Com
- Ascend
- Bay Networks
- CheckPoint
- Cisco
- Generic RADIUS
- Livingstone
- Network Associate's Sniffer
- Shiva
- TeleBit
- U.S. Robotics

Attribute List Group Box

The Attribute List group box is where you assign response group attributes to the response group object.

Group Members

Clickable list of the response names assigned to the response group.

Add/Remove button

Opens the Response Group Items Dialog, from which you can add responses to or remove responses from the response group.

Tasks Related to the Response Group Properties Dialog

The following tasks are related to the Response Group Properties Dialog:

- [Configure a Response Group](#) (see page 373)
- [Response Group Property Editing](#) (see page 374)
- [Response Group Deletion](#) (see page 374)

Response Group Items Dialog

The Response Group Items Dialog is where you add responses to or remove responses from the response group.

Response Group Items Dialog Prerequisites

The following prerequisites must be met in order to successfully create a response group using the Response Group Items Dialog:

- You must have at least one existing response to add to the response group.

Navigate to the Response Group Items Dialog

To access the Response Group Item Dialog in the Policy Server User Interface to create or edit a response group, from the Response Group Properties Dialog, click Add/Remove.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Response Group Items Dialog Fields and Controls

The Response Group Items Dialog contains the following fields and controls:

Current Members

A clickable list of responses that are part of the group.

Available Members

A clickable list of responses that may be added to the group.

Chapter 31: Variable-related Dialogs Reference

This section contains the following topics:

[Static Variable Editor Dialog](#) (see page 717)
[Request Context Variable Editor Dialog](#) (see page 719)
[User Context Variable Editor Dialog](#) (see page 721)
[POST Variable Editor Dialog](#) (see page 725)
[Web Services Variable Editor Dialog](#) (see page 727)
[HTTP Header Dialog](#) (see page 730)
[Variable Lookup Dialog](#) (see page 731)

Static Variable Editor Dialog

The Static Variable Editor Dialog is where you create and edit static variables, which consist of a name and a fixed value.

Note: When you define a static variable, you must specify a variable return (data) type in addition to the value.

More information:

[Static Variables](#) (see page 426)

Static Variable Editor Dialog Prerequisites

The following prerequisites must be met in order to create or edit static variables in the Static Variable Editor Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or Manage Domain Objects privilege.
- The Policy Domain in which the variable is to be configured must exist.

More information:

[Administrator Scope and Tasks](#) (see page 227)

Navigate to the Static Variable Editor Dialog

How you access the Static Variable Editor Directory Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Static Variable Editor Directory

1. Do one of the following:

If you have the Manage System and Domain Objects privilege:

- In the Object pane, click on the Domains tab.
- Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

If you have the Manage Domain Objects privilege:

- In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

2. Click on the + symbol to the left of the Variables entry to expand the list of variable types.
3. Select the Static Variables entry.
4. From the menu bar, select Edit, Create Variable.

Static Variable Editor Dialog Fields and Controls

Name field

Name of the static variable object. This name will be used to identify the variable when incorporating it into a response or policy expression.

The value of the Name field can be no more than 255 characters.

Description field

(Optional) Description of the variable.

The contents of the Description field must be less than 1K.

Return Type drop down list

Indicates the type of data represented by the variable. Possible return types include the following:

Boolean

The variable returns a value of 1 (true) or 0 (false).

Date

The variable returns a date in the format of YYYY-MM-DD.

Number

The variable returns a number, including integers or decimals.

String

The variable returns a valid string of characters or a null value.

Value field

Value of the static variable. This value is substituted for the variable at runtime wherever you reference the variable in a policy expression or response.

Null check box

Indicates that the value of the static variable is null. If the Null option is set, the Value field is grayed-out.

Tasks Related to the Static Variable Editor Dialog

The following task is related to the Static Variable Editor Dialog:

- [Create a Variable](#) (see page 433)

Request Context Variable Editor Dialog

The Request Context Variable Editor Dialog is where you create and edit request context variables.

More information:

[Request Context Variables](#) (see page 426)

Request Context Variable Editor Dialog Prerequisites

The following prerequisites must be met in order to create or edit request context variables in the request context variable Editor Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or Manage Domain Objects privilege.
- The Policy Domain in which the variable is to be configured must exist.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Request Context Variable Editor Dialog

How you access the Request Context Variable Editor Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Request Context Variable Editor Dialog

1. Do one of the following:

If you have the Manage System and Domain Objects privilege.

- In the Object pane, click on the Domains tab.
- Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

If you have the Manage Domain Objects privilege.

- In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

2. Click on the + symbol to the left of the Variables entry to expand the list of variable types.
3. Select the Request Context Variables entry.
4. From the menu bar, select Edit, Create Variable

Request Context Variable Editor Dialog Fields and Controls

Name field

Name of the request context variable object. This name will be used to identify the variable when incorporating it into a response or policy expression.

The value of the Name field can be no more than 255 characters.

Description field

(Optional) Description of the variable.

The contents of the Description field must be less than 1K.

Property drop down list

Portion of the request context that provides the value of the variable. The Property is selected from a pull-down list that includes:

Action

The action requested by the user. For example, a typical action when accessing a web resource is GET.

Resource

Resource part of the user access request. For example, if the user attempts to access `http://server.company.com/dir1/index.html`, the resource portion of the request context is `/dir1/index.html`.

Server

Server part of the user access request. For example, if the user attempts to access `http://server.company.com/dir1/index.html`, the server portion of the request context is `http://server.company.com`.

Note: For more information about request context, see the *Developer's Guide for C* or the *Developer's Guide for Java*.

Tasks Related to the Request Context Variable Editor Dialog

The following task is related to the Request Context Variable Editor Dialog:

- [Create a Variable](#) (see page 433)

User Context Variable Editor Dialog

The User Context Variable Editor Dialog is where you create and edit user context variables.

More information:

[User Context Variables](#) (see page 426)

User Context Variable Editor Dialog Prerequisites

The following prerequisites must be met in order to create or edit user context variables in the User Context Variable Editor Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or Manage Domain Objects privilege.
- The Policy Domain in which the variable is to be configured must exist.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the User Context Variable Editor Dialog

How you access the User Context Variable Editor Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the User Context Variable Editor Dialog

1. Do one of the following:

If you have the Manage System and Domain Objects privilege:

- In the Object pane, click on the Domains tab.
- Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

If you have the Manage Domain Objects privilege:

- In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

2. Click on the + symbol to the left of the Variables entry to expand the list of variable types.
3. Select the User Context Variables entry.
4. From the menu bar, select Edit, Create Variable.

User Context Variable Editor Dialog Fields and Controls

Name field

Name of the user context variable object. This name will be used to identify the variable when incorporating it into a response or policy expression.

The value of the Name field can be no more than 255 characters.

Description field

(Optional) Description of the variable.

The contents of the Description field must be less than 1K.

Item drop down list

Indicates the portion of the user context that provides the value of the variable. The item is selected from a pull down list which includes:

Directory Entry Property

Attribute of a user directory entry.

Directory Namespace

Namespace of the directory, such as LDAP:, WinNT:, or ODBC:.

Directory Path

Directory path of the user directory connection where the user's context was established. The path uses the following notation:

<directory namespace> // <server>

where *<directory namespace>* is a namespace such as LDAP:, WinNT:, or ODBC:.

and *<server>* is the name of server on which the directory server is installed, for example:

server.company.com

Directory Server

Name of the server on which the directory server is installed, for example:

server.company.com

Is User Context Initialized

Boolean value that indicates whether or not the user's identity is available and the user context has been established.

Session ID

Session ID that has been or will be assigned to the user's session, depending on whether or not the session has been established.

User Path

Full user path in the following notation:

`<directory namespace> // <server> / <user DN>`

where `<directory namespace>` is a namespace such as LDAP:, WinNT:, or ODBC:, `<server>` is the name of the server on which the directory server is installed, and `<user DN>` is the distinguished name of the user. For example:

LDAP://server.company.com/uid=user1,ou=people,o=company.com

User Property

Attribute of a user.

Username

Full distinguished name of the user.

Return Type drop down list

Indicates the type of data represented by the variable. Possible return types include the following:

Boolean

The variable returns a value of 1 (true) or 0 (false).

Date

The variable returns a date in the format of YYYY-MM-DD.

Number

The variable returns a number, including integers or decimals.

String

The variable returns a valid string of characters or a null value.

Property field

For user context variables with an Item field selection of Directory Entry Property or User Property, this field contains the name of the attribute of the directory or user that provides the value of the variable.

DN field

For user context variables with an Item field selection of Directory Entry Property, this field provides the distinguished name of the directory entry.

Buffer Size field

For user context variables with an Item field selection of Directory Entry Property or User Property, this field contains the size (in bytes) of the buffer that will store the variable.

Note: For more information about user context, see the *Developer's Guide for C* or the *Developer's Guide for Java*.

Tasks Related to the User Context Variable Editor Dialog

The following task is related to the User Context Variable Editor Dialog:

- [Create a Variable](#) (see page 433)

POST Variable Editor Dialog

The POST Variable Editor Dialog is where you create and edit Form Post variables.

Note: Form Post variables are not supported by EJB or Servlet Agents. Do not use Form Post variables in policies enforced by EJB or Servlet Agents.

More information:

[Form Post Variables](#) (see page 427)

Post Variable Editor Dialog Prerequisites

The following prerequisites must be met in order to create or edit post variables in the Post Variable Editor Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or Manage Domain Objects privilege.
- The Policy Domain in which the variable is to be configured must exist.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Post Variable Editor Dialog

How you access the Post Variable Editor Dialog in the Policy Server User Interface depends on your administrative privileges.

To navigate to the Post Variable Editor Dialog

1. Do one of the following:

If you have the Manage System and Domain Objects privilege.

- In the Object pane, click on the Domains tab.
- Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

You have the Manage Domain Objects privilege.

- In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.
2. Click on the + symbol to the left of the Variables entry to expand the list of variable types.
 3. Select the Form Post Variables entry.
 4. From the menu bar, select Edit, Create Variable.

Post Variable Editor Dialog Fields and Controls

Name field

Name of the user form post variable object. This name will be used to identify the variable when incorporating it into a response, or policy expression.

The value of the Name field can be no more than 255 characters.

Description field

(Optional) Description of the variable.

The contents of the Description field must be less than 1K.

Form Field Name field

Name of the POST variable contained in the form.\

Tasks Related to the Post Variable Editor Dialog

The following tasks are related to the Post Variable Editor Dialog:

- [Form Post Variables](#) (see page 427)
- [Create a Variable](#) (see page 433)

Web Services Variable Editor Dialog

The Web Services Variable Editor Dialog is where you create and edit Web Services variables.

The Dialog contains two tabs:

General

Specifies general information about the Web services variable.

SOAP Document

Specifies information about the SOAP document that will be forwarded to the Web service.

HTTP Headers

Lists the HTTP headers associated with the Web services variable.

More information:

[Web Services Variables](#) (see page 427)

Web Services Variable Editor Dialog Prerequisites

The following prerequisites must be met in order to create or edit web services variables in the Web Services Variable Editor Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects or Manage Domain Objects privilege.
- The Policy Domain in which the variable is to be configured must exist.
- Web Service Variable Resolver must be configured.

More information:

[Policy Server Administrators](#) (see page 223)

[Configure the Web Service Variable Resolver](#) (see page 432)

Navigate to the Web Services Variable Editor Dialog

How you access the Web Services Variable Editor Dialog in the Policy Server User Interface depends on your administrative privileges.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Web Services Variable Editor Dialog

1. Do one of the following:

If you have the Manage System and Domain Objects privilege:

- In the Object pane, click on the Domains tab.
- Expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

If you have the Manage Domain Objects privilege:

- In the Object pane, expand the policy domain in which you want to create a variable by clicking on the + symbol to the left of the policy domain icon.

2. Click on the + symbol to the left of the Variables entry to expand the list of variable types.
3. Select the Web Service Variables entry.
4. From the menu bar, select Edit, Create Variable.

Web Services Variable Editor Dialog Fields and Controls—General Tab

There are three group boxes within the Web Services Variable Editor Dialog—General Tab.

- [Remote URL Group Box](#) (see page 728)
- [Return Query Group Box](#) (see page 729)
- [Web Service Credentials Group Box](#) (see page 729)

Remote URL group box

URL field

URL of the Web Service that will be used to resolve the Web Service variable.

SSL Connection check box

If selected, indicates whether or not the connection from the Policy Server to the Web Service will use SSL.

Note: You must select the SSL Connection check box even if you enter a complete URL starting with http or https.

Return Query group box

XPath field

Defines the XPath query that the Policy Server will use to extract the value of the Web Service variable from the SOAP document returned by the Web Service.

For more information about XPATH, see the XPATH specification available at:

<http://www.w3.org/TR/xpath.html>

Web Service Credentials group box

Require Credentials check box

If selected, indicates whether or not credentials must be presented to the Web Service by the Policy Server.

Use Clear Text Password check box

If the Require Credentials check box is selected, this setting indicates whether or not credentials will be sent to the Web Service in clear text. When this check box is selected, the Policy Server adds the password, in the clear, to the WS-Security digest. Otherwise, the Policy Server uses the base64-encoded SHA-1 digest of the password.

Note: If the Policy Server connects to a Web Service over an SSL connection, it is safe to send password in the clear, because the entire message is encrypted. However, not every Web Service may support base64-encoded SHA-1 password digests, in which case the combination of SSL encryption and clear text password may be required.

Username field

Indicates the user name that the Policy Server will use when connecting to the Web Service.

Password field

Indicates the password that the Policy Server will use when connecting to the Web Service.

Confirm Password field

Confirms the contents of the Password field.

The Web Services Variable Editor Dialog allows you to add user-defined HTTP headers.

Web Services Variable Editor Dialog Fields and Controls—SOAP Document Tab

SOAP Header

Defines the header of the SOAP message that will be forwarded to the Web Service.

Variable button

Opens the Variable Lookup Dialog from where you can lookup existing variables to include as part of the SOAP header.

SOAP Body

Defines the body of the SOAP message that will be forwarded to the Web Service.

Variable button

Opens the Variable Lookup Dialog from where you can lookup existing variables to include as part of the SOAP message.

More information:

[Variable Lookup Dialog](#) (see page 731)

Web Services Variable Editor Dialog Fields and Control - HTTP Headers Tab

HTTP Headers

Clickable list of the HTTP headers associated with the Web services variable.

Create

Opens the HTTP Header Dialog from which you can create an HTTP header.

Edit

Opens the HTTP Header Dialog from which you can edit an HTTP header.

Remove

Removes the selected HTTP header.

Tasks Related to the Web Services Variable Editor Dialog

The following task is related to the Web Services Variable Editor Dialog:

- [Create a Variable](#) (see page 433)

HTTP Header Dialog

The HTTP Header Dialog is where you create and edit HTTP Headers.

HTTP Header Dialog Prerequisites

The following prerequisites must be met in order to successfully create or edit an HTTP header response attribute using the HTTP Header Dialog:

- You must be in the process of creating or editing a Web service variable using the Web Services Variable Editor Dialog.

Navigate to the HTTP Header Dialog

From the HTTP Headers tab on the Web Services Variable Editor Dialog in the Policy Server User Interface, click Create. If an existing HTTP header is selected, click Edit.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

HTTP Header Dialog Fields and Controls

Name

Specifies the name of the HTTP header.

Value

Specifies the value of the HTTP header.

Variable Lookup Dialog

The Variable Lookup Dialog is where you select existing variables and insert them into policy expressions or responses.

Variable list

Selectable list of all existing variables in a domain, sorted by type. Click on the + symbol next to the variable type to expand the list of existing variables.

Chapter 32: Policy and Related Dialogs Reference

This section contains the following topics:

[Policy Dialog](#) (see page 733)

[Expression Editor](#) (see page 738)

[Rule Items Dialog](#) (see page 742)

[Set Response Dialog](#) (see page 744)

[Policy Expression Dialog](#) (see page 745)

Policy Dialog

Policies define how users interact with resources. When you create policies in the Policy Server User Interface, you link together (bind) objects that identify users, resources, and actions associated with the resources.

Policy Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- Before you can create a policy object, you must create one or more of the following Policy Server objects:
 - User Directory Object(s)
 - Policy Domain Objects(s)
 - Realm Objects(s)
 - Rule Object(s)

Policy objects may also see any of the following Policy Server Objects:

- Response Object(s)
- Response Group Object(s)

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Policy Dialog

To navigate to the Policy Dialog in the Policy Server User Interface, from the Domains tab of the Policy Server User Interface, right-click a domain object and select Create Policy.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Policy Dialog Fields and Controls

Name field

Name of the policy.

Description field

(Optional) Brief description of the policy.

Enabled check box

Select this check box to enable the policy. If the check box is not marked with a check, the policy is disabled. If you disable a policy, the policy does not fire.

Tabs

The Policy Dialog also contains a number of tabs. For information about the fields and controls contained on these tabs, see the following:

- [Policy Dialog—Users Tab](#) (see page 735)
- [Policy Dialog—Rules Tab](#) (see page 735)
- [Policy Dialog—IP Addresses Tab](#) (see page 736)
- [Policy Dialog—Time Tab](#) (see page 737)
- [Policy Dialog—Expression Tab](#) (see page 737)
- [Policy Dialog—Advanced Tab](#) (see page 738)

Policy Dialog—Users Tab

The Users tab is where you bind users and groups to the policy.

Directory tabs

The Users tab contains a listing of users and groups that are bound to the policy. If the current policy domain contains more than one user directory, the directories appear as subtabs within the User tab. In the image above, three user directories (Sample WinNT, Sample LDAP, and Sample ODBC) have been included in the policy domain that contains the policy.

Add/Remove button

Opens the Users/Group Dialog from where you can specify the users and groups that are bound to the policy. For more information, see [Users/Groups Dialog](#) (see page 912).

More information:

[Users/Groups Dialog](#) (see page 912)

Policy Dialog—Rules Tab

The Rules tab is where you add rules to and remove rules from the policy.

Rules list

Lists rules that are currently added to the policy. The Rules list has the following columns:

Rule column

Lists the names of rules that have been added to the policy.

Realm column

Lists the Resource Filter associated with the associated rule.

Response column

Lists a response or response group that has been associated with the rule.

Add/Remove Rules button

Opens the Rule Items Dialog. This Dialog allows you to add rules to the policy.

Set Response button

Opens the Set Response Dialog. This Dialog allows you to associate a response or response group with a rule.

Set Global Resp button

Opens the Set Global Response Dialog. This Dialog allows you to associate a global response with a rule.

Remove Response button

Removes the association between the selected rule/response or response group.

Note: If you use the Remove Response button to remove an association between a rule included in a policy, and a response (response group), the response (response group) remains in the policy store. However, the response (response group) is no longer associated with the policy.

More information:

[Rule Items Dialog](#) (see page 742)

[Set Response Dialog](#) (see page 744)

[Set Global Response Dialog](#) (see page 767)

Policy Dialog—IP Addresses Tab

The IP Addresses tab is where you optionally add, edit and delete IP addresses, ranges of IP addresses, subnet masks and hostname to be included in the policy. If specified, the policy only fires for users who access the policy's resources from the specified IP address(es).

IP Address list

Lists IP address entries included in the policy. The IP Address list has the following columns:

IP Address column

Lists IP addresses included in the policy. For IP address ranges, lists the first IP address in the range.

Subnet Mask column

Lists subnet masks included in the policy.

End of Range column

Lists the last IP addresses in ranges of IP addresses. The beginning of the range of IP addresses is indicated by the value in the IP Address column of the same row in the table.

Name column

Lists host names included in the policy.

Add button

Opens the Add an IP Address Dialog from where you can specify IP addresses, ranges of IP addresses, subnet masks and hostname to be included in the policy.

Edit button

Opens the Edit and an IP Address Dialog. This Dialog is identical to the Add an IP Address Dialog.

Remove button

Removes the selected IP address, range of IP addresses, subnet mask, or hostname from the policy.

More information:

[Add an IP Address Dialog](#) (see page 916)

Policy Dialog—Time Tab

The Time tab is where you specify time restrictions for the policy.

Set button

Opens the Time Dialog. This Dialog allows you to specify time restrictions for the policy.

Remove button

Removes time restrictions from the policy.

More information:

[Time Dialog](#) (see page 905)

Policy Dialog—Expression Tab

The Expression tab is where you can optionally specify an expression that the Policy Server evaluates at runtime to determine whether or not the policy applies to a request for resources.

Edit button

Opens the Policy Expression Dialog from where you can create a policy expression.

Clear button

Clears the existing expression from the policy.

More information:

[Policy Expression Dialog](#) (see page 745)

Policy Dialog—Advanced Tab

The Advanced tab is where you optionally specify an active policy created using APIs to enable custom functionality.

Edit button

Opens the Active Policy Editor. This Dialog allows you to specify an active policy.

Clear button

Clears the existing expression from the policy.

More information:

[Active Policy Editor Dialog](#) (see page 910)

Tasks Related to the Policy Dialog

The following tasks are related to the Policy Dialog:

- [Configure a Policy](#) (see page 378)
- [Enable and Disable Policies](#) (see page 390)
- [Advanced Policy Options](#) (see page 391)
- [Policy Binding Establishment](#) (see page 396)
- [Policy Deletion](#) (see page 405)

Expression Editor

The Expression Editor is where you bind an LDAP search expression to a policy. Search expressions can bind users to a policy based on attributes that appear in user, group, and organization profiles.

Expression Editor Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must edit an existing policy object, or create a new policy object.
- The policy object must be contained in a policy domain that includes an LDAP directory connection.

More information:

[Policy Server Administrators](#) (see page 223)

[Configure a Policy](#) (see page 378)

[Domains](#) (see page 307)

Navigate to the Expression Editor

You can access the Expression Editor from the Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Expression Editor

1. Open the Policy Dialog.
2. Select the Users tab.
3. If applicable, select the tab associated with the LDAP directory.

Note: If the policy domain that contains the policy object includes multiple user directory connections, the Users tab will contain a tab for each directory connection.

4. Click the Add/Remove button.

The Policy Users/Groups Dialog opens.

5. Click the Create New Expression or Edit Search Expression button.

The Expression Editor opens.

More information:

[Navigate to the Policy Dialog](#) (see page 734)

Expression Editor Fields and Controls

There are three group boxes within the Expression Editor:

- [Where to Search Group Box](#) (see page 740)
- [Condition Group Box](#) (see page 740)
- [Infix Notation Group Box](#) (see page 741)

Where to Search Group Box

The Where to Search group box is where you specify the type attribute for the search.

Where to Search drop down list

Type of attribute for the search. The following options are available:

User Attribute

Indicates that the LDAP search is limited to matches in user entries.

Group Attribute

Indicates that the LDAP search is limited to matches in group entries.

Org Attribute

Indicates that the LDAP search is limited to matches in organization entries (organizations and organizational units).

Any Attribute

Indicates that the LDAP search includes all entries in the directory.

Condition Group Box

The Condition group box is where you create conditions to add to the LDAP search expression.

Attribute (left) combo box

Specifies the attribute for the search condition. The types of attributes that are available in this group box change according to the value you select from the drop down list in the Where to Search group box. If the attribute name that you want to use does not appear in the drop down list, you can type an attribute name in the field.

Operator (middle) drop down list

Specifies the operator for the search condition:

- equals
- approx
- greater than or equal

- less than or equal
- contains

Value (right) field

Specifies the desired value to completes the condition.

Add button

Adds the specified search condition to the LDAP search expression.

For example, to create a condition in an expression that includes an ou=marketing attribute, you can specify Organization in the Attribute combo box list, choose equals from the Operator drop down list, and enter a value of marketing in the Value field.

Infix Notation Group Box

The Infix Notation group box is where you view and manipulate the conditions of the LDAP expression.

Condition list

Lists all the conditions in the search expression using infix notation.

Note: By default, the second and all subsequent conditions are added to the Infix Notation group box with an and operator. You can group conditions and change between and operators and or operators using the associated control buttons to the right.

And/Or button

With the second or subsequent condition selected in the Condition list, toggles the operator between and and or.

Not button

Adds a not operator to the condition selected in the Condition list.

() button

With two or more conditions selected in the Condition list, adds parentheses to the selected conditions.

Modify button

Opens the Edit Dialog, in which you can modify the condition selected in the Condition list. (The Edit Dialog contains the same controls as the Condition group box.)

Remove button

Removes the condition selected in the Condition list.

LDAP Notation Group Box

The LDAP Notation group box displays the entire LDAP expression by showing the conditions in the Infix Notation group box using the appropriate LDAP notation.

Tasks Related to the Expression Editor

The following task is related to the Expression Editor:

- [Add LDAP Expressions to Policies](#) (see page 387)

Rule Items Dialog

The Rule Items Dialog provides an interface for specifying rules and rule groups to be included in the policy.

Rule Items Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must edit an existing policy object, or create a new policy object.

More information:

[Policy Server Administrators](#) (see page 223)

[Policies](#) (see page 375)

Navigate to the Rule Items Dialog

You can access the Rule Items Dialog from the Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Items Dialog

1. Open the Policy Dialog.
2. Select the Rules tab.
3. Click the Add/Remove Rules button.

The Rule Items Dialog opens.

More information:

[Navigate to the Policy Dialog](#) (see page 734)

Rule Items Dialog Fields and Controls

Current Members list

Lists the rules and rule groups currently included in the policy. You can select multiple rules and rule groups by holding the CTRL or SHIFT key when clicking items.

Available Members list

Lists the rules and rule groups that are available for inclusion in the policy. Available rules and rule groups consist of all of the rules associated with realms in the policy domain, as well as all of the rule groups defined for the policy domain. You can select multiple rules and rule groups by holding the CTRL or SHIFT when clicking items.

Search button

Allows you to use a search filter when looking for a rule in the Available Members list.

When you click the Search button, the space to the left of the button displays the Filter fields.

Enter a string in the Filter field. Click Go and the items displayed in the Available Members list will be limited according to the string you entered. Click All to show all rules and rule groups. Click the button with the two arrows to return to the default state of the dialog.

Previous/Next/All buttons

When a policy domain contains enough rules and rule groups to require multiple pages of information, the previous and next buttons page through the available items.

Left Arrow button

Moves selected rules and rule groups from the Available Members list to the Current Members list.

Right Arrow button

Moves selected rules and rule groups from the Current Members to the Available Members list.

More information:

[Domains](#) (see page 307)

Tasks Related to the Rule Items Dialog

The following tasks are related to the Rule Items Dialog:

- [Configure a Rule for Web Agent Actions](#) (see page 334)
- [Configure a Rule for Authentication Event Actions](#) (see page 336)

- [Configure a Rule for Authorization Event Actions](#) (see page 337)
- [Configure a Rule for Impersonation Event Actions](#) (see page 339)
- [Configure a Rule for an Affiliate Agent](#) (see page 340)

Set Response Dialog

This Dialog lists the responses and response groups that have been configured for the policy's policy domain. You can select a single response or response group from the list to be associated with a rule or rule group in a policy.

Set Response Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must edit an existing policy object, or create a new policy object.

More information:

[Policy Server Administrators](#) (see page 223)
[Policies](#) (see page 375)

Navigate to the Set Response Dialog

You can access the Set Response Dialog from the Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Set Response Dialog

1. Open the Policy Dialog.
2. Select the Rules tab.
3. Click the Set Response button.

More information:

[Navigate to the Policy Dialog](#) (see page 734)

Set Response Dialog Fields and Controls

Response list

List the responses and response groups currently configured for the policy's policy domain. The list contains the following columns:

Name column

Lists the names of responses and response groups that exist in the policy's policy domain.

Agent Type column

Lists the Agent types associated with the responses.

Description column

Lists the descriptions associated with the responses.

Search button

Allows you to specify a search filter when looking for a response or response group.

When you click the Search button, the space to the left of the button displays the Filter field.

Enter a string in the Filter field. Click Go and the items displayed in the dialog will be limited according to the string you entered. Click All to show all responses and response groups. Click the button with the two arrows to return to the default state of the dialog.

Previous/Next/All buttons

When a policy domain contains enough responses and response groups to require multiple pages of information, the previous and next buttons page through the available items.

Tasks Related to the Set Response Dialog

The following task is related to the Set Response Dialog:

- [Associate a Rule with a Response or Response Group](#) (see page 380)

Policy Expression Dialog

The Policy Expression Dialog is where you create boolean expressions that incorporate variable objects in policy decisions. For more information, see the *Federation Security Services Guide*.

Policy Expression Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must edit an existing policy object, or create a new policy object.

More information:

[Policy Server Administrators](#) (see page 223)

[Policies](#) (see page 375)

Navigate to the Policy Expression Dialog

You can access the Policy Expression Dialog from the Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Policy Expression Dialog

1. Open the Policy Dialog.
2. Select the Expression tab.
3. Click the Edit button.

The Policy Expression Dialog opens.

More information:

[Navigate to the Policy Dialog](#) (see page 734)

Policy Expression Dialog Fields and Controls

The Policy Expression Dialog contains three group boxes:

- [Condition Group Box](#) (see page 747)
- [Infix Notation Group Box](#) (see page 748)
- [Resulting Notation Group Box](#) (see page 748)

Condition Group Box

The Condition group box is where you create conditions to add to the policy expression.

Attribute (left) field

Specifies the variable name for the policy expression condition. Type a value, or click the search icon to select a variable object using the Variable Lookup Dialog.

Operator (middle) drop down list

Specifies the operator for the expression condition. Selections include:

equals

Indicates that the variable specified in the first portion of the condition must equal the value in the second portion of the condition.

greater

Indicates that the variable specified in the first portion of the condition must be greater than the value in the second portion of the condition.

less

Indicates that the variable specified in the first portion of the condition must less than the value in the second portion of the condition.

greaterOrEqual

Indicates that the variable specified in the first portion of the condition greater than or equal to the value in the second portion of the condition.

lessOrEqual

Indicates that the variable specified in the first portion of the condition must less than or equal to the value in the second portion of the condition.

IsNull

Indicates that the variable specified in the first portion of the condition must have a null value.

Value field

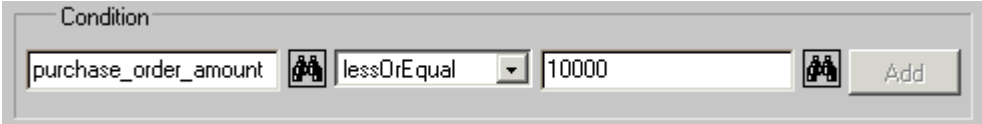
Specifies a value for the policy expression condition. Type a value, or click the search icon to select an existing variable object using the Variable Lookup Dialog.

Note: When entering a string value, the string must be contained in quotes.

Add button

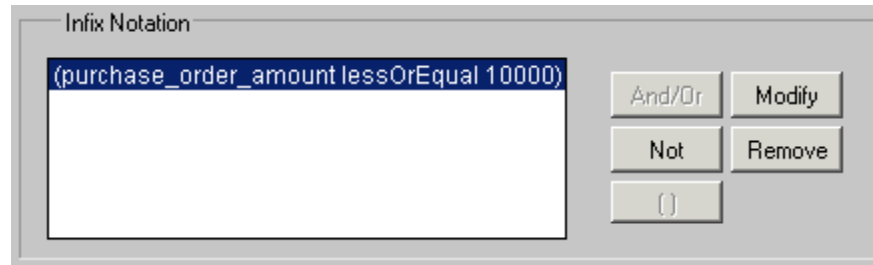
Adds the condition to the Expression list in the Infix Notation group box.

For example, if you enter the following condition and click Add,



The screenshot shows a dialog box titled "Condition". It contains a text field with the value "purchase_order_amount", a search icon, a dropdown menu with "lessOrEqual" selected, another text field with the value "10000", a second search icon, and an "Add" button.

The following appears in the Infix Notation group box:



Infix Notation Group Box

The Infix Notation group box is where you view and manipulate the conditions of the policy expression.

Condition list

Displays the conditions that have been added to the policy expression.

And/Or button

With the second or subsequent condition selected in the Condition list, toggles the operator between and and or.

Not button

Adds a not operator to the condition selected in the Condition list.

() button

With two or more conditions selected in the Condition list, adds parentheses to the selected conditions.

Modify button

Opens the Edit Dialog, in which you can modify the condition selected in the Condition list. (The Edit Dialog contains the same controls as the Condition group box.)

Remove button

Removes the condition selected in the Condition list.

Resulting Notation Group Box

The Resulting Notation group box displays the complete policy expression that results from the condition entries in the Infix Notation group box.

Tasks Related to the Policy Expression Dialog

The following task is related to the Policy Expression Dialog:

- [Expressions in Policies](#) (see page 413)

Chapter 33: Global Policies, Rules, and Responses Dialogs Reference

This section contains the following topics:

[Global Rule Dialog](#) (see page 751)

[Global Response Dialog](#) (see page 755)

[Global Response Attribute Editor Dialog](#) (see page 757)

[Global Policy Dialog](#) (see page 761)

[Rule Items Dialog](#) (see page 765)

[Set Global Response Dialog](#) (see page 767)

Global Rule Dialog

The SiteMinder Global Rule Dialog is where you create and edit SiteMinder global rules.

Global Rule Object Prerequisites

The following prerequisites must be met in order to successfully create a global rule object using the Global Rule Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects.

Navigate to the Global Rule Dialog

To navigate to the Global Rule Dialog in the Policy Server User Interface, from the Global Policies tab of the Policy Server User Interface, select Edit, Global Policies, Create Rule.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Global Rule Dialog Fields and Controls

Name field

Name of the global rule.

Description field

Optionally, a description of the global rule.

Resource Group Box

The Resource group box is where you select the resources to which this global rule will apply.

Agent field

Name of the agent or agent group for which the global rule should apply.

Lookup button

Opens the Set Agent/Group Dialog from which you pick an agent or agent group from a list of all such existing objects or search for one by name.

If you specify an agent group, in a global rule, and you have also configured domain-specific rules associated with the same resource, you can adversely affect system performance by effectively duplicating processing steps. When configuring global rules and global policies, be sure to consider domain-specific rules that may duplicate the responses generated by global rules. Note that in such cases, only one response is returned to the agent, since the Policy Server automatically deletes duplicate responses before passing information back to a requesting Agent.

Resource Filter field

Specifies the path for the resource filter that will determine the resources affected by the global rule. This field functions as a root for locating resources.

Perform regular expression pattern matching check box

Specifies that a resource can be a specific file or an expression that uses resource matching with regular expressions.

More information:

[Resource Matching and Regular Expressions](#) (see page 341)

Allow/Deny and Enable/Disable Group Box

The Allow/Deny and Enable/Disable group box is where you specify whether the global rule is enabled or disabled.

Enabled check box

If set, the rule is enabled. If not set, the rule is disabled and the rule will not fire.

Action Group Box

The Action group box is where you specify the type of event and the specific action that must take place for the global rule to fire:

Authentication Events radio button

Specifies that the rule should fire for the authentication event selected from the drop-down list box that appears to the right of the radio button when this entry is selected:

OnAuthAccept

Occurs when a user successfully authenticates. This event may be used to redirect a user after a successful authentication

OnAuthAttempt

Occurs when a user fails to authenticate because no user name was supplied.

OnAuthChallenge

May be used in custom authentication schemes to trigger a response.

OnAuthReject

Occurs when a user that is bound to a policy fails to authenticate. The user must be bound to the policy for the rule to fire.

OnAuthUserNotFound

This event is only used to trigger Active Responses. This event should not be used to trigger any response other than an Active Response.

Authorization Events radio button

Specifies that the rule should fire for the authorization event selected from the drop-down list box that appears to the right of the radio button when this entry is selected

OnAccessAccept

Occurs when SiteMinder successfully authorizes a user to access the resource.

OnAccessReject

Occurs when SiteMinder rejects a user because the user is not authorized to access the resource.

Advanced Group Box

The Advanced group box is where you define additional global rule settings. This group box contains two tabs:

Time Restrictions tab

Allows you to manage time restrictions for the rule (that is, when the rule should and should not fire).

Set button

Opens the Time Dialog in which you can add, or if they already apply, edit time restrictions for the rule.

Remove button

Remove all time restrictions set for the rule (grayed-out when no time restrictions are present).

Active Rules tab

Allows you to manage active rules for dynamic authorization based on external business logic.

Active Rule field

Displays the active rule, if one has been defined.

Edit button

Opens the Active Rule Editor Dialog in which you can define a new, or edit an existing active rule.

Clear button

Remove an existing active rule (grayed-out when none exists).

More information:

[Time Dialog](#) (see page 905)

[Active Rule Editor Dialog](#) (see page 909)

Tasks Related to the Global Rule Dialog

The following tasks are related to the Global Rule Dialog:

- [Global Rule Objects](#) (see page 440)
- [Global Rule Deletion](#) (see page 445)
- [Add Global Rules and Global Responses to a Global Policy](#) (see page 449)

Global Response Dialog

The SiteMinder Global Response Dialog is where you configure global responses.

Global Response Object Prerequisites

The following prerequisites must be met in order to successfully create a global response using the Global Response Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Global Responses Dialog

To navigate to the Global Responses Dialog in the Policy Server User Interface, from the Global Policies tab of the Policy Server User Interface, select Edit, Global Policies, Create Response.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Global Response Dialog Fields and Controls

Name

Name of global response.

Description

(Optional) A description of the global response.

Agent Type Group Box

The Agent Type group box is where you identify agent type for the response by clicking on the appropriate radio button and selecting from the drop-down list box. The Agent Type for a rule group corresponds to the type of agent to which the response will pass information.

SiteMinder drop down list

Specifies the type of SiteMinder Agent to which the response will pass information:

- Web Agent
- Affiliate Agent
- EJB Agent

- Servlet Agent
- AffiliateMinder

More information:

[Types of Responses](#) (see page 355)

Attribute List Group Box

The Attribute List group box is where you assign response attributes to the response object.

Attribute list

Clickable list of the attribute names assigned to this response.

Create button

Opens the Response Attribute Editor Dialog, from where you can create a new response attribute and add it to the response.

Edit button

Opens the response attribute selected in the Attribute List for editing in the Response Attribute Editor Dialog.

Remove button

Removes the selected response attribute from the Attribute List.

Create button

Opens the Global Response Attribute Editor Dialog, which is similar to the Response Attribute Editor Dialog.

More information:

[Response Attributes](#) (see page 359)

[Global Response Attribute Editor Dialog](#) (see page 757)

Tasks Related to the Global Response Dialog

The following tasks are related to the Global Response Dialog:

- [Configure a Global Response](#) (see page 445)
- [Add Global Rules and Global Responses to a Global Policy](#) (see page 449)

Global Response Attribute Editor Dialog

The Global Response Attribute Editor Dialog is where you configure global response attributes.

Global Response Attribute Editor Dialog Prerequisites

The following prerequisites must be met in order to successfully create a global response attribute using the Global Response Attribute Editor Dialog:

- You must be in the process of creating or editing a global response object in the Global Response Dialog.

Navigate to the Global Response Attribute Editor Dialog

To access the Response Attribute Editor Dialog in the Policy Server User Interface to create a new global response attribute, from the Global Response Dialog, click the Create button to create a new attribute.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

SiteMinder Global Response Attribute Editor Dialog Fields and Controls

The Global Response Attribute Editor Dialog contains the following controls and fields:

Attribute drop-down list

Lists the type of response attribute to add to the Web Agent response.

Attribute Setup tab

Shows the attribute kind and field.

Advanced tab

Displays and allows you to edit the script generated from the values entered in the Attribute Setup tab.

More information:

[Response Attributes](#) (see page 359)

[Global Response Attribute Editor Dialog - Attribute Setup Tab](#) (see page 758)

[Global Response Attribute Editor Dialog - Advanced Tab](#) (see page 760)

Attribute Caching Group Box

The Attribute Caching group box is where you specify whether the agent should cache the attribute value or periodically recalculate the value.

Cache value radio button

Agent should cache attribute value.

Recalculate value every *<integer>* seconds radio button

Attribute value varies over time; agent should recalculate at the specified interval.

More information:

[Configure Response Attribute Caching](#) (see page 363)

Global Response Attribute Editor Dialog - Attribute Setup Tab

The Attribute Setup tab is where you specify the attribute kind and fields.

Attribute Kind Group Box

The Attribute Kind group box is where you specify the type of response attribute.

Static radio button

If selected, the attribute returns the static value defined in the Variable Value (header variable) or Cookie Value (cookie variable) field.

User Attribute radio button

If selected, the attribute returns profile information stored in the user attribute specified in the Attribute Name field from a user's entry in a user directory.

DN Attribute radio button

If selected, the attribute returns profile information from a directory object associated with a DN in an LDAP or ODBC user directory (specified in the DN Spec and Attribute Name fields).

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Active Response radio button

If selected, the attribute returns values from a customer supplied library that is based on the SiteMinder Authorization API (defined in the Library Name, Function Name, and (optional) Parameters fields).

Allow Nested Groups check box

(Active only when the DN Attribute option is selected in the Attribute Kind group box). If set, this option allows SiteMinder to return a response with an attribute contained within a group that is nested in a group specified by a policy. Nested groups often occur in complex LDAP deployments.

Attribute Fields Group Box

The Attribute Fields group box contains fields that specify information about the selected attribute type. The fields in this group box are context-sensitive, being determined according to the selection made in the Attribute Kind group box.

Variable Name field

Defines the name of the attribute SiteMinder returns in a header variable.

Note: Some of the attributes you can select from the Attribute drop-down list use default names and do not require you to enter a Variable Name.

Cookie Name field

Defines the name of the attribute SiteMinder returns in a cookie.

Variable Value field

Defines the static text that will be returned as the second half of the attribute name/value pair SiteMinder returns in a header variable.

Cookie Value field

Defines static text for the value of the attribute name/value pair SiteMinder returns in a cookie.

Attribute Name field

Defines the attribute in a user directory that provides the attribute value.

DN Spec field

Defines the distinguished name of the user group from which SiteMinder should retrieve the user attribute. The DN must be related to the users for whom you want to return values to the consumer.

Lookup button

Opens the SiteMinder User Lookup Dialog to locate the user group and select a DN to populate the DN Spec field.

Library Name field

Defines the name of the shared library that will retrieve values for the response attribute.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default PATH for your system to search for the shared library at run-time.

You should not include a file extension for the shared library.

For example, if you are using a Windows platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory, you simply need to enter lib in the Library Name field.

Function Name field

Defines the name of the function in the shared library that will be called to retrieve values for the response attribute.

Parameters field

(Optional) Defines the list of parameters the Policy Server passes to the function in the shared library.

Global Response Attribute Editor Dialog - Advanced Tab

The Advanced tab displays the script that SiteMinder generates based on your entries in the Attribute Setup tab and allows you to edit it.

Script field

Initially displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can manually edit the contents of the Script field.

Note: Once you make any changes in the Script field, you cannot return to the Attribute Setup tab.

For example, you may be constructing an Active Response that returns the name of the attribute. So, instead of the standard:

```
Name=<@lib="MyLib.lib" func="Func" param="Param"@>
```

You may want to remove the Name= so that the Script field contains:

```
<@lib="MyLib.lib" func="Func" param="Param"@>
```

You can also copy the contents of the Script field and paste the contents into the Script field for another response.

Note: If you copy the contents of the Script field, when creating the destination attribute, select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab *before* moving to the Advanced tab. Once you paste into the Script field, you can't return to the Attribute Setup tab.

Tasks Related to the Global Response Attribute Editor Dialog

The following tasks are related to the Global Response Attribute Editor Dialog:

- [Response Attributes for Global Responses](#) (see page 446)

Global Policy Dialog

The Global Policy Dialog is where you configure global policies.

Global Policy Object Prerequisites

The following prerequisites must be met in order to successfully create a global policy object using the Global Policy Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects.
- Before you can create a global policy object, you must create one or more Global Rule Objects.
- If you want to associate responses with your global rules, you must also have created one or more global response objects.

Navigate to the Global Policy Dialog

To navigate to the Global Policy Dialog in the Policy Server User Interface, from the Global Policies tab of the Policy Server User Interface, select Edit, Global Policies, Create Policy.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Global Policy Dialog Fields and Controls

Name field

Name of the global policy object.

Description field

Optionally, a brief description of the global policy.

Enabled check box

If the check box is marked with a check, the global policy is enabled. If the check box is not marked with a check, the policy is disabled. If you disable a policy, the policy does not fire.

Tabs

For information about the fields and controls contained on the tabs in the Global Policy Dialog, see one of the following:

- [Global Policy Dialog—Users Tab](#) (see page 762)
- [Global Policy Dialog—Rules Tab](#) (see page 762)
- [Global Policy Dialog—IP Addresses Tab](#) (see page 763)
- [Global Policy Dialog—Time Tab](#) (see page 764)
- [Global Policy Dialog—Advanced Tab](#) (see page 764)

Global Policy Dialog—Users Tab

The Users tab is informational only. It simply serves as a reminder that global policies apply to all user directories.

Global Policy Dialog—Rules Tab

The Rules tab is where you specify the global rules and (optionally) global responses in a global policy.

Rule list

Selectable list of the rules and associated responses currently in the global policy.

Add/Remove Rules button

Opens the Rule Items Dialog, from where you can add global rules to, and remove global rules from, the global policy.

Set Response button

(Active when a global rule is selected in the Rule list.) Opens the Set Response Dialog, from where you can choose a global response to associate with a global rule selected in the Rule list.

Remove Response button

(Active when a global rule with an associate response is selected in the Rule list.) Removes the response associated with a global rule selected in the Rule list.

More information:

[Rule Items Dialog](#) (see page 742)

[Set Global Response Dialog](#) (see page 767)

Global Policy Dialog—IP Addresses Tab

The IP Addresses tab is where you optionally specify IP addresses (in the form of hostnames, IP addresses, ranges of IP addresses, and subnet masks) of users requesting protected resources in order for a global policy to fire. For example, if you specify a range of IP addresses, only those users who login in from one of the specified IP addresses will trigger the global policy.

IP Address list

Selectable list of IP address requirements for the global policy. List columns are:

IP Address

This column lists IP addresses included in the policy.

Subnet Mask

This column lists subnet masks included in the policy.

End of Range

This column lists the last IP addresses in ranges of IP addresses. The beginning of the range of IP addresses is indicated by the value in the IP Address column of the same row in the table.

Name

Host name included in the policy.

Add button

Opens the Add an IP address Dialog from where you can add an IP address, range of IP addresses, subnet mask, or hostname to your global policy.

Edit button

Opens the Edit and an IP Address Dialog from where you can edit the IP address entry selected in the IP Address list. This Dialog is functionally identical to the Add an IP Address Dialog.

Remove button

(Active when an IP address entry is selected in the IP Address list.) Removes the IP address entry selected in the IP Address list from the global policy.

More information:

[Add an IP Address Dialog](#) (see page 916)

Global Policy Dialog—Time Tab

The Time tab is where you manage time restrictions for the global policy (that is, when the rule should and should not fire)

Set button

Opens the Time Dialog, from which you can add, or if they already apply, edit time restrictions for the global policy.

Remove button

Removes time restrictions from the policy.

More information:

[Time Dialog](#) (see page 905)

Global Policy Dialog—Advanced Tab

The Advanced tab is where you add, edit, and remove an active policy for the global policy. Active policies enable custom functionality.

Expression field

Displays the active policy expression.

Edit button

Opens the SiteMinder Active Policy Editor Dialog. This Dialog allows you to specify an active policy created using APIs.

Clear button

Clears the existing expression from the policy.

More information:

[Active Policy Editor Dialog](#) (see page 910)

Tasks Related to the Global Policy Dialog

The following tasks are related to the Global Policy Dialog:

- [Configure Global Policies](#) (see page 449)
- [Configure Global Policy Objects](#) (see page 449)
- [Enable and Disable Global Policies](#) (see page 450)
- [Specify IP Addresses for Global Policies](#) (see page 451)
- [Add and Remove Global Policy Time Restrictions](#) (see page 454)

Rule Items Dialog

The Rule Items Dialog is where you specify the global rules in a global policy. You can add any number of global rules to a global policy.

Rule Items Dialog Prerequisites

You must be in the process of creating or editing a global policy object.

More information:

[Configure Global Policy Objects](#) (see page 449)

Navigate to the Rule Items Dialog

You can access the Rule Items Dialog from the Global Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Rule Items Dialog

1. Open the Global Policy Dialog.
2. Select the Rules tab.
3. Click the Add/Remove Rules button.

The Rule Items Dialog opens.

More information:

[Navigate to the Global Policy Dialog](#) (see page 761)

Rule Items Dialog Fields and Controls

Current Members list

Lists global rules that are in the global policy.

Available Members list

Lists all available global rules. Move items from this list to the Current Members list to add them to the rule group.

Note: You can select multiple rules and rule groups by holding the CTRL or SHIFT key and clicking on items in one of the Members lists. When you select multiple items and click one of the Arrow buttons, all of the selected items are moved to the other list.

Arrow buttons

Move selected entries from one list to the other:

Left Arrow button

Moves selected items from the Current Members list to the Available Members list.

Right Arrow button

Moves selected items from the Available Members list to the Current Members list.

Search button

The following button allows you to use a search filter when looking for a global rule in the Available Members list.

When you click the Search button, the space to the left of the button displays the Filter field.

Enter a string in the Filter field. Click Go and the items displayed in the Available Members list will be limited according to the string you entered. Click All to show all global rules. Click the button with the two arrows to return to the default state of the dialog.

Previous/Next/All buttons

When there are enough global rules to require multiple pages of information, the previous and next buttons page through the available items.

Tasks Related to the Rule Items Dialog

The following task is related to the Rule Items Dialog:

- [Add Global Rules and Global Responses to a Global Policy](#) (see page 449)

Set Global Response Dialog

This dialog contains a list of the global responses that have been configured. You can select a single response from the list to be associated with a rule or rule group in a policy or global policy.

Set Global Response Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.
- You must be in the process of editing an existing policy or global policy object, or creating a new policy or global policy object.

More information:

[Configure Global Policy Objects](#) (see page 449)

Navigate to the Set Global Response Dialog

You can access the Set Global Response Dialog from the Global Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Set Global Response Dialog

1. Open the Global Policy Dialog.
2. Select the Rules tab.
3. Click the Set Response button.

The Set Global Response Dialog opens.

More information:

[Navigate to the Global Policy Dialog](#) (see page 761)

Set Global Response Dialog Fields and Controls

Global Response list

Displays the names, associated Agent types, and descriptions of existing global responses. Select the global response that you want to associate with a global rule in a global policy by clicking an entry in this list.

Search button

The following button allows you to use a search filter when looking for a global response.

When you click the Search button, the space to the left of the button displays the filter field.

Enter a string in the Filter field. Click Go and the items displayed in the dialog will be limited according to the string you entered. Click All to show all global responses. Click the button with the two arrows to return to the default state of the dialog.

Previous/Next/All buttons

When enough global responses exist to require multiple pages of information, the previous and next buttons page through the available items.

Tasks Related to the Set Global Response Dialog

The following task is related to the Set Global Response Dialog:

- [Add Global Rules and Global Responses to a Global Policy](#) (see page 449)

Chapter 34: Password Services Dialog Reference

This section contains the following topics:

[Password Policy Dialog](#) (see page 769)

Password Policy Dialog

The Password Policy Dialog is where you configure password policies.

Password Policy Dialog Prerequisites

The following prerequisites must be met in order to successfully create and edit password policies using the Password Policy Dialog:

- Your SiteMinder administrator account must have Manage Password Policies privilege for the domain for which you want to configure password policies.
- The user directory connection that specifies the users to whom you want the password policy to apply must be fully configured and see an LDAP user store.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Password Policy Dialog to Create a New Policy

There are multiple ways to access the Password Policy Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Password Policy Dialog to create a new password policy

1. In the Object pane, click the System tab.
2. In the System tab, click Password Policies.
3. From the menu bar select Edit, Create Password Policy.

Navigate to the Password Policy Dialog to Modify an Existing Policy

There are multiple ways to access the Password Policy Dialog in the Policy Server User Interface. The following procedure is the method recommended by CA.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Password Policy Dialog to modify an existing password policy:

1. In the Object pane, select the System tab.
2. In the System tab, click Password Policies.
3. In the list pane, right-click the password policy you want to edit and select Properties of the Password Policy from the popup menu.

Password Policy Dialog Fields and Controls

Name field

The name of the password policy.

Description field

Optionally, a brief description of the password policy.

The Password Policy Dialog also contains the following tabs:

General

Allows you to bind the password policy to all or part of a user directory, and enable the password policy.

Expiration

Lets you configure criteria for password expiration such as the maximum number of failed login attempts or the amount of time an account can be inactive before it is disabled.

Composition

Enables you to specify rules governing the composition of valid passwords such as minimum length and the minimum numbers that comprise a password.

Regular Expressions

Allows you to specify regular expressions that valid passwords must match or not match.

Restrictions

Allows you to set restrictions for reusing passwords, creating similar passwords, and using specific words.

Advanced

Allows you to specify the order in which password policies that apply to the same user directory or namespace should be evaluated and set password preprocessing options (for example, removing white space).

More information:

[Password Policy Dialog—General Tab](#) (see page 771)

[Password Policy Dialog—Expiration Tab](#) (see page 774)

[Password Policy Dialog—Composition Tab](#) (see page 777)

[Password Policy Dialog—Regular Expressions Tab](#) (see page 779)

[Password Policy Dialog—Restrictions Tab](#) (see page 782)

[Password Policy Dialog—Advanced Tab](#) (see page 785)

Password Policy Dialog—General Tab

The General tab is where you configure general properties for your password policy.

User Directory Information Group Box

The User Directory Information group box is where you specify the LDAP user directory connection and users within that directory to which the password policy will apply.

Directory drop-down list box

The user directory connection to which the password policy will apply.

Password Policy applies to the whole directory radio button

Specifies that the password policy should apply to the entire user directory.

Password Policy applies to part of the Directory radio button

Specifies that the password policy should apply to only part of the user directory.

When this option is selected, other related controls are activated:

Lookup button

Opens the User Lookup Dialog from where you can select the users within the user directory for which the password policy should apply.

Path and Class information fields

Show the path and class of the namespace you selected in the User Lookup Dialog.

Allow Nested Groups check box

Specifies whether the password policy is associated with nested groups within your LDAP directory.

More information:

[User Directories](#) (see page 141)

[User Lookup Dialog](#) (see page 589)

Password Policy Enabled State Group Box

The Password Policy Enabled State group box is where you enable or disabled the password policy.

Enabled check box

Specifies whether the password policy is enabled.

Redirect Group Box

The Redirect group box is where you define where users should be redirected if they enter a password that is deemed invalid by the password policy.

Redirection URL field

Specifies the URL to which users should be redirected when the password policy determines that an invalid password has been entered in the Redirection URL field.

To enable redirection, you must specify the virtual path to the Password Services CGI or servlet. By default, the Policy Server fills in the path to the Password Services CGI:

`http://myserver.mycompany.org/siteminderagent/pwcgi/
smpwservicescgi.exe`

If you want to use the Password Services servlet, you should specify its path:

`http://myserver.mycompany.org/siteminderagent/pwservlet/PSWDChangeServlet`

If you choose to set up a custom Password Services directory on a non-default web server, be sure to do the following:

Copy the `smpwservicescgi.exe` to a location accessible by the non-default web server.

- The CGI extracts an `agent_path` from the Redirect URL to replace the default `/siteminderagent/pw` path. When you setup a custom Password Services directory, create two virtual directories: `pw` and `pwcgi`. The latter is a CGI directory. For example:

`<custom directory>/pwcgi`

`<custom directory>/pw`

When a user is redirected to the Password Services CGI or servlet, it takes the information from the Policy Server, determines why the password is invalid, and displays a form that provides information or requests additional credentials from the user.

Make sure that the Password Services CGI or servlet is not protected. If SiteMinder is protecting directories above the servlet, create a realm that specifies the following:

- Resource Filter if using CGI: `siteminderagent/pwcgi/smpwservicescgi.exe`
- Resource Filter if using Servlet:
`siteminderagent/pwservlet/PSWDChangeServlet`
- Default Resource Protection: Unprotected

Do not create a policy for this realm.

Note: If a user who is accessing resources through an Agent that is not using an SSL connection must change passwords, the user's new password information will be received over the non-secure connection. To provide a secure change of passwords, set up a password policy that redirects the user over an SSL connection using the Redirection URL field.

Password Policy Dialog—Expiration Tab

The Expiration tab is where you configure events that disable user accounts.

Track Successful Logins check box

Enables and disables tracking of successful logins, including the time of the last login. If you enable this check box, the Policy Server writes the login information to the user directory.

When you enable this check box, the fields in the Password Expires from Inactivity group box are enabled.

Track Failed Logins check box

Enables and disables tracking of failed user login attempts. If you enable this check box, the Policy Server writes the login information to the user directory.

When you enable this check box, the fields in the Incorrect Password group box are enabled.

Authenticate on Login Tracking Failure check box

Enables and disables logins when user tracking fails. By default, this check box is disabled and if user activity cannot be written to the user directory, users are not allowed to login. However, if you enable this check box, users may login, even if password data cannot be written to the user directory.

This check box is disabled if the Track Login Details check box is not enabled.

More information:

[Configure Password Expiration](#) (see page 485)

Password Expires if Not Changed Group Box

The Password Expires if Not Changed group box is where you configure behavior for passwords that have expired and have not been changed by the owner. Optionally, you can specify how far in advance users are warned that their password will expire.

After *number* Days field

Specifies the number of days after a password expires that SiteMinder waits before disabling the user or forcing a password change.

Note: SiteMinder does not disable an account when the number of days specified in the After *number* Days field in the Password Expires from Inactivity group box is reached. A user must attempt to login to the account before SiteMinder disables it.

Disable User radio button

If selected, Specifies that when a user's password expires, SiteMinder should disable that user's account. Disabled users must be enabled using the User Management Dialog (see the *SiteMinder Policy Server Management Guide*).

Force Password Change radio button

If selected, specifies that when a user's password expires, SiteMinder should force a password change when the user next attempts to log in.

Issue expiration warnings for <number> days field

Specifies how many days in advance a user is notified that their password is about to expire.

Note: If you have a password policy enabled and users enter the wrong password in the optional "Change Password" screen, they are redirected back to the login page and do not get an error message since their current session is valid.

More information:

[Password Services Dialog Reference](#) (see page 769)

Incorrect Password Group Box

The Incorrect Password group box is where you can specify how many failed logins are allowed before a user account is disabled. Additionally, you can specify how long the account is disabled before a user can attempt to log in again.

Note: This group box is only available if the Track login details check box is enabled.

Account will be disabled after field

Specifies the number of consecutive failed log in attempts a user can make before the user account is disabled. Limiting the number of unsuccessful attempts protects against programs designed to access a resource by repeatedly trying passwords until the correct one is found. If a user fails to login correctly after the specified number of attempts, the Policy Server disables the user's account. The account must be re-enabled by an administrator.

If you use an HTML forms authentication scheme with the default login.fcc template that ships with the SiteMinder Agent, you should set the smretries parameter in the login.fcc file to 0, so that the password policy determines the number of retries allowed based on the value you enter in this field.

Note: The SMTRYNO cookie may be used during authentication. For example, it can be used to track and display a message that indicates the current number of failed login attempts. If you need the SMTRYNO cookie to be set, then smretries may be set to a value higher than the number of tries allowed before Password Services disablement (for example, @smretries=6 if the number of tries allowed in the Password Policy is 3). This will make sure that a user is disabled instead of redirected to the .unauth page, but it will still allow the SMTRYNO counter to be set.

Important! This value should be set carefully. The Policy Server searches for users by checking all of the user directories bound to a policy. If three users with the same user name exist in separate directories, the Policy Server checks the password against each user name until it finds a match. For each user name the password does not match, the Policy Server records a failed attempt. If the number of failures is set too low, the Policy Server may incorrectly disable an account.

After <number> minutes field

Specifies the length of time (in minutes) that a user must wait before he or she is allowed another login attempt or their account is re-enabled (see below). If the user enters another incorrect password, the Policy Server disables the account again. The user must wait the specified amount of time before trying again.

Allow 1 login attempt radio button

If selected, specifies that when a user enters an incorrect password, they are allowed another login attempt after the specified number of minutes.

Re-enable account radio button

If selected, specifies that when a user enters an incorrect password, their account is re-enabled after the specified number of minutes.

More information:

[HTML Forms Authentication Schemes](#) (see page 244)

Password Expires from Inactivity Group Box

The Password Expires from Inactivity group box is where you specify a period of time between user login attempts after which a user account is considered inactive. You can also use this group box to specify an action should a user whose account is considered inactive successfully log in.

Note: This group box is only available if the Track login details check box is enabled.

After number Days field

Specifies the number of days of inactivity after which a user's password expires.

Disable User radio button

If selected, specifies that when a user's password expires due to inactivity, the Policy Server disables the user. Disabled users must then be enabled using the User Management Dialog (see the *SiteMinder Policy Server Management Guide*).

Force Password Change radio button

If selected, specifies that when a user's password expires due to inactivity, the Policy Server forces a password change when the user next attempts to log in.

Password Policy Dialog—Composition Tab

The Composition tab is where you configure password composition settings.

More information:

[Configure Password Composition](#) (see page 487)

Password Length Group Box

The Password Length group box is where you configure the length of user passwords. Longer passwords are more secure, but can be cumbersome for users to use.

Minimum Length field

Specifies the minimum length for user passwords.

Maximum Length field

Specifies the maximum length for user passwords.

Note: When configuring password length, you should consider values set in the Content Minimums group box. If the number of required characters set in the Content Minimums group box exceeds the maximum length for a password, all passwords will be rejected. For example, if Letters is set to 6 and Digits is set to 6, all passwords must contain at least 12 characters (6 letters and 6 digits). In this example scenario, if a maximum password length of 8 characters is specified, the Policy Server will reject all passwords.

Repeating Characters Group Box

The Repeating Characters group box is where you specify the maximum number of identical characters that can appear consecutively in a password. For example, if this value is set to 3, then “aaaa— can not appear anywhere in the password. However, “aaa— is acceptable within a password. You should set this value to ensure that users cannot enter passwords made up entirely of a single character.

Maximum field

Specifies the maximum number repeating characters that can appear in passwords.

Content Minimums Group Box

The Content Minimums group box is where you configure the types of characters required in passwords by specifying whether each character type should be allowed and, if so, the minimum number of those characters that should be in each password. For example, a policy could require that passwords contain all numbers, disallow any punctuation, at least three letters, or at least two numbers and four letters characters.

Note: When configuring password composition settings, consider the maximum password length when determining values for character requirements. If the total number of letters and/or numbers required exceeds the maximum password length then all passwords will be rejected. For example, if Letters is set to 6 and Digits is set to 6, all passwords must contain at least 12 characters (6 letters and 6 digits). In this example scenario, if a maximum password length of 8 characters is specified, the Policy Server will reject all passwords.

The Content Minimums group box contains several fields, each with an Allow check box and a Minimum entry box. Disallow any of the character types from passwords by deselecting their **Allow** check box. Specify the minimum number of any character type by entering an integer value in its Minimum field. By default all character types are allowed (that is the option is checked), and there is no minimum number of each character type specified.

The character type fields are:

Upper case letters

Specifies whether to allow upper case alphabetic characters and, if so, the minimum number a password must contain.

Lower case letters

Specifies whether to allow lower case alphabetic characters and, if so, the minimum number a password must contain.

Digits

Specifies whether to allow numbers and, if so, the minimum a password must contain.

Letters or Digits

Specifies whether to allow either letters or numbers (upper or lower case, a-Z; 0-9) and if so, the minimum number a password must contain. If this setting is set in conjunction with Letters or Digits, characters can satisfy both requirements. For example, if Digits and Either are set to 4, the password "1234—" is a valid password.

Punctuation

Specifies whether to allow punctuation marks, and if so, the minimum number a password should contain. These can be periods, commas, exclamation marks, slashes, dashes, and hyphens.

Note: The characters that are considered as punctuation characters conform to POSIX standards.

Non-printables

Specifies whether to allow non-printable characters, and if so, the minimum number a password should contain. These characters cannot be displayed on a computer screen.

Important! Use this setting with caution. Certain browsers do not support non-printable characters.

Non-alphanumeric

Specifies whether to allow non-alphanumeric characters such as punctuation marks and other symbols located on the keyboard (“@—”, “\$—”, and “*—”) and if so, the minimum number a password should contain. Non-printable characters are also included. A non-alphanumeric character also satisfies Punctuation and Non-printable character requirements.

Note: The characters that are considered as non-alphanumeric characters conform to POSIX standards.

Password Policy Dialog—Regular Expressions Tab

The Regular Expressions tab is where you configure password regular expression matching.

More information:

[Password Regular Expressions](#) (see page 488)

Passwords MUST Match These Expressions Group Box

The Passwords MUST Match These Expressions Group Box is where you create and manage required regular expressions that a password must match to be acceptable. You can create multiple expressions in which case passwords must match *all* specified expressions to be acceptable.

Example: The following regular expression definition could be used to require that all passwords start with an upper or lower case letter:

Tag:	MustStartAlpha
Regular Expression:	[a-zA-Z].*

The group box contains a clickable list of existing required expressions and controls that allow you to add new expressions or to edit or remove exiting expressions selected from the list.

Add button

Click to add a new expression to the list of required expressions. The Password Regular Expression Dialog opens. In this dialog, enter details of the new regular expression in the fields provided:

Tag

A descriptive tag for the expression (1-32 alphanumeric characters, no white space).

Regular Expression

A regular expression.

Remove button

Click to remove one or more selected entries from the list of required expressions.

Edit button

Click to edit a single entry selected from the list of required expressions. The Password Regular Expression Dialog opens. In this dialog, edit the information about the existing regular expression in the fields provided:

Tag

A descriptive tag for the expression (1-32 alphanumeric characters, no white space).

Regular Expression

A regular expression.

More information:

[Regular Expressions Syntax](#) (see page 488)

Passwords MUST NOT Match These Expressions Group Box

The Passwords MUST NOT Match These Expressions Group Box is where you create and manage forbidden regular expressions that a password must not match to be acceptable. You can create multiple forbidden expressions in which case passwords must match *none* of the specified expressions to be acceptable.

Example: The following regular expression definition could be used to require that no passwords end with a digit:

	NoTrailingDigit
Tag:	
Regular Expression:	.*[0-9]\$

The group box contains a list of existing forbidden expressions and controls that allow you to add new expressions or to edit or remove exiting expressions selected from the list.

Add button

Click to add a new expression to the list of forbidden expressions. The Password Regular Expression Dialog opens. In this dialog, enter details of the new regular expression in the fields provided:

Tag

A descriptive tag for the expression (1-32 alphanumeric characters, no white space).

Regular Expression

A regular expression.

Remove button

Click to remove one or more selected entries from the list of required expressions.

Edit button

Click to edit a single entry selected from the list of forbidden expressions. The Password Regular Expression Dialog opens. In this dialog, edit the information about the existing regular expression in the fields provided:

Tag

A descriptive tag for the expression (1-32 alphanumeric characters, no white space).

Regular Expression

A regular expression.

More information:

[Regular Expressions Syntax](#) (see page 488)

Password Policy Dialog—Restrictions Tab

The Restrictions tab is where you configure password restrictions.

More information:

[Configure Password Restrictions](#) (see page 490)

Reuse Group Box

The Reuse Group Box is where you specify how much time must pass and how many passwords must be used before a user can reuse an old password.

If you specify a length of time and number of passwords, both criteria must be satisfied before a password can be reused. For example, you can configure a password policy which requires users to wait 365 days and specify 12 passwords before reusing a password.

After a year, if only six passwords have been used, another six would have to be used before the user can reuse the first password.

Minimum number of days before reuse field

Specifies how many days a user must wait before reusing a password.

Minimum number of passwords before reuse field

Specifies how many passwords must be used before a password can be reused.

Change Required Group Box

The Change Required group box is where you specify how much a new password must differ from the old.

Percent different from last password field

The percentage of characters a new password must contain that differ from characters in the previous password. If the value is set to 100, the new password may contain no characters that were in the previous password, unless Ignore sequence when checking for differences is set to 0. For examples of how this parameter works with Ignore sequence when checking for differences, see the following table.

Ignore sequence when checking for differences check box

Ignores the position of the characters in the password when determining the percentage.

For example, if a user's initial password is BASEBALL12 and the Ignore sequence when checking for differences check box is selected, a user cannot choose 12BASEBALL as the new password. If the check box is deselected, 12BASEBALL is an acceptable password because each letter occurs in a different position. For examples of how this parameter works with Percent different from last password, see the following table.

For increased security, Ignore sequence when checking for differences check box should be selected.

Passwords	Percent different	Ignore sequence	Accepted
BASEBALL12 (Old)	0	1	Y
12BASEBALL		0	Y
BASEBALL12 (Old)	100	1	N
12BASEBALL		0	Y
BASEBALL12 (Old)	0	1	Y
12SOFTBALL		0	Y
BASEBALL12 (Old)	90	1	N
12SOFTBALL		0	Y
BASEBALL12 (Old)	100	1	N
12SOFTBALL		0	N

Profile Attributes Group Box

The Profile Attributes group box is where you can configure SiteMinder to prevent users from using personal information in their passwords.

Match Length field

Specifies the minimum sequence length the password policy compares to attributes in the user's directory entry. For example, if this value is set to 4, the Policy Server checks to see that the password is not composed of the last four digits of the user's telephone number.

Dictionary Group Box

The Dictionary group box is where you can configure the password policy to check passwords against a user-defined dictionary of forbidden passwords.

The Policy Server dictionary file is a list of strings that cannot be used in passwords. You can include common passwords such as SECRET or words commonly used in the users' industry. For example, a bank may disallow words like *Account*, *Savings*, *Checking*, and *Money*. Strings in the dictionary may not be embedded in a password.

Note: The dictionary file must be located in a directory that all Policy Servers can access.

Note: The last line of the dictionary file used by Password Services must be followed by a carriage return or it will not be included in the dictionary search.

Path field

Specifies the full path and name of the dictionary file.

Match Length

Controls the length of strings compared against values in the dictionary file. The comparison ignores the case (upper/lower) of the strings. If the Match Length field is left blank or set to zero, only passwords that match a string in the dictionary exactly will be rejected. If the match length is greater than zero, passwords entries will be rejected if both of the following are true:

- The password includes a substring which starts with the same series of characters as a dictionary entry.
- The number of consecutive matching characters are greater than or equal to the number specified in the Match Length field.

For example, consider a dictionary file that contains the following:

- lion
- tiger
- bear

If the Match Length field is set to 4, the following will result:

- "TeddyBear" will be rejected because Bear matches the bear entry in the dictionary file.
- "prestige" will be rejected because tige matches the first four characters of the tiger entry in the dictionary file.
- "Geiger Counter" will be accepted since iger does not include the first letter of the tiger entry in the dictionary file.

Password Policy Dialog—Advanced Tab

The Advanced tab is where you configure advanced password policy options.

More information:

[Configure Advanced Password Options](#) (see page 491)

Password Pre-Processing Group Box

The Password Pre-Processing group box is where you configure options that determine how passwords should be processed prior to processing and storage:

Do Not Force Case | Force Upper Case | Force Lower Case radio buttons

Specify whether passwords should be forced to upper or lower case before processing and storage. Choose a case forcing option by clicking the Force Upper Case or Force Lower Case radio button. Otherwise ensure that the Do Not Force Case radio button (the default) is selected.

Important! Be careful to ensure that any case forcing option that you specify is consistent with any case-related composition requirements you have defined.

Remove Leading White Space check box

Select to remove leading white space from passwords before processing.

Remove Trailing White Space check box

Select to remove trailing white space removed from passwords before processing.

Remove Embedded White Space check box

Select to remove all embedded white space before processing.

Note: Some user directory implementations automatically strip leading and/or trailing white space from attribute values (in which user passwords are stored) before storing them regardless of the settings you specify in your password policy.

More information:

[Configure Password Composition](#) (see page 487)

Password Policy Priority Group Box

The Password Policy Priority group box is where you configure options that determine how multiple policies that will apply to the same user directory or namespace are applied:

Evaluation Priority field

Specifies the evaluation priority for the password policy. The value should be in the range 0 (the default) to 999. Applicable policies are evaluated in descending order (999 first; 0 last).

Apply Lower Priority Password Policies check box

Determines whether lower priority password policies are applied after this one.

Tasks Related to the Password Policy Dialog

The following tasks are related to the Password Policy Dialog:

- [Password Services Overview](#) (see page 479)
- [Important Considerations](#) (see page 482)
- [Change the Default Redirect URL](#) (see page 483)
- [Create Password Policies](#) (see page 483)
- [Configure Password Expiration](#) (see page 485)
- [Configure Password Composition](#) (see page 487)
- [Password Regular Expressions](#) (see page 488)
- [Configure Password Restrictions](#) (see page 490)
- [Configure Advanced Password Options](#) (see page 491)
- Password Policy Troubleshooting
- [Regular Expressions Syntax](#) (see page 488)

Chapter 35: SAML 1.x Affiliate Dialog Reference

This section contains the following topics:

[Affiliate Dialog](#) (see page 787)
[Navigate to the Affiliates Dialog](#) (see page 787)
[Affiliate Dialog Fields and Controls](#) (see page 788)
[Affiliate Dialog Prerequisites](#) (see page 790)
[Affiliate Dialog--Users Tab](#) (see page 791)
[Affiliate Dialog--Assertions Tab](#) (see page 791)
[Affiliate Dialog--Session Tab](#) (see page 793)
[Affiliate Dialog--Attributes Tab](#) (see page 793)
[Affiliate Dialog--IP Addresses Tab](#) (see page 797)
[Affiliate Dialog--Time Tab](#) (see page 797)
[Affiliate Dialog--Advanced Tab](#) (see page 798)
[Tasks Related to the Affiliate Dialog](#) (see page 798)

Affiliate Dialog

The Affiliate dialog is where you configure a SAML 1.x consumers.

Navigate to the Affiliates Dialog

You can access the Affiliate Properties dialog or Affiliate dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Affiliate Properties dialog

1. If necessary, in the Object pane, click on the Domains tab.
2. In the Domains tab, click the affiliate domain to which you want to add a consumer.
3. From the menu bar, select:

Edit *affiliate_domain_name*, Create Affiliate

To access the Affiliate Dialog to edit an existing consumer

1. If necessary, click on the Domains tab.
2. Expand the affiliate domain that contains the consumer that you want to edit.
3. Click on the Affiliates entry.
4. In the List pane, click the consumer that you want to edit.
5. From the menu bar, select:
Edit, Properties of Affiliate

Affiliate Dialog Fields and Controls

The following fields and controls are on the Affiliate dialog:

Name

Defines the name of the consumer. This name should be unique across all affiliate domains.

Description

Defines a brief description for a consumer.

Password

Defines the password that a consumer uses to identify itself to the producer site so it can retrieve an assertion.

Confirm Password

Confirms the password entered in the Password field.

Enabled check box

If set, activates the consumer. This check box must be marked for the Policy Server and Federation Web Services to authenticate users trying to access consumer resources.

Use Secure URL

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:
`http(s)://idp_server:port/affwebservices/secure/securedirect`
2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory web_agent_home/affwebservices/WEB-INF.

Allow Notification

If set, SiteMinder provides event notification services for a site where the SAML Affiliate Agent is acting as the consumer. When set, SiteMinder can receive event notifications from the consumer about which affiliate resources a user has accessed.

Note: The Notification service is not supported for consumer at which the SAML credential collector acting as a consumer.

Authentication URL

Defines the protected URL used to authenticate users who do not have a session at the producer.

When a user who has not logged in at the producer requests a protected consumer resource, the user is sent to the Authentication URL. This URL must point to the redirect.jsp file -- for example,

`http://myserver.mysite.com/siteminderagent/redirectjsp/redirect.jsp`

myserver

Identifies the web server with the Web Agent Option Pack or the SPS federation gateway. This redirect.jsp file is included with the Web Agent Option Pack and with the SPS federation gateway.

Important! You must create a policy to protect the AuthenticationURL.

The Affiliates dialog also contains the following tabs:

Users tab

Lists the users or groups of users for the consumer. Users included in a consumer can be authenticated for access to consumer resources, and the assertion generator can create SAML assertions that include attribute information for the users.

Assertions tab

Specifies the audience, validity duration, and skew time of a SAML assertion generated by the SAML assertion generator.

Session tab

Enables the sharing of session information between the producer and the consumer.

Attributes tab

Configures consumer attributes, which pass user attributes, DN attributes, or static data from the Policy Server to the consumer in an assertion.

IP addresses tab

Configures IP addresses, ranges of IP addresses, or subnet masks that users must use in order to access a consumer site. If IP addresses have been specified for a consumer, only users who access the consumer site from the appropriate IP addresses will be accepted by the consumer site as producer users.

Time tab

Sets time restrictions for a consumer. When you add a time restriction, the consumer only functions during the period specified in the time restriction. If a user attempts to access a consumer resource outside of the period specified by the time restriction, the consumer does not produce SAML assertions.

Advanced tab

Optionally, configures a plug-in to customize the content of the SAML assertion generated by the Assertion Generator.

Affiliate Dialog Prerequisites

To successfully create a consumer using the Affiliate Dialog, your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege for the affiliate domain in which the consumer is to be added.

Affiliate Dialog--Users Tab

The Users tab is where you specify the users and groups that can access resources at the consumer. You can only add users and groups from directories that are included in the affiliate domain in which the consumer exists.

After a user is associated with an consumer object, the assertion generator can create SAML assertions for the user. Then the assertion acts as the user's credentials for access to consumer resources.

User Directory

Lists the users and groups that are allowed access resources at the consumer site.

Add/Remove

Opens the Users/Groups dialog from where you can add and remove users and groups.

You may only add users and groups from directories that were included in the affiliate domain.

Affiliate Dialog--Assertions Tab

The Assertions tab is where you define the SAML assertion generated by the Assertion Generator. The dialog contains the following fields:

SAML Profile

Specifies the SAML profile used for sending an assertion. The options are:

Artifact

The artifact is a 42-byte hex encoded ID that references an assertion stored in the session server on the producer-side Policy Server. An artifact lets a consumer retrieve an assertion document from a producer.

POST

The POST profile embeds a SAML response with the assertion in an HTML form. The form is posted by the user's browser at the destination consumer site. SAML POST profile is only supported for SAML version 1.1.

SAML Version

Specifies the version of the SAML protocol in use. The choices are SAML 1.0 and 1.1. SAML artifact can support either version. SAML POST can only support SAML 1.1.

Assertion Consumer URL

(Required for SAML POST binding, optional for SAML Artifact binding)

Specifies the destination site URL to which the user's browser must send a generated assertion. The default URL varies depending on the SAML binding and whether SiteMinder or the SAML Affiliate Agent is at the consumer site.

- SAML 1.x URL when SiteMinder is the affiliate at the consumer site, the URL is typically the SAML credential collector, which is a component of the Federation Web Services application at the consumer.

`https://consumer_server:port/affwebservices/public/samlcc`

- SAML 1.0 URL when the SAML Affiliate Agent is at the consumer site, set this field to the SSLInterceptorURL that is specified for the SAML Affiliate Agent Configuration. Example:

`https://consumer_server:port/affagent/affiliatesite/test1.htm`

consumer_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Note: For SAML 1.x artifact binding, the value of the Assertion Consumer URL field takes precedence over the value of the SMCONSUMERURL query parameter, which must be part of the intersite transfer URL that a user selects to initiate single sign-on.

Audience

Defines the URL of the document that describes the terms and conditions of the agreement between the producer and the consumer. This value is included in the assertion passed to the consumer and may be used for validation purposes. Also, the consumer may parse the actual audience document to obtain relevant information.

If the SAML Affiliate Agent is the consumer, the value entered here must match the value entered for the Assertion Audience setting in the AffiliateConfig.xml file, the configuration file for the SAML Affiliate Agent. For any other SAML consumer, the value entered must match that of the Audience field configured for the SAML authentication scheme.

Validity Duration

Defines the amount of time, in seconds, that the assertion is valid. If the consumer does not receive the assertion during the assertion's valid window, the consumer considers the assertion invalid.

Skew Time

Defines the difference, in seconds, between the system clock time of the producer and the system clock time of the consumer. The skew time is added to the validity duration.

Note: Times are relative to GMT.

Sign Assertion

Enables the producer to digitally sign the assertion with its private key. This adds an additional level of security to the assertion in addition to the assertion response being passed across a secure back-channel.

Signing Alias

(Optional) Specifies the alias associated with a specific private key in the SiteMinder key database. By completing this field, you are indicating which private key the Producer should use to sign assertions or assertion responses.

Note: It is recommended that the private key already be stored in the key database before you specify its associated alias in this field.

Limits: an alphanumeric character string

Affiliate Dialog--Session Tab

The Session tab is where you enable sharing of session information between the producer and the SAML Affiliate Agent consumer.

Note: Sessions only apply when the SAML Affiliate Agent is the consumer.

The controls on the Session tab are as follows:

Shared Sessioning check box

If an consumer implements a shared session solution, this check box enables the sharing of session information between the producer and the SAML Affiliate Agent.

Sync Interval

Defines the frequency (while the user is active at the consumer) at which the SAML Affiliate Agent contacts the producer to validate session status in seconds.

Affiliate Dialog--Attributes Tab

The Attributes tab is where you configure whether attributes user attributes, DN attributes, or static data are passed from the producer to the consumer in an assertion. Attributes can be used by servlets, Web applications, or other custom applications to display customized content or enable other custom features.

When used with Web applications, attributes can be used for fine-grained access control by limiting what a user can do at the consumer. For example, you can send an attribute called Authorized Amount and set it to a maximum dollar amount that the user can spend at the consumer.

Attribute List

Lists the available attributes.

Create

Opens the Affiliate Attribute Editor dialog from where you configure attributes.

More Information:

[Affiliate Attribute Editor Dialog Fields and Controls](#) (see page 795)

Affiliate Attribute Editor Dialog

To configure attributes to include in SAML assertions for federated communication, you configure the specific attribute name/value pairs in the Affiliate Attribute Editor dialog.

Affiliate Attribute Editor Prerequisites

To successfully edit an attribute using the Affiliate Attribute Editor dialog, you must be in the process of creating or editing an affiliate object using the Affiliate dialog.

Navigate to the Affiliate Attribute Editor Dialog

Use one of the following methods to access the Affiliate Attribute Editor Dialog:

To create a new attribute

1. Begin at the Affiliate dialog.
2. Click Create on the Attributes tab.

To edit an existing attribute

1. Begin at the Attributes tab.
2. Select an attribute in the Attribute List
3. Click the Edit button.

Affiliate Attribute Editor Dialog Fields and Controls

Attribute drop down list--Specifies whether the attribute is passed to the consumer as an Affiliate-HTTP-Header-Variable or Affiliate-HTTP-Cookie-Variable response attribute.

Affiliate Attribute Editor Dialog--Attribute Setup Tab

The Attribute Setup tab of the Attribute Editor dialog contains the following fields:

Attribute Kind Group Box

The Attribute Kind group box contains radio buttons that specify the attribute type. The fields on this dialog are as follows:

Static radio button

If selected, the attribute returns the static value defined in the Variable Value (header variable) or Cookie Value (cookie variable) field.

User Attribute radio button

If selected, the attribute returns the information stored in the user attribute specified in the Attribute Name field from a user's entry in a user directory.

DN Attribute radio button

If selected, the attribute returns profile information from a directory object associated with a DN in an LDAP or ODBC user directory.

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Allow Nested Groups check box

If set, nested groups are allowed when selecting the DN Attribute type. Enabled if the DN Attribute radio button is selected.

Attribute Fields Group Box

The Attribute Fields group box contains fields that specify information about the selected attribute type. The fields in this group box are context-sensitive and are determined according to the selection made in the Attribute Kind group box.

The fields are as follows:

Variable Name

Defines the name of the attribute SiteMinder returns to the consumer in a header variable.

Cookie Name

Defines the name of the attribute SiteMinder returns to the consumer in a cookie.

Variable Value

Defines static text for the value of the attribute name/value pair SiteMinder returns to the consumer in a header variable.

Cookie Value

Defines static text for the value of the attribute name/value pair SiteMinder returns to the consumer in a cookie.

Attribute Name

Defines the attribute in a user directory that provides the attribute value.

DN Spec

Defines the distinguished name of the user group from which SiteMinder should retrieve the user attribute. The DN must be related to the users for whom you want to return values to the consumer.

Lookup button

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

Affiliate Attribute Editor Dialog--Advanced Tab

The Advanced tab of the attribute editor contains the following field:

Script

Displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into this field for another response.

Note: If you copy and paste the contents of the Script field for another attribute, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

Affiliate Dialog--IP Addresses Tab

Specifies a single IP address, range of IP addresses, or a subnet mask of users that can access a consumer. If IP addresses are specified for an consumer, only users who access the consumer from the appropriate IP addresses will be accepted by the consumer.

The IP Address tab of the Affiliate dialog contains the following fields:

IP Address list

Lists IP address entries:

IP Address column

Lists IP addresses that can access the consumer. For IP address ranges, lists the first IP address in the range.

Subnet Mask column

Lists subnet masks of users that can access the consumer.

End of Range column

Lists the last IP addresses in ranges of IP addresses of users that can access the consumer. The beginning of the range of IP addresses is indicated by the value in the IP Address column of the same row in the table.

Name column

Lists host names of users that can access the consumer.

Add Button

Opens the Add an IP Address dialog from where you can add IP address entries.

Affiliate Dialog--Time Tab

The Time tab is where you specify time restrictions for generating assertions for a consumer. When you add a time restriction, a user must attempts to access an consumer during the specified time period. If the user attempts to access a resource outside of the period, the producer does not generate a SAML assertion.

Configuring time restrictions for assertions is similar to configuring restrictions for policies.

Note: Time restrictions are based on the system clock of the server on which the SiteMinder Policy Server is installed.

The controls on the Time tab are as follows:

Set button

Opens the Time Dialog from where you can specify time restrictions for the consumer.

Remove button

Removes time restrictions for generating an assertion.

Affiliate Dialog--Advanced Tab

The Advanced tab is where you can optionally configure an Assertion Generator plug-in to customize the content of the SAML assertion generated by the Assertion Generator.

The fields on the Advanced tab of the Affiliate dialog are as follows:

Full Java Class Name

Defines the Java class name of the plug-in that will be invoked by the Assertion Generator at run time.

This plug-in is invoked by the Assertion Generator at run time. The plug-in class can parse and modify the assertion, and then return the result to the Assertion Generator for final processing.

Only one plug-in is allowed for each consumer. For example, `com.mycompany.assertiongenerator.AssertionSample`

Parameters

(Optional) Defines a string of parameters that gets passed to the plug-in as a parameter at run time. The string can contain any value; there is no specific syntax to follow.

Tasks Related to the Affiliate Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 36: SAML 1.x Authentication Scheme Properties

This section contains the following topics:

[Authentication Scheme Dialog Prerequisites](#) (see page 799)

[Navigate to the Authentication Scheme Dialog](#) (see page 799)

[Authentication Scheme Dialog--SAML Artifact Template](#) (see page 800)

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 801)

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 808)

[Authentication Scheme Dialog--SAML POST Template](#) (see page 808)

[Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab](#) (see page 809)

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 814)

[Tasks Related to the SAML 1.x Authentication Scheme Dialog](#) (see page 814)

Authentication Scheme Dialog Prerequisites

To successfully configure a SAML 1.x authentication scheme (artifact or POST):

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Authentication Scheme Dialog

You can access the Authentication Scheme dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog, do the following:

1. Begin at the menu bar.
2. Select Edit, System Configuration, Create Authentication Scheme

The Authentication Scheme Dialog opens.

To access the Authentication Scheme Dialog for an existing authentication scheme:

1. In the object pane, click on Authentication Schemes.
2. In the Authentication Scheme List pane, select an existing authentication scheme.
3. Right click and select Properties of Authentication Scheme.

Authentication Scheme Dialog--SAML Artifact Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML artifact authentication scheme for SAML 1.x. After the scheme is configured, you can assign it to a realm.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

More Information:

[Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab](#) (see page 645)

[Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab](#) (see page 652)

Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x artifact authentication is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML Artifact Template are as follows:

Affiliate Name

Names of the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name of a consumer in the Affiliate domain at the producer site.

Important! For the SAML artifact profile, the producer sends the assertion over a protected back channel to the consumer. If you are using basic authentication to protect the back channel, the value of this field is the name of the consumer. If you are using client certificate authentication for the back channel, the value of this field is the alias of the client certificate stored in the smkeydatabase.

Redirect Mode drop-down list

Indicates the method by which the SAML credential collector redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data (default)

Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects--one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

—`Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

—`Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

PersistAttributes

Redirects the user through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose PersistAttributes and the assertion contains attributes that are left blank, a value of NULL is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

Password

Defines the password that the consumer uses to identify itself to the producer.

This password must match the password entered for the consumer at the producer site.

Verify Password

Confirms the password that you specified in the Password field.

Company Source ID

Defines the source ID of the producer.

The SAML specification standard defines a source ID as a 20-byte binary, hex-encoded number that identifies the producer. The consumer uses this ID to identify an assertion issuer.

Enter a number that matches the Source ID that the producer uses. At the producer, this ID is located in the SAML assertion generator properties file, `AMAssertionGenerator.properties`, which is located in the following directory:

policy_server_home/config/properties

The default Company Source ID value is:

b818452610a0ea431bff69dd346aeff83128b6a

SAML Version

Specifies the version of the SAML specification with which the assertion is generated. The choices are 1.0 or 1.1. SAML 1.1 is the default.

There must be a match between the SAML protocol and assertion versions used by the producer or consumer. For example, if a producer using SAML 1.1 receives a SAML 1.0 request, it returns an error. If a consumer using SAML 1.1 receives a SAML 1.0 assertion embedded into a SAML 1.1 protocol element, it returns an error.

Assertion Retrieval URL

Defines the URL of the service at the producer where the consumer retrieves the SAML assertion. Specifically, this URL identifies the location of the Assertion Retrieval Service, which must be a URL accessed over an SSL connection.

If the assertion retrieval service at the producer is part of a realm using the Basic over SSL authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/assertionretriever`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

If the assertion retrieval service at the producer is part of a realm using the X.509 client certificate authentication scheme, the default URL is:

`https://idp_server:port/affwebservices/certassertionretriever`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Authentication

Specifies the authentication method for the realm at the producer that contains the Assertion Retrieval Service. The value of this field determines the type of credentials that the SAML credential collector at the consumer must provide to access the Assertion Retrieval Service and retrieve the SAML assertion.

The Assertion Retrieval Service retrieves the assertion from the Policy Server at the producer, and then sends it to the consumer across an SSL back channel. We recommend protecting the Assertion Retrieval Service.

Select one of the following:

- Basic Auth if the Assertion Retrieval Service at the producer is part of a realm protected by a Basic over SSL authentication scheme.
- Client Cert if the Assertion Retrieval Service at the producer is part of a realm protected by an X.509 client certificate authentication scheme.

When you select this option, configure access to the Assertion Retrieval Service with a client certificate.

- No Auth if the Assertion Retrieval Service is unprotected.

Depending on the authentication method you select, there can be some additional configuration required.

- When you select Basic Auth, no additional configuration is required at the producer or consumer, unless the certificate of the Certificate Authority that established the SSL connection is not in the consumer's smkeydatabase. If the appropriate certificate is not in the key database, you must import it.
- If you select Client Cert, there are several configuration steps at the consumer and the producer that you must complete to access the Assertion Retrieval Service with the client certificate.

For client certificate authentication, you can use non-FIPS 140 encrypted certificates to secure the back channel even if the Policy Server is operating in FIPS-only mode. However, for FIPS-only installations use certificates only encrypted with FIPS 140-compatible algorithms.

Audience

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.companya.com/SampleAudience`

Issuer

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, `producerA.ca.com`

The consumer accepts assertions from only this issuer. The administrator at the producer determines the issuer.

Note: The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:
`siteminder_home/Config/properties/AMAssertionGenerator.properties`

Search Data XPATH

Defines an Xpath query that tells the authentication scheme where to look in the SAML assertion for data used to look up a user in the user directory. The Xpath query and the Search Specification value map data from a SAML assertion to a user store entry.

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

Select the namespace and click Edit to the right of the search specification scroll box. The SiteMinder Authentication Scheme Namespace Mapping dialog box opens. In this dialog, you map the data collected by the Xpath query to a namespace attribute.

Example:

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Use  
rname/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/  
SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query string extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site.

Using the following abbreviated syntax extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax:

```
"substring-after(//SMprofile/NVpair[1]/text(),"header:uid=")"
```

D-Sig Alias

Specifies the alias of the public key in the smkeydatabase that is used to verify the digital signature which signed the assertion. This alias is associated with the certificate in the smkeydatabase. When you list the certificates in the smkeydatabase, the type of certificate used for verification is a CertificateEntry or KeyEntry certificate.

Namespace

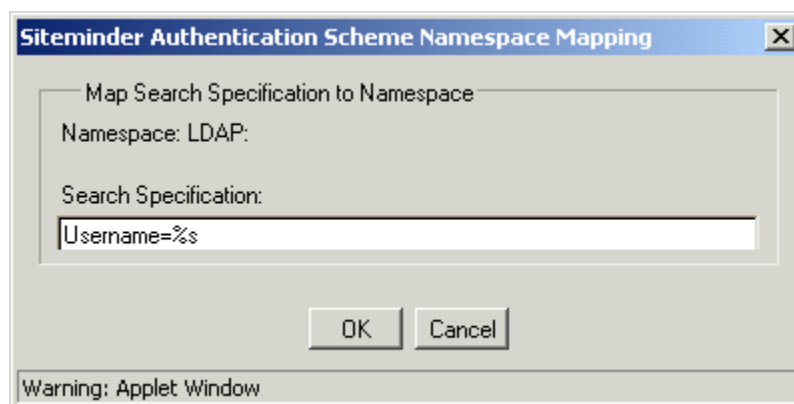
Selectable list of namespace types and defined search specifications.

Search Specification

Enter the attribute that the authentication scheme uses to search a namespace.

Edit button

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.



For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. Compare this string against the user directory to find the correct record for authentication.

Additional Configuration

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for an SAML artifact scheme is where you define information about the shared library that processes SAML 1.x artifact authentication.

The fields on the Advanced tab are as follows:

Library

This field contains the name of the shared library that processes SAML artifact authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for HTML Forms authentication is `smauthsaml`.

Parameter

This field displays the information you entered in the Scheme Setup tab.

Authentication Scheme Dialog--SAML POST Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML POST Template authentication scheme for the SAML 1.x POST profile. After configuring the authentication scheme and assign it to a realm.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

More Information:

[Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab](#) (see page 653)

[Authentication Scheme Dialog--SAML POST Template--Advanced Tab](#) (see page 658)

Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab

The Scheme Setup tab for SAML 1.x POST is where you specify how the consumer communicates with the producer to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields on the Scheme Setup tab of the SAML POST Template are as follows:

Affiliate Name

Names the consumer. Enter an alphabetic string, for example, CompanyA.

The name you enter here must match a name for a consumer in the Affiliate domain at the producer site.

SAML Version

Specifies the SAML version (inactive; the value defaults to 1.1, indicating that POST profile assertions are compliant with SAML version 1.1).

The SAML producer and consumer must be generating and consuming assertions and responses that are the same version.

Audience

Defines the audience for the SAML assertion.

The Audience identifies the location of a document that describes the terms and conditions of the business agreement between the producer and the consumer. The administrator determines the audience at the producer site. It also must match the audience specified for the consumer at the producer site.

The audience value must not exceed 1 K.

To specify the audience, enter a URL. This element is case sensitive. For example:

`http://www.ca.com/SampleAudience`

Redirect Mode

Specifies the method by which the SAML credential collector servlet redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data

(Default). Redirects user through an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

Redirects user through an HTTP 302 redirect with a session cookie and additional cookie data configured at the site that produced the assertion.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The SAML Credential Collector transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:

`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by setting different attribute objects in the request object before redirecting the user to the target application.

The service creates two `java.util.HashMap` objects—one to store all header attributes and one to store all cookie attributes. It uses distinguished attribute names to represent each hashmap object:

- `Netegrity.HeaderAttributeInfo` attribute represents the hashmap that contains header attributes.

- `Netegrity.CookieAttributeInfo` attribute represents the hashmap that contains cookie attributes.

Two other `Java.lang.String` attributes are set by the assertion consuming service to pass the user identity to the custom application:

- `Netegrity.smSessionID` attribute represents the SiteMinder session ID

- `Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the hashmap objects.

PersistAttributes

User is redirected through an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

Assertion Consumer URL

Specifies the URL of the assertion consumer (synonymous with the SAML credential collector). This URL is where the user's browser must POST the generated assertion.

The default URL is:

`http://consumer_server:port/affwebservices/public/samlcc`

consumer_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway. For example:

`http://www.discounts.com:85/affwebservices/public/samlcc`

Issuer

Identifies the producer that issues assertions for the consumer. This element is case sensitive.

For example, producerA.ca.com

The consumer accepts assertions from only this issuer. The administrator determines the issuer at the producer.

Note: The value that you enter for the issuer must match the value of the AssertionIssuerID at the producer site. This value is specified in the AMAssertionGenerator.properties file located in:
policy_server_home/Config/properties/AMAssertionGenerator.properties

Dsig Issuer DN

Specifies the distinguished name of the certificate issuer that signs the SAML POST response. The SAML POST response must be digitally signed by the producer. When the consumer receives the response, it must verify the signature using the data in this parameter and the Serial Number parameter. These two parameters list the issuer of the certificate who signed the response.

The public key used to verify the signature must be in the smkeydatabase. More information about the smkeydatabase can be found in Using Key Databases for Federation Security Services.

Serial Number

Specifies the serial number (a hexadecimal string) of the certificate of the consumer in the smkeydatabase key store. This value is used with the Dsig Issuer DN to locate the certificate for digitally signing the SAML POST response.

Search Data XPATH

Defines an XPATH query that the authentication scheme applies to the SAML assertion. The query tells the authentication scheme where to look in the assertion document. The data obtained by the query is used to look up a user in the user directory.

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```


Example

The following query extracts the text of the Username attribute from the assertion:

```
"/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMlogin/Username/text()"
```

"//Username/text()" extracts the text of first Username element in the SAML assertion using abbreviated syntax.

Other examples:

```
"substring-after(/Assertion/AttributeStatement/Attribute/AttributeValue/SMContent/SMprofile/NVpair[1]/text(),"header:uid=")"
```

This query extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site. The string "substring-after(/SMprofile/NVpair[1]/text(),"header:uid=")" extracts the text of the header attribute named "uid" configured as the first attribute in the Affiliate dialog at the producer site using abbreviated syntax.

Namespace

Selectable List of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

Edit button

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification, which maps data from a SAML assertion to a user store entry, which enables the SAML authentication scheme to locate the correct user record for authentication.

Use %s in the entry as a variable representing the assertion data collected by the Search Data XPATH query.

For example, the XPATH query retrieves the value of user1 for the Username attribute in the SAML assertion. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user directory to find the correct record for authentication.

Additional Configuration

Selecting this button displays the SAML 1.x Auth Scheme Properties dialog for the authentication scheme. From this dialog, you define the message consumer plug-in, redirect URLs for failed authentication, and the target federation resource.

Authentication Scheme Dialog--SAML POST Template--Advanced Tab

The Advanced tab of the Authentication Scheme Dialog for a SAML 1.x POST scheme is where you define information about the shared library that processes SAML 1.x POST authentication.

The fields on the Advanced tab are as follows:

Library

This field contains the name of the shared library that processes SAML POST authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML POST authentication is `smauthsaml`.

Parameter

This field displays the information you entered in the Scheme Setup tab.

Tasks Related to the SAML 1.x Authentication Scheme Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 37: SAML 2.0 Service Provider Reference

This section contains the following topics:

- [SAML Service Provider Dialog](#) (see page 815)
- [Navigate to the SAML Service Provider Dialog](#) (see page 815)
- [SAML Service Provider Dialog Prerequisites](#) (see page 816)
- [SAML Service Provider Properties Dialog Fields and Controls](#) (see page 816)
- [SAML Service Provider Dialog--Users Tab](#) (see page 819)
- [SAML Service Provider Dialog--Name IDs Tab](#) (see page 819)
- [SAML Service Provider Dialog--General Tab](#) (see page 821)
- [SAML Service Provider Dialog--SSO Tab](#) (see page 824)
- [Restrictions Dialog for an SP](#) (see page 827)
- [SAML Service Provider Dialog--Attributes Tab](#) (see page 829)
- [SAML Service Provider Dialog--SLO Tab](#) (see page 832)
- [SAML Service Provider Dialog--IPD Tab](#) (see page 833)
- [SAML Service Provider Dialog--Encryption Tab](#) (see page 834)
- [SAML Service Provider Dialog--Advanced Tab](#) (see page 835)
- [Tasks Related to the SAML Service Provider Attribute Editor Dialog](#) (see page 839)

SAML Service Provider Dialog

If you have purchased SiteMinder Federation Security Services, you can use the SAML Service Provider Dialog to configure a SAML 2.0 Service Provider (SP).

Navigate to the SAML Service Provider Dialog

Use one of the following methods to access the SAML Service Provider dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To create a new SAML Service Provider

1. Begin at the menu bar.
2. Select Edit *affiliate_domain_name*, Create SAML Service Provider.

To access an existing SAML Service Provider

1. In the object pane, click on the Domains tab.
2. In the list pane, right-click the domain object you want to edit.
3. From the pop-up menu, select Properties of Domain.

SAML Service Provider Dialog Prerequisites

The following prerequisites must be met in order to successfully create a policy or affiliate domain using the SAML Service Provider dialog:

- Your SiteMinder administrator account must have the Manage Domain Objects privilege.
- The Affiliate Domain in which the SAML Service Provider is to be configured must preexist.
- The Policy Server Session Server must be configured. The session store is where assertions are stored before they are forwarded to the Federation Web Services application.
- Federation Web Services must be installed on a system.

Federation Web Services can be installed by the Web Agent Option Pack or by the SPS federation gateway.

- The federated Service Provider site must be set up as a SAML 2.0 Service Provider. The SAML assertions generated at the Policy Server must be forwarded to an application at the Service Provider that can receive and interpret the assertions. The SAML Affiliate Agent and the SAML Credential Collector (installed with the Web Agent Option Pack or SPS federation gateway) can both act as SAML consumers.

Note: For information about the SAML Affiliate Agent, see the *SAML Affiliate Agent Guide*.

SAML Service Provider Properties Dialog Fields and Controls

The SAML Service Provider Properties dialog contains the following fields and controls:

Name

Name of the Service Provider. This name should be unique across all affiliate domains.

Description

Optionally, a brief description of the Service Provider.

Authentication URL

Protected URL used to authenticate users trying to access Service Provider resources.

When a user who has not logged in at the Identity Provider requests a protected Service Provider resource, the user is sent to the Authentication URL. This URL must point to the redirect.jsp file, for example,

`http://myserver.idpA.com/siteminderagent/redirectjsp/redirect.jsp`

myserver

Identifies the web server with the Web Agent Option Pack or the SPS federation gateway. The redirect.jsp file is included with the Web Agent Option Pack or SPS federation gateway installed at the Identity Provider site.

Important! You must protect the AuthenticationURL; however, do this after you add a Service Provider to an affiliate domain because you have to exit the SAML Service Provider Properties dialog.

Enabled

Enables the Policy Server and Federation Web Services to support authentication of Service Provider resources.

Use Secure URL

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:
`http(s)://idp_server:port/affwebservices/secure/securedirect`
2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory `web_agent_home/affwebservices/WEB-INF`.

Application URL

(Optional) Identifies the protected URL for a custom web application that is used to supply user attributes to the SiteMinder Single Sign-on service. The application can be on any host in your network.

Attributes from the web application specified in this field are made available to the Assertion Generator and then placed in the SAML assertion by an Assertion Generator plug-in. You must write and integrate the plug-in with SiteMinder.

The Federation Web Services application supplies sample web applications that you can use as a basis for your web application. They are:

`http://idp_server:port/affwebservices/public/sample_application.jsp`

`http://idp_server:port/affwebservices/public/unsolicited_application.jsp`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

The SAML Service Provider dialog also contains the following tabs:

Users tab

Configures the users or groups of users for the Service Provider. Configured users can be authenticated for access to Service Provider resources; the assertion generator can create SAML assertions that include entitlement information for these users.

Name IDs tab

Specifies the Name Identifier to be used when corresponding with this Service Provider.

General tab

Specifies general information about the Service Provider.

SSO tab

Configures the Single Sign-On (SSO) aspects of a Service Provider.

Attributes tab

Configures assertion attribute statements, which define the user attributes, DN attributes, or static data that are passed from the Policy Server to the Service Provider in SAML 2.0 assertions.

SLO tab

Configures the Single Logout (SLO) aspects of the Service Provider.

IPD tab

Configures the Identity Provider Discovery Profile.

Encryption tab

Configures XML encryption.

Advanced tab

(Optional) Configures an Assertion Generator plug-in.

SAML Service Provider Dialog--Users Tab

The Users tab is where you specify the users and groups that can be granted access to resources at the Service Provider. The Assertion Generator at the Identity Provider can generate assertions for these users for authentication purposes.

You can only add users and groups from directories that are included in the affiliate domain in which the Service Provider exists.

The controls on the Users tab are as follows:

Add/Remove

Opens the Users/Groups Dialog, which allows you to add and remove users and groups to the Users/Groups list box.

Users/Groups

Lists the users and groups that are included or excluded by the Policy that governs Assertion production for a particular user directory. The list box contains one tab for each user directory in the affiliate domain.

SAML Service Provider Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name Identifier, which names a user in a unique way in the assertion. The format of the Name Identifier establishes the type of content used for the ID. For example, the format might be the User DN, in which case the content might be a uid.

The Name IDs tab contains the following fields:

Name ID Format

Specifies the Name Identifier format. Pick one of the following:

- Unspecified
- EmailAddress
- X.509 Subject Name
- Windows Domain Qualified Name
- Entity Identifier

- Persistent Identifier
- Transient Identifier

For a description of each format, see Section 8.3 of the *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification (sstc-saml-core-2.0-cd-04.pdf).

Name ID Type Group Box

The Name ID group box contains radio buttons that specify the Name Identifier type. The choices are as follows:

Static radio button

Indicates that the Name Identifier is specified by the value in the Static Value field. Activates the Static Value field; disables other controls.

User Attribute radio button

Indicates that the Name Identifier resides in the user attribute specified in the Attribute Name field. Activates the Attribute Name field; disables other controls.

DN Attribute radio button

Indicates that the Name Identifier will be specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

Allow Nested Groups check box

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

Name ID Fields Group Box

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

Static Value field

Specifies the static text value that will be used for all name identifiers for this Service Provider.

User Attribute

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

DN Spec

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

Lookup button

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

Other Controls**SAML Affiliation**

Specifies a SAML Affiliation for the Service Provider to join. Select from any configured SAML Affiliation object. If an Affiliation is selected, the other controls on the tab are grayed out (the Affiliation settings will be used instead).

SAML Service Provider Dialog--General Tab

The General tab lets you specify general information about the Service Provider.

The settings on this tab are as follows:

SP ID

Specifies a URI that uniquely identifies the Service Provider, such as, sp.example.com.

IdP ID

Specifies a URI that uniquely identifies the Identity Provider, such as idp.ca.com. This URI value becomes the value of the Issuer field in the assertion.

SAML Version

Specifies the SAML version (disabled; the value defaults to 2.0, indicating that assertions sent to this SP ID must be compliant with SAML version 2.0).

Skew Time

Specifies the number of seconds (as a positive integer) added and subtracted from the current clock time to account for Service Providers with clocks that are not synchronized with the Policy Server acting as an Identity Provider. The skew time and the Validity Duration determine how SiteMinder FSS calculates the total time that an assertion is valid.

To determine the assertion validity, the skew time is subtracted from the assertion generation time (IssueInstant) to get the NotBefore time. The skew time is then added to the validity duration and the IssueInstant to get the NotOnOrAfter time. The following equations illustrate how the skew time is used:

- **NotBefore=IssueInstant - Skew Time**
- **NotOnOrAfter=Validity Duration + Skew Time + IssueInstant**

Times are relative to GMT.

For more information about assertion validity, see the *Federation Security Services Guide*.

D-Sig Info Group Box

Enables you to specify digital signature processing information.

Disable Signature Processing

If set, all signature processing for this Service Provider (both signing and verification of signatures) is disabled for the Service Provider.

Note: Signature processing *must* be enabled in a production environment. Select the Disable Signature Processing option only for debugging purposes.

Issuer DN

Specifies the distinguished name of the issuer of the Service Provider certificate used for signature verification of messages coming from that Service Provider. This value is used with the Serial Number to locate the certificate of the Service Provider in the SMKeyDatabase key store.

Important! This field is only enabled when either the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab.

Serial Number

Specifies the serial number (a hexadecimal string) of the Service Provider certificate in the SmKeyDatabase key store. This certificate verifies the signature of messages coming from that Service Provider. This value is used with the Issuer DN to locate the certificate.

Important! This field is only enabled when either the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab.

Signing Options

Displays the Signing Options dialog. From this dialog you can configure the settings for digital signing, such as the signing alias and the signature algorithm.

Other Controls**Configure Backchannel Authentication**

Click this button to implement single sign-on with the HTTP-Artifact binding.

SAML Service Provider Dialog--Backchannel Properties

The Backchannel Properties dialog is where you configure the password for the secure backchannel over which the Identity Provider sends the SAML assertion to the Service Provider.

The fields on this dialog are:

Password

Specifies the password to be used for Service Provider access across the back-channel.

Confirm Password

Confirmation of the entry in the Password field above.

SAML Service Provider Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information for the SAML Service Provider. The fields on this tab are as follows:

Audience

Specifies the URI of the audience the IdP sends in the assertion (to be compared with the audience specified in the authentication scheme on the Service Provider). For example, sp.ca.com.

Assertion Consumer Service

Specifies the URL of the service that receives assertions at the Service Provider. The default for SiteMinder is:

`http://<sp_server:port>/affwebservices/public/saml2assertionconsumer`

sp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Note: This field is disabled if indexed endpoints have been defined. In that case, you can only edit this URL by modifying an indexed entry in the Assertion Consumer Service dialog, accessed by selecting the ellipsis button.

Ellipsis button

Opens the Assertion Consumer Services dialog. From this dialog, you can do the following:

- Add an indexed Assertion Consumer Service entry. You can add multiple entries; there is no limit.
- Edit an existing indexed Assertion Consumer Service entry
- Remove an existing indexed Assertion Consumer Service entry

Bindings Group Box**HTTP-Artifact**

Indicates the artifact binding is supported at the Service Provider for single sign-on.

Artifact Encoding

Specifies the encoding used for the artifact binding: Form or URL. If you select URL, the artifact is added to a URL-encoded query string. If you select Form, the artifact is added to a hidden form control in a form.

Override system generated IdP Source ID

Allows you to specify an IdP Source ID in the associated field. The default is an SHA-1 hash of the IdP ID. Values must be a 40-digit hexadecimal number. To override this generated value, select this checkbox and enter a 40 character hexadecimal value for the IdP Source ID.

Note: If you select HTTP-Artifact, configure the backchannel settings on the General tab.

HTTP-Post

Indicates that the POST binding is supported at Service Provider for single sign-on.

Other Controls**Enhanced Client and Proxy Profile**

Activates support for the SAML 2.0 Enhanced Client and Proxy (ECP) Profile. This profile is used if the SP and IdP are not communicating directly.

Require Signed AuthnRequests

Indicates that AuthnRequest messages sent by the Service Provider must be signed to be accepted. If you select this check box, the Identity Provider cannot send unsolicited responses, securing a trust between the Identity Provider and the Service Provider.

Important! If you enable this feature, you also have to complete the Issuer DN and Serial Number parameters on the General tab for validating the signature of the AuthnRequest.

Signing Options

Displays the Signing Options dialog. From this dialog you can configure the settings for digital signing, such as the signing alias and the signature algorithm.

Authentication Level

Specifies the minimum level at which the user must have authenticated to gain access to a SiteMinder realm. If the user has authenticated at this level or higher, the Identity Provider generates an assertion for the user. If the user is not authenticated at this level or higher, they are redirected to the Authentication URL to authenticate at this level.

Validity Duration

Specifies a number of seconds (a positive integer) for which a generated assertion is valid.

In a test environment, you might want to increase the Validity Duration value above 60, the default, if you see the following message in the Policy Server trace log:

Assertion rejected (_b6717b8c00a5c32838208078738c05ce6237) – current time (Fri Sep 09 17:28:33 EDT 2005) is after SessionNotOnOrAfter time (Fri Sep 09 17:28:20 EDT 2005)

Note: This property applies only to SSO messages -- it is *not* the same as setting specified in the Validity Duration field on the SLO tab.

AuthnContext Class Ref

Defines the URI provided in the AuthnContextClassRef element to describe how the requesting user authenticated. Be sure that you specify a value that is legal based on the SAML specifications and is appropriate for the Authentication Level specified for the Service Provider. We recommend that you accept the default value, which is urn:oasis:names:tc:SAML:2.0:ac:classes:Password

You can specify another value, however, you then must configure a custom SAML response element on the Advanced tab.

Allow Creation of New User Identifier

Permits the Identity Provider to create a value for the NameID in an assertion if the AuthnRequest message received from the Service Provider does not include a NameID but the AllowCreate attribute in the AuthnRequest message is set to true. The newly generated attribute for the NameID is included in the assertion.

Restrictions button

Opens the Restrictions dialog from where you can configure IP address and time restrictions on the assertion generation policy.

More Information:

[Restrictions Dialog for an SP](#) (see page 827)

Assertion Consumer Services Dialog

The Assertion Consumer Services dialog is where you configure an indexed Assertion Consumer Service entry.

The fields on this dialog are:

Index

Specifies the index number for the URL.

Limits: unique integer from 0-65535.

Binding

Specifies the binding you are using for single sign-on for this end point. Choose either HTTP-Artifact or HTTP-POST.

If single sign-on is initiated by an unsolicited request at the Identity Provider, and the link that triggers the request includes the ProtocolBinding query parameter, the binding specified this query parameter overrides the value you choose for this field.

Assertion Consumer Service URL

Specifies the URL for the Assertion Consumer Service. The value must begin with URI must start with http:// or https: //

Restrictions Dialog for an SP

The Restrictions dialog is where you configure IP address and time restrictions on the assertion generation policy for the Service Provider.

Navigate to the Restrictions Dialog

To access the Restrictions dialog in the Policy Server User Interface, click the Restrictions button on the SAML Service Provider dialog SSO tab.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More Information:

[SAML Service Provider Dialog--SSO Tab](#) (see page 824)

Restrictions Dialog Prerequisites

You must be in the process of configuring single sign-on for a Service Provider from the Service Provider Properties dialog.

Restrictions Dialog Fields and Controls

The Restrictions dialog has the following fields and controls

IP Addresses Group Box

IP Addresses

A list of restricted IP addresses configured for the policy for the Service Provider resources. The Policy Server User Interface allows you to specify an IP address, range of IP addresses, or a subnet mask of the Web server on which a user's browser must be running for the user to access a Service Provider.

Add

Opens an empty Add IP Address dialog from where you can create a new IP address restriction.

Edit

Opens the IP restriction entry selected in the IP Addresses list for editing in the Add IP Address dialog.

Remove

Deletes the IP restriction entry selected in the IP Addresses list.

Time Restrictions Group Box

Set Button

Opens the Time dialog so you can configure the availability of the Resource Partner's availability. When you add a time restriction, the Service Provider functions only during the period specified.

Tasks Related to the Restrictions Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

SAML Service Provider Dialog--Attributes Tab

The Attributes tab is where you configure SAML assertion attribute statements. You also indicate whether the attribute will be used for calls to the Attribute Authority or for single sign-on.

The fields on this dialog are as follows:

Attribute

Selectable list of configured attributes.

Create

Opens the Service Provider Attribute Editor dialog from where you can configure Service Provider entitlement attributes.

Edit

Opens the attribute entry selected in the Attribute List for editing in the SAML Service Provider Attribute Editor dialog.

Remove

Deletes the attribute entry selected in the Attribute List.

Allow Nested Groups

Allows nested groups to be searched in an LDAP Directory.

Navigating to the SAML Service Provider Attribute Editor Dialog

To access the SAML Service Provider Attribute Editor dialog in the Policy Server User Interface, click the Create or Edit button on the SAML Service Provider dialog Attributes tab.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

More Information:

[SAML Service Provider Dialog--Attributes Tab](#) (see page 829)

SAML Service Provider Attribute Editor Dialog Prerequisites

You must be in the process of configuring a SAML Service Provider from the SAML Service Provider dialog.

SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab

Attribute field

Specify the format for the attribute that will be part of a SAML assertion. Options are:

- unspecified
- basic
- uri

Refer to the SAML 2.0 specification for definitions of these formats.

Attribute Kind Group Box

The Attribute Kind group box contains radio buttons that allow you to specify the attribute type:

Static

Returns data that remains constant. Use a static attribute to include as part of an assertion. Specify static values in the Variable Name and Variable Value fields.

User Attribute

Returns profile information from a user entry in a user directory. Specify the name for the attribute in the Variable Name field and the user directory attribute in the Attribute Name field.

Note: For the SiteMinder IdP to return values from a user directory, configure the user directory in the SiteMinder User Directory pane.

DN Attribute

Returns profile information from a directory object in a user directory. Groups and Organizational Units (OUs) that are part of a user DN are examples of directory objects whose attributes can be treated as DN attributes. For example, you can use a DN attribute to return a company division for a user, based on the user's membership in a division.

Activates the Variable Name, DN Spec, and Attribute Name, which you configure to add the attribute to the assertion.

Selecting a DN Attribute also activates the Allow Nested Groups check box. Allowing nested groups lets SiteMinder return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Note: For the Identity Provider to return an assertion containing DN attributes, the user directories must be configured in the SiteMinder User Directory pane.

Allow Nested Groups

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute is selected.

Note: If the attribute is configured to be encrypted, the Encryption Certificate settings in the Encryption tab will be enabled.

Encrypted

Instructs SiteMinder to encrypt the DN attribute.

Retrieval Method

Specifies the intended use of the attribute.

Limits:

- SSO

Indicates that the attribute is used for single sign-on.

- Attribute Service

Indicates the attribute is for use by the Attribute Authority to complete requests from an attribute query.

Attribute Fields

Complete the fields for the Attribute Kind that you selected. Different fields become available depending on the attribute type (static, user attribute, DN attribute).

Variable Name

Designates the name for the attribute that SiteMinder returns in the assertion. Complete this field for any attribute type.

Variable Value

(Static only) Defines the static text as the value for the name/value pair. Enter a value.

Attribute Name

(User and DN Attributes only) Specifies the user directory attribute SiteMinder uses for the name/value pair.

DN Spec

(DN Attribute only) Specifies the DN of the user or user group and the name of the user attribute in the Attribute Name field.

SAML Service Provider Attribute Editor Dialog--Advanced Tab

The Advanced tab of the Service Provider Attribute Editor dialog contains the following field:

Script

Displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into the Script field for another response.

Note: If you copy and paste the contents of the Script field for another entitlement, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

SAML Service Provider Dialog--SLO Tab

The SLO tab is where you configure Single Logout (SLO) properties of the SAML Service Provider.

HTTP-Redirect

Specifies whether the IdP-initiated Single Logout Profile over HTTP is supported at the Service Provider.

Request Expiration Group Box

Validity Duration

Specifies the number of seconds for which a SLO request is valid.

Note: This property applies only to SLO messages -- it is *not* the same as that specified in the Validity Duration field on the SSO tab.

URLs Group Box

SLO Location URL

(Required) Specifies the URL of the single logout service at the Service Provider. The default URL is:

`http://sp_server:port/affwebservices/public/saml2slo`

sp_server:port

Specifies the server and port number at the Service Provider that is hosting the Web Agent Option Pack or the SPS federation gateway.

SLO Response Location URL

(Optional) Specifies the URL of the single logout service at the Service Provider. The purpose of having a Response Location URL is for a configuration where there is one service for single logout requests and one service for single logout responses.

For SiteMinder, this value is always the same as the SLO Location URL:

`http://sp_fws_server:port/affwebservices/public/saml2slo`

where *sp_fws_server:port* is the server at the Service Provider where the Federation Web Services application is installed.

For third-party vendors, the URL represents the service handling single logout responses.

SLO Confirm URL

Specifies the URL that the Identity Provider or Service Provider redirects the user when the single logout request is complete. This value needs to be a local resource and not a resource in a federated partner's domain. For example, if the local domain is acme.com, the SLO confirm page cannot be in the example.com domain.

Relay State Overrides SLO Confirm URL

(Optional) Replaces the URL in the SLO Confirm URL field with the value of the Relay State query parameter included with the single logout request to the SLO service. This check box gives you more control over the single logout confirmation target because using the Relay State query parameter lets you dynamically define the confirmation URL for SLO requests.

SAML Service Provider Dialog--IPD Tab

The IPD tab is where you configure the Identity Provider Discovery profile (IPD), which enables the Service Provider to determine which Identity Provider the principals are using with the Web browser single sign-on profile.

The fields on this dialog are as follows:

Enable

Indicates that the Identity Provider Discovery profile is enabled.

Service URL

Specifies the URL of the Identity Provider Discovery Profile servlet. The default URL is:

`https://idp_server:port/affwebservices/public/saml2ipd/*`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

Common Domain

Specifies the common cookie domain (must be a subset of the host in the Service URL).

Persistent Cookie

Indicates that the cookie should be persistent.

SAML Service Provider Dialog--Encryption Tab

The Encryption tab is where you configure XML encryption for a SAML assertion. If you enable encryption, all data in the assertion is encrypted, including all attribute statements.

Note: If you do not want to encrypt the entire assertion but want to encrypt only individual attribute statements, go to the Attributes tab, select or create an attribute, and check the Encrypted check box for the individual attribute.

The fields on the encryption tab are as follows:

Encrypt Name ID

Specifies the Name ID in the assertion is encrypted.

Encrypt Assertion

Enables encryption of the assertion.

Encryption Block Algorithm

Specifies the block algorithm to be used for encryption. Choose one of the following:

- triledes (the default)
- aes-128
- aes-256

Encryption Key Algorithm

Specifies the key algorithm to be used for encryption. Choose one:

- rsa-v15 (the default)
- rsa-oaep

Note: The minimum key size required to use the rsa-oaep encryption algorithm is a 1024 bits.

Encryption Public Key Certificate Group Box

This group box is where you specify the location of the public certificate of the Service Provider.

Note: You must complete both fields in this group box if either the Encrypt Name ID or Encrypt Assertion option is set or any SAML assertion attribute needs encryption.

Issuer DN

Specifies (along with the Serial Number) the Issuer DN of the certificate of the Service Provider in the key store.

Serial Number

Specifies (along with the Issuer DN) the serial number of the certificate of the Service Provider in the key store.

SAML Service Provider Dialog--Advanced Tab

The Advanced tab is where you optionally configure a custom-developed Assertion Generator plugin developed using the Assertion Generator Plugin API.

Assertion Generator Plugin Group Box**Full Java Class Name**

Specifies the fully qualified Java class name of the Assertion Generator Plugin.

Parameters

If a value is entered in the Full Java Class Name field, specifies a string of parameters to be passed to the specified plugin.

Proxy Group Box

Server

If your network has a proxy server between the client and the system where Federation Web Services is running (that is, the system with the Web Agent Option Pack or the SPS federation gateway) specify the scheme and authority portions of the URL, such as *protocol:authority*. The scheme is http: or https: and the authority is *//host.domain.com* or *//host.domain.com:port*. For example, *http://example.ca.com*.

Status Redirect URLs and Modes Group box

Customize Validity duration

Displays the Customize Validity duration dialog which lets you define the value of the SessionNotOnOrAfter parameter in the assertion.

Additional URL Configuration

Allows you to specify additional redirect URLs for HTTP status errors. The user can be redirected to a custom error page for further action.

Click this button to open the Additional URL Configuration dialog, which is described in the next topic.

SAML 2.0--Additional URL Configuration Dialog

The Additional URL Configuration dialog is where you configure redirect URLs to handle HTTP 500, 400, 405, and 403 error conditions. If any of these errors occur, redirect URLs can send the user to an application or a customized error page for further action.

Redirection to these customized URLs can take place only when enough information about the Service Provider is configured at the Identity Provider. For example, if during a request there is an issue in retrieving the certificate information from smkeydatabase, the user is redirected to Server Error URL specified. However, if a request contains an invalid SP ID, no redirection happens and the HTTP error code 400 is returned to the browser.

The Additional URL Configuration dialog lets you configure redirect URLs for error handling. This dialog includes the following settings:

Enable Server Error URL

Specifies the URL where the user is redirected when an HTTP 500 Server error occurs. A user can encounter a 500 error because an unexpected condition prevents the web server from fulfilling the client request. If this type of error occurs, the user is sent to the specified URL for further processing.

Example: *http://www.redirectmachine.com/error_pages/server_error.html*

Enable Invalid Request URL

Specifies the URL where the user is redirected when an HTTP 400 Bad Request or a 405 Method Not Allowed error occurs. A user can encounter a 400 error because a request is malformed or a 405 error because the web server does not allow a particular method or action to be performed. If these types of errors occur, the user is sent to the specified URL for further processing.

Example: http://www.redirectmachine.com/error_pages/invalidreq_error.html

Enable Unauthorized Access URL

Specifies the URL where the user is redirected when an HTTP 403 Forbidden error occurs. A user can encounter a 403 error because the URL in a request is pointing to the wrong target, such as a directory instead of a file. If this error occurs, the user is sent to the specified URL for further processing.

Example:

http://www.redirectmachine.com/error_pages/unauthorized_error.html

Mode Fields

For each URL, you can select a mode by which the browser redirects the user.

302 No Data

Redirects the user using an HTTP 302 redirect. When the user is redirected to the specified custom error page URL, they are redirected only with a session cookie and no information appended to the redirect URL.

HTTP POST

Redirects the user using the HTTP POST protocol. When the user is redirected to the custom error page URL, the following data, if available, is appended to the redirect URL and then posted to custom error page:

- TARGET
- AUTHREASON
- CONSUMERURL
- SAMLRESPONSE
- IDPID
- SPID
- PROTOCOLBINDING

Customize Session Validity Duration

The settings in the Customize Validity Duration dialog determine whether the IdP adds the SessionNotOnOrAfter attribute to a SAML assertion. If the attribute is in the assertion, the dialog lets you determine the duration of the session between the user and the IdP.

This configuration setting is available only at the SiteMinder IdP. It only instructs the IdP what value it should set for the SessionNotOnOrAfter parameter in the assertion. The setting does not set any timeout value at the SP.

Important! If SiteMinder is acting as an SP, it ignores the SessionNotOnOrAfter value. Instead, a SiteMinder SP sets session timeouts based on the realm timeout that corresponds to the configured SAML authentication scheme that protects the target resource.

The options are:

SP Session Validity Duration

Determines the use and duration of the SessionNotOnOrAfter attribute in an assertion.

Options:

Use Assertion Validity

Calculates the SessionNotOnOrAfter value based on the assertion validity duration.

Omit

Instructs the IdP not to include the SessionNotOnOrAfter parameter in the assertion.

IDP Session

Calculates the SessionNotOnOrAfter value based on the IdP session timeout. The timeout is configured in the IdP realm for the authentication URL . Using this option can synchronize the IdP and SP session timeout values.

Custom

Lets you specify a custom value for the SessionNotOnOrAfter parameter in the assertion. If you select this option, enter a time in the SAML_SP_CUSTOM_TIME_OUT property.

Custom Assertion Session Duration (HH:MM)

Specifies the amount of time set for the SessionNotOnOrAfter parameter in the assertion. This setting lets you designate any amount of time that suits your environment.

Tasks Related to the SAML Service Provider Attribute Editor Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 38: SAML 2.0 Authentication Scheme Properties Reference

This section contains the following topics:

[Authentication Scheme Dialog -- SAML 2.0 Template](#) (see page 841)
[SAML 2.0 Authentication Properties--Scheme Setup Tab](#) (see page 842)
[SAML 2.0 Authentication Properties--Advanced Tab](#) (see page 844)
[Navigate to the SAML 2.0 Auth Scheme Properties Dialog](#) (see page 844)
[SAML 2.0 Auth Scheme Properties Dialog](#) (see page 844)
[SAML 2.0 Auth Scheme Properties Dialog Prerequisites](#) (see page 844)
[SAML 2.0 Auth Scheme Properties Dialog Fields and Controls](#) (see page 845)
[SAML 2.0 Auth Scheme Properties Dialog--Users Tab](#) (see page 846)
[SAML 2.0 Auth Scheme Properties Dialog--SSO Tab](#) (see page 847)
[SAML 2.0 Auth Scheme Properties Dialog--SLO Tab](#) (see page 852)
[SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab](#) (see page 853)
[SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab](#) (see page 854)
[SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab](#) (see page 854)
[SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab](#) (see page 854)
[SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab](#) (see page 856)
[SAML 2.0--Additional URL Configuration](#) (see page 858)
[Tasks Related to the SAML 2.0 Auth Scheme Properties Dialog](#) (see page 859)

Authentication Scheme Dialog -- SAML 2.0 Template

If you have purchased SiteMinder Federation Security Services, you can configure the SAML 2.0 Template authentication scheme.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

SAML 2.0 Authentication Properties--Scheme Setup Tab

The Scheme Setup tab for SAML 2.0 authentication lets you specify how the Service Provider communicates with the Identity Provider to retrieve the assertion.

The fields on this tab are:

SP ID

Specifies a URI that uniquely identifies the Service Provider.

Note: The value that you enter must match the value of the ID specified for the corresponding Service Provider object that you configure at the Identity Provider.

IdP ID

Specifies a URI that uniquely identifies the Identity Provider from which assertions for this Service Provider are issued.

The Service Provider accepts assertions from only this IdP.

Note: The value that you enter for the issuer must match the value of the IdP ID configured at the Identity Provider site.

SAML Version

Specifies the SAML version (disabled; the value defaults to 2.0, indicating that assertions sent to this IdP ID must be compliant with SAML version 2.0).

D-Sig Info Group Box

Contains fields and controls that allow you to specify digital signature information:

Disable Signature Processing

Disables all signature processing, that is, signing and verification of signatures, for this Service Provider.

Note: Signature processing *must* be enabled in a production environment. Select the Disable Signature Processing option only for debugging purposes.

Signing Options

Displays a dialog with the settings for digital signature configuration, specifically the Signing Alias and the Signature Algorithm.

Issuer DN

Specifies the distinguished name of the issuer of the certificate. This value is used with the Serial Number to locate the certificate of the Service Provider in the SMKeyDatabase file.

Note: This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

Serial Number

Specifies the serial number (a hexadecimal string) of the certificate of the Service Provider in the key store. This value is used with the IssuerDN to locate the certificate in the SMKeyDatabase key store.

Note: This field is enabled only if the HTTP Post option is set on the SSO tab or the HTTP Redirect Binding option is set on the SLO tab. If signature processing is disabled, this field is inactive.

Additional Configuration

Opens the SAML 2.0 Auth Scheme Properties dialog. From this dialog, you can specify additional configuration details for the authentication scheme.

SAML 2.0 Authentication Properties--Advanced Tab

The Advanced tab of the Authentication Scheme Properties dialog for SAML 2.0 is where you define information about the shared library that processes SAML 2.0 authentication.

Library

Contains the name of the shared library that processes SAML 2.0 authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for SAML 2.0 authentication is `smauthsaml`.

Parameter

Displays the information you entered in the Scheme Setup tab.

Navigate to the SAML 2.0 Auth Scheme Properties Dialog

To access the SAML 2.0 Auth Scheme Properties dialog in the Policy Server User Interface, click the Additional Configuration button on the Authentication Scheme Dialog--SAML 2.0 Template--Scheme Setup Tab.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

SAML 2.0 Auth Scheme Properties Dialog

The SiteMinder SAML 2.0 Auth Scheme Properties dialog is where you specify configuration details for the SAML 2.0 Template authentication scheme.

SAML 2.0 Auth Scheme Properties Dialog Prerequisites

You must be in the process of configuring a SAML 2.0 Template Authentication scheme from the Authentication Scheme dialog.

SAML 2.0 Auth Scheme Properties Dialog Fields and Controls

The SAML 2.0 Auth. Scheme Properties dialog contains the following tabs:

User

Defines how to collect user information from information in a SAML 2.0 assertion.

SSO

Specifies the Single Sign-On (SSO) bindings supported by the Service Provider.

SLO

Determines the Single Logout (SLO) properties for the SAML Service Provider.

Encryption

Defines the XML encryption configuration for SAML assertions sent to this Service Provider.

Attributes

Specifies the attributes that the SAML Requester can request from an Attribute Authority.

NameIDs

Specifies the Name ID that will be in an attribute query sent from the SAML requester (SP) to the Attribute Authority (IdP).

Backchannel

Defines the configuration of the secure back channel between the Service Provider and the Identity Provider.

Advanced

Specifies advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors.

SAML 2.0 Auth Scheme Properties Dialog--Users Tab

The Users tab is where you configure how to obtain the user information from a SAML 2.0 assertion and use this information to authenticate a user.

User Disambiguation Group Box

Xpath Query

Specifies an XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

Example

To obtain the attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/  
AttributeValue/text()
```

Namespace

Displays a selectable list of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

Edit

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user store to find the correct record for authentication.

SAML Affiliation

(Optional) specifies a SAML Affiliation for the Identity Provider to join. Select from any configured SAML Affiliation object. If an Affiliation is selected, the remaining controls are dimmed and the Affiliation settings are used instead.

User Disambiguation Group Box

The Authentication Scheme Namespace Mapping dialog box is where you specify the attribute that the authentication scheme uses to search a namespace.

Search Specification

Specifies the attribute that the SAML 2.0 authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the LoginID.

SAML 2.0 Auth Scheme Properties Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information.

Redirect Mode

Specifies the method by which the Service Provider redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data (default)

User is redirected via an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

User is redirected via an HTTP 302 redirect with a session cookie and additional cookie data configured for the Service Provider at the Identity Provider.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The service that collects the credentials (SAML 2.0 Assertion Consumer Service or WS-Federation Security Token Consumer Service) transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application--that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by an attribute object in the request before redirecting the user to the target application. The service that consumes assertions sends the attributes by creating a `java.util.HashMap` object. The attribute that contains the `HashMap` of SAML attributes is "Netegrity.AttributeInfo."

Two other `Java.lang.String` attributes are set by the service that consumes assertions to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the `hashmap` objects.

PersistAttributes

User is redirected via an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

SSO Service

Specifies the URI of the Single Sign-On service at an Identity Provider. This is the location where the AuthnRequest service redirects an authnrequest message, which contains the Service Provider's ID. The default URL is:

`http://idp_host:port/affwebservices/public/saml2sso`

Audience

Specifies the audience for the SAML assertion. The Audience is a URL that identifies the location of a document that describes the terms and conditions of the business agreement between the Identity Provider and the Service Provider. The audience is determined by the administrator at the Identity Provider site. It also must match the audience specified for the Service Provider at the Identity Provider site.

The audience value should not exceed 1K and is case-sensitive. For example:

`http://www.ca.com/SampleAudience`

Target

Specifies the target resource URI at the destination Service Provider site.

Relay State Overrides Target

(Optional) Replaces the value specified in the Target field with the value of the Relay State query parameter for SP-initiated or IdP-initiated single sign-on. This check box gives you more control over the target because using the Relay State query parameter lets you dynamically define the target.

Bindings Group Box**HTTP-Artifact**

Enables the artifact binding (when enabled, the following associated controls are activated).

Override system generated IdP Source ID

Allows you to specify an IdP Source ID in the associated field. The default is an SHA-1 hash of the IdP ID. Values must be a 40-digit hexadecimal number.

Resolution Service

Specifies the URL of the Identity Provider's Artifact Resolution Service. The default URL is:

`http://host:port/affwebservices/saml2artifactresolution`

Sign ArtifactResolve

Indicates that the artifact resolve message, the request sent by the Service Provider to retrieve the original SAML message, must be signed. If you check this box, the artifact resolve message must be signed or the Identity Provider will not accept it.

If you select this check box, the Identity Provider must be configured to require a signed artifact resolve message.

Note: Digital signature processing must be enabled to sign the artifact resolve message.

Require Signed ArtifactResponse

Specifies that the Service Provider will only accept the artifact response, which contains the original SAML message, if the response is signed.

If you select this check box, the Identity Provider must be configured to sign the artifact response.

Note: Digital signature processing must be enabled to process the signed response.

Authentication

Specifies the type of authentication that protects the realm that contains the Artifact Resolution Service. The authentication scheme determines the type of credentials the Service Provider must present to access the Artifact Resolution Service to retrieve the assertion.

Choose one:

Basic

(Default) Uses the SP ID and the password specified in the SSO tab of the authentication scheme properties for credentials. No additional configuration is required, unless the connection to the Artifact Resolution Service is an SSL connection. Then, the certificate of the Certificate Authority who enabled the SSL connection must be in the Service Provider's AM.keystore.

Client Cert

Uses the SP ID and password specified in the SSO tab of the authentication scheme properties for credentials to look up the certificate in the key store. If you select this option, you have to configure access to the Artifact Resolution Service using a client certificate.

No Auth

Indicates that no authentication is required.

Index

Enabled upon selecting HTTP-Artifact check box, this field assigns an AssertionConsumerServiceIndex parameter for the artifact binding. If you have multiple endpoints in a federated network, you can assign an index for the Assertion Consumer Service so the Identity Provider knows where to send the response. Enter an integer in the range of 0-65535.

Http-Post

Indicates that the POST binding is enabled for the Identity Provider.

Enforce Single Use Policy check box

Enforces the single use policy is enforced, preventing SAML 2.0 assertions that arrive via POST binding from being re-used at a Service Provider to establish a second session.

Index

Enabled upon selecting HTTP-Post check box, this field assigns an AssertionConsumerServiceIndex parameter for the artifact binding. If you have multiple endpoints in a federated network, you can assign an index for the Assertion Consumer Service so the Identity Provider knows where to send the response. Enter an integer in the range of 0-65535.

Other Controls**Enhanced Client and Proxy Profile**

Enables processing of requests using the SAML 2.0 Enhanced Client and Proxy (ECP) Profile.

Sign AuthRequests

Enables the signature of AuthnRequest messages (which request authentication from an Identity Provider).

Allow IdP to Create New User Identifier

If the Service Provider sends an AuthnRequest message to the Identity Provider to get an assertion, checking this box sets the AllowCreate attribute in the AuthnRequest message to true. The AllowCreate attribute instructs the Identity Provider to generate a new value for the NameID, provided that the AllowCreate feature is enabled at the Identity Provider. This new value for the NameID is included in the assertion sent back to the Service Provider.

SAML 2.0 Auth Scheme Properties Dialog--SLO Tab

The SLO tab is where you configure Single Logout (SLO) properties for the SAML Service Provider being protected by the authentication scheme.

Bindings Group Box

HTTP-Redirect

Specifies whether the IdP-initiated Single Logout Profile over HTTP is supported at the Service Provider.

Request Expiration Group Box

Validity Duration

Specifies the number of seconds for which a SLO request is valid.

Note: This property is different from that specified in the Validity Duration field on the SSO tab).

Other Controls

SLO Location URL

Required. Specifies the URL of the single logout service at the Identity Provider. The default URL is:

`http://idp_server:port/affwebservices/public/saml2slo`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

SLO Response Location URL

Optional. Specifies the URL of the single logout service at the Identity Provider. The purpose of having a Response Location URL is for a configuration where there is one service for single logout requests and one service for single logout responses.

For SiteMinder, this value is always the same as the SLO Location URL:

`http://idp_fws_server:port/affwebservices/public/saml2slo`

idp_server:port

Identifies the web server and port hosting the Web Agent Option Pack or SPS federation gateway.

For third-party vendors, the URL will represent the service handling single logout responses.

SLO Confirm URL

Specifies the URL that the Identity Provider or Service Provider redirects the user when the single logout request is complete. This value needs to be a local resource and cannot be a resource in a federated partner's domain. For example, if the local domain is ca.com, the SLO confirm page cannot be in the example.com domain.

Relay State Overrides SLO Confirm URL

(Optional) Replaces the URL in the SLO Confirm URL field with the value of the Relay State query parameter included with the single logout request to the SLO service. This check box gives you more control over the single logout confirmation target because using the Relay State query parameter lets you dynamically define the confirmation URL for SLO requests.

SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab

The Encryption tab is where you configure XML Encryption configuration for SAML assertions sent to this Service Provider.

Encryption Restrictions Group Box**Require Encrypted Name ID**

Indicates that the Name ID in the assertion must be encrypted. If the Name ID is not encrypted, the assertion will not be accepted.

Require Encrypted Assertion

Indicates that the entire assertion must be encrypted. If the assertion is not encrypted, it will not be accepted.

Decryption Private Key Group Box**Alias**

Specifies an alias for the private key in the smkeydatabase used to decrypt encrypted data in the assertion received from the Identity Provider.

SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab

In the Attributes tab, you configure the attributes that the SAML requester can request from an Attribute Authority. These are the attributes that the SAML requester can include in the attribute query and are then returned by the Attribute Authority in its response.

A description of the contents of this tab can be found in the [SAML 2.0 Attribute Query Reference](#) (see page 891).

SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name ID to be included in the attribute queries sent from the SAML requester (SP) to the Attribute Authority (IdP).

This tab is only configurable if the Attribute Query feature is enabled, that is, the Enabled checkbox on the Attributes tab is checked.

A description of the contents of this tab can be found in [Name IDs Dialog Fields and Controls for Attribute Queries](#) (see page 894).

More Information:

[Name IDs Dialog Fields for Attribute Queries](#) (see page 894)

SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab

The Backchannel tab defines the configuration of the secure back channel for two functions:

- Single Sign-on between a Service Provider and Identity Provider
- Attribute Queries and Responses between a SAML Requester and an Attribute Authority

The following required fields serve the same function for both purposes, as follows:

Authentication

Specifies the authentication method used across the back channel. Options are:

Basic (over SSL)

Indicates that the Single Sign-on or Attribute Service is part of a realm protected by a Basic over SSL authentication scheme.

If you select this option, no additional configuration other than the remaining required fields in this dialog, is required unless you want to use Basic over SSL. If you select Basic over SSL, you must ensure that the certificate of the Certificate Authority that enabled the SSL connection is in the smkeydatabase. If it is not, import the certificate into the smkeydatabase.

Client Cert

Indicates that the Single Sign-on or Attribute Service is part of a realm protected by an X.509 client certificate authentication scheme.

For single sign-on, if you select this option, you must also configure access to the Assertion Retrieval Service with a client certificate.

You can use non-FIPS 140 encrypted certificates to secure the backchannel even if the Policy Server is operating in FIPS-only mode. However, for FIPS-only installations use certificates only encrypted with FIPS 140-compatible algorithms.

NoAuth

Indicates that the Single Sign-on or Attribute Service is not protected. If you select this option, no authentication is required.

SP Name

Identifies the Service Provider object. This name must match a Service Provider or SAML Requester name specified at the Identity Provider or Attribute Authority.

If you are using basic authentication as the authentication scheme for the backchannel, the value of this field is the name of the Service Provider. If you are using client certificate authentication for the backchannel, the value of this field should be the alias of the client certificate stored in the smkeydatabase.

Password

Specifies the password used by the Identity Provider or Attribute Authority to access the Service Provider or SAML Requester through the back-channel. Enter a valid string between 3 and 255 characters.

Confirm Password

Confirms the entry in the Password field.

SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab

The Advanced tab is where you specify advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors during authentication.

The tab contains the following fields and controls:

Message Consumer Plugin Group Box

Full Java Class Name

(Optional) Specifies the fully qualified Java class name of a class which implements a Message Consumer Plug-in interface for the authentication scheme.

Parameter

If a value is entered in the Full Java Class Name field, this field specifies a string of parameters the API passes to the specified plug-in.

Status Redirect URLs and Modes Group Box

Assertion-based authentication can fail at the site that consumes assertions for various reasons. If authentication does fail, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, SiteMinder can be configured to redirect the user to a provisioning system, which could create a user account based on the information found in SAML assertion.

The following options redirect the user to a configured URL based on the condition that caused the failure.

Redirect URL for the User Not Found Status

(Optional) Identifies the URL where SiteMinder redirects the user when the user is not found. The user not found status applies when the single sign-on message has no LoginID or the user directory does not contain the LoginID.

Redirect URL for the Invalid SSO Message Status

(Optional) Identifies the URL where SiteMinder redirects the user if one of the following conditions occur:

- The single sign-on message is not well formed, according to the XML specification
- The single sign-on message is invalid based on rules specified by the SAML schemas
- The consumer requires an encrypted assertion but the single sign-on message does not contain an encrypted assertion.

Redirect URL for the Unaccepted User Credential (SSO Message) Status

(Optional) Identifies the URL where SiteMinder redirects the user for all other error conditions other than when a user is not found or the single sign-on message is invalid. The assertion is valid, but SiteMinder does not accept the message for certain reasons, such as:

- XML digital signature validation fails
- XML decryption operation fails
- XML validation of conditions fails, such as an expired message or an audience mismatch
- None of the assertions in SSO message contain an authentication statement.

Mode

Specifies the method by which SiteMinder redirects the user to the redirect URL. The options are:

302 No Data (default)

Redirects user with an HTTP 302 redirect with a session cookie, but no other data.

Http Post

Redirects user using HTTP Post protocol.

Additional URL Configuration

Allows you to specify additional redirect URLs for HTTP status errors. The user can be redirected to a custom error page for further action.

Click this button to open the Additional URL Configuration dialog, which is described in the next topic.

Proxy Group Box**Server**

If your network has a proxy server between the client and the system where Federation Web Services is running (that is, the system where Web Agent Option Pack is installed) specify the scheme and authority portions of the URL, such as *protocol:authority*. The scheme is http: or https: and the authority is //host.domain.com or //host.domain.com:port. For example, http://example.ca.com.

SAML 2.0--Additional URL Configuration

The Additional URL Configuration dialog is where you configure redirect URLs to handle HTTP 500, 400, 405, and 403 error conditions. If any of these errors occur, redirect URLs can send the user to an application or a customized error page for further action.

Redirection to these customized URLs can take place only when enough information about the Identity Provider is provided to the Service Provider. For example, if during a request there is an issue in retrieving certificate information from smkeydatabase, the user is redirected to Server Error URL specified. However, if a request contains an invalid IdP ID, no redirection happens and the HTTP error code 400 is returned to the browser.

The Additional URL Configuration dialog lets you configure redirect URLs for error handling. This dialog includes the following settings:

Enable Server Error URL

Specifies the URL where the user is redirected when an HTTP 500 Server error occurs. A user can encounter a 500 error because an unexpected condition prevents the web server from fulfilling the client request. If this type of error occurs, the user is sent to the specified URL for further processing.

Example: `http://www.redirectmachine.com/error_pages/server_error.html`

Enable Invalid Request URL

Specifies the URL where the user is redirected when an HTTP 400 Bad Request or a 405 Method Not Allowed error occurs. A user can encounter a 400 error because a request is malformed or a 405 error because the web server does not allow a particular method or action to be performed. If these types of errors occur, the user is sent to the specified URL for further processing.

Example: `http://www.redirectmachine.com/error_pages/invalidreq_error.html`

Enable Unauthorized Access URL

Specifies the URL where the user is redirected when an HTTP 403 Forbidden error occurs. A user can encounter a 403 error because the URL in a request is pointing to the wrong target, such as a directory instead of a file. If this error occurs, the user is sent to the specified URL for further processing.

Example:

`http://www.redirectmachine.com/error_pages/unauthorized_error.html`

Mode Fields

For each URL, you can select a mode by which the browser redirects the user.

302 No Data

Redirects the user using an HTTP 302 redirect. When the user is redirected to the specified custom error page URL, they are redirected only with a session cookie and no information appended to the redirect URL.

HTTP POST

Redirects the user using the HTTP POST protocol. When the user is redirected to the custom error page URL, the following data, if available, is appended to the redirect URL and then posted to custom error page:

- TARGET
- AUTHREASON
- CONSUMERURL
- SAMLRESPONSE
- IDPID
- SPID
- PROTOCOLBINDING

Tasks Related to the SAML 2.0 Auth Scheme Properties Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 39: WS-Federation Resource Partner Properties Reference

This section contains the following topics:

[Resource Partner Properties Dialog](#) (see page 861)
[Resource Partner Properties Dialog Prerequisites](#) (see page 861)
[Navigate to the Resource Partner Properties Dialog](#) (see page 862)
[Resource Partner Properties Dialog Fields and Controls](#) (see page 863)
[Resource Partner Properties Dialog--Users Tab](#) (see page 865)
[Resource Partner Properties Dialog--Name IDs Tab](#) (see page 866)
[Resource Partner Properties Dialog--General Tab](#) (see page 868)
[Resource Partner Properties Dialog--SSO Tab](#) (see page 869)
[Resource Partner Properties Dialog--Attributes Tab](#) (see page 871)
[Resource Partner Properties Dialog--Signout Tab](#) (see page 873)
[Resource Partner Properties Dialog--Advanced Tab](#) (see page 874)
[Tasks Related to the Resource Partner Properties Dialog](#) (see page 875)

Resource Partner Properties Dialog

If you have purchased SiteMinder Federation Security Services, you can use the Resource Partner Properties Dialog to configure a WS-Federation Resource Partner.

More Information:

[Resource Partner Properties Dialog Fields and Controls](#) (see page 863)

Resource Partner Properties Dialog Prerequisites

The following prerequisites must be met to successfully establish an affiliate domain with a Resource Partner:

- Your SiteMinder administrator account must have the Manage Domain Objects privilege.
- The Affiliate Domain in which the Resource Partner is to be configured must pre-exist.

- Federation Web Services must be installed on a system.
Federation Web Services can be installed by the Web Agent Option Pack or by the SPS federation gateway.
- The assertion generated at the Account Partner must be forwarded to an application at the Resource Partner that can receive and interpret it.

Navigate to the Resource Partner Properties Dialog

You can access the Resource Partner Properties dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Resource Partner Properties dialog

1. If necessary, in the Object pane, click on the Domains tab.
2. In the Domains tab, click the affiliate domain to which you want to add a Resource Partner.
3. From the menu bar, select:
Edit *affiliate_domain_name*, Create Resource Partner

To access the Resource Partner Properties dialog to edit an existing object

1. If necessary, click on the Domains tab.
2. Expand the affiliate domain that contains the Resource Partner that you want to edit.
3. Click on the Resource Partners entry.
4. In the List pane, click the partner that you want to edit.
5. From the menu bar, select:
Edit, Properties of Resource Partner

Resource Partner Properties Dialog Fields and Controls

The Resource Partner Properties dialog contains the following fields and controls:

Name

Name of the Resource Partner. This name should be unique across all affiliate domains.

Description

(Optional) a brief description of the Resource Partner.

Authentication URL

Defines the protected URL used for authentication purposes when a user tries to access Resource Partner resource but does not have a SiteMinder session. When a user who has not logged in at the Account Partner requests a protected Resource Partner resource, the user is sent to the Authentication URL. This URL must point to the redirect.jsp file. For example:

`http://myserver.mysite.com/siteminderagent/redirectjsp/redirect.jsp`

myserver

Identifies the web server with the Web Agent Option Pack or the SPS federation gateway.

Important! You must create a policy to protect the Authentication URL; however, do this after you add a Resource Partner to an affiliate domain because you have to exit the Resource Partner Properties dialog.

Enabled

This check box must be marked for the Policy Server and Federation Web Services to support authentication of Resource Partner resources.

Use Secure URL

The Use Secure URL feature instructs SiteMinder's Single Sign-on Service to encrypt only the SMPORTALURL query parameter that it appends to the Authentication URL prior to redirecting the user to establish a SiteMinder session. After the user is authenticated, the Authentication URL redirects the user back to the destination specified by the SMPORTALURL query parameter.

Encrypting the SMPORTALURL prevents a malicious user from modifying the value and redirecting authenticated users to a malicious web site instead of the originally requested target.

If you select this checkbox, you must do the following:

1. Set the Authentication URL field to the following URL:
`http(s)://idp_server:port/affwebservices/secure/securedirect`
2. Protect the securedirect web service with a SiteMinder policy.

If the SiteMinder Producer/Identity Provider/Account Partner serves more than one Consumer/Service Provider/Resource Partner, it probably authenticates different users for these different partners. As a result, for each Authentication URL that uses securedirect, this web service must be included in a different realm for each partner.

To associate the securedirect service with different realms, modify the web.xml file to create different resource mappings; you cannot copy the securedirect web service to different locations on your server. This file is located in the directory `web_agent_home/affwebservices/WEB-INF`.

Application URL

(Optional) Protected URL for a custom web application that is used to supply user attributes to the SiteMinder Single Sign-on service. The application can be on any host in your network.

Attributes from the web application specified in this field are made available to the Assertion Generator and then placed in the SAML assertion by an Assertion Generator plug-in. You must write and integrate the plug-in with SiteMinder.

The FWS application supplies the sample web applications that you can use as a basis for your web application. These applications are located as follows:

`http://ap_server:port/affwebservices/public/sample_application.jsp`

`http://ap_server:port/affwebservices/public/unsolicited_application.jsp`

ap_server:port

Specifies the server and port number of the system at the Account Partner that is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

The Resource Partner Properties dialog also contains the following tabs:

Users tab

Configures the users or groups of users that can be authenticated for access to Resource Partner resources; the assertion generator can create SAML assertions that include information about these users.

Name IDs tab

Specifies the Name Identifier to be used when corresponding with this Resource Partner.

General tab

Specifies general information about the Resource Partner.

SSO tab

Configures the Single Sign-On (SSO) profile of a Resource Partner.

Attributes tab

Configures assertion attribute statements, which define the user attributes, DN attributes, or static data that are passed from the Account Partner to the Resource Partner in assertions.

Signout tab

Configures the WS-federation signout feature.

Advanced tab

(Optional) Configures an Assertion Generator plug-in.

Resource Partner Properties Dialog--Users Tab

The Users tab is where you specify the users and groups that can be granted access to resources at the Resource Partner. The Assertion Generator at the Account Partner can generate assertions for these users for authentication purposes.

You can only add users and groups from directories that are included in the affiliate domain in which the Resource Partner exists.

In this dialog, you will find the following fields and controls:

Add/Remove

Opens the Users/Groups dialog, where you can add and remove users and groups to the Current Members list box.

Users/Groups

Lists the users and groups that are included or excluded by the policy that governs assertion production for a particular user directory. The list box contains one tab for each user directory in the affiliate domain.

AND Users/Groups

Lets you configure a policy with an OR or AND relationship between the users or groups of users to instruct the Policy Server when to fire the rules included in a policy. For example, you could define the following group relationship in a policy:

Authorize if Member of (Accounting AND Marketing AND Collections) OR (Legal AND Finance AND Support) OR (Shipping)

In this example, the Policy Server would only fire the policy when the user is a member of all the groups in the AND relationship or a member of the groups in the OR relationship.

More Information:

[AND Users/Groups Check Box](#) (see page 382)

[Specify AND/OR Relationships between Users/Groups](#) (see page 384)

Resource Partner Properties Dialog--Name IDs Tab

The Name IDs tab is where you configure the Name Identifier, which names a user in a unique way in the SAML assertion. The format of the Name Identifier establishes the type of content used for the ID. For example, the format might be the User DN, in which case the content might be a uid.

Name ID Format

Specifies the format for the Name ID. Pick one of the following options:

- EmailAddress
- UPN
- Common Name

For a description of each format, see *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* specification (sstc-saml-core-2.0-cd-04.pdf).

Name ID Type Group Box

The Name ID group box contains radio buttons that specify the Name Identifier type. The selections are:

Static

Indicates that the Name Identifier is the value in the Static Value field. Activates the Static Value field; disables other controls.

User Attribute

Indicates the Name Identifier is specified in the Attribute Name field. Activates the Attribute Name field; disables other controls.

DN Attribute

Indicates the Name Identifier is specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

Allow Nested Groups

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

Name ID Fields Group Box

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

Static Value

Specifies the static text value that will be used for all name identifiers for this Service Provider.

User Attribute

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

DN Spec

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

Lookup button

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

Resource Partner Properties Dialog--General Tab

The General tab is where you specify identity information for the Account Partner and Resource Partner other general information.

Resource Partner ID

Specifies a URI that uniquely identifies the Resource Partner, such as, rp.example.com.

Account Partner ID

Specifies a URI that uniquely identifies the Account Partner, such as ap-ca.com. This becomes the Issuer field in the SAML assertion.

Skew Time

Specifies the number of seconds (as a positive integer) to be subtracted from the current time to account for Resource Partners that have clocks that are not synchronized with the Policy Server acting as an Account Partner.

D-Sig Verification Info Group Box

Determines digital signature functions.

Disable Signature Processing

If set, all signature processing for this Resource Partner (both signing and verification of signatures) is disabled for the Resource Partner.

Note: Signature processing *must* be enabled in a production environment. Disabling it by setting the Disable Signature Processing option should be used for debugging only.

Signing Alias

(Optional) Specifies the alias associated with a specific private key in the SiteMinder key database. By completing this field, you are indicating which private key the Account Partner should use to sign assertions and assertion responses.

Note: It is recommended that the private key already be stored in the key database before you specify its associated alias in this field.

Limits: an alphanumeric character string

Resource Partner Properties Dialog--SSO Tab

The SSO tab is where you configure Single Sign-On (SSO) information for the WS-Federation Resource Partner.

Authentication Method

Specifies the URI for the AuthenticationMethod attribute placed in the SAML assertion. Specify an authentication method as dictated by the SAML specification and one that is appropriate for the Authentication Level specified for the Resource Partner.

For example, if the authentication level is certificate plus basic, the default Authentication Level is 15. The default value is Password. For this default value, the URI would in the assertion would be:

`urn:oasis:names:tc:SAML:1.0:am:classes:password.`

Validity Duration

Specifies a number of seconds (a positive integer) for which a generated assertion is valid. The default is 60 seconds.

In a test environment, you may want to increase the Validity Duration value above 60, the default, if you see the following message in the Policy Server trace log:

Assertion rejected (_b6717b8c00a5c32838208078738c05ce6237) –current time (Fri Sep 09 17:28:33 EDT 2006) is after SessionNotOnOrAfter time (Fri Sep 09 17:28:20 EDT 2006)

Security Token Consumer Service

Specifies the URL of the service at the Resource Partner that receives security token response messages and extracts the assertion. The default location for the SiteMinder service is:

`https://rp_server:port/affwebservices/public/wsfeddispatcher`

rp_server:port

Identifies the web server and port at the Resource Partner hosting the Web Agent Option Pack or SPS federation gateway. These components provide the Federation Web Services application.

Note: The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to the appropriate serviced based on the query parameter data. Although there is a wsfedsecuritytokenconsumer service, the wsfeddispatcher service is recommended for the entry in this field.

Authentication Level

Specifies the minimum level at which the user must have authenticated to gain access to a SiteMinder realm. If the user has authenticated at this level, the Account Partner will generate an assertion for the user. If the user is not authenticated at this level, they are redirected to the Authentication URL to authenticate at this level.

Note: The user must have authenticated at this level or greater.

Restrictions

Opens the Restrictions dialog from where you can configure IP address and time restrictions on the assertion generation policy.

More Information:

[Restrictions Dialog for a Resource Partner](#) (see page 870)

Restrictions Dialog for a Resource Partner

The Restrictions dialog is where you configure IP address and time restrictions for the assertion generation policy for the Resource Partner.

Restrictions Dialog Prerequisites

You must be in the process of configuring Single Sign-On for a Resource Partner from the Resource Partner Properties dialog.

Restrictions Dialog Field and Controls

The IP Addresses group box on the Restrictions dialog contains the following fields:

IP Addresses

A list of restricted IP addresses configured for the policy for the Resource Partner resources. The Policy Server User Interface allows you to specify an IP address, range of IP addresses, or a subnet mask of the Web server on which a user's browser must be running for the user to access a Resource Partner.

Add

Opens an empty Add IP Address dialog from where you can create a new IP address restriction.

Edit

Opens the IP restriction entry selected in the IP Addresses list for editing in the Add IP Address dialog.

Remove

Deletes the IP restriction entry selected in the IP Addresses list.

The Time Restrictions Group Box contains the following field:

Set

Opens the Time dialog so you can configure the availability of the Resource Partner's availability. When you add a time restriction, the Resource Partner functions only during the period specified.

Resource Partner Properties Dialog--Attributes Tab

The Attributes tab is where you configure SAML assertion attribute statements. You also indicate whether the attribute will be used for calls to the Attribute Authority or for single sign-on.

The controls on this tab are as follows:

Attribute List

Selectable list of configured attributes.

Create

Opens the Service Provider Attribute Editor dialog from where you can configure Service Provider entitlement attributes.

Edit

Opens the attribute entry that you selected in the Attribute List for editing.

Remove

Deletes the attribute entry selected in the Attribute List.

Allow Nested Groups

Allows nested groups to be searched in an LDAP Directory.

Resource Partner Attribute Dialog Prerequisites

You must be in the process of configuring a WS-Federation Resource Partner from the Resource Partner Properties dialog.

Resource Partner Attribute Dialog--Attribute Setup Tab

The Attribute Setup tab contains the following fields and controls:

Attribute

Specify the format for the attribute that will be part of a SAML assertion. Options are:

- EmailAddress
- UPN
- CommonName
- Group
- NameValue

Attribute Kind

The Attribute Kind group box contains radio buttons that allow you to specify the type of attribute that will be added to the assertion:

Static

Indicates that the attribute kind is the value in the Variable Name field. Activates the Variable Value field; disables other controls in the dialog.

User Attribute

Indicates that the DN identifier is the value of Attribute Name field. Activates the Attribute Name field; disables other controls in the dialog.

DN Attribute

Indicates that the name identifier will be specified by an attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups field, and disables the Static Value field.

Note: If you select the DN Attribute radio button, you may also select the Allow Nested Groups check box. Selecting this check box allows SiteMinder to return an attribute from a group that is nested in another group specified by a policy. Nested groups often occur in complex LDAP deployments.

Allow Nested Groups

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

Note: If the attribute is configured to be encrypted, the Encryption Certificate settings in the Encryption tab will be enabled.

Attribute Fields

Complete the fields for the Attribute Kind that you selected. Different fields become available depending on the attribute type (static, user attribute, DN attribute).

Variable Name

Designates the name for the attribute that SiteMinder returns in the assertion. Complete this field for any attribute type.

Variable Value

(Static only) Defines the static text as the value for the name/value pair. Enter a value.

Attribute Name

(User and DN Attributes only) Specifies the user directory attribute SiteMinder uses for the name/value pair.

DN Spec

(DN Attribute only) Specifies the DN of the user or user group and the name of the user attribute in the Attribute Name field.

Resource Partner Attribute Dialog--Advanced Tab

The Advanced tab of the Resource Partner Attribute dialog contains the following field:

Script

Displays the script that SiteMinder generates based on your entries in the Attribute Setup tab. You can copy the contents of this field and paste them into the Script field for another response.

Note: If you copy and paste the contents of the Script field for another entitlement, you must select the appropriate radio button in the Attribute Kind group box of the Attribute Setup tab.

Resource Partner Properties Dialog--Signout Tab

The Signout tab lets you enable the WS-Federation signout feature, which terminates all sessions for a particular user at each Resource Partner where the user has a session.

The fields and controls on this dialog are as follows:

Enable Signout

Enables the WS-Federation signout feature for the Resource Partner.

SignoutCleanupURL

Specifies a URL at the Resource Partner where the user's browser is redirected to terminate the user session.

Federation Web Services redirects the user to the signout cleanup URL after the user's session is completely removed at the Account Partner and all Resource Partner sites. If the signout is initiated at the Resource Partner, the logout confirmation page should be an unprotected resource at the Resource Partner site. If signout is initiated at an Account Partner site, the logout confirmation page should be an unprotected resource at the Account Partner site.

SignoutConfirmURL

Specifies the URL at the Account Provider where the signout message and Resource Provider cleanup locations are sent as post data to the SignoutConfirm JSP page.

The default URL is:

`http://ap_server:port/affwebservices/public/signoutconfirmurl.jsp`

ap_server:port

Specifies the server and port number of the system at the Account Partner that is hosting the Web Agent Option Pack or the SPS federation gateway, depending on which component is installed in your federation network.

The signoutconfirmurl.jsp is included with the Web Agent Option Pack or SPS federation gateway. If you move this page from the default directory, make sure you put it in a location where it is accessible to the servlet engine you are using for the Federation Web Services application.

Resource Partner Properties Dialog--Advanced Tab

The Advanced tab is where you can configure a custom-developed Assertion Generator plugin developed using the Assertion Generator Plugin API. This task is optional.

The Assertion Generator Plugin Group Box has the following fields:

Full Java Class Name

Specifies the fully qualified Java class name of the Assertion Generator Plugin.

Parameters

Specifies a string of parameters to be passed to the specified plugin when the Full Java Class Name field is configured.

Tasks Related to the Resource Partner Properties Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 40: WS-Federation Authentication Scheme Reference

This section contains the following topics:

[Authentication Scheme Properties Dialog--WS-Federation Template](#) (see page 877)

[WS-Federation Auth. Scheme Properties Dialog Prerequisites](#) (see page 878)

[Navigate to the Auth. Scheme Properties Dialog--WS Federation Template](#) (see page 878)

[Federation Authentication Scheme Dialog--Scheme Setup Tab](#) (see page 879)

[WS-Federation Authentication Scheme Dialog--Advanced Tab](#) (see page 880)

[WS-Federation Auth. Scheme Fields and Controls](#) (see page 881)

[Auth. Scheme Properties--WS-Federation--Users Tab](#) (see page 882)

[Auth. Scheme Properties--WS-Federation--SSO Tab](#) (see page 883)

[Auth. Scheme Properties--WS-Federation--Signout Tab](#) (see page 885)

[Auth. Scheme Properties--WS-Federation--Advanced Tab](#) (see page 886)

[Tasks Related to the WS-Federation Authentication Scheme Properties Dialog](#) (see page 887)

Authentication Scheme Properties Dialog--WS-Federation Template

If you have purchased SiteMinder Federation Security Services, you can configure the WS-Federation authentication scheme. After the scheme is configured, you can assign it to a realm.

Scheme Common Setup Group Box

This group box lets you identify the authentication scheme.

Name

Designates a name for the authentication scheme.

Note: When Active Directory Application Mode (ADAM) is used as a policy store, the maximum length for the authentication scheme name is 22 characters.

Description

Provides a description of the authentication scheme.

Protection Level

Allows single sign-on for authentication schemes of equal or lower protection levels within the same policy domain, while requiring additional authentication to access resources with higher protection level schemes.

Limits: 1 and 1000.

Authentication schemes have a default protection level that you may need to change. High protection levels should be used for critical resources, while lower level schemes should be used for commonly accessible resources.

More Information:

[Federation Authentication Scheme Dialog--Scheme Setup Tab](#) (see page 879)

[WS-Federation Authentication Scheme Dialog--Advanced Tab](#) (see page 880)

WS-Federation Auth. Scheme Properties Dialog Prerequisites

To successfully configure a WS-Federation authentication scheme:

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Auth. Scheme Properties Dialog--WS Federation Template

You can access the Authentication Scheme dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Authentication Scheme Dialog for a new user directory connection, do the following:

1. Begin at the menu bar.
2. Select Edit, System Configuration, Create Authentication Scheme

The Authentication Scheme Dialog opens.

To access the Authentication Scheme Dialog for an existing authentication scheme:

1. In the object pane, click on Authentication Schemes.
2. In the Authentication Scheme List pane, select an existing authentication scheme.
3. Right click and select Properties of Authentication Scheme.

Federation Authentication Scheme Dialog--Scheme Setup Tab

The Scheme Setup tab for WS-Federation authentication is where you specify how the Resource Partner communicates with the Account Partner to retrieve the assertion, authenticate a user based on that assertion, then direct the user to the target resource.

The fields in the dialog box are as follows:

Resource Partner ID

Specifies a URI that uniquely identifies the Resource Partner.

Note: The value that you enter must match the value of the ID specified for the corresponding Resource Partner object that you establish at the Account Partner.

Account Partner ID

Specifies a URI that uniquely identifies the Account Partner from which assertions for this Resource Partner are issued.

The Resource Partner will accept assertions only from this Account Partner.

Note: The value that you enter for the issuer must match the value of the Account Partner ID configured at the Account Partner site.

Skew Time

Determines the number of seconds (as a positive integer) to be subtracted from the current time to account for Resource Partners that have clocks that are not synchronized with the Policy Server acting as an Account Partner.

D-Sig Verification Group Box

Contains fields and controls that allow you to specify signature verification information.

Disable Signature Processing

Disables all signature processing, that is, signing and verification of signatures, for this Resource Partner.

Caution: Signature processing must be enabled in a production environment. Disabling it by setting the Disable Signature Processing option should be used for debugging only.

Alias

Specifies a unique string of only alphanumeric characters, which locates the certificate in the Resource Partner's smkeydatabase used to verify the digital signature of the assertion. This certificate is associated with the private key stored at the Account Partner. An alias entry is required for signature verification.

Additional Configuration

Opens the WS-Fed Auth Scheme Properties dialog. From this dialog, you can specify additional configuration details for the authentication scheme.

WS-Federation Authentication Scheme Dialog--Advanced Tab

The Advanced tab of the Authentication Scheme Properties dialog for WS-Federation is where you define information about the shared library that processes WS-Federation authentication.

The fields on the Advanced tab are as follows:

Library

Designates the name of the shared library that processes WS-Federation authentication. Do not change this value, unless you have a custom authentication scheme, written using the SiteMinder Authentication API.

The default shared library for WS-Federation authentication is smauthsaml.

Parameter

Displays the information you entered in the Scheme Setup tab.

WS-Federation Auth. Scheme Fields and Controls

When you click on the Additional Configuration button from the main Authentication Scheme Properties window, the WS-Federation Auth. Scheme Properties dialog is displayed.

The WS-Federation Auth. Scheme Properties dialog contains the following tabs:

Users

Specifies how to obtain the user information from an assertion in an incoming <RequestSecurityTokenResponse> message for authentication purposes.

SSO

Defines the single sign-on bindings supported by the Resource Partner.

Signout

Enables the Resource Partner to respond to a signout request sent from an Account Partner.

Advanced

Defines advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors.

Auth. Scheme Properties--WS-Federation--Users Tab

The Users tab defines how to obtain user information from an assertion in an incoming <RequestSecurityTokenResponse> message.

The tab includes the following fields and controls:

User Disambiguation Group Box

Xpath Query

XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

Xpath queries should not contain namespace prefixes. The following is an invalid Xpath query:

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

The valid Xpath query is:

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

Example

To obtain the attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/AttributeValue/text()
```

Namespace

Selectable List of namespace types and defined search specifications from which you can select namespace (user directory) type and then define a search specification for user disambiguation.

Edit

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification that defines the attribute that the authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user store to find the correct record for authentication.

Authentication Scheme Namespace Mapping Dialog

The Authentication Scheme Namespace Mapping dialog is where you specify the attribute that the authentication scheme uses to search a namespace. The dialog contains the following field:

Search Specification

Specifies the attribute that the WS-Federation authentication scheme uses to search a namespace. Use %s in the entry as a variable representing the LoginID.

Auth. Scheme Properties--WS-Federation--SSO Tab

The SSO tab is where you define how the Resource Partner processes single sign-on requests.

The tab contains the following fields:

Redirect Mode

Indicates the method by which the Resource Partner redirects the user to the target resource. If you select 302 No Data or 302 Cookie Data, no other configuration is required. If you select Server Redirect or PersistAttributes, there is additional configuration required.

302 No Data

(Default). User is redirected via an HTTP 302 redirect with a session cookie, but no other data.

302 Cookie Data

User is redirected via an HTTP 302 redirect with a session cookie and additional cookie data configured for the Resource Partner at the Account Partner.

Server Redirect

Enables header and cookie attribute information, which is received as part of a SAML assertion, to be passed to the custom target application. The service that collects the credentials (SAML 2.0 Assertion Consumer Service or WS-Federation Security Token Consumer Service) transfers the user to the target application URL by using server-side redirect technology. Server-side redirects are part of the Java Servlet specification, and are supported by all the standard-compliant servlet containers.

To use this mode, you must follow these requirements:

- The URL you specify for this mode must be relative to the context of the servlet that is consuming the assertion, which is typically `/affwebservices/public/`. The root of the context is the root of the Federation Web Services application, typically `/affwebservices/`.

All target application files need to be in the application's root directory. This directory is either:

—Web Agent: `web_agent_home\webagent\affwebservices`

—SPS federation gateway:
`sps_home\secure-proxy\Tomcat\webapps\affwebservices`

- You need to define realms, rules, and policies to protect target resources. The realms must be defined with at least the value `/affwebservices/` in the resource filter.
- You must have a custom Java or JSP application on the server that is serving the Federation Web Services application—that is, the server where the Web Agent Option Pack or SPS federation gateway is installed.

Java servlet technology allows applications to pass information between two resource requests using the `setAttribute` method of the `ServletRequest` interface.

The service that consumes assertions sends the user attribute to the target application by an attribute object in the request before redirecting the user to the target application. The service that consumes assertions sends the attributes by creating a `java.util.HashMap` object. The attribute that contains the `HashMap` of SAML attributes is "Netegrity.AttributeInfo."

Two other `Java.lang.String` attributes are set by the service that consumes assertions to pass the user identity to the custom application:

—`Netegrity.smSessionID` attribute represents the SiteMinder session ID

—`Netegrity.userDN` attribute represents the SiteMinder user DN.

The custom target application at the customer site can read these objects from the HTTP request object and can make use of the data found in the `hashmap` objects.

PersistAttributes

User is redirected via an HTTP 302 redirect with a session cookie, but no other data. Additionally, this mode instructs the Policy Server to store attributes extracted from an assertion in the session store so they can be supplied as HTTP header variables. For additional configuration, see the instructions for using SAML attributes as HTTP headers.

Note: If you choose `PersistAttributes` and the assertion contains attributes that are left blank, a value of `NULL` is written to the session store. This value acts as a placeholder for the empty attribute and it is passed to any application using the attribute.

Target

Specifies the target resource URI at the Service Provider destination site.

Enforce Single Use Policy

Ensures the single use policy is enforced, selecting this option prevents assertions from being re-used at a Resource Partner to establish a second session.

Auth. Scheme Properties--WS-Federation--Signout Tab

The Signout tab lets you configure the Resource Partner to respond to a signout request sent from an Account Partner.

The fields and controls on this dialog are as follows:

Enable Signout

Enables WS-Federation signout.

SignoutURL

If signout is enabled, specifies the URL of the signout servlet at the Account Partner. The following default URL is the recommended entry:

`https://<ap_service:port>/affwebservices/public/wsfeddispatcher`

Note: The WSFedDispatcher Service receives all incoming WS-Federation messages and forwards the request processing to the appropriate service based on the query parameter data. Although there is a wsfedsignout service, use the wsfeddispatcher URL for the SignoutURL.

Auth. Scheme Properties--WS-Federation--Advanced Tab

The Advanced tab is where you specify advanced configuration for the Message Extension Consumer API and optional redirect URLs for assertion processing errors during authentication.

The tab contains the following fields and controls:

Message Consumer Plugin Group Box

Full Java Class Name

(Optional) Specifies the fully qualified Java class name of a class which implements a Message Consumer Plug-in interface for the authentication scheme.

Parameter

If a value is entered in the Full Java Class Name field, this field specifies a string of parameters the API passes to the specified plug-in.

Status Redirect URLs and Modes Group Box

Assertion-based authentication can fail at the site that consumes assertions for various reasons. If authentication does fail, Federation Security Services provides functionality to redirect the user to different applications (URLs) for further processing. For example, when user disambiguation fails, SiteMinder can be configured to redirect the user to a provisioning system, which could create a user account based on the information found in SAML assertion.

The following options redirect the user to a configured URL based on the condition that caused the failure.

Redirect URL for the User Not Found Status

(Optional) Identifies the URL where SiteMinder redirects the user when the user is not found. The user not found status applies when the single sign-on message has no LoginID or the user directory does not contain the LoginID.

Redirect URL for the Invalid SSO Message Status

(Optional) Identifies the URL where SiteMinder redirects the user if one of the following conditions occur:

- The single sign-on message is not well formed, according to the XML specification
- The single sign-on message is invalid based on rules specified by the SAML schemas
- The consumer requires an encrypted assertion but the single sign-on message does not contain an encrypted assertion.

Redirect URL for the Unaccepted User Credential (SSO Message) Status

(Optional) Identifies the URL where SiteMinder redirects the user for all other error conditions other than when a user is not found or the single sign-on message is invalid. The assertion is valid, but SiteMinder does not accept the message for certain reasons, such as:

- XML digital signature validation fails
- XML decryption operation fails
- XML validation of conditions fails, such as an expired message or an audience mismatch
- None of the assertions in SSO message contain an authentication statement.

Mode

Specifies the method by which SiteMinder redirects the user to the redirect URL. The options are:

302 No Data (default)

Redirects user with an HTTP 302 redirect with a session cookie, but no other data.

Http Post

Redirects user using HTTP Post protocol.

Tasks Related to the WS-Federation Authentication Scheme Properties Dialog

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 41: Attribute Authority and Query Reference

This section contains the following topics:

[SAML 2.0 Attribute Authority Reference](#) (see page 889)

[SAML 2.0 Attribute Query Reference](#) (see page 891)

[Tasks Related to Attributes and Attribute Queries](#) (see page 896)

SAML 2.0 Attribute Authority Reference

You can configure an Identity Provider to act as an Attribute Authority. The Authority provides attributes to a SAML Requester so the requester can authorize a user based on the retrieved attributes.

Attribute Svc Dialog

You configure the Attribute Authority in the Attribute Svc tab of the SAML Service Provider Properties dialog.

Note: The attributes used by the Attribute Authority are configured on the [Attributes tab of the SAML Service Provider Properties dialog](#) (see page 829).

Attribute Svc Dialog Prerequisites

To successfully configure a SAML 2.0 Attribute Authority:

- You must have installed the Policy Server and Policy Server Option Pack.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Attribute Svc Dialog

To access the Attribute Svc dialog, do the following

1. Log on to the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

2. Select an existing Service Provider object or create a new one.
The SAML Service Provider Properties Dialog opens.
3. Select the Attribute Svc tab.

Attribute Svc Dialog Fields and Controls

The following fields and controls are located on the Attribute Svc tab:

Enabled

Lets the Service Provider act as an Attribute Authority, able to generate attribute assertions based on a query message from a SAML requester.

Validity Duration

Specifies the number of seconds that the assertion is valid.

Default: 60 seconds

Require Signed Attribute Query

Indicates that the Attribute Authority requires a digitally signed attribute query from the SAML Requester.

Signing Options

Designates the Attribute Authority's signing requirements for attribute assertions and responses.

Sign Assertion

Instructs the Attribute Authority to sign only the attribute assertion. The SAML response is not signed.

Sign Response

Instructs the Attribute Authority to sign only the SAML response.

Sign Both

Instructs the Attribute Authority to sign the attribute assertion and the SAML response.

Sign Neither

Instructs the Attribute Authority not to sign the attribute assertion nor the SAML response.

User Lookup Group Box**Namespaces**

Defines search specifications for user directory namespaces. The search specification is used by the Attribute Authority to locate the user locally using the NameID provided in the subject of the Attribute Query message.

Edit

Lets you access the Authentication Scheme Namespace Mapping dialog after selecting a namespace.

Note: At least one search specification must be provided.

Attribute Service Namespace Mapping Dialog

The Attribute Service Namespace Mapping dialog is where you specify the attribute that the Attribute Authority uses to search a namespace. This dialog contains the following field:

Search Specification

Specifies the attribute that the Attribute Authority uses to search a namespace. Use %s in the entry as a variable representing the name ID.

More Information:

[SAML Service Provider Dialog--Attributes Tab](#) (see page 829)

[SAML Service Provider Dialog--Encryption Tab](#) (see page 834)

SAML 2.0 Attribute Query Reference

The SAML Requester, which in a SiteMinder environment is a Service Provider, sends an attribute query to the Attribute Authority. You configure the contents of the attribute query in the Attributes tab of a given SAML 2.0 authentication scheme.

Attributes Dialog Prerequisites

To successfully configure an attribute query:

- You must install the Web Agent and the Web Agent Option Pack or the SPS federation gateway to use the Federation Web Services application.
- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege.

Navigate to the Attributes Dialog

To access the Attributes dialog to configure a new attribute query do the following:

1. Log on to the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

2. Select an existing SAML 2.0 authentication scheme or create a new one.

The Authentication Scheme Properties Dialog opens.

3. Click on Additional Configuration.

The SAML 2.0 Auth Scheme Properties dialog opens.

4. Click the Attributes tab.

Attributes Dialog for Attribute Queries

In the Attributes tab, you configure the attributes that the SAML Requester can retrieve from an Attribute Authority. These attributes will be included in the attribute query sent to the Attribute Authority.

The dialog contains the following fields and controls:

Supported Attributes Group Box

Lists the attributes that can be requested from a SAML Attribute Authority along with the associated script stored in the policy store.

Attribute Query Group Box

Enabled

Checks to generate attribute queries.

Sign Attribute Query

Instructs the SAML Requester to sign the attribute query prior to sending it to the Attribute Authority.

Require Signed Assertions

Indicates that the SAML Requester only accepts attribute assertions that are signed by the Attribute Authority. The assertion is rejected if it is not signed.

Attribute Service

URL of the Attribute Service at the Attribute Authority. Must be a valid URL of less than 1024 characters.

Get All Attributes

Indicates that the attribute query should retrieve all attributes configured at the Attribute Authority. Checking this box results in an attribute query without any attributes; however, the Attribute Authority returns all the configured attributes, that is, attributes associated with the Attribute Service retrieval method.

In some cases, checking this box avoids some performance overhead because less checking is done when constructing the query and verifying that requested attributes are still valid at the Attribute Authority. However, if there are a large number of configured attributes at the Attribute Authority and encryption is used, then checking this box might degrade performance.

Add/Edit Attribute Dialog**Local Name**

Specifies the unique local name of the attribute used only by the authentication scheme and the federation attribute variables that you configure. The value of this field must match the Local Name field for the federation attribute variable, which is used in the policy expression that will be used when configuring the authentication scheme. Enter a unique alphanumeric string between 1 and 1024 characters.

Note: You can have attributes with the same name but with different name formats.

The Local Name field enables you to change the name of the requested attribute without changing the policy expression that protects the resource.

Example:

Attribute configuration:

- Local Name:local1
- Attribute Name: phone
- Name Format:unspecified

Federation Attribute Variable configuration:

- Name=fedvar1
- IdP ID=AttributeAuthority1
- Local Name=local1

Policy expression:

- (fedvar1=="STATIC_VALUE")

If you change only the Attribute Name from **phone** to **title**, only the requested attribute changes; the federation variable and the policy expression do not have to change.

Attribute Name

Specifies the unique name that the SAML requester includes in the attribute query. This value must match the value of the Variable Name field specified for an attribute configured at the Attribute Authority.

Name Format

Defines the format of the attribute. The options are:

- unspecified
- basic
- uri

Refer to the SAML 2.0 specification for definitions of these formats.

Name IDs Dialog Fields for Attribute Queries

The Name IDs tab is where you configure the Name ID included in the attribute query sent from the SAML Requester to the Attribute Authority.

This tab is only configurable if the Attribute Query feature is enabled, that is, the Enabled checkbox on the Attributes tab is checked.

The fields on this dialog are as follows:

Name ID Format

Specifies the format of the Name ID. This value must match the format of the expected Name ID at the Attribute Authority or the request will fail.

Name ID Type

This field defines the type of attribute used for the Name Identifier.

Static

Indicates that the Name Identifier is the value in the Static Value field.
Activates the Static Value field; disables other controls.

User Attribute

Indicates that the Name Identifier is the value in the Attribute Name field.
Activates the Attribute Name field; disables other controls.

DN Attribute

Indicates that the Name Identifier is an attribute associated with a DN.
Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

Allow Nested Groups

Indicates that nested groups are allowed when selecting the DN. Enabled if the DN Attribute Radio Button is selected.

Name ID Fields Group Box

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

Static Value

Specifies the static text value that will be used for all name identifiers for this Service Provider.

User Attribute

Specifies the name of the user attribute which contains the name identifier, or the attribute associated with a group or organizational unit DN.

DN Spec

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

Federation Attribute Variable Editor Reference

You can include a federation attribute variable for inclusion in a policy expression that protects a resource at a SAML Requester site.

The fields for the Attribute Variable Editor dialog are as follows:

Name

Names the variable.

Description

Provides a way to describe the variable.

Return Type

Specifies the type of data used to express the variable supported by the Policy Server. Options are:

- Boolean
- Date
- Number
- String

IdP ID

Using the IdP ID value as a reference, this parameter identifies the SAML 2.0 authentication scheme that contains the configuration for this federation attribute. When you configure a SAML 2.0 authentication scheme, IdP ID is one of the fields you set. This field for the federation attribute variable lists all IdP ID values for all configured SAML 2.0 authentication schemes. Select the IdP ID associated with authentication scheme that defines the attribute you want to use.

Local Name

Unique name that identifies the variable to the SAML 2.0 authentication scheme. This value must match the value you entered when adding an attribute in the Attributes tab of the SAML 2.0 Authentication Scheme.

Tasks Related to Attributes and Attribute Queries

For tasks related to this dialog, see the *Federation Security Services Guide*.

Chapter 42: SAML Affiliations Reference

This section contains the following topics:

[SAML Affiliation Properties Dialog](#) (see page 897)
[SAML Affiliation Dialog Prerequisites](#) (see page 897)
[Navigate to the SAML Affiliation Dialog](#) (see page 897)
[SAML Affiliation Dialog Fields and Controls](#) (see page 898)
[SAML Affiliation Dialog--Name IDs Tab](#) (see page 899)
[SAML Affiliation Dialog--Users Tab](#) (see page 900)
[SAML Service Providers Tab](#) (see page 902)
[SAML Auth Schemes Tab](#) (see page 902)

SAML Affiliation Properties Dialog

The SAML Affiliation Dialog is where you configure a SAML affiliation.

This dialog is only available if you purchased Federation Security Services.

SAML Affiliation Dialog Prerequisites

To successfully create an affiliation, your SiteMinder administrator account must have the Manage System Objects privilege.

Navigate to the SAML Affiliation Dialog

You can access the SAML Affiliation dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the SAML Affiliation dialog to create a new affiliation

1. In the Object pane, click the System tab.
2. In the System tab, click SAML Affiliations.

Note: If the SAML Affiliations entry is not visible in the System tab, from the menu bar select View, SAML Affiliations.

3. From the menu bar select Edit, Create SAML Affiliation.

To access the SAML Affiliation dialog to modify an existing affiliation

1. In the Object pane, click the System tab.
2. In the System tab, click SAML Affiliations.
3. In the list pane, either:
 - Double-click the SAML affiliation you want to edit.
 - Right-click the SAML affiliation you want to edit and select Properties of SAML Affiliation from the pop-up menu.

SAML Affiliation Dialog Fields and Controls

The SAML Affiliation dialog contains the following fields:

Name

Names the affiliation. This name should be unique across all affiliate domains.

Description

(Optional) Describes the affiliation.

Affiliation ID

Specifies a URI that uniquely identifies the affiliation.

SAML Version

Specifies the SAML version (inactive; the value defaults to 2.0, indicating that assertions sent to Service Providers that are members of this affiliation must be compliant with SAML version 2.0).

The SAML Affiliation dialog also contains the following tabs:

Name IDs

Specifies the name identifier for the principal to be used when corresponding with SAML Service Providers that are members of the affiliation.

Users

Configures the users or groups of users for the SAML Service Providers that are members of the affiliation. Configured users can be authenticated for access to Service Provider resources; the assertion generator can create SAML assertions that include entitlement information for these users.

SAML Service Providers

Provides a read-only listing of all SAML Service Providers that are members of the affiliation.

SAML Auth Schemes

Provides a read-only listing of all SAML 2.0 authentication schemes that are members of the affiliation.

SAML Affiliation Dialog--Name IDs Tab

The Name IDs tab is where you configure the name ID that names the principal and is used for user disambiguation by Service Providers and Identity Providers that are members of this affiliation.

Name ID Format

Specifies the format of the name identifier format. For example, the name ID might take the format of an email address, such as johndoe@example.com.

Pick one of the following:

- Unspecified
- EmailAddress
- X.509 Subject Name
- Windows Domain Qualified Name
- Entity Identifier
- Persistent Identifier
- Transient Identifier

Name ID Type Group Box

The Name ID group box contains radio buttons that specify the Name Identifier type.

Static

Indicates the Name Identifier is the value in the Static Value field. Activates the Static Value field; disables other controls.

User Attribute

Indicates the Name Identifier is the value in the Attribute Name field. Activates the Attribute Name field; disables other controls.

DN Attribute

Specifies the Name Identifier as the attribute associated with a DN. Activates the User Attribute field, the DN Spec field, and the Allow Nested Groups check box; disables the Static Value field.

Allow Nested Groups

Enables nested groups when selecting the DN spec. Enabled if the DN Attribute Radio Button is selected.

Name ID Fields Group Box

Contains fields that specify information about the selected Name Identifier. The fields in this group box are context-sensitive, being determined according to the Name ID Type selection.

Static Value

Specifies the static text value that will be used for all name identifiers for this Service Provider.

User Attribute

Specifies the name of the user attribute from within a user directory, which contains the name identifier, or the attribute associated with a group or organizational unit DN.

DN Spec

Specifies the group or organizational unit DN to be used for obtaining the associated attribute to be used as the name identifier.

Lookup button

Opens the SiteMinder User Lookup dialog to locate the user group and select a DN to populate the DN Spec field.

SAML Affiliation Dialog--Users Tab

For the entity acting only as an Identity Provider, the information on the Users tab is not relevant. You do not have to complete the fields on this tab.

If the entity is acting as an Identity Provider and a Service Provider, then the information on the Users tab is used when this system is acting as a Service Provider. As a Service Provider, it uses the information to obtain information from an incoming SAML 2.0 assertion to locate a user record and authenticate the user.

User Disambiguation Group Box**Xpath Query**

Specifies an XPath query that the authentication scheme applies to the assertion to obtain the LoginID.

The default XPath query used when none is configured, is:

```
/Assertion/Subject/NameID/text()
```

Example:

To obtain the an attribute called "FirstName" from the assertion for authentication, the XPath query is:

```
/Assertion/AttributeStatement/Attribute[@Name="FirstName"]/AttributeValue/text()
```

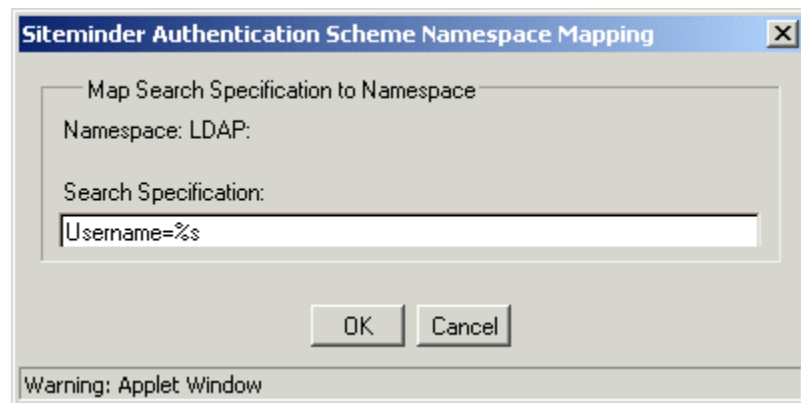
To extract the text of first Username element in the SAML assertion, use the abbreviated syntax "//Username/text()"

Namespace

Displays a selectable list of namespace types and defined search specifications from which you can select namespace (user directory) and then define a search specification for user disambiguation.

Edit

Opens the Authentication Scheme Namespace Mapping dialog where you can enter a Search Specification which defines the attribute that the authentication scheme uses to search a namespace in the Namespace list.

SiteMinder Authentication Scheme Namespace Mapping dialog

The Authentication Scheme Namespace Mapping dialog is where you specify the attribute that the authentication scheme uses to search a namespace.

Search Specification

Specifies the attribute that the SAML 2.0 authentication scheme uses to search a namespace. Use %s as the entry representing the LoginID.

For example, the LoginID is user1. If you specify Username=%s in the Search Specification field, the resulting string is Username=user1. This string is checked against the user store to find the correct record for authentication.

SAML Service Providers Tab

The SAML Service Providers tab provides a read-only list of all SAML Service Providers that belong to the affiliation. You add Service Providers to the affiliation's membership or remove them through the SAML Service Provider dialog.

SAML Auth Schemes Tab

Provides a read-only listing of all SAML 2.0 authentication schemes that are members of an affiliation. You add authentication schemes to the affiliation's membership or remove them through the SAML 2.0 Auth Scheme Properties dialog.

Chapter 43: Registration Dialog Reference

Chapter 44: Shared Dialog Reference

This section contains the following topics:

[Shared Dialog Reference Overview](#) (see page 905)

[Time Dialog](#) (see page 905)

[Active Rule Editor Dialog](#) (see page 909)

[Active Policy Editor Dialog](#) (see page 910)

[Users/Groups Dialog](#) (see page 912)

[Add an IP Address Dialog](#) (see page 916)

Shared Dialog Reference Overview

This chapter describes utility Dialogs that provide functionality common to more than one area of the Policy Server User Interface.

Time Dialog

The Time Dialog is where you configure time restrictions for rules, policies (and global rules and policies) and affiliates. A time restriction indicates specific time periods during which a rule or policy should or should not fire or when users should be able to access an affiliate. For more information, see:

Rules

[Add Time Restrictions for Rules](#) (see page 344)

Policies

[Add and Remove Time Restrictions for Policies](#) (see page 394)

Affiliates

Configure Time Restrictions for Affiliates.

Note: Time restrictions are based on the system clock of the server on which the Policy Server is installed. For more information, see [How the Web Agent and Policy Server Calculate Time](#) (see page 83).

Time Dialog Prerequisites

The following prerequisites must be met in order to successfully set time restrictions for rules using the Time Dialog:

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must be in the process of configuring a rule, global rule, policy, global policy, or affiliate.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Time Dialog

To access the Time Dialog in the Policy Server User Interface and create a rule, from the Rule Dialog or Global Rule Dialog, click the Set button in the Advanced group box on the Time Restrictions tab.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

Time Dialog Fields and Controls

The Time Dialog contains five group boxes:

- [Effective Starting Date Group Box](#) (see page 907)
- [Expiration Date Group Box](#) (see page 907)
- [Set Starting Date Group Box](#) (see page 907)
- [Set Expiration Date Group Box](#) (see page 908)
- [Hourly Restrictions Group Box](#) (see page 908)

Effective Starting Date Group Box

The Effective Starting Date group box is where you can configure a specific date before which a rule or policy will not fire. If no starting date is specified (the default), the rule or policy is effective immediately.

Display Field

Displays the effective starting date for the rule or policy, if one is set. If no date is set, "<now>" indicates that the rule or policy is effective immediately.

Select Button

Displays the Set Starting Date group box (to the right of the Effective Starting Date and Expiration Date group boxes), where you can manage starting dates for a rule or policy.

Expiration Date Group Box

The Expiration Date group box is where you can configure a specific date after which a rule or policy will not fire. If no expiration date is specified (the default), the rule or policy will not expire.

Display Field

Displays the expiration date for the rule or policy, if one is set. If no date is set, "<never>" indicates that the rule or policy will not expire.

Select Button

Displays the Set Expiration Date group box (to the right of the Effective Starting Date and Expiration Date group boxes), where you can manage starting dates for a rule or policy.

Set Starting Date Group Box

Displayed when the Select button in the Effective Starting Date group box is clicked, the Set Starting Date group box is where you specify starting dates for rule or policies. Its contents change based on the option you select.

<now> radio button

Sets the rule or policy effective as soon as you close the Time Restriction Dialog.

Select Date radio button

Sets a specific starting date on which the rule or policy should become effective. When selected, day, month and year fields with corresponding up and down arrow buttons appear. Use these controls to set a starting date for the rule or policy.

Set Expiration Date Group Box

Displayed when you click the Select button in the Effective Starting Date group box, the Expiration Date group box is where you specify expiration dates for rules or policies. Its contents change based on the option you select.

<never> radio button

Sets the rule or policy effective indefinitely.

Select Date radio button

Sets a specific date on which the rule or policy should expire. When selected, day, month and year fields with corresponding up and down arrow buttons appear. Use these controls to set an expiration date for the rule or policy.

Hourly Restrictions Group Box

The Hourly Restrictions group box is where you specify the hours and days of the week on which the rule or policy will fire.

Each block in the table represents an hour. When a block is yellow, the rule or policy will not fire during that hour, and the rule or policy will not apply to the resources governed by it. When a block is blue, the rule or policy will fire during that hour.

Grid blocks

Click on blocks in the time table to change the blocks from Blue (Fire Rule), to Yellow (Do Not Fire).

Each block in the table represents an hour. When a block is yellow, the rule will not fire during that hour, and the rule will not apply to the resources governed by it.

Hour Blocks (blocks in the body of the table)

Toggle between blue (fire rule or policy) and yellow (do not fire). The numbered buttons along the top of the Hourly Restrictions group box allow you to toggle the value of an entire column of Grid Blocks with a single click.

Day buttons (Monday, Tuesday, and so on)

Toggle an entire row (day) between blue (fire rule or policy) and yellow (do not fire). The buttons representing the days of the week along the left side of the Hourly Restrictions group box allow you to toggle the value of an entire row of Grid Blocks with a single click.

Hour buttons (across the top of the table)

Toggle an entire column (hour of the day) between blue (fire rule or policy) and yellow (do not fire).

Always Fire button

Change all blocks in the table to blue (fire rule or policy).

Never Fire button

Change all blocks in the table to yellow (never fire).

Reset button

Returns the time restriction settings to their previously saved state.

Tasks Related to the Time Dialog

The following tasks are related to the Time Dialog:

- [Add Time Restrictions for Rules](#) (see page 344)
- [Configure Time Restrictions for Global Rules](#) (see page 444)
- [Add and Remove Time Restrictions for Policies](#) (see page 394)
- [Add and Remove Global Policy Time Restrictions](#) (see page 454)

Active Rule Editor Dialog

The Active Rule Editor is where you specify active rules for rules.

Active Rule Editor Dialog Prerequisites

The following prerequisites must be met in order to successfully configure an active rule using the Active Rule Editor Dialog:

- You must be in the process of configuring a rule in the Rule Dialog or a global rule in the Global Rule Editor Dialog.

Navigate to the Active Rule Editor Dialog

To access the Active Rule Editor Dialog to create an active rule, click the Edit button in the Advanced group box Active Rule tab on the Rule Dialog.

Active Rule Editor Dialog Fields and Controls

Library Name field

The name of the shared library that supports the Active Rule.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default path for your system to search for the shared library at run-time.

You should not include a file extension for the shared library. For example, if you are using a Windows platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory, you simply need to enter lib in the Library Name field.

Function Name field

The name of the actual function in the shared library that implements the Active Rule.

Function Parameter(s) field

(Optional) List of parameters to be passed to the function in the shared library.

Generated Script field

Displays the generated active rule script as you enter information in the other fields.

Tasks Related to the Active Rule Editor Dialog

The following tasks are related to the Active Rule Editor Dialog:

- [Configure an Active Rule](#) (see page 346)

Active Policy Editor Dialog

An active policy is used for dynamic authorization based on external business logic. An active policy is included in the authorization decision by having the Policy Server invoke a function in a customer-supplied shared library. This shared library must conform to the interface specified by the Authorization API (available separately with the Software Development Kit; if installed, see the *Developer's Guide for C* for more information).

Active Policy Editor Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must edit an existing policy object, or create a new policy object.

More information:

[Policy Server Administrators](#) (see page 223)
[Policies](#) (see page 375)

Navigate to the Active Policy Editor

You can access the Active Policy Editor from the Policy Dialog in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Active Policy Editor

1. Open the Policy Dialog.
2. Select the Advanced tab.
3. Click the Edit button.

The Active Policy Editor opens.

More information:

[Navigate to the Policy Dialog](#) (see page 734)

Active Policy Editor Fields and Controls

Library Name

Name of the shared library that supports the Active Policy.

You can include a path in the Library Name, however, it is not required. If you do not specify a path, the Policy Server uses the default path for your system to search for the shared library at run-time. You should not include a file extension for the shared library.

For example, if you are using a Windows platform and your shared library is called lib.dll, and is located in the \SiteMinder\bin\ directory which is added to your system path when you install the Policy Server, you simply need to enter lib in the Library Name field.

Function Name

Name of the actual function in the shared library that implements the Active Policy.

Parameter(s)

(Optional) List of parameters to be passed to the function in the shared library.

Generated Script

System-generated display of the generated string that the Policy Server will use to call the active policy.

Tasks Related to the Active Policy Editor

The following task is related to the Active Policy Editor:

- [Configure a Global Active Policy](#) (see page 454)

Users/Groups Dialog

The Users/Groups Dialog provides an interface for defining users and groups for:

Policies

Specifies users and groups to be explicitly included in or excluded from a policy.

Affiliates

Specifies users and groups that are allowed access resources at an affiliate site.

SAML Service Providers

Specifies users and groups that are allowed access resources at a Service Provider site.

Users/Groups Dialog Prerequisites

Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.

More information:

[Policy Server Administrators](#) (see page 223)

Navigate to the Users/Groups Dialog

There are multiple ways to access the Users/Groups Dialog from either the Policy, Affiliate or SAML Service Provider Dialogs in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To access the Users/Groups Dialog

1. From the Policy Dialog, Affiliate Dialog, or SAML Service Provider Dialog, select the Users tab.
2. If applicable, select the tab associated with a particular user directory connection.

Note: If the policy domain that contains the policy object includes multiple user directory connections, the Users tab will contain a tab for each directory connection.

3. Click the Add/Remove button.

The Users/Groups Dialog opens.

Users/Groups Dialog Fields and Controls

Current Members

Lists the users and groups currently included in the policy, affiliate, or SAML Service Provider.

Available Members

Lists the users and groups that are available for inclusion in the policy, affiliate, or SAML Service Provider.

Entry

Specify users or groups to be added to the policy, affiliate, or SAML Service Provider without using the Current and Available Members lists.

For LDAP directories, enter a valid DN in the Manual Entry field.

When entering a manual entry for an LDAP user directory, you can also select one of the following from the Action drop down list:

Search Users

Indicates that the LDAP search specified in the Entry field is limited to matches in user entries.

Search Groups

Indicates that the LDAP search specified in the Entry field is limited to matches in group entries.

Search Organizations

Indicates that the LDAP search specified in the Entry field is limited to matches in organization entries.

Search Any Entry

Indicates that the LDAP search specified in the Entry field is limited to matches in user, group, and organization entries.

For Microsoft SQL Server, Oracle and WinNT directories, enter a user name in the Manual Entry field.

For an Microsoft SQL Server or Oracle, you can enter a SQL query, instead. For example:

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

The Policy Server will perform the query as the database user specified in the Username field of the Credentials and Connection tab for the user directory. When constructing the SQL statement for the Manual Entry field, you need to be familiar with the database schema for the user directory. For example, if you are using the SmSampleUsers schema and want to add specific users, you could select from the SmUser table.

If the manual entry query does not contain a WHERE statement, the WHERE statement from the Init User query field is appended. For example:

Init User: select Name from SmUser where Name = '%s'

Manual Entry: Select Name from customers

Result: select Name from customers where Name = '%s'

If the manual entry query contains a WHERE statement, the portion of the query following the Init User WHERE statement is appended. For example:

Init User: select Name from SmUser where Name = '%s'

Manual Entry: Select Name from customers where balance > 1000

Result: select Name from customers where balance > 1000 and Name = '%s'

Note: For an LDAP directory, you can enter all in the Manual Entry field to bind the policy, affiliate, or SAML Service Provider to the entire LDAP directory.

Create Expression button and Edit Expression button

Open the Expression Editor. The Expression Editor allows you to bind an LDAP search expression to the policy, affiliate, or SAML Service Provider.

Note: The Create Expression and Edit Expression buttons are only available for LDAP directory connections. If you selected a directory connection on the Users tab associated with an LDAP directory connection, the Expression Editor group box appears to the right of the Manual Entry group box.

Search button

The following button opens a search Dialog associated with the type of user directory.

See one of the following:

- [Search LDAP/AD Dialog](#) (see page 584)
- [Search Database Dialog](#) (see page 586)
- [Search WinNT Directory Dialog](#) (see page 587)

Previous/Next/All buttons

When a directory contains enough users and groups to require multiple pages of information, the previous and next buttons page through the available users and groups.

Click the All button to display all users and groups in a single scrolling list.

Left and Right Arrow Buttons

Select users or groups in the Available Members list and click the Left Arrow to move the selected items to the Current Members list.

Select users or groups from the Current Members list and click the Right Arrow to move them to the Available Members list.

You can select multiple user and groups by holding the CTRL or SHIFT key and clicking on items in one of the Members lists. When you select multiple items and click one of the Arrow buttons, all of the selected items are moved to the other list.

Exclude button

Explicitly excludes selected users in the Current Members list. If you exclude users or groups, the following symbol appears to the left of the entry in the Current Members list.

Add to Current Members button

For items you specify in the Manual Entry group box, click this button to add items to the Current Members list.

Validate Entry

(Active Directory) When adding a manual entry for Active Directory, this setting requires that the entry be validated before being added to the Current Members list. If the entry does not represent a user or group in the directory, the entry is not added to the policy.

More information:

[Expression Editor](#) (see page 738)

[Policy Dialog—Users Tab](#) (see page 735)

Tasks Related to the Users/Groups Dialog

The following tasks are related to the Users/Groups Dialog:

- [Add Users to a Policy](#) (see page 379)
- [Exclude a User or Group from a Policy](#) (see page 381)
- [Add Users by Manual Entry](#) (see page 385)
- [Enhance Policy Server's LDAP Authorization Performance](#) (see page 386)
- [Add LDAP Expressions to Policies](#) (see page 387)
- [Bind Policies to Users with the Manual Entry Field](#) (see page 398)

Add an IP Address Dialog

This Dialog allows you to specify an IP address, range of IP addresses, subnet mask or a host name for use in policy and affiliate objects.

Add an IP Address Dialog Prerequisites

- Your SiteMinder administrator account must have the Manage System and Domain Objects privilege or the Manage Domain Objects privilege.
- You must be in the process of creating or editing an existing policy object or affiliate.

Navigate to the Add an IP Address Dialog

You can access the Add an IP Address Dialog from either the Policy or Affiliate Dialogs in the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

To navigate to the Add an IP Address Dialog

1. From the Policy Dialog or Affiliate Dialog, select the IP Addresses tab.
2. Click the Add button.

The Add an IP Address Dialog opens.

Add an IP Address Dialog Fields and Controls

The contents of the Add an IP Address Dialog change based on the option you select.

Single Host radio button

If selected, the IP address entered in the associated IP Address fields is added to the policy or affiliate.

Note: You can use the Single Host option to add multiple individual IP addresses to the policy or affiliate.

IP Address fields

Each field specifies one of the four portions of the IP address.

DNS Lookup button

Opens a DNS Lookup Dialog. Enter a host name in the dialog and click OK. The IP address of the host name appears in the Add an IP Address Dialog.

Host Name radio button

If selected, the host name specified in the associated Host Name field is added to the policy or affiliate.

Host Name field

Host name to be included in the policy or affiliate.

Subnet Mask radio button

If selected, the IP address and subnet mask entered using the associated controls is added to the policy or affiliate.

IP Address fields

Each field specifies one of the four portions of the IP address.

DNS Lookup button

Opens a DNS Lookup Dialog. Enter a host name in the dialog and click OK. The IP address of the host name appears in the Add an IP Address Dialog.

Subnet Mask slider

Specifies the subnet mask.

Subnet Mask arrow buttons

Move the slider to the left or right by one position for fine-tuning the subnet mask value.

Range radio button

If selected, the range of IP addresses specified in the associated controls are added to the policy or affiliate.

From IP Address fields

Each field specifies one of the four portions of the IP address that starts the range of addresses.

DNS Lookup button

Opens a DNS Lookup Dialog. Enter a host name in the dialog and click OK. The IP address of the host name appears in the Add an IP Address Dialog.

To IP Address fields

Each field specifies one of the four portions of the IP address that completes the range of addresses.

DNS Lookup button

Opens a DNS Lookup Dialog. Enter a host name in the dialog and click OK. The IP address of the host name appears in the Add an IP Address Dialog.

Tasks Related to the Add an IP Address Dialog

The following task is related to the Add an IP Address Dialog:

- [Specify Allowable IP Addresses for Policies](#) (see page 391)

Chapter 45: TransactionMinder-related User Interface Elements

This section contains the following topics:

[Overview](#) (see page 919)

Overview

If you have purchased and installed TransactionMinder, a number of elements are added to the Policy Server User Interface:

Authentication schemes (added to the Authentication Scheme Dialog)

- XML Document Credential Collector Template
- XML Digital Signature Template
- SAML Session Ticket Template
- WS Security Template

Variables (added to the variable list and with corresponding variable editor Dialogs)

- SAML Assertion variables
- Transport variables
- XML Agent variables
- XML Body Variables
- XML Envelope Header Variables

For descriptions of these elements, see the *TransactionMinder Operations Guide*.

Chapter 46: Troubleshooting

This section contains the following topics:

[Policy Server Hangs When Web Agent Connection Is Idle](#) (see page 921)

[Policy Server Does Not Connect to LDAP Stores Using SSL with AKI](#) (see page 922)

[Troubleshooting SSL Authentication Schemes](#) (see page 923)

Policy Server Hangs When Web Agent Connection Is Idle

Platform: Solaris only

Symptom:

When the Policy Server sends a request to a Web Agent that is no longer connected, the Policy Server hangs while waiting for a response.

Solution:

To solve this problem, enable the following Policy Server environment variable by setting its value to one:

`SM_ENABLE_TCP_KEEPALIVE`

When the TCP keep-alive environment variable is set to one and the Web Agent connection is idle, the Policy Server sends keep-alive packets to the Web Agent. If the Web Agent does not respond to the keep-alive packets, the Policy Server closes the connection and does not hang.

If Transport Layer Interface (TLI) logging is enabled, the enabled/disabled state of the TCP keep-alive environment variable is logged. TLI logging is enabled by setting a Web Server environment variable. The Policy Server keep-alive environment variable is disabled by default.

On Solaris, there are two TCP keep-alive parameters to set:

`tcp_time_wait_interval`

(milliseconds) Specifies how long to wait before sending the first keep-alive packet.

`tcp_keepalive_interval`

(milliseconds) Specifies how long to wait for a response before sending the next keep-alive packet.

To display the value of a TCP keep-alive parameter, use the following operating system command:

```
ndd -get/dev/tcp tcp_parameter_name
```

To enable TCP keep-alive in the Policy Server on a Solaris platform

1. Set the time interval to wait before sending the first keep-alive packet, as follows:

```
ndd -set/dev/tcp tcp_time_wait_interval timewait_value
```

timewait_value

Specifies a value in milliseconds for the `tcp_time_wait_interval` parameter.

2. Set the time interval to wait before sending subsequent keep-alive packets, as follows:

```
ndd -set/dev/tcp tcp_keepalive_interval keepalive_value
```

keepalive_value

Specifies a value in milliseconds for the `tcp_keepalive_interval` parameter.

3. Set the value of the Policy Server TCP keep-alive environment variable to one: `SM_ENABLE_TCP_KEEPALIVE`.

Policy Server Does Not Connect to LDAP Stores Using SSL with AKI

Platform: Solaris only

Problem

The Policy Server does not connect to LDAP stores that use SSL certificates with the AKI extension.

Solution

To solve this problem, follow these steps:

1. Install the patch corresponding to your version of Solaris:

- Solaris 8: 119209-12
- Solaris 9: 119211-12
- Solaris 10: 119213-12

Patches for Solaris 8 and 9

Deploy libraries in the following directory: `/usr/lib/mps/secv1`.

Patch for Solaris 10

Deploys libraries in the following directory: `/usr/lib/mps`.

Note: Since a default installation of Solaris 10 contains the necessary libraries already, you may not need this patch.

2. Verify the following:
 - NSS libraries are version 3.9.3 or later.
 - NSPR libraries are version 4.5 or later.
3. Stop the Policy Server.
4. Copy the following five libraries to the ../siteminder/lib directory, overwriting the existing libraries:
 - libplc4.so
 - libnspr4.so
 - libplds4.so
 - libssl3.so
 - libnss3.so
5. Restart the Policy Server.

The Policy Server can now connect to LDAP stores that use SSL certificates with the AKI extension.

Troubleshooting SSL Authentication Schemes

Configuring the SSL Advanced Authentication Schemes requires Web Servers to be properly configured to use SSL. Most of the problems you may encounter configuring Authentication Schemes over SSL connections are likely to be SSL configuration issues. Therefore, the first step in troubleshooting Authentication Schemes over SSL is to verify that SSL is properly configured and working. This is done without the interaction of the SiteMinder Web Agent so that these components can be individually analyzed.

Determine SSL Connection Ability

The first step in troubleshooting Authentication Schemes over SSL is to verify that SSL is properly configured and working. This is done without the interaction of the SiteMinder Web Agent so that these components can be individually analyzed.

To determine whether you are able to establish an SSL connection

1. Disable the SiteMinder Web Agent protecting the realm for which you want to use an authentication scheme over SSL.

Note: For information about disabling a Web Agent, see the *Web Agent Guide*.

2. Using your browser, go to one of the following URLs (using a browser with a certificate):

- https://web_server_name:port (Netscape Web Servers)
- https://web_server_name:port/<SSL Virtual Directory> (IIS Web Servers)
- https://web_server_name:port (Apache Web Servers)

If this SSL connection is configured to require certificates, you will be prompted to select a certificate.

If you are unable to successfully establish this SSL connection, then see [SSL Configuration](#) (see page 924) for more information on configuring SSL. If you were able to establish this connection, but have not been successful in configuring SiteMinder, see [SSL Troubleshooting](#) (see page 928).

SSL Configuration

It is imperative that SSL be configured and working properly before using SiteMinder. In order to make an SSL connection, you must be able to trust the certificate authority of an incoming certificate. For example, if a browser presents a certificate that was signed by VeriSign, you must have a VeriSign Certificate Authority installed and trusted in the Web Server. In addition to trusting client certificates that are presented, the server itself must have a certificate to present to the clients. The clients have to trust the Certificate Authority that issued the certificate. This allows for mutual authentication.

Once these certificates have been installed, you can configure the Web Server to use SSL and *require* certificates, if desired. For detailed SSL configuration information, see the documentation provided with your web server software. This section contains step-by-step instructions for configuring your Web Server and Web browser to successfully establish an SSL connection. If you have correctly configured SSL, but are still having problems making the connection, see the common problems at the end of the section.

Install the Netscape Web Server Certificate

If you have not already done so, you will need to generate a key for your Web Server. This is done through a command line utility. See the Netscape documentation for the details.

To install the Netscape Web Server Certificate

1. Once you create a key, request a certificate. This is done under Keys & Certificates: Request Certificate in Netscape Server Administration.
2. After creating the certificate request, go to the Certificate Authority and load the request. The Certificate Authority then creates a Server Certificate for you.

3. Once the Server Certificate is created, load it into the Web Server. This is done under Keys & Certificates: Install Certificate. When filling out this information be sure to select This Server under the Certificate For section.
4. Once the certificate is loaded, verify that it is there and trusted by going to Manage Certificates. Look for the ServerCert certificate. Select the Certificate and verify that it is trusted.
5. Click OK and restart the Web Server.

Configure the Netscape Web Server to use SSL

After installing the Netscape Web Server Certificate, you must configure the Netscape Web Server to use SLL by requiring certificates.

To require certificates for your SSL Web Server

1. In Netscape Server Administration, click Admin Preferences.
2. Click Encryption On/Off and ensure that Encryption is on.
3. If you are running the *Certificate* or *Certificate with Basic Authentication Scheme*, you must require certificates. This is done under Encryption Preferences setting where Require Certificates must be set to On. From a browser that has a Certificate installed, verify that you can get to https://servername:port.

Note: Do not turn on Required Certificates for the *Certificate* or *Basic Authentication Scheme*.

Enable the Web Server to Trust Client Certificates in Netscape

If a certificate authority is already installed in the Web Server, go on to the next section. Otherwise, install a certificate for the Certificate Authority on the SSL Web Server.

To enable the Web Server to Trust Client Certificates in Netscape

1. Obtain the Certificate Authority's certificate and either keep it on your screen or save it to a file.
2. In Netscape Server Administration, select Keys & Certificates.
3. Click Install Certificate.
4. In the Certificate For field, fill out the Server Security Chain.
5. In the Certificate Name field, enter a description.
6. If you saved your Certificate Authority's certificate to a file, enter the file name in the Message is in this file field; otherwise, select the Message text (with headers) radio button and paste the certificate in the Message text (with headers) field.
7. Click OK and restart the Web Server.

Establish Trust for the Netscape Certificate Authority

If a certificate authority is installed in the Web Server, you can establish trust between the two.

To establish trust for the Netscape Certificate Authority

1. In Netscape Server Administration, select Keys & Certificates.
2. Select Manage Certificates.
3. Select the Certificate Authority. The system displays a dialog detailing the certificate.
4. Select Trust.
5. Click OK and restart the Web Server.

Enable the Web Server to Trust Client Certificates in Windows

You must trust your client certificates by installing the appropriate Certificate Authority Certificates.

SSL Web Servers must have certificates for each Certificate Authority. Major certificate authorities may already be installed. You can configure certificates in Windows operating systems by using the Certificates snap-in. For information, see your Windows documentation.

Configure the IIS Web Server to use SSL

To verify that a secure port is enabled on the Web Server, open the Management Console, right-click on the Web Server, and view the Web site tab, where you can see the SSL Port. Generally, the number of the secure port is 443. Verify that the port number is installed.

Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

The advanced authentication schemes can create virtual directories in the Web Server. These directories are automatically configured to require SSL and certificates according to the specifications of each authentication scheme. For testing purposes, you can create a test virtual directory. You can configure this virtual directory to require certificates through the Directory Security tab, Secure Communications. When testing the virtual directory, verify that the browser is asked for a certificate:

`https://servername:port/virtual directory`

Install the IIS Web Server Certificate

If you have not already done so, generate a key for your Web server through the Management Console, Key Manager, Access the Key Manager.

Note: Note this process may be slightly different for IIS 3 and IIS 4.

Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

To install the IIS Web Server Certificate

1. In the Management Console, right-click the Web Server and select Properties.
2. Click the Directory Security tab.
3. In the Secure Communications panel, click Key Manager.
4. Under Key, select Create New Key and a Wizard will guide you through the process.

Once you create a key, you can request a certificate using the file created in the steps mentioned earlier. Go to the Certificate Authority and request a certificate for this server. You will need to paste the certificate request information generated in Step 1 in order to receive a certificate. Once you received a certificate, go back to Management Console, Directory Security and click Key Manager to install the certificate for the key described in the next step.

5. Right-click the key name and select Install Certificate.
6. Restart the Web Server.

Enable the Web Server to Trust Client Certificates in Apache

If a certificate authority is already installed on your web server, go on to the next section. Otherwise, install a certificate for the CA on the SSL Web Server as follows.

To enable the Web Server to trust client certificates in Apache

1. Download and build the following Apache components:
 - Apache
 - OpenSSL
 - Mod_SSL
 - Mod_so - This module is included as part of Apache, but it must be enabled during the build of the Web server.
 - RSAREf-2.0

See the *Policy Server Installation Guide* for details about installing the web server.

2. Copy the CA certificate into the `apache/conf/ssl.crt` directory in x509 b64 format.
3. Run `make` in the `apache/conf/ssl.crt` directory.
4. Restart the Web Server.

Installing the Apache Web Server Certificate

The process for installing a certificate on an Apache Web Server varies with individual configurations. Consult the documentation for Mod_SSL and OpenSSL for details about how to configure these components.

SSL Troubleshooting

The following sections detail the most common problems encountered when dealing with SSL authentication schemes.

There Was No Prompt for a Certificate

If you were not prompted for a certificate, verify that SSL is configured appropriately. If the Web Agent is installed, disable the Web Agent. The first step is to verify a simple SSL connection.

To determine whether you are able to establish an SSL connection

1. Disable the SiteMinder Web Agent protecting the realm for which you want to use an authentication scheme over SSL.

Note: For information about disabling a Web Agent, see the *Web Agent Guide*.

2. Using your browser, go to one of the following URLs (using a browser with a certificate):
 - `https://web_server_name:port` (Netscape Web Servers)
 - `https://web_server_name:port/<SSL Virtual Directory>` (IIS Web Servers)
 - `https://web_server_name:port` (Apache Web Servers)

If this SSL connection is configured to require certificates, you will be prompted to select a certificate.

After Following Previous Procedure, Still No Certificate Prompt

There are five additional steps to take if you are still not receiving a certificate prompt.

- [Verify that all web servers are configured to use SSL and require certificates](#) (see page 929)
- [Verify the following settings for each SiteMinder Virtual Directory](#) (see page 929)

- [Check the web server's certificate expiration](#) (see page 930)
- [Verify browser certificate validity](#) (see page 930)

Verify That All Web Servers Are Configured to Use SSL and Require Certificates

For Netscape Web Servers

1. In the Netscape Server Administration, click Admin Preferences.
2. Click Encryption On/Off and verify that the encryption is on, then click OK.
3. Click Encryption Preferences and verify that Required Certificates is set.
4. Restart the Web Server.

For IIS Web Servers

Verify that the virtual directories *SMGetCredCert*, *SMGetCredCertOptional*, *SMGetCredNoCert* are created and have the correct settings.

- *SMGetCredCert* - Require Certificates will be selected
- *SMGetCredCertOptional* - Accept Certificates will be selected
- *SMGetCredNoCert* - Do not accept certificates will be selected

Note: As part of the SiteMinder SSL Authentication setup, SiteMinder configures SSL virtual directories based on the type of SSL connection required by the authentication scheme.

Verify the Following Settings for each SiteMinder Virtual Directory

To verify the following settings for each SiteMinder Virtual Directory

1. In the Management Console, right-click a virtual directory and select Properties.
2. Click the Directory Security tab.
3. Click Edit Secure Communications.

Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

For Apache Web Servers

In the httpd.conf file, be sure to set SSLVerifyClient as follows:

- For Basic over SSL: SSLVerifyClient none
- For Certificate or Basic: SSLVerifyClient optional
- For Certificate/Certificate and Basic: SSLVerifyClient require

Note: For Apache Web servers where Certificates are required or optional, the "SSL Verify Depth 10" line in the httpd.conf file must be uncommented.

Check the Web Server's Certificate Expiration

Netscape Servers

1. In the Netscape Server Administration, click Keys & Certificates.
2. Click Manage Certificates.
3. Click ServerCert.
4. Verify that it is trusted, and has not expired. If it does not exist, or has expired, you will need to request a new certificate by following the steps in [Install the Netscape Web Server Certificate](#) (see page 924).

IIS Servers

1. In the Management Console, right-click the Web Server and select Properties.
Important! If you are accessing this graphical user interface on Windows Server 2008, open the shortcut with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.
2. Click the Directory Security tab.
3. In the Secure Communications panel, click Key Manager.
4. Select a key to view its properties and verify that the key has not expired.
5. If you need to make any changes, restart the Web Server.

Apache Servers

If an Apache Web Server certificate expires, you will receive an error messages at server startup that indicates the certificate has expired.

Verify Browser Certificate Validity

Internet Explorer Browsers

1. In the IE browser menu select View, Internet Options.
2. Select Certificates, Personal.
3. Verify that the certificate is listed there and that it is valid. If it is not there or is not valid, install a new certificate.

After Certificate Prompt, Authentication Failure Received

Apache Web Servers

- Verify that the SSL Web Server contains the certificate authority of the certificate supplied.
- Verify that the SSL Web Server Trusts the certificate authority of that certificate.
- Ensure the SSL Verify Depth 10 is uncommented.

Netscape Web Servers

Verify that the Certificate Authority for the certificate is listed and that the Trust for the certificate has not expired. If it is not there or is not valid, install a new CA certificate.

IIS Web Servers

Verify that the certificate is listed and that it is valid. If it is not present or is not valid, install a new certificate. If you are able to get to the destination directory, then certificates are installed correctly.

Verify Correct Policy Server and Web Agent Configuration

After completing the steps in the previous topic based on your specific web server, verify your policy server and web agent configuration.

To verify correct policy server and web agent configuration

1. Check that the Policy Server is created correctly.
2. Check that the Web Agent contains the correct Policy Server information.
3. Verify that the Web Agent is enabled.
4. Restart the Web Agent and Policy Server.

SiteMinder Policy Should Allow Access, but SSL-Authentication Failed Message Received

In this situation, there is a Policy that is being called, but the user is incorrectly being denied access. This can result from a number of configuration errors. Common errors include:

- The SSL Server is not configured to Require Client Certificates. Therefore, the client is not passing a certificate; thereby disabling SiteMinder authentication process. You can verify this is the situation by enabling the logging option in the Web Agent. The log should indicate that the user is unknown. To correct this problem, turn on Require Certificates in the SSL Web Server.
- The Policy was not created properly. Check the Policy's users and be sure that the selection is correct.
- For Apache Web server, ensure the SSL Verify Depth is set properly and uncommented.

More information:

[Configure a Policy Domain](#) (see page 309)

[Certificate Mapping](#) (see page 293)

Error Not Found Message Received

This is generally caused from the Authentication Scheme Parameter being configured improperly. The redirect is not configured properly so the Web Server is unable to find the SSL Web Agent component.

More information:

[Authentication Schemes](#) (see page 231)

Running Certificate or Basic but Cannot Enter Basic credentials.

On Netscape Web Servers, the *Certificate or Basic* scheme requires the Web Server to have encryption turned on, but does not require certificates. Be sure that in the Encryption Preferences section of the Netscape Server Administration, the Require Certificate setting is set to No.

Chapter 47: LanMan User Directories

This section contains the following topics:

- [About LanMan User Directories](#) (see page 933)
- [LanMan Directory Connection Prerequisites](#) (see page 933)
- [Configure a LanMan Directory Connection](#) (see page 934)
- [Failover for Windows User Directories](#) (see page 935)
- [LanMan User Directory Search Criteria](#) (see page 936)

About LanMan User Directories

In a Windows environment, the Policy Server enumerates and manages the resources in a directory service through the Microsoft Active Directory Service Interface (ADSI) layer. This layer abstracts the capabilities of directory services from different network providers in a distributed computing environment. However, the current version of ADSI has its own limitations which can adversely affect the performance of the Policy Server.

With ADSI, every Windows directory request must always pass through the Primary Domain Controller (PDC) first. This compounds the network traffic that the PDC must handle. A custom solution to this dilemma is for the Policy Server to channel Windows directory requests to Backup Domain Controllers (BDCs) while bypassing the PDC. The Policy Server handles this sort of custom solution by using LanMan directory connections.

The LanMan user directory connection option allows you to specify a failover list of BDCs used for each user directory lookup in the Windows Registry. Using a LanMan directory connection, the Policy Server sends Windows directory requests to the first active BDC in the Registry list, rather than forcing requests to pass through the PDC.

LanMan Directory Connection Prerequisites

The following conditions must be met before the Policy Server can use a LanMan directory connection to access user data in a Windows directory:

- The file SmDsLanman.dll must reside in the Policy Server installation directory. The default location is:
`<Installation Directory>\netegrity\siteminder\bin\`
- You must configure a connection to the LanMan directory.

More information:

[Configure a LanMan Directory Connection](#) (see page 934)

Configure a LanMan Directory Connection

There are two procedures for configuring a LanMan directory connection in the Policy Server User Interface:

- [Configure Registry Keys for a LanMan Directory Connection](#) (see page 934)
- [Configure a LanMan User Directory Connection](#) (see page 935)

Configure Registry Keys for a LanMan Directory Connection

The first procedure in configuring a LanMan directory connection is configuring the appropriate registry keys.

To configure registry keys for a LanMan directory connection

1. From the Windows Start menu, select Run.
The Run dialog opens.
2. Enter regedit in the Run dialog and click OK.
Windows opens the Registry Editor.
3. Modify the following registry key:
 - In
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds,
the NameSpaces key is set to the following string value:
"LDAP:,ODBC:,OCI:,WinNT:,Custom:,AD:"
 - Add the following string to the value data for the NameSpaces registry key:
Lanman.
4. Create the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds\Lanman_DC
5. Create a registry key of the NT Domain Name under the Lanman_DC key:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds\Lanman_DC\<NT_domain_name>
For example:
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds\Lanman_DC\MyDomain
6. Create a registry value named NumUserDir of type DWORD under the newly created NT Domain key. For the value data, enter the actual number of separate sets of user directories (maximum 16) in this NT domain.

7. Create String registry values of UserDir0, UserDir1, ..., UserDirN, in sequential order starting from 0, for each failover list of BDCs.
8. Enter comma delimited strings for each failover list. SmDsLanman will read the lists and will find the first active BDC in each failover list to look up NT users and groups.
9. Repeat steps 5 through 7 for other NT domains.
10. Restart the Policy Server services.

For more information on starting and stopping the Policy Server, see the *Policy Server Management* guide.

Configure a LanMan User Directory Connection

To configure a LanMan User Directory Connection

1. Open the SiteMinder User Directory dialog.
2. In the Name field, enter the name of the new user directory connection.
3. Optionally, in the Description field, type a brief description of the user directory connection.
4. In the Directory Setup group box, from the NameSpace drop-down list box, select LanMan.
5. The contents of the SiteMinder User Directory dialog change to support a Windows user directory connection.
6. In the Domain Controller Key field, enter the name of one of the NT Domains that you set up in the registry keys.

More information:

[User Directories](#) (see page 141)

[Configure Registry Keys for a LanMan Directory Connection](#) (see page 934)

Failover for Windows User Directories

The list of registry keys you create for the LanMan user directory connection determines failover order.

LanMan User Directory Search Criteria

LanMan directory connections are a type of Windows user directory connection. A LanMan directory connection functions similarly to a regular Windows connection, with the exception of which actual Domain Controller handles requests. This does not affect the procedure for executing a user directory search.

More information:

[Search Directories](#) (see page 211)

Chapter 48: Customizing Registration Services

This section contains the following topics:

[Customizing Registration Services Overview](#) (see page 937)

[The Cascading Style Sheet](#) (see page 937)

[Properties Files](#) (see page 941)

[JavaServer Pages \(JSPs\)](#) (see page 949)

[Search Customization](#) (see page 950)

[Task Selector Page Customization](#) (see page 958)

[Collect Additional Attributes](#) (see page 963)

[Localize Registration Services](#) (see page 970)

Customizing Registration Services Overview

Registration Services is a subset of DMS functionality. See the *SiteMinder Web Agent Quarterly Release 2 Release Notes* for more information.

This chapter explains how to customize Registration Services, including User Self Registration. You can customize the following components:

Cascading Style Sheets

Determine the appearance of fonts and colors in Registration Services pages

Properties Files

Define information about users, and control elements that are displayed on Registration Services pages

JavaServer Pages

Comprise Registration Services presentation.

The Cascading Style Sheet

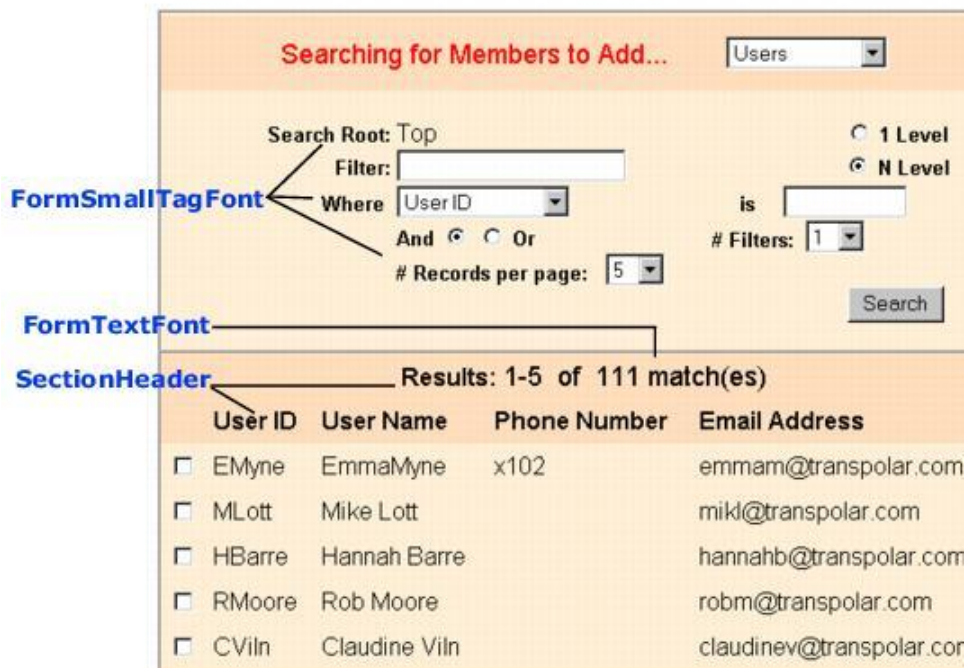
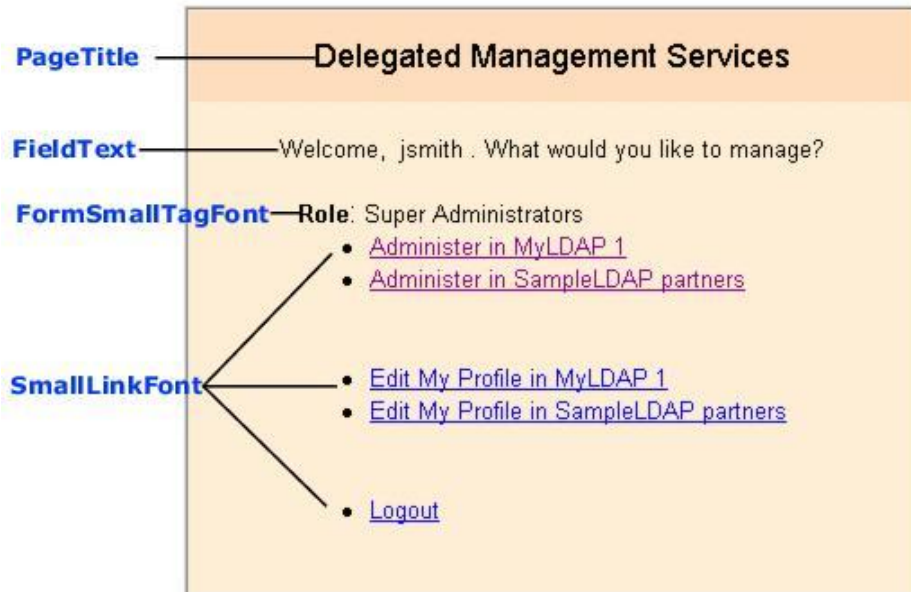
A Cascading Style Sheet (CSS) determines the appearance of Registration Services pages. The CSS contains style definitions for elements, such as fonts and color, in an HTML or JSP page. A CSS is used to maintain a consistent format appearance across multiple pages that reference the same CSS. A change in the CSS is reflected in all of the pages.

The Cascading Style Sheet defines the following elements:

Element	Style Name	Description
Background color	.lightBackground	The predominant color of the Registration Services screens. The color is expressed in the CSS as a hexadecimal code.
Border color	.darkBackground	The border color in the Registration Services screens. This color is expressed in the CSS as a hexadecimal code.
Field name	.Field Text	Used for field names on collector and submitter pages, such as user_data_collector.jsp. Also used for the welcome message on the Registration Space Selector Page.
Small header	.FormSmallTagFont	Used for field names on search pages and small headers.
Section label	.SectionHeader	Used for section headings, such as User Profile Information.
Page title	.PageTitle	Used for the title at the top of the collector and submitter pages, such as user_data_collector.jsp.
Medium header	.FormTextFont	Used for medium headers and large text, such as the number of matches in search results pages.
Warning messages	.WarningFont	Used for warning messages when using the forgotten password feature.
Required field indicator	.RequiredFieldFont	Used for the asterisk (*) that indicates a required field.
Links	.SmallLinkFont	Used for links in the Registration Selector and Task Selector pages.
Search title	.Form TitleFont	Used for the title of standard search pages.

Note: The Registration Services forms that users log into do not use the Cascading Style Sheet. To modify the appearance of the login forms, edit the .FCC files, which are installed in the Forms directory where Registration Services is installed.

The following figures illustrate how the styles CSS are used:



Cascading Style Sheet Customization

The CSS contains classes. Classes define style variations that are applied to text. For example, you could create a class that displays text in blue font. Each time you applied the class to text in the HTML, the text would be displayed in blue.

The CSS file, which is called `dms_style.css`, is located in the `dmspages` directory where Registration Services is installed. For example:

- For Windows: `C:\Program Files\Netegrity\<Web Agent Installation>\Samples\dmspages`
- For UNIX: `/export/smuser/netegrity/<webagent_installation>/samples/dmspages`

Note: After modifying the Cascading Style Sheet, the JSPs must be recompiled for the changes to take effect.

To customize the CSS, you can modify the existing class definitions or create new definitions.

Class Definitions

Classes are defined as follows:

```
.SmallLinkFont{
    font-family: sans-serif, serif, monospace;
    font-size: 80%;
    font-weight: normal;
}
```

In this example:

- `.SmallLinkFont` is the style name
- The curly braces (`{}`) start and end the style definition
- The pairs within the curly braces are the properties that define the style
- The style properties are separated by semi-colons (`;`)

To use classes, the HTML pages must include additional code that tells the browser when to use the class. The HTML code that must be added resembles the following:

```
<span class="SmallLinkFont">
```

where `span` applies the style from the CSS, and `class="SmallLinkFont"` tells the browser which style to use.

To modify a class, change the definition of the style properties. For example, to change the font to bold, change the following property in the class definition from `font-weight: normal;` to `font-weight: bold;`.

When a browser renders an HTML page that contains the `<span class="SmallLinkFont">` it will display the text in bold.

Links Between The Cascading Style Sheet to Registration Services Pages

In Registration Services, the style sheet is linked to an include file, `inc_header.jsp`. This include file, which initializes the variables needed to make requests, is linked to the Registration Services pages.

The following HTML and JSP code links the style sheet to `inc_header.jsp`:

```
<style>
<%@ include file="dms_style.css" %>
</style>
```

Properties Files

Properties files are text files that contain name/value pairs. The Registration Services JSPs use these name/value pairs to obtain information about users and global properties. Registration Services includes four types of properties files:

Locale

Determines the text, such as field names, error messages, and links, which is displayed on Registration Services pages.

Presentation

Determines which elements, such as fields and links, appear in the Registration Services pages, based on an administrator's role.

Object

Define LDAP attribute names for profile data, such as address and title.

Mapping

Maps a set of properties files to a Registration Space.

Locale Properties Files

The locale properties file (dms_en_US.properties) is a text file that contains name/value pairs. An identifier, which is referenced in the JSP pages, is paired with a value, which is typically a text string. The values can be modified to localize Registration Services. For example, dms_en_US.properties may include the following pair:

TITLE_ID=Title

When Registration Services requests the value of TITLE_ID to display in a page, it reads the locale properties files and returns "Title."

The locale properties file defines the following elements:

- Field names
- Dialog titles
- Error messages
- Links and button text
- Text messages
- List box items

To modify the text in the Registration Services pages, change the text strings that are paired with the appropriate identifiers.

For example, to add the word "optional" to the label for the Title field, change the text string as follows:

TITLE_ID=Title (optional)

Note: After modifying the locale properties file, you must recompile the JSPs for the changes to take effect.

Presentation Properties Files

Presentation properties files determine which elements, such as fields and links, Registration Services displays based on a user's role. For example, using presentation properties files, you can make the Manage Organizations link on the Task Selector page appear for Super Administrators but not for Organization Administrators.

The presentation properties files contain object identifiers, which correspond to elements in the Registration Services pages. The object identifiers are paired with administrative roles. For roles where the field or link is visible, the role is followed by an R, for Read access. For example, the City field for organizations is represented in the organization properties file as:

```
CITY_ID=Admin:R,OrgAdmin:R
```

Note: The presentation and object properties files describe different qualities of the same object—visibility and LDAP attribute definition, respectively. The object identifiers used in the presentation properties files must match the object identifiers in the tables properties files.

To make the City field in the example above visible only to Super Administrators, delete the R that follows OrgAdmin, for example:

```
CITY_ID=Admin:R,OrgAdmin
```

Registration Services includes the following properties files, which are located in the presentation directory where the default properties files are installed:

global.properties

Contains the presentation privileges for the links in the Task Selector page.

search.properties

Contains the presentation privileges for the attributes on which users can search. search.properties includes attributes for users, organizations, and groups.

user.properties

Contains the presentation privileges for the fields in the pages used to add and modify users.

Note: The properties files group.properties and org.properties, which are located in the presentation directory are used by DMS2.

In the presentation properties files, administrative roles are represented as follows:

Role	Representation
Super Administrator	Admin

Role	Representation
Organization Administrator	OrgAdmin
Users who can modify their own profiles	SelfAdmin
Self-registered users	SelfReg

Object Properties Files

The user.properties file defines the LDAP attributes in user profiles. This file contains identifiers mapped to LDAP attribute names. For example, the attribute for address appears as follows:

```
ADDRESS_ID=postaladdress
```

To modify an attribute name, change the value that is paired with the identifier. Do not modify the identifier.

The user.properties file is stored in the tables directory where the default properties files are installed—for example:

```
C:\Program Files\Netegrity\<Web Agent Installation>\properties\default\tables
```

Note: The tables directory contains org.properties, group.properties, and manage.dorg.properties, which are used by DMS2. Modifying these files does not affect Registration Services.

Identifier	Description
OBJECTCLASS	Identifies the object classes for organizations. The object class defines the types of attributes that an entry can contain. The attribute can be required or optional. Each LDAP directory includes a set of pre-defined object classes. See the documentation for your directory server for information on the default object classes.
USER_ID	Identifies the LDAP attribute where Registration Services stores the user ID.

■ PASSWORD_ID	Identifies the LDAP attribute where a user's password is stored.
■ NEW_PASSWORD_ID	Note: The specified attribute name should match the Password Attribute that you configured when you set up the connection to your LDAP directory in the SiteMinder Policy Server.
■ CONFIRM_PASSWORD_ID	
■ CONFIRM_NEW_PASSWORD_ID	
FIRSTNAME_ID	Identifies the LDAP attribute that stores a user's first name.
LASTNAME_ID	Identifies the LDAP attribute that stores a user's last name.
COMMONNAME_ID	Identifies the LDAP attribute that stores a user's common name.
TITLE_ID	Identifies the LDAP attribute that stores a user's title.
ADDRESS_ID	Identifies the LDAP attribute that stores a user's street address.
CITY_ID	Identifies the LDAP attribute that stores the user's city.
STATE_ID	Identifies the LDAP attribute that stores the user's state
COUNTRY_ID	Identifies the LDAP attribute that stores the user's country.
ZIP_ID	Identifies the LDAP attribute that stores the user's ZIP code.
PHONE_ID	Identifies the LDAP attribute that is used to store the user's telephone number.
FAX_ID	Identifies the LDAP attribute that is used to store the user's FAX number.
EMAIL_ID	Identifies the LDAP attribute that is used to store the user's e-mail address.
ORG_ID	Identifies the LDAP attribute where the organization to which a user belongs is stored.
ATTR_PEOPLE_CONTAINER_ID	Identifies the attribute name for the container where user profiles are stored—for example, ou. The attribute name is paired with the value of DEFAULT_NAME_USER_ID to form the DN of the user container.
DEFAULT_NAME_PEOPLE_ID	Specifies the name of the user container—for example, People. The container name is paired with the value of ATTR_PEOPLE_CONTAINER_ID to form the DN of the user container.
OBJECTCLASS_PEOPLE_CONTAINER_ID	Identifies the objectclass for the users container.

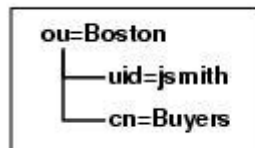
How Containers are Specified for Users

Registration Services stores managed users in containers under organizations where they are created in the LDAP directory. For example, user profiles may be stored in the container, `ou=People`, under an organization, such as `ou=Sample Organization`. When a user is added, Registration Services creates this container under the specified organization automatically (if the container doesn't already exist).

The default containers is `ou=People`.

You can modify these container names to reflect your directory structure.

You can specify `None` to add user profiles directly under the organization. For example:



The `user.properties` file contains attribute/value pairs that identify where user profiles are stored.

Note: If you created an additional set(s) of properties files and mapped it to a Registration Space, modify the object properties files in the `tables` directory where you created the new set of properties files.

The following attribute/value pairs in the object properties files identify the containers:

```
ATTR_USER_CONTAINER_ID=ou
DEFAULT_NAME_USER_ID=People
OBJECTCLASS_OBJECTTYPE_CONTAINER_ID=objectclass,top,organizationalunit
```

To modify the name of the container for users, change the value that is paired with the variable. Do not change the variable. For example, to change the name of the Users container, modify "Users" as shown:

```
ATTR_USER_CONTAINER_ID=ou
DEFAULT_NAME_USER_ID=Employees
OBJECTCLASS_OBJECTTYPE_CONTAINER_ID=objectclass,top,organizationalunit
```

To specify `None` for users, change the value of the *attribute identifier* and the *container name* to `None`, as shown:

```
ATTR_USER_CONTAINER_ID=None
DEFAULT_NAME_USER_ID=None
OBJECTCLASS_OBJECTTYPE_CONTAINER_ID=objectclass,top,organizationalunit
```

More information:

[Map Properties Files](#) (see page 947)

Map Properties Files

Mapping properties files allow you to customize the default locale, presentation, and object properties files, and map them to specific Registration Spaces. Customizing properties files enables you to do the following, without changing the JSPs:

- Use different types of LDAP directories, such as NDS and Netscape LDAP
- Change the text that is displayed in the JSP pages, such as field labels and page titles
- Modify which types of administrators can view a profile's attributes, such as password
- Manipulate different object schema

You can have multiple sets of properties files, each linked to a different Registration Space. The registration space properties files map a Registration Space identifier (*SPID*) to a sub-directory where properties files for the specified environment are located.

For example, if a user accesses the following URL:

`http://myserver.myorg.org/servlet/DMS/Admin/sales/user_data_collector.jsp`

the Registration Space properties files map sales in the URL to a subdirectory in the Properties directory where Registration Services is installed.

If no mapping is specified, Registration Services uses the properties files under the default folder where the properties files are installed—for example:

For Windows

`C:\Program Files\Netegrity\<Web Agent Installation>\Samples\Properties\default`

For UNIX

`export/smuser/netegrity/<webagent_installation>/samples/properties/default`

Registration Services includes the following mapping properties files in the properties directory:

reg space_tables.properties

Maps Registration Space identifiers to object properties files, which are contained in a tables directory.

reg space_presentation.properties

Maps Registration Space identifiers to presentation properties files, which are contained in a presentation directory.

reg space_locale.properties

Maps Registration Space identifiers to locale properties files, which are contained in a locale directory.

To map properties files to a Registration Space

1. Create a new directory under the properties directory where Registration Services is installed—for example:

- For Windows:

C:\Program Files\Netegrity\<Web Agent installation>\Samples\your-directory

- For UNIX:

export/smuser/netegrity/<web agent installation>/samples/properties/your-directory

Note: To easily identify the directory, give it the same name as the Registration Space to which it will apply.

2. From the properties directory, copy the directory (or directories), which contains the properties files that you want to modify, into the directory that you created. For example, if you want to modify presentation and object properties files, copy the presentation and tables directories into the new directory.
3. In the mapping properties file for the properties file type(s) that you copied in step 2, add the following:

spid=directory name

where *spid* is the Registration Space identifier that you specified when creating the environment, and *directory name* is the name of the properties directory that you created.

JavaServer Pages (JSPs)

Registration Services is composed of JavaServer Pages (JSPs) that define presentation. There are several types of JSPs, which perform different functions in Registration Services:

Initiator pages

Accessed directly by users. Initiator pages include:

Collector pages

Collect user data. For example, the `user_data_collector.jsp` page initiates the create and modify user functions. The information collected by these JSPs is posted to submitter pages.

Submitter pages

Submit data to the DMS servlet. These pages are invoked by the collector pages.

Special Application pages

Used by applications within Registration Services, such as the launch and forgotten password applications.

Object Profile pages

Define the presentation of form elements, such as text fields and list boxes, for specific object profiles. These pages consist mostly of HTML code. Object profile JSPs are included by the collector JSPs.

Object Table pages

Process object-specific actions, such as add, modify, delete, and lookup. For example, `table_profile_user.jsp` processes the following actions:

- Add user
- Modify user
- Delete user
- Lookup user

The following table lists the Registration Services initiator pages:

JSP Page	Description
<code>forgotpassword.jsp</code>	Displays a challenge question for users who have forgotten their password and collects the answer.
<code>launch.jsp</code>	Contains a list of tasks that a user can perform for a specific Registration Space. Each task is represented as an HTML link.

JSP Page	Description
manage_data.jsp	Collects search parameters.
manage_data_submitter.jsp	Submits a search request to the Policy Server.
newpassword.jsp	Presents a temporary password to users who have correctly answered their forgotten password question. Once users log in with the temporary password, they are immediately prompted to enter a new password.
submitusername.jsp	Submits a username to forgotpassword.jsp.
user_data_collector.jsp	Collects user data and posts it to user_data_submitter.jsp
user_data_submitter.jsp	Creates or modifies directory entries for groups.
welcome.jsp	Contains a list of tasks that the user can perform across multiple Registration Spaces.

Search Customization

The search page enables administrators to locate users.

Some common customization scenarios include:

- [Modify the Search Fields](#) (see page 950)
- [Change the Search Attributes](#) (see page 953)
- [Specify Search Limits](#) (see page 954)
- [Customize the Search Results Fields](#) (see page 954)

Modify the Search Fields

The JSP page that enables administrators to search for users in their directory is `manage_data.jsp`. It includes search fields in which administrators can specify search criteria. For example, you can select an attribute, such as `uid`, and specify a value for that attribute.

You can modify the fields that appear on the search page by modifying `manage_data.jsp`, which is installed in the `samples\dmspaces` directory where Registration Services is installed.

Note: If you created additional pages directories for specific registration spaces, modify the `manage_data.jsp` page in the appropriate directories.

Each field in the search page is controlled by a parameter in `manage_data.jsp`.

Note: The display of the # Records per Page field is set in `profile_search_init.jsp`.

The section of `manage_data.jsp` that controls the display of these fields resembles the following:

```
boolean searchFixedRoot = true;
String searchRoot = (String)request.getAttribute("orgpath");
boolean searchNoFilter = false;
boolean searchNoFilterLines = false;
boolean searchFixedFilterLines = false;
boolean searchNoScope = false;
int searchScope = 1;
int searchDisplayColumn = 0;
```

➤

The following table describes the parameters in the previous section of `manage_data.jsp`.

Parameter	Description
<code>searchObject</code>	Used by DMS.
<code>searchFixedRoot</code>	Determines whether or not the Search Root field is writable. This boolean can have the following values: true (default)—Makes the Search Root field Read Only. Users cannot modify the search root manually. false—Makes the Search Root field writable. Administrators can manually enter the search root where they would like to begin a search.
<code>searchRoot</code>	Determines the search root that Registration Services uses. The <code>searchRoot</code> parameter is paired with an expression that retrieves the current organization's DN. If no current organization is selected, the DN of the highest organization for which the administrator has privileges is used.
<code>searchNoFilter</code>	Controls the display of the Filter field. The Filter field enables administrators to manually enter search parameters. This boolean can have the following values: true—Removes the Filter field from the search page. false (default)—Adds the Filter field to the search page.

Parameter	Description
searchNoFilterLines	Controls the display of the Where and Is fields. The Where field allows administrators to select an attribute to search. The Is field enables administrators to specify a value for the selected attribute. This boolean can have the following values: true—Removes the Where and Is fields from the search page. false (default)—Adds the Where and Is fields to the search page.
searchFixedFilterLines	Controls the display of the #Filters field and the And and Or radio buttons. The #Filters field enables administrators to specify the number of filters that they want to set. The And radio button joins multiple search filters. The Or radio button causes Registration Services to search for results that match at least one of the search filters. This boolean can have the following values: true—Removes the #Filters field and the And and Or radio buttons from the search page. false (default) —Adds the #Filters field and the And and Or radio buttons to the search page.
searchNoScope	Controls the display of the 1 Level and N Level radio buttons. These buttons determine how many levels of the directory the administrator wants to search. This boolean can have the following values: true—Removes the 1 Level and N Level radio buttons from the search page. false (default) —Adds the 1 Level and N Level radio buttons to the search page. Note: If you remove these radio buttons, Registration Services will perform N-level searches, unless you change the value of searchScope.
searchScope	Controls the default scope for searches. The values for this parameter are: 2—sets the default search scope to n-levels 1 (default)—sets the default search to one level

More information:

[Customize the Results Per Page](#) (see page 957)

Change the Search Attributes

The attributes that are displayed in the Where fields are determined by the search.properties presentation file. The search.properties file contains entries for the attributes associated with organizations, users, and groups. Each entry identifies an attribute and specifies which type of Registration Services administrators can search on that attribute. The entries have the following format:

IDENTIFIER=Administrative role:R

IDENTIFIER

Identifies the attribute.

The identifier must match identifiers in the object and locale properties files. The object properties file maps the identifier to an attribute in your LDAP directory. The locale properties file maps the identifier to a text string, which is displayed in the drop-down list in the Where field.

Administrative role

Is a comma-delimited list of administrative roles. By default, only Super Administrators and Organization Administrators can search for managed objects. The default list contains the following roles:

Admin

Indicates Super Administrators

OrgAdmin

Indicates Organization Administrators

R

Indicates that the attribute is visible (and therefore, searchable) for users who have the corresponding role.

For example:

`TITLE_ID=Admin:R,OrgAdmin:R`

In this case, the Title attribute is visible in the list of searchable attributes that is displayed for Super Administrators and Organization Administrators. Both types of administrators can search on a user's title.

You can change the displayed attributes based on the administrator's role by adding or removing the R. For example, you may only want to allow Super Administrators to search for users based on their title. In this case, you would remove the R that is paired with OrgAdmin for TITLE_ID as follows:

`TITLE_ID=Admin:R,OrgAdmin`

Note: Restart the servlet engine that Registration Services is using for changes to the properties files to take effect.

Specify Search Limits

Setting search limits can improve performance by preventing large scale searches, which can tie up system resources.

Search limits can be set in three places:

- In the object properties file, `global.properties`
- The user directory properties in the SiteMinder Policy Server User Interface
- The LDAP directory

When Registration Services performs a search, it first checks to see if a search limit is set in the `global.properties` file. If no limit is set, Registration Services then checks to see if limits are set in the user directory properties in SiteMinder. If there are no specified values in the Policy Server, the LDAP directory's settings (if any) apply.

For information on setting search limits in SiteMinder, see [User Directories](#) (see page 141). For information on setting search limits in the LDAP directory, see the documentation for your directory server.

In `global.properties`, you can set two limits:

SEARCH_TIMEOUT

Ends the search after a specified amount of time. This value is specified in seconds.

SEARCH_LIMIT

Ends the search after a specified number of entries have been searched.

Customize the Search Results

You can customize the following elements of the search results:

- The page where Registration Services displays the results, as described in [Modify the Search Results Page](#) (see page 955) in the following section.
- The attributes that Registration Services displays in the results list, as described in [Specify Attributes in the Search Results](#) (see page 956).
- The number of results per page, as described in [Customize the Results Per Page](#) (see page 957).

Modify the Search Results Page

By default, Registration Services displays the results of a search query at the bottom of the search page. You can modify where Registration Services displays the search results by changing `manage_data.jsp`.

The following statements tell Registration Services to display the search results at the bottom of the search page:

```
<%
    if( action.equalsIgnoreCase("INITSEARCH"))||
        action.equalsIgnoreCase( "DOSEARCH" ) ||
    action.equalsIgnoreCase("DELETE") ||
    action.equalsIgnoreCase("JOIN") ||
        action.equalsIgnoreCase("DISJOIN") ||
    action.equalsIgnoreCase("GET_SEARCH_OBJECT_ATTRS")
    ) { %>
```

To display search results in a different page, comment out the all of the statements in the code above *except* the first statement. To comment out these lines, add two forward slashes (`//`) to the beginning of each line and after (`"INITSEARCH"`), as shown:

```
<%
    if( action.equalsIgnoreCase("INITSEARCH") //||
        //action.equalsIgnoreCase( "DOSEARCH" ) ||
        //action.equalsIgnoreCase("DELETE") ||
        //action.equalsIgnoreCase("JOIN") ||
            action.equalsIgnoreCase("DISJOIN") ||
        //action.equalsIgnoreCase("GET_SEARCH_OBJECT_ATTRS")
    ) { %>
```

Note: The changes to this section of code are indicated with bold text.

The resulting search page resembles the following:

Results: 1-5 of 68 match(es)			
User ID	User Name	Phone Number	Email Address
☐ EMyne	EmmaMyne	x102	emmam@transpolar.com
☐ MLott	Mike Lott		mikl@transpolar.com
☐ HBarre	Hannah Barre		hannahb@transpolar.com
☐ RMoore	Rob Moore		robm@transpolar.com
☐ CViln	Claudine Viln		claudinev@transpolar.com
Add User Edit... Delete 			

Specify Attributes in the Search Results

When Registration Services searches for a user, it displays a set of attributes in the results. For example, when an administrator searches for a user, Registration Services displays the user's ID, name, phone number and email address by default. You can specify which attributes are displayed.

The JSP page that determines which attributes are displayed in the search results is `table_profile_manage.jsp`. The section of `table_profile_manage.jsp` that contains the appropriate code appears after the following comments:

```
//**** Construct search filter and search results attributes
//**** based on the search object (org, user or group)
//*****
```

Note: Registration Services enables you to search for users. Modifying search results settings for organizations and groups in `table_profile_manage.jsp` will not affect Registration Services search pages.

To customize the displayed attributes, modify the identifiers that are specified in statements that resemble the following:

```
searchResultFieldIds.addElement( (String)("USER_ID"));
```

In this statement:

- `searchResultFieldIds.addElement`—Adds an attribute to the results list
- `(String)`—Specifies the type of attribute (in this case, a text string)
- `("USER_ID")`—Is the identifier of the attribute

Note: The attribute identifier should match an identifier in `user.properties`.

The following section of the code determines which user attributes are displayed in the search results:

```
else if( searchObject.equalsIgnoreCase("USER") )
{
    searchResultFieldIds.addElement( (String)("USER_ID"));
    searchResultFieldIds.addElement(
        (String)("COMMONNAME_ID"));
    searchResultFieldIds.addElement(
        (String)("PHONE_ID"));
    searchResultFieldIds.addElement(
        (String)("EMAIL_ID"));
    showOrgPath=false;
}
```


In the sample above, `showOrgPath` does not correspond to an attribute in the object properties files for users. If `showOrgPath` is set equal to `true`, Registration Services displays the user's DN in the search results.

Customize the Results Per Page

The Registration Services search page includes a #Records per page field that allows administrators to select the number of results that are displayed per page.

You can customize the items that appear in the #Records per Page list box in the `profile_search_init.jsp`. The code that determines these items appears below:

```
<select name="N_MAX_ITEMS">

.

.

.

<option value="5" <% if (searchResultLines <= 5)
    out.print("selected"); %> >5</option>

<option value="10" <% if ((searchResultLines > 5) &&
    (searchResultLines <= 10)) out.print("selected"); %> >10
</option>

<option value="20" <% if ((searchResultLines > 10) &&
    (searchResultLines <= 20)) out.print("selected"); %> >20
</option>

<option value="50" <% if (searchResultLines > 20)
    out.print("selected"); %> >50</option>

</select>
```

In these statements:

- `value` indicates the value to be returned if the option is selected
- `<% if (searchResultLines > #)out.print("selected"); %>` is JSP code that preserves the selected option when `manage_data.jsp` is reloaded—for example, when Registration Services posts the search results to that page.

To modify a number in the #Records per Page list, modify one of the option statements, as follows:

```
<option value="30" <% if ((searchResultLines > 20) &&
                        (searchResultLines <= 30)) out.print("selected"); %> >30
</option>
```

Task Selector Page Customization

The Task Selector page displays the tasks that an administrator can perform for a selected Registration Space. These tasks are HTML links that take the administrators to pages where they can carry out their task. The displayed links are based on the administrator's role.

You can customize the displayed links, including modifying the link text, changing the displayed tasks based on administrative role, and adding new links.

Modify the Links

You can modify the links on the Task Selector page by customizing the link text and changing which administrators can see a link, based on their role.

Change The Text of Links

The text that comprises the links is determined by the `dms_en_US.properties`. The `dms_en_US.properties` is located in the locale directory under the `properties/default` directory.

The `dms_en_US.properties` file contains identifiers paired with text strings. The identifiers are referenced in the JSP code in `launch.jsp`. When Registration Services renders `launch.jsp` and builds the task links, it adds the text string where ever it finds the paired identifier.

These are the entries for the default Registration Services links in `dms_en_US.properties`:

```
MANAGEUSER_ID=Manage Users
```

```
MODIFYUSER_ID=Edit My Profile
```

```
LOGOUT_ID=Logout
```

To modify the text that appears as links, change the text that is paired with the identifier. For example, to change "Manage Users" to "Manage Employees," modify the text paired with `MANAGEUSER_ID` as follows:

```
MANAGEUSER_ID=Manage Employees
```

Note: Do not modify the identifier. The identifiers are also used in the object and presentation properties files.

More information:

[Locale Properties Files](#) (see page 942)

Change the Tasks Displayed For Administrators

By modifying the presentation properties file, `global.properties`, you can change which tasks are displayed to administrators based on their roles.

The `global.properties` file contains entries that have the following format:

IDENTIFIER=Administrative role:R

IDENTIFIER

Identifies the task. The identifier matches the identifier in the locale properties file.

Administrative role

Is a comma-delimited list of administrative roles.

R

Indicates that the task is displayed for users who have the corresponding role.

For example:

`MANAGEUSER_ID=Admin:R,OrgAdmin:R`

This entry tells Registration Services to display the Manage Users link for Super Administrators and Organization Administrators.

To change the displayed tasks based on the administrators role, add or remove the `R`.

For example, to remove the Manage Groups link from an Organization Administrator's Task Selector Page, delete the `R` paired with `OrgAdmin`:

`MANAGEGROUP_ID=Admin:R,OrgAdmin:`

Add Tasks to the Task Selector Page

Registration Services uses three files to define a task link:

The locale properties file (`dms_en_US.properties`)

Determines the link text

The presentation properties file (`global.properties`)

Determines whether or not the link is displayed based on administrative role

The Task Selector page (`launch.jsp`)

Sets up the link

To add a new task to the Task Selector page:

1. Make sure that the task has an identifier and has entries in the locale properties file and `global.properties`:

- In the locale properties file, the entry should resemble the following:

IDENTIFIER=Link Text

where *IDENTIFIER* identifies the task and *Link Text* is the text string that will be the link in the Task Selector page.

- In `global.properties`, the entry should resemble the following:

IDENTIFIER=Admin:R,OrgAdmin:R

where *IDENTIFIER* identifies the task.

Note: The identifier for the task must be the same in the locale and presentation properties file.

2. In `launch.jsp`, define a form:

```
<form name="FormName" action="/servlet/  
DMS<%=request.getAttribute("adminFilter")%>  
"targetpage.jsp" method="post">  
  
<input type="hidden" name="ACTION" value="ActionName">  
  
<input type="hidden" name="ORGPATH" value="<%=orgpath%>">  
  
</form>
```

where:

- *FormName* is the name of the form that you are creating
- *targetpage.jsp* is the name of the page where users are directed when they click on the link
- *ActionName* is the name of the action administrators will perform, such as NEW if administrators are adding a new object

3. Define the JavaScript:

```
<script language="JavaScript">
<!--
function FunctionName() {
    document.FormName.submit();
}
//-->
</script>
```

where:

- *FunctionName* is the name of the function that you are defining. The function name should begin with the word Submit, for example—SubmitAddUser
- *FormName* is the name of the form that you created in step 1

4. Add the link:

```
<ul align="center" type="disc">
<% if (accessProperties.getVisible(role, "IDENTIFIER"))
{ %>
    <jsp:setProperty name="localebean" property="key"
value="IDENTIFIER"/>
<li><font class="SmallLink"><a href="javascript:Function
Name()"><jsp:getProperty name="localebean"
property="label"/></a></font></li>
```

where:

- *IDENTIFIER* is the name of the identifier for the task.
This identifier must match identifiers in the locale properties file and in global.properties. The locale properties file provides the text for the link. The global.properties file determines which types of administrators can view the link on their Task Selector page.
- *FunctionName* is the name of the function you defined in step 2.

Example: Adding a Register New Users Link

The following code adds the link "Register New Users" to the Task Selector Page:

- In dms_us_EN.properties:
NEWUSER_ID=Register New Users
- In global.properties:
NEWUSER_ID=Admin:R,OrgAdmin:R
- In launch.jsp:

Note: Only the sections of launch.jsp which relate specifically to the Register New Users link are displayed below.

```
</head>

<body bgcolor="#ffffff" text="#000000">

<form name="AddUserForm" action="/servlet/DMS
    <%=request.getAttribute("adminFilter")%>
    user_data_collector.jsp" method="post">
    <input type="hidden" name="ACTION" value="NEW">
    <input type="hidden" name="ORGPATH" value="<%=orgpath%>">
</form>
.
.
.
<script language="JavaScript">
<!--
function SubmitAddUser() {
    document.AddUserForm.submit();
}
.
.
.
//-->
</script>
<p>
<center>
.
.
.
```

```

<table width="100%" border=0 cellpadding=0 cellspacing=0>
  <tr>
    <td width="15%">&nbsp;</td>
    <td nowrap>
      <ul align="center" type="disc">
        <% if (accessProperties.getVisible(role,
          "NEWUSER_ID")) { %>
          <jsp:setProperty name="localebean" property="key"
            value="NEWUSER_ID"/>
          <li><span class="SmallLinkFont"><a href="java-
            script:SubmitAddUser()"><jsp:getProperty
              name="localebean" property="label"/></a>
            </span></li>
          <% } %>
        </ul>
      </td>
    </tr>
  </table>
</center>
</body>
</html>

```

Collect Additional Attributes

You can customize Registration Services to collect additional attributes.

To collect additional attributes

1. Extend the object class (if necessary).
2. Add the additional attributes to the properties files.
3. Add new fields to collect the new attributes in the collector pages.

Each of these steps is described in the following sections.

Extend the Object Class

The object class specifies the required and optional attributes that define an entry. An LDAP entry, such as a user profile, may belong to more than one object class. For example, the default object classes for a user in Registration Services are `top`, `person`, `organizationalperson`, `inetorgperson`. Each object class extends the attributes that a user's profile can contain.

Note: Each LDAP Directory includes a set of object classes. See your directory server's documentation for more information.

The user.properties file contains the object class attribute. You can add an additional value(s) to this attribute to extend the object class for users.

Note: The org.properties and group.properties files are used by DMS2. Modifying these files does not affect Registration Services.

To add values to the object class attribute, append the additional value(s) to the values paired with the OBJECTCLASS attribute, using a comma (,).

The following example includes the additional object class value pilotorganization to the object class attribute for organizations:

```
OBJECTCLASS=objectclass,top,organizationalunit,pilotorganization
```

After extending the object class, add the additional attributes that you want to collect from the new object class to the appropriate properties files, as described in the following section.

More information:

[Object Properties Files](#) (see page 965)

Add Attributes to Properties Files

For Registration Services to collect additional attributes, the attributes must be added to the appropriate properties files. There are three types of properties files to which the attribute must be added:

Object

Pairs identifiers with LDAP attribute names. These files determine which attributes make up an object profile.

Presentation

Pairs identifiers with administrative roles and specifies whether or not users with those roles have privileges to view the attribute

Locale

Pairs identifiers with text strings. The text strings are used in the displayed pages

Note: The identifier for the attribute must be the same in the object, presentation, and locale properties files.

More information:

[Properties Files](#) (see page 941)

Object Properties Files

To add an attribute to an object properties file, specify an identifier paired with an LDAP attribute name as follows:

IDENTIFIER=AttributeName

where:

- *IDENTIFIER* is an identifier that you select. It must match the identifier for the attribute in the presentation and locale properties files.
- *AttributeName* is the name of an LDAP attribute. The attribute must be included in one of the object classes for the type of object for which you are adding the attribute. See your LDAP directory server's documentation for a list of attributes.

Presentation Properties Files

To add an attribute to a presentation properties file, specify an identifier paired with administrative roles. The roles are associated with an R (for readable) for administrators who have the privilege to view the attribute. The attribute/value pair has the following format:

IDENTIFIER=Role1:R,Role2:R

- *IDENTIFIER* is an identifier that you select. It must match the identifier for the attribute in the object and locale properties files.
- *Role* is an administrative role.
- R indicates that the attribute is displayed for administrator's who have the role with which it is associated.

More information:

[Administrative Roles](#) (see page 45)

Locale Properties File

To add the attribute to the locale properties file, `dms_en_US.properties`, add an identifier paired with a text string. The text string will be used as the field name in the page used to collect the new attribute.

The attribute/value pair should have the following format:

IDENTIFIER=Text String

where *IDENTIFIER* is an identifier that you select. It must match the identifier for the attribute in the object and presentation properties files.

Add Fields to Collector Pages

Registration Services uses collector pages to enable users to add or modify a user's profile. The collector page references a profile page, which defines the presentation of user profiles by specifying the form elements, such as text fields and list boxes, that are displayed in the collector pages.

To collect additional user profile attributes, you must add additional fields to `profile_user.jsp`.

To add a new text field

1. Add the following JSP code, which determines whether or not users can view the field based on their administrative role:

```
<%  
bVisible = accessProperties.getVisible(role, "IDENTIFIER");  
if (bVisible==true) { %>
```

In this statement:

- `bVisible` is a boolean. If the value of `accessProperties.getVisible(role, "IDENTIFIER");` is true, Registration Services displays the field.
- `accessProperties.getVisible` tells the `AccessTableBean` to determine whether or not the field should be visible based on the entry in the presentation properties file.
- `role` is a parameter for the `getVisible` method. It identifies the administrator's role
- `IDENTIFIER` is the identifier that is specified in the presentation properties file for the attribute that you are collecting

2. Add the following JSP code, which determines the field label that is displayed in the collector JSP page:

```
<jsp:setProperty name="localebean" property="key"  
value="IDENTIFIER" />  
<jsp:getProperty name="localebean" property="label"/>
```

In these statements:

- `IDENTIFIER` is the identifier that is specified in the locale properties file for the attribute that you are collecting.
- `jsp:setProperty` and `jsp:getProperty` are the methods that call the locale bean, pass it the identifier, and retrieve the paired text string, which is displayed on the HTML form built by the JSP.

3. Add the HTML and JSP code that creates the input field:

```
<tr>  
<td>  
<%
```

```
String attributeValue = properties.getValue("IDENTIFIER");
if ( attributeValue != null ) {
%>
<td><input type="text" size="42" maxLength="40"
name="IDENTIFIER" value= "<%= attributeValue %>">
</td>
<% } else { %>
<td><input type="text" size="42" maxLength="40"
name="IDENTIFIER">
</td>
<% } %>
<td><input type="hidden" size="42" maxLength="40"
name="INFO_IDENTIFIER">
</td>
</tr>
<% } %>
```

In these statements:

- *IDENTIFIER* is the identifier that is specified in the object properties file for the attribute that you are collecting.
 - *attributeValue* holds the value returned by the `properties.getValue` method. *attribute* represents the name of the attribute that you are collecting. This attribute may have a value if an administrator is modifying an existing profile.
 - `if ( attributeValue != null )` tells Registration Services to display the value returned by `properties.getValue` if the value is not null. If the value is not null, the `<input>` statement that follows the `if` statement tells Registration Services to display the returned value.
 - `<% } else { %>` tells Registration Services what to display if the value returned by `properties.getValue` is null. In this case, Registration Services displays an empty text box for the attribute, as defined by the `<input>` statement following the `else` statement.
 - The third `<input>` statement holds a value if `properties.getValue` returns a value. This value is used as a flag for the `checkForEmptyValues` method. The `checkForEmptyValuesMethod` checks to see if the attribute has a value before and after a profile is modified. If the value exists before the modify action but not after, Registration Services removes the attribute from the profile.
4. Add the attribute identifier to the `checkForEmptyValues` method. This method is located in `newuserjs.jsp`, which contains Javascript for managing users.

These files are included by the `user_data_collector.jsp` page.

Add the attribute identifier, as follows:

- a. Locate the function declaration for `checkForEmptyValues`:
`function checkForEmptyValues()`
- b. Add the attribute identifier to the list of identifiers, as follows:
`listOfIds.add("IDENTIFIER");`

Example: Collecting the Drink Attribute

The following example describes the steps taken to collect users' favorite drinks and add that information to their profiles:

1. Add an additional value to the user object class attribute in the object properties file, `user.properties`.

The modified object class attribute is:

`OBJECTCLASS=objectclass,top,person,organizationalperson,inetorgperson,newpilotperson`

`newpilotperson` is an object class that is provided with iPlanet Directory Server. One of the attributes contained in this object class is `drink`, which is used to store a user's favorite beverage.

2. Add the drink attribute to the appropriate properties files, as described in the following table:

Properties File	Entry
<code>user.properties</code> object properties file	<code>DRINK_ID=drink</code>
<code>user.properties</code> presentation properties file	<code>DRINK_ID=Admin:R,OrgAdmin:R</code>
<code>dms_en_US.properties</code> locale properties file	<code>DRINK_ID=Favorite Drink</code>

3. Create a new field in profile_user.jsp:

[illegible]

4. Add the attribute to the `newuser.js.jsp` for the `checkForEmptyValues` method:

```
<%  
    // Collect all attributes that will be checked for empty values  
    Vector listOfIds = new Vector();  
    /**  
    listOfIds.add("PASSWORD_ID"); //never remove password!!  
    **/  
    listOfIds.add("DRINK ID");
```

The rendered HTML form resembles the following:

User Profile Information	
First Name	
Last Name *	
Title	
ZIP	
Country	U.S.A. ▼
Phone Number	
Fax Number	
Email Address	
Favorite Drink	

Localize Registration Services

You can localize Registration Services by modifying the following components:

- Locale properties file
- The FCC files used to collect user credentials.
- The inc_header.jsp and dmserrorpage.jsp files

Each form for user entry, such as user_data_collector.jsp, includes an SMLOCALE variable. The SMLOCALE variable identifies the properties file that Registration Services uses. For example:

```
<INPUT type=HIDDEN name="SMLOCALE" value="en_US">
```

In this code, the value that is paired with SMLOCALE tells Registration Services to use the sample file, dms_en_US.properties.

Note: The value for SMLOCALE is specified as “*lg-CN*” where *lg* is the letter abbreviation for language and *CN* is the two letter abbreviation for country.

dms_en_US.properties, which is the locale properties file that is installed with Registration Services, contains English text strings. To localize Registration Services, create a locale properties file that contains text strings in the desired language.

In addition to specifying the SMLOCALE variable, you may also want to modify the character encoding for the JSP pages. Character encoding is specified in inc_header.jsp and dmserrorpage.jsp.

More information:

[Configure Registration Services for Different Languages](#) (see page 971)

Configure Registration Services for Different Languages

An overview of the steps for localizing Registration Services is listed below.

To configure registration services for different languages

1. Create a new locale properties file based on the properties file installed with Registration Services.
2. Modify the FCC files to reference the correct locale properties file and character encoding.
3. Run the DMS configuration wizard.

Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

4. Modify the authentication schemes created by the configuration wizard to point to the correct FCC files, as described in [Configure Registration Services To Use The Appropriate FCC Files](#) (see page 973).
5. Modify the default character encoding (if necessary) in inc_header.jsp and dmserrorpages.jsp, as described in [Change the Character Encoding in The JSPs](#) (see page 974).

More information:

[Modify the Properties Files](#) (see page 972)

[Modify the FCC Files](#) (see page 973)

Modify the Properties Files

The first procedure in configuring registration services for different languages involves modifying the properties files.

To modify the properties files

1. In the locale directory rename `dms_en_US.properties`. The name of the properties file should have the following format:

`dms_lg_CN.properties`

where *lg* is the two character abbreviation for language and *CN* is the two character abbreviation for country. These values are case-sensitive.

This naming convention is important because the `SMLOCALE` variable in FCC files uses the language and country in the filename to identify the Properties file.

2. Open the properties file in a text editor.

The properties file contains name/value pairs that consist of an identifier, the equal sign (=), and a text string.

3. Customize the text strings following the equal sign (=).

For example, to change the field name “Last Name” to “Surname”, edit this line:

`LASTNAME_ID=Last Name`

so that it appears as follows:

`LASTNAME_ID=Surname`

4. Save the changes.
5. Stop and restart the servlet engine.

Modify the FCC Files

After modifying the properties files, you should modify the FCC files in order to configure registration services for different languages.

To modify the FCC files

1. Open one of the forms in the Forms directory that is provided for internationalization.

Registration Services installs the following forms, which can be modified:

login-dms-i.fcc

Lets a user login to Registration Services

loginandregister-dms-i.fcc

Lets a user login or register as a new user

loginandregisterwithforgottenpassword-dms-i.fcc

Lets a user select a question and answer that Registration Services uses when the user forgets his or her password. The user receives a temporary password by answering the question correctly when prompted.

2. In each of the FCC files, modify the value for SMLOCALE in the following line:

```
<INPUT type=HIDDEN name="SMLOCALE" value="en-US">
```

The value should have the following format:

lg_CN

where *lg* is the two character abbreviation for language and *CN* is the two character abbreviation for country. These values are case-sensitive.

In the example above, US is the abbreviation for the country, United States of America, and en is the abbreviation for the English language.

3. Specify a character encoding by modifying the following statement at the top of the FCC:

```
<!-- SiteMinder Encoding=UTF-8; -->
```

Configure Registration Services To Use The Appropriate FCC Files

The FCC file, which creates the initial user login screen, identifies which locale properties file Registration Services should use for text strings.

To localize Registration Services, you need to configure Registration Services to retrieve FCC files for the appropriate language from the Forms directory.

To configure registration services to use the appropriate FCC files

1. In the Policy Server User Interface, run the DMS configuration wizard to create a new Registration Space. Take note of the special identifier.

Note: If you do not want to create a new Registration Space, you can modify the authentication schemes for an existing environment.

Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your SiteMinder component.

2. In the System tab, click Authentication Schemes.

In the right pane, SiteMinder lists the available authentication schemes.

3. Double-click one of the authentication schemes that the DMS configuration wizard created.

The authentication schemes are:

DMSv2 0 Self Reg

DMSv2 0 Admin

DMSv2 0 Deny

DMSv2 special identifier Admin

4. In the Target field, change the default target to include the form that you want to use. The sample FCC files that are provided for internationalization end with the letter i.

For example, change

/siteminderagent/dmsforms/login-dms.fcc

to

/siteminderagent/dmsforms/login-dms-i.fcc

5. Click OK.
6. Repeat steps 3-5 until you have changed the Target field in the DMS authentication schemes to point to the forms that you want to use.

Change the Character Encoding in The JSPs

By default, the JSPs are encoded in ISO-8859-1 encoding. You can modify the encoding for these pages by changing the following files:

inc_header.jsp

This file is included by the JSPs that include character encoding.

dmserrorpage.jsp

This file is used to display error messages.

To change the character encoding

1. Navigate to the pages directory where Registration Services is installed—for example:
 - For Windows: C:\Program Files\Netegrity\SiteMinder Web Agent\Samples\dmspages
 - For UNIX: /export/smuser/netegrity/<webagent_installation>/samples/dmspages
2. Open inc_header.jsp in a text editor.
3. Change the encoding for JSPs in the following statement:


```
<%! String charset = "ISO-8859-1"; %>
```

For example, to specify UTF-8 encoding, modify the statement, as follows:

```
<%! String charset = "UTF-8"; %>
```
4. Change the encoding for HTML pages in the following statement:


```
<%@ page contentType="text/html; charset=ISO-8859-1" %>
```

For example, to specify UTF-8 encoding, modify the statement, as follows:

```
<%@ page contentType="text/html; charset=UTF-8" %>
```
5. Save inc_header.jsp.
6. Open dmserrorpage.jsp.
7. Repeat steps 3-4.
8. Save dmserrorpage.jsp.

Chapter 49: CA SSO/WAC Integration

This section contains the following topics:

[Overview](#) (see page 977)

[SiteMinder and CA SSO Integration Architectural Examples](#) (see page 978)

[Prerequisites Before Integrating SiteMinder and CA SSO](#) (see page 983)

[Configure Single Sign-On from SiteMinder to CA SSO](#) (see page 984)

[Configure Single Sign-On from CA SSO Client to SiteMinder](#) (see page 987)

[Configure Single Sign-On from CA SSO to SiteMinder](#) (see page 988)

[Configure an smetssocookie Web Agent Active Response Attribute](#) (see page 990)

[Configure an smauthetssso Custom Authentication Scheme](#) (see page 992)

Overview

SiteMinder provides single sign-on from SiteMinder to CA SSO environments. Users log into a SiteMinder or CA SSO environment, and once authenticated by SiteMinder, are authenticated for both environments. Authenticated users can access protected resources in either environment without having to re-enter credentials, as long as they are authorized. User authorization is based on the policies in effect within each environment.

When allowing users to access secure resources, SiteMinder and CA SSO each maintain user credentials in their own session stores. They also have their own proprietary session credentials that cannot be read by the other and, thus, user credentials are maintained separately. Since these credentials reside in different stores, to enable single sign-on, the SiteMinder Policy Server and CA SSO Policy Server must be part of the same cookie domain and share the same user or authentication store.

In this single sign-on configuration, the SiteMinder and CA SSO Policy Servers can be on the same or on different machines. SiteMinder can contain a Web Agent, Secure Proxy Server, or both. You use a Web Agent or Secure Proxy Server based on your own SiteMinder environment. CA SSO uses the Web Access Control (WAC) Web Agent, and you do not need to modify your current environment to enable single sign-on with SiteMinder.

Note: You should be familiar with both SiteMinder and CA SSO before configuring single sign-on between the products.

For a list of supported components, search for the SiteMinder r6.0 SP6 Platform Support matrix on the Technical Support web site at <http://ca.com/support>.

More information:

[SiteMinder and CA SSO Integration Architectural Examples](#) (see page 978)

SiteMinder and CA SSO Integration Architectural Examples

The following are three examples of single sign-on between SiteMinder and CA SSO environments:

1. A user authenticates to SiteMinder using a Web browser and then accesses an CA SSO-protected resource (see [Example 1: User Accesses SiteMinder-Protected Resource Before CA SSO](#) (see page 979)).
2. A user authenticates to CA SSO through a desktop CA SSO Client and then accesses a SiteMinder-protected resource using a Web browser (see [Example 2: Authenticated CA SSO Client User Accesses SiteMinder Resource](#) (see page 980)).
3. A user authenticates to CA SSO using a Web browser and then accesses a SiteMinder-protected resources (see [Example 3: User Accesses CA WAC-Protected Resource Before SiteMinder](#) (see page 982)).

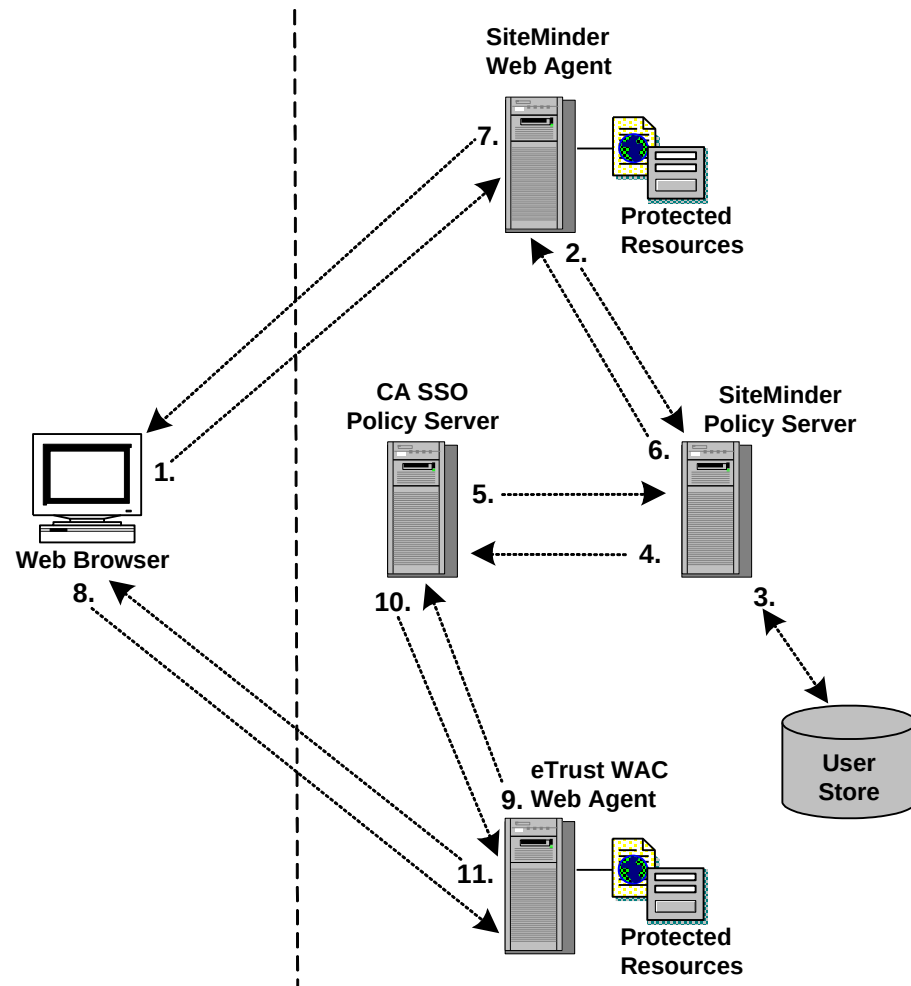
Note: In these examples, the SiteMinder and CA SSO Policy Servers authorization access steps to protected resources are omitted for clarity.

More information:

[Configure Single Sign-On from CA SSO Client to SiteMinder](#) (see page 987)

Example 1: User Accesses SiteMinder-Protected Resource Before CA SSO

The following example illustrates a user accessing SiteMinder-protected resource before a WAC-protected resource:

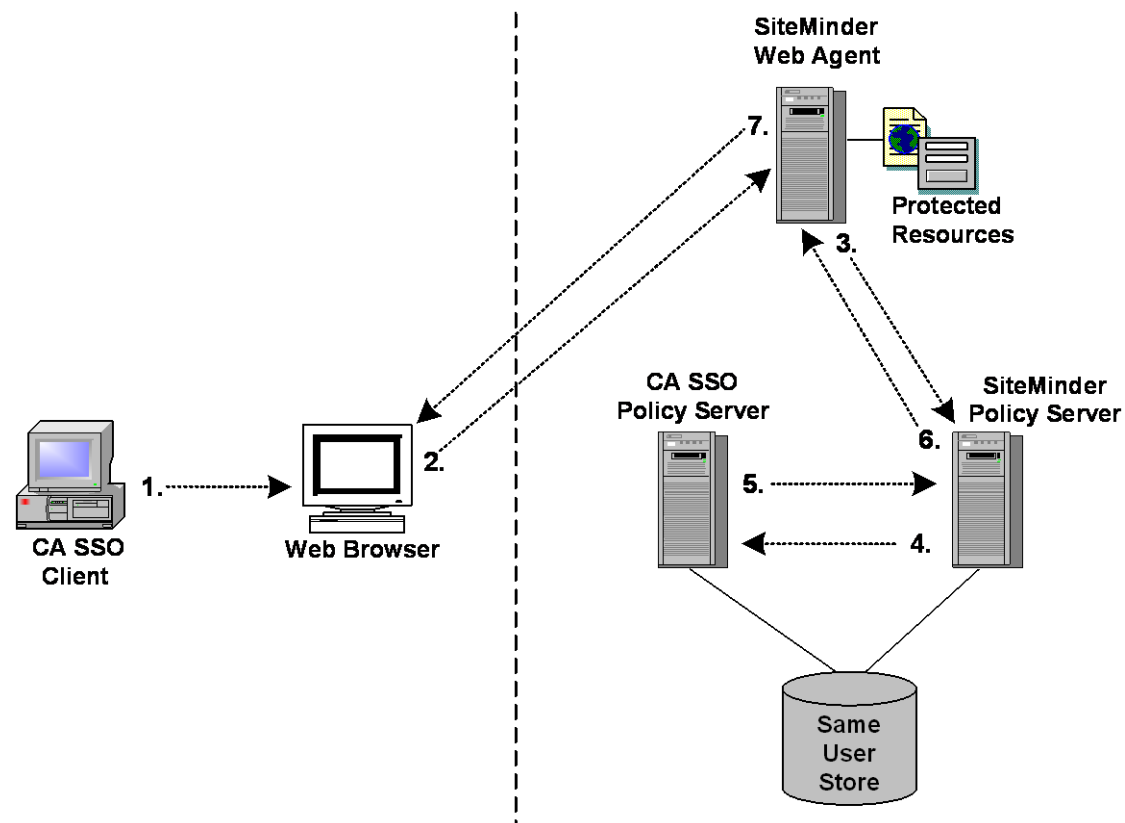


1. The user tries to access a SiteMinder-protected resource and the SiteMinder Web Agent/Secure Proxy Server intercepts the request. The user provides the Agent/SPS with authentication credentials.
2. The Web Agent/Secure Proxy Server forwards the credentials to the SiteMinder Policy Server for validation.
3. The SiteMinder Policy Server makes sure that the user's credentials are valid in the user store.
4. After successful authentication, the SiteMinder Policy Server requests that the CA SSO Policy Server to issue and return an CA SSO cookie for the SiteMinder user.
5. The CA SSO Policy Server validates the user and forwards the user's CA SSO Web authentication credentials to the SiteMinder Policy Server.

6. The SiteMinder Policy Server forwards the CA SSO Web authentication credentials to the SiteMinder Web Agent/Secure Proxy Server.
7. The SiteMinder Web Agent/Secure Proxy Server sets the CA SSO Web authentication and SiteMinder cookies in the user's browser and returns the resource to the user.
8. The user tries to access a CA SSO resource and the CA WAC Web Agent intercepts the request.
9. The CA WAC Web Agent validates the user's CA SSO Web authentication cookie credentials with the CA SSO Policy Server.
10. The CA SSO Policy Server tells the CA WAC Web Agent that the user has valid credentials.
11. The CA WAC Web Agent allows the user to access the SSO-protected resource.

Example 2: Authenticated CA SSO Client User Accesses SiteMinder Resource

The following example illustrates an authenticated CA SSO client user accessing a SiteMinder protected resource:

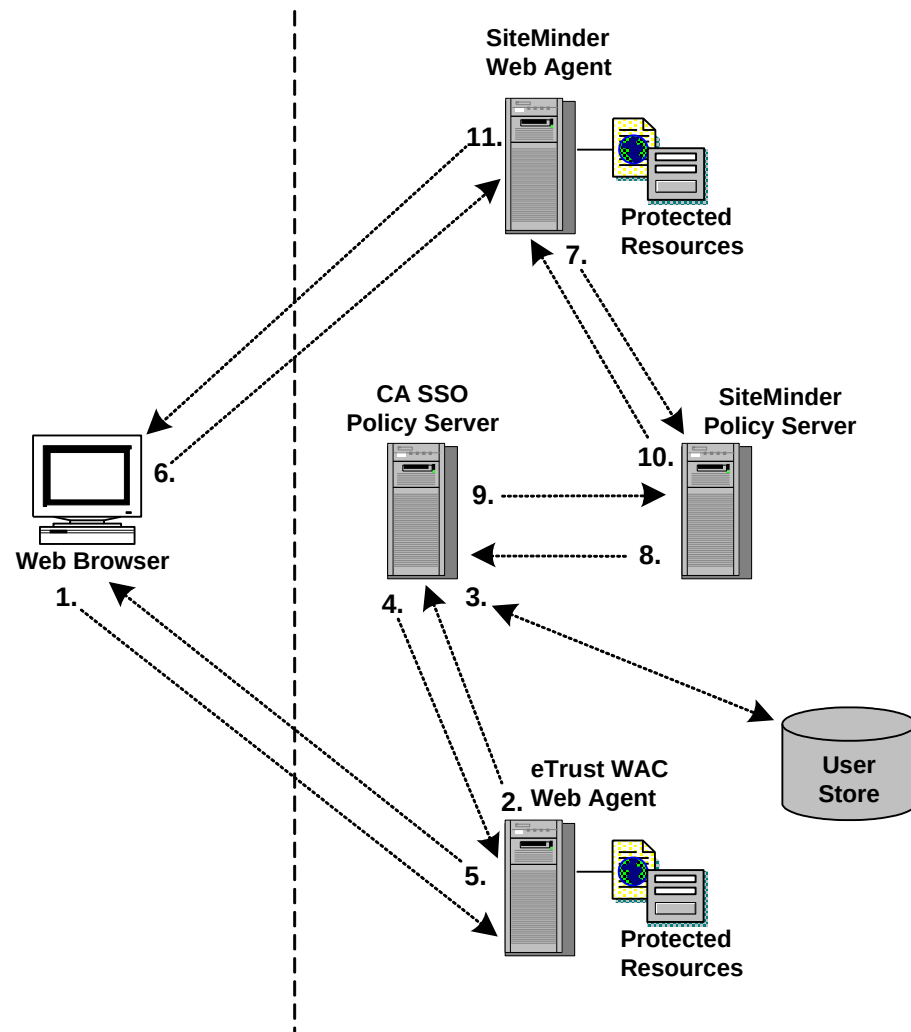


1. An authenticated CA SSO Client user launches a Web browser. While this is happening, the CA SSO Client places an CA SSO Web authentication cookie into the browser.
2. The user tries to access a SiteMinder-protected resource using the Web browser and the request is intercepted by the SiteMinder Web Agent/Secure Proxy Server.
3. The SiteMinder Web Agent/Secure Proxy Server forwards the CA SSO Web authentication cookie to the SiteMinder Policy Server.
4. The SiteMinder Policy Server forwards the CA SSO Web authentication cookie to the CA SSO Policy Server.
5. The CA SSO Policy Server validates the CA SSO Web authentication cookie and returns the user name to the SiteMinder Policy Server.
6. The SiteMinder Policy Server verifies the returned user name in the SiteMinder user store, then issues a corresponding SiteMinder cookie and returns it to the SiteMinder Web Agent/Secure Proxy Server.
7. The SiteMinder Web Agent/Secure Proxy Server returns the requested resource to the user, who now has the authentication cookie credentials necessary for SiteMinder and CA SSO environments.

Example 3: User Accesses WAC-Protected Resource Before SiteMinder

The following example illustrates a user accessing a WAC-protected resource before SiteMinder.

Note: The example assumes the environment is using a IIS6 WAC Agent. An IIS6 WAC Agent is the only platform that the following example supports.



1. The user tries to access a CA SSO-protected resource and the WAC Web Agent intercepts the request. The user provides the Agent with authentication credentials.
2. The Web Agent forwards the credentials to the CA SSO Policy Server for validation.
3. The CA SSO Policy Server makes sure that the user's credentials are valid in the user store.
4. The CA SSO Policy Server forwards the user's CA SSO Web credentials to the CA WAC Web Agent.

5. The CA WAC Web Agent sets the user's CA SSO Web authentication cookie in the Web browser.
6. The user tries to access a SiteMinder-protected resource and the SiteMinder Web Agent/Secure Proxy Server intercepts the request.
7. The SiteMinder Web Agent/Secure Proxy Server forwards the user's CA SSO Web authentication credentials to the SiteMinder Policy Server.
8. The SiteMinder Policy Server forwards the user's CA SSO Web authentication credentials to the CA SSO Policy Server.
9. The CA SSO Policy Server validates the user's CA SSO Web authentication credentials and forwards the user name back to the SiteMinder Policy Server.
10. The SiteMinder Policy Server verifies the returned user name in the SiteMinder user store, then issues a corresponding SiteMinder cookie and returns it to the SiteMinder Web Agent/Secure Proxy Server.
11. The SiteMinder Web Agent/Secure Proxy Server sets the SiteMinder cookies in the user's browser and allows the user to access the requested resource.

Prerequisites Before Integrating SiteMinder and CA SSO

Before configuring single sign-on from SiteMinder to CA SSO, CA SSO Client to SiteMinder, or CA SSO to SiteMinder, do the following:

1. Install and configure CA SSO.

Note: When installing the CA SSO Policy Server, gather the following:

- An SSO administrator name and password. The SiteMinder Policy Server uses the administrator name and password when authenticating to the CA SSO Policy Server through the smauthetsso authentication scheme.
- The SSO ticket encryption key. The SiteMinder Policy Server smetssocookie active response requires this value.

2. Upgrade the Policy Server to 6.0 SP 4.

For instructions, see the *SiteMinder Upgrade Guide*.

Note: When enabling single sign-on, be sure that the SiteMinder Policy Server and CA SSO Policy Server are part of the same cookie domain and share the same user or authentication store.

3. Upgrade the Policy Server policy store to 6.0 SP 4.

For instructions, see the *SiteMinder Upgrade Guide*.

4. Install the Web Agent or the 6.0 Secure Proxy Server.

More information:

[Configure an smauthetsso Custom Authentication Scheme](#) (see page 992)

[Configure an smetssocookie Web Agent Active Response Attribute](#) (see page 990)

Configure Single Sign-On from SiteMinder to CA SSO

SiteMinder provides single sign-on from SiteMinder to CA SSO environments.

To enable single sign-on from SiteMinder to CA SSO

Web Agent or Secure Proxy Server Configuration Steps

Enable the SiteMinder SSO Plug-in installed with the Web Agent or Secure Proxy Server:

For the 6.x QMR 4 IIS 6.0 or Apache 2.0 Web Agent

- Remove the comment (#) character from the following line in the WebAgent.conf file:

#LoadPlugin=<Path to eTSSOPlugin.dll or libetssoplugin.so>
- To locate the WebAgent.conf for IIS 6.0/Apache 2.0, see the *SiteMinder Web Agent Guide*.
- For instructions on enabling/disabling the plug-in, see the *SiteMinder Web Agent Guide*.

Note: After you modify the WebAgent.conf file, restart the Web server if it is running so that the new configuration settings take effect.

For the 6.0 Secure Proxy Server

- Remove the comment (#) character from the following line in the WebAgent.conf file, located in
<SPS_install_dir>\proxy-engine\conf\defaultagent\WebAgent.conf:

#LoadPlugin=<Path to eTSSOPlugin.dll or libetssoplugin.so>

Note: After you modify the WebAgent.conf file, restart the Secure Proxy Server if it is running so that the new configuration settings take effect.

CA WAC Web Agent Verification Steps

1. Configure the domain in the CA WAC Web Agent's webagent.ini file by setting the following parameter:

DomainCookie=<domain>

where <domain> is the same domain (for example, test.com) for the CA SSO and SiteMinder Web Agents.

The file is installed in the following location on the CA WAC Web Agent machine:

C:\Program Files\CA\WebAccessControl\WebAgent\webagent.ini

2. Verify the following Web server and the authentication method settings in the webagent.ini file:
 - The "Authentication methods" and "The default authentication method" parameters should be configured as SSO.
 - The WebServerName, PrimaryWebServerName, AgentName, NTLMPath and Secure should point to the machine where CA SSO Web Access Control is installed.
 - The ServerName attribute should point to the IP Address of the machine where the CA SSO Policy Server is installed.

Note: For more information about configuring the CA WAC Web Agent, see the CA WAC documentation.

CA SSO Policy Manager Verification Steps

1. Ensure that the SiteMinder and CA SSO Policy Servers use the same user or authentication store.
2. Make sure you have the following:
 - An SSO administrator name and password. The SiteMinder Policy Server uses the administrator name and password when authenticating to the CA SSO Policy Server through the smauthetsso authentication scheme.
 - The SSO ticket encryption key, as it is needed by the SiteMinder Policy Server's smetssocookie active response.

Note: For more information about configuring the Policy Manager, see the CA SSO documentation.

Policy Server Configuration Steps

1. Create a Web Agent, Agent Configuration Object, and Host Configuration Object using the Policy Server User Interface. For more information, see the *Policy Server Installation Guide* and *Web Agent Installation Guide*.
2. Configure the SiteMinder and CA SSO Policy Servers to use the same user or authentication store.

For SiteMinder user store configuration instructions, see this guide.
For the CA SSO authentication store, see the CA SSO documentation.
3. Configure an smetssocookie (certificate) custom active response.
4. Create a domain, realm, and rules using the Policy Server User Interface to protect any resource with the SiteMinder Web Agent.

Note: When creating the rules, append the smetssocookie custom active response to them.

Overall Verification Steps

1. Configure the user with credentials to access resources protected by the SiteMinder Web Agent and the CA WAC Web Agent.
2. Restart the SiteMinder Policy Server and Web server hosting the Policy Server User Interface.
3. Access the resource protected by the SiteMinder Web Agent and provide this Web Agent with the appropriate user credentials.
4. After gaining access to this resource, in the same browser session, request a resource protected by the CA WAC Web Agent.

You should gain access to this resource without being prompted for credentials.

More information:

[Configure an smauthetsso Custom Authentication Scheme](#) (see page 992)

[Configure an smetssocookie Web Agent Active Response Attribute](#) (see page 990)

[Domains](#) (see page 307)

[Grouping Resources in Realms](#) (see page 315)

[Rules](#) (see page 327)

[Configure a Rule for Web Agent Actions](#) (see page 334)

Configure Single Sign-On from CA SSO Client to SiteMinder

Site Minder provides single sign-on from the CA SSO Client to SiteMinder.

To enable single sign-on from a CA SSO Client to SiteMinder:

Policy Server Configuration Steps

1. Configure the smauthetsso custom authentication scheme using the Policy Server User Interface.
2. Create a domain, realm, and rules using the Policy Server User Interface to protect any resource with the SiteMinder Web Agent.
3. Configure the smauthetsso custom authentication scheme to protect a resource.
4. Create a policy that grants access to the protected resource to users who already have access the browser protected by the CA SSO Client.

CA SSO Client Verification Steps

In the CA SSO Client's SsoCInt.ini file, set the following:

DomainNameServer=<eSSO_WA_FQDN> <SM_WA_FQDN>

<eSSO_WA_FQDN>

(Optional) Specifies the CA WAC Web Agent's fully qualified domain name

<SM_WA_FQDN>

Specifies the SiteMinder Web Agent's fully qualified domain name.

Example:

DomainNameServer=http://abc.mycompany.com http://xyz.mycompany.com

The SsoCInt.ini file is installed in the following location on the CA SSO Client machine:

C:\Program Files\CA\CA SSO\Client

Note: For more information about configuring the CA SSO Client, see the CA SSO documentation.

Overall Verification Steps

1. Restart the CA SSO Client, SiteMinder Policy Server, and Web server hosting the Policy Server User Interface.
2. Access the protected browser through the SSO Client and enter the URL of the resource protected by the SiteMinder Policy Server.

You should be able to access the resource without being rechallenged by SiteMinder.

More information:

[Configure an smauthetsso Custom Authentication Scheme](#) (see page 992)

[Domains](#) (see page 307)

[Grouping Resources in Realms](#) (see page 315)

[Rules](#) (see page 327)

Configure Single Sign-On from CA SSO to SiteMinder

SiteMinder provides single sign-on from CA SSO to SiteMinder.

To enable single sign-on from CA SSO to SiteMinder

Policy Server Configuration Steps

1. Configure the smauthetsso custom authentication scheme using the Policy Server User Interface.
2. Create a domain, realm, and rules using the Policy Server User Interface to protect any resource with the SiteMinder Web Agent.

For more information, see Domains, Grouping Resources in Realms, or Rules.

3. Configure the smauthetsso custom authentication scheme to protect a resource.

CA WAC Web Agent Verification Steps

1. Configure the domain in the CA WAC Web Agent's webagent.ini file by setting the following parameter:

DomainCookie=<domain>

where <domain> is the same domain (for example, test.com) for the CA SSO and SiteMinder Web Agents.

The file is installed in the following location on the CA WAC Web Agent machine:

C:\Program Files\CA\CAWebAccessControl\WebAgent\webagent.ini

2. Verify the following Web server and authentication method settings in the webagent.ini file:
 - The "Authentication methods" and "The default authentication method" parameters should be configured as SSO.
 - The WebServerName, PrimaryWebServerName, AgentName, NTLMPath and Secure should point to the machine where CA SSO Web Access Control is installed.
 - The ServerName attribute should point to the IP Address of the machine where the CA SSO Policy Server is installed.
 - For more information about configuring the CA WAC Web Agent, see the CA SSO documentation.

Note: For more information about configuring the CA WAC Web Agent, see the CA WAC documentation.

Web Agent or Secure Proxy Server Configuration Steps

1. Enable the SSO plug-in installed with the Web Agent or Secure Proxy Server, so that SSO Client cookies can be authenticated, by removing the comment character (#) from the following line in the WebAgent.conf file:

```
#LoadPlugin=path_to_eTSSOPlugin.dll | path_to_libetssoplugin.so
```

Note: The WebAgent.conf file is located as follows:

6.x QMR 4 IIS 6.0 or Apache 2.0 Web Agent

See the Web Agent Configuration Guide.

6.0 Secure Proxy Server

SPS_install_dir\proxy-engine\conf\defaultagent\

SPS_install_dir

Secure Proxy Server installation directory

2. Restart the Policy Server.

Overall Verification Steps

1. Restart the CA WAC Web Agent, SiteMinder Policy Server, and Web server hosting the Policy Server User Interface.
2. Access a resource protected by the CA WAC Web Agent and provide valid credentials.
3. Access a resource protected by the SiteMinder Web Agent in the same browser.

You should be able to access the resource without being rechallenged by SiteMinder.

More information:

[Configure an smauthetsso Custom Authentication Scheme](#) (see page 992)

Configure an smetssocookie Web Agent Active Response Attribute

The smetssocookie Web Agent active response generates an SSO cookie to a Web browser to allow a SiteMinder-authenticated user to access CA WAC or SSO protected content without having to reauthenticate.

To configure an smetssocookie Web Agent response attribute

1. In the SiteMinder Response dialog, make sure the SiteMinder radio button is selected in the Agent Type group box and Web Agent is selected in the drop-down list.

2. Click Create.

The SiteMinder Response Attribute Editor dialog opens.

3. From the Attribute drop down list, select the WebAgent-HTTP-Cookie-Variable response attribute for the Web Agent response.

4. In the Attribute Setup tab, select the Active Response radio button.

Your selection from the Attribute drop-down list and the response attribute type radio button you select determine the available fields in the Attribute Fields group box.

5. In the Attribute Fields group box, specify the following:

Cookie Name

Defines the name for the SSO cookie. Required Value: SSOTK

Library Name

Defines the library name for the smetssocookie. Required Value: smetssocookie

Function Name

Defines the function name for the GenEtssCookie. Required value: GenEtssCookie

The capitalization in the function name is significant and must match GenEtssCookie.

Parameters

Defines an ordered set of tokens separated by semicolons in the following format:

<CA_PS_Host_Name>;<SSO_Auth_Host>;<SSO_AuthMethod>;<EncryptionKey>

Note: For improved legibility, you can enter a space before or after any token.

<CA_PS_Host_Name>

Defines the host name of the CA SSO Policy Server.

<SSO_Auth_Host>

Defines the SSO authentication host name in the CA Policy Manager. You specify this host name by going to Web Access Control Resources, Configuration Resources, Authentication Host.

Default: SSO_Authhost

SSO_Authhost is the default value for the CA SSO configuration that you must enter here.

<SSO_AuthMethod>

Defines the SSO authentication method. Required value: SSO

<EncryptionKey>

Defines the ticket encryption key for the SSO authentication host name in the CA Policy Manager. For more information, see Properties, Authentication Information, Advanced in CA Policy Manager for the host.

6. Click OK to save your changes.
7. Attach this active response to an OnAuthAccept rule.

More information:

[Response Attribute Editor Dialog](#) (see page 707)

Configure an smauthetsso Custom Authentication Scheme

The CA SSO SiteMinder (smauthetsso) authentication scheme allows the SiteMinder Policy Server to validate CA SSO authentication credentials so that a user already authenticated in a CA SSO/WAC environment does not need to re-authenticate to SiteMinder. This custom authentication scheme accepts a CA SSO Cookie as a login credential; has it validated by an CA SSO Policy Server; extracts the user name from it; and verifies that the name is present in the SiteMinder user store. You can set this authentication scheme in a cookie, cookieorbasic, or cookieorforms mode.

You can configure one CA SSO Policy Server to failover to another CA SSO Policy Server when it fails for some reason. To configure failover, specify a comma-separated list of CA SSO Policy Servers in the parameter field on the Scheme Setup tab on the Authentication Scheme Dialog.

To configure an smauthetsso custom authentication scheme

1. Open the Authentication Scheme Dialog.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.

2. Complete the fields described in [Authentication Scheme Dialog Fields and Controls](#) (see page 609).

From the Authentication Scheme Type drop-down list, select Custom Template.

3. In the Scheme Setup tab, specify the following:

Library

Defines the name of the smauthetsso custom authentication scheme.

Required value: smauthetsso

Secret

Defines the Password of the CA SSO Policy Server administrator that you specify in the Parameter field.

Confirm Secret

Defines the administrator password once again.

Parameter

Defines an ordered set of tokens separated by semicolons in the following format:

<Mode> [; <Target>] ; <AdminID> ; <CAPS_Host_A> [, <CAPS_Host_B>] ...

Examples:

cookie; SMPS_sso; myserver.myco.com

cookieorforms; /siteminderagent/forms/login.fcc; SMPS_sso;
myserver.myco.com

Note: For improved legibility, you can enter a space before or after any token.

<Mode>

Determines the type of credentials that the authentication scheme accepts: cookie, cookieorbasic, or cookieorforms.

cookie

Only CA SSO cookies are acceptable.

cookieorbasic

Determines the login name and password to use with Basic authentication when you do not provide a CA SSO Cookie or the smauthetsso custom authentication scheme fails for any reason.

cookieorforms

Determines the login name and password to use with Forms authentication when you do not provide a CA SSO Cookie or the smauthetsso custom authentication scheme fails for any reason.

<Target>

(Optional) Defines the pathname of the .fcc file as is used by the HTML Forms authentication scheme.

Note: The <Target> is required for the cookieorforms mode and is not allowed for the cookie or cookieorbasic modes.

<AdminID>

Defines the user name of the administrator for the CA SSO Policy Server. You specify this administrator's password in the Secret and Confirm Secret fields. When authenticating to the CA SSO Policy Server, SiteMinder uses this administrator's user name and password to request validation of CA SSO cookies.

<CAPS_Host_A>

Specifies the name of the host, where the CA SSO Policy Server resides.

[, <CAPS_Host_B>] ...

(Optional) Specifies a comma-separated list of host names, one for each host on which the failover CA SSO Policy Servers reside.

4. Click OK to save your changes and close the Authentication Scheme Dialog.

You can assign the authentication scheme to a realm now.

More information:

[Navigate to the Authentication Scheme Dialog](#) (see page 608)

[HTML Forms Authentication Schemes](#) (see page 244)

[Authentication Scheme Dialog—Custom Template—Scheme Setup Tab](#) (see page 644)
[Realms](#) (see page 315)

Appendix A: SiteMinder Kerberos Authentication

This section contains the following topics:

[Kerberos Overview](#) (see page 995)

[How To Configure SiteMinder Kerberos Authentication](#) (see page 996)

[Kerberos Configuration Examples](#) (see page 1002)

[Verify that a Resource is Protected](#) (see page 1010)

[Troubleshooting SiteMinder Kerberos Authentication](#) (see page 1011)

Kerberos Overview

Kerberos is a standard protocol, designed at MIT, to provide a means of authentication between a client and a server on an open network. The Kerberos protocol protects messages from eavesdropping and replay attacks. Kerberos uses shared secrets, symmetric keys, and Kerberos services. Microsoft Windows operating environments use Kerberos V5 as the default authentication package. Solaris 10 also includes Kerberos V5.

In a Kerberos environment, user accounts and service accounts are named principals. Kerberos uses a trusted third party (the Key Distribution Center, or KDC) to mediate message exchanges between principals. The purpose of the Key Distribution Center is to reduce the risks inherent in exchanging keys.

Kerberos authentication is based on messages that request and deliver tickets. The Key Distribution Center processes two types of tickets:

- Ticket-Granting Ticket (TGT) — used internally by the KDC to transport a requestor's credentials to the ticket-granting service (TGS).
- Session Ticket — used by the ticket-granting service (TGS) to transport the requestor's credentials to the target server or service.

Kerberos uses keytab files for logging in to the KDC. Keytab files consist of pairs of Kerberos principals and encrypted keys derived from a Kerberos password.

The Kerberos protocol message exchange can be summarized in a simplified way as follows:

1. When a user logs in, the client contacts the KDC Authentication Service, requesting a short-lived message (the ticket-granting ticket) containing the user identity information.
2. The KDC authentication service generates the TGT and creates a session key that the client can use to encrypt communication with the ticket-granting service.

3. When a user requests access to local or network resources, the client presents the ticket-granting ticket (TGT), an authenticator, and the Service Principal Name (SPN) of the target server to the KDC.
4. The ticket-granting service examines the ticket-granting ticket and the authenticator. If these credentials are acceptable, the ticket-granting service creates a service ticket, which includes the user identity information copied from the TGT. The service ticket is sent back to the client.

Note: The ticket-granting service cannot determine whether the user is granted access to the target resource. The ticket-granting service only authenticates the user and returns the session ticket.

5. After the client has the session ticket, the client sends the session ticket and a new authenticator to the target server, requesting access to a resource.
6. The server decrypts the ticket, validates the authenticator, and grants the user access to the resource.

How To Configure SiteMinder Kerberos Authentication

Kerberos authentication supports various configuration scenarios, depending on the host environments of the client and server. Although each scenario is slightly different, implementing Kerberos authentication in a SiteMinder environment requires the following administrator tasks:

- Configuration at the domain controller for the Kerberos Key Distribution Center (KDC)
- Configuration at the Policy Server
- Configuration on the SiteMinder administrative UI of a custom authentication scheme
- Configuration at the web server
- Configuration at the Windows workstation

Kerberos KDC Configuration at the Domain Controller

When using Kerberos, the domain controller is the key distribution center (KDC) for the Kerberos Realm. In a pure Windows environment, a Kerberos Realm is equivalent to a Windows Domain. The domain controller host provides storage for the user, service accounts, credentials, the Kerberos ticketing services, and Windows Domain services.

A keytab file is required for Kerberos authentication, which lets users authenticate with the KDC without being prompted for a password. The keytab file is created with the ktpass utility. The ktpass command tool utility is a Windows support tool. The ktadd utility is the equivalent on UNIX.

KDC configuration for SiteMinder on the domain controller host (Windows or UNIX) follows this general sequence:

1. Create a user account. This account is for logging in to the workstation.
2. Create a service account for the web server for logging in to the web server host.
3. Create a service account for the Policy Server for logging in to the Policy Server host.
4. Associate the web server account with a web server principal name.
5. Create a keytab file, which is transferred to the web server host.
6. Associate the Policy Server account with a Policy Server principal name.
7. Create another keytab file, which is transferred to the Policy Server host.
8. Specify that the web server and Policy Server accounts are Trusted for Delegation.

Important! For any service to use Kerberos protocol, be sure to create the Service Principal Name (SPN) in a standard format, that is, service/fqdn_host@REALM_NAME.

More information:

[KDC Configuration on Windows 2003 Example](#) (see page 1002)

[KDC Configuration on UNIX Example](#) (see page 1005)

Kerberos Authentication Configuration at the Policy Server

In addition to the standard Policy Server configuration, Kerberos authentication requires the following steps:

- Add these three parameters to the Agent Configuration Object:

Parameter	Value
KCCExt	.kcc
HttpServicePrincipal	Specifies the web server principal name. Example: HTTP/win2k3iis6.test.com@TEST.COM
SmpsServicePrincipal	Specifies the Policy Server principal name. Example: smps/win2kps.test.com@TEST.COM

- Configure a Kerberos configuration file (krb5.ini) and place it in the system root path on Windows and in /etc/krb5/ on UNIX.
- Deploy the keytab file created on the KDC that contains the Policy Server principal credentials to a secure location on the Policy Server.

Important! If the Policy Server is installed on Windows and the KDC is deployed on UNIX, be sure to perform the additional configuration on the Policy Server host using the [Ksetup utility](#) (see page 1001).

More information:

[Kerberos Configuration at the Policy Server on Windows Example](#) (see page 1006)

[Kerberos Configuration at the Policy Server on UNIX Example](#) (see page 1008)

Kerberos Authentication Configuration at the Web Server

Configuring a Windows or UNIX web server to support Kerberos authentication follows these general steps:

1. Install a SiteMinder Web Agent with SiteMinder Kerberos Authentication Scheme support.
2. Register a trusted host with the Policy Server and configure the Web Agent.
3. Configure a Kerberos configuration file (krb5.ini):
 - Configure the KDC for the Kerberos realm (domain).
 - Configure krb5.ini to use the keytab file containing the credentials of the web server principal.
 - Place krb5.ini in the system root path on Windows and in /etc/krb5/ on UNIX.
4. Deploy the keytab file (created on the KDC) containing the web server credentials to a secure location on the web server.

Important! If the web server is installed on Windows and the KDC is deployed on UNIX, be sure to perform additional configuration on the web server using the [Ksetup utility](#) (see page 1001).

Kerberos Authentication Configuration at the Windows Workstation

To support Kerberos authentication, several Internet Explorer settings are required, and the workstation host is added to the KDC domain.

Important! If the KDC is deployed on UNIX, be sure to perform the additional required configuration on the workstation using the [Ksetup utility](#) (see page 1001).

To configure the Windows workstation to support Kerberos authentication

1. Add the host for the Windows workstation to the KDC domain.
2. Log in to the host using user account created on the KDC.
3. Configure Internet Explorer to pass credentials automatically:
 1. Initiate an instance of the IE web browser.
 2. Select the Internet options menu.
 3. Select the Security tab.
 4. Select Local intranet tab.
 5. Click Sites and select all three checkboxes.
 6. Select the Advanced tab and add `http://*.domain.com` to local intranet zone.
 7. Select the Custom level tab under security settings and select Automatic logon only in intranet zone under the User Authentication tab.
 8. Select the Advanced tab from Internet options and select the Enable Integrated Windows authentication (requires restart) option.
 9. Close the browser.

Configure a Kerberos Authentication Scheme

A custom authentication scheme is required to support Kerberos authentication in the SiteMinder environment.

To configure a Kerberos authentication scheme

1. Log in to the Policy Server User Interface.

Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.
2. Select the System tab.

A list of system-related objects appears.

3. Right-click Authentication Schemes and click Create Authentication Scheme.

The Authentication Scheme Properties dialog appears.

4. Select Custom Template from the Authentication Scheme Type list.

Custom Template settings appear.

5. Enter **smauthkerberos** in the Library field.

6. Enter the following values in the Parameter field. Enter the values in the order listed below delimited by a semicolon:

- a. The name of the machine hosting web server and target fields
- b. The Policy Server principal name from the Kerberos domain
- c. The mapping between user principal and the user store search filter

LDAP Example 1:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;(uid=%{UID})`

LDAP Example 2:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;(uid=%{UID})`

AD Example 1:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;(cn=%{UID})`

AD Example 2:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;(cn=%{UID})`

ODBC Example 1:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;%{UID}`

ODBC Example 2:

`http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpps/win2kps.test.com@TEST.COM;%{UID}`

7. Click OK.

The Kerberos Authentication scheme is saved and appears in the Authentication Scheme List.

Associate this authentication scheme with any realm whose protected resources use Kerberos authentication.

Configure Kerberos External Realm on Windows Host

For the Windows workstation to use a Kerberos KDC deployed on UNIX, you must configure both the Kerberos KDC server and the workstation.

In the Kerberos realm, create a host principal for the Windows host. Use the following command:

```
kadmin.local: addprinc host/machine-name.dns-domain_name.
```

For example, if the Windows workstation name is W2KW and the Kerberos realm name is EXAMPLE.COM, the principal name is host/w2kw.example.com.

Because a Kerberos realm is not a Windows domain, the KDC operating environment must be configured as a member of a workgroup, which happens automatically when you follow this process:

1. Remove the host from the Windows domain.
2. Add the test user, for example, testkrb, to the local user database.
3. Add the Kerberos Realm:

```
ksetup /SetRealm EXAMPLE.COM
```

4. Restart the host.

5. Add the KDC :

```
ksetup /addkdc EXAMPLE.COM rhasmit
```

6. Set a new password:

```
ksetup /setmachpassword password
```

Note: The password used here is same as the one used while creating the host principal account in the MIT KDC.

7. Restart the host.

Note: Whenever changes are made to the external KDC and realm configuration, a restart is required.

8. Set the Realm Flag

```
ksetup /SetRealmFlags EXAMPLE.COM delegate
```

9. Run AddKpasswd:

```
ksetup /AddKpasswd EXAMPLE.COM rhasmit
```

10. Use Ksetup to configure single sign on to local workstation accounts by defining the account mappings between the Windows host accounts to Kerberos principals. For example:

```
ksetup /mapuser testkrb@EXAMPLE.COM testkrb  
ksetup /mapuser * *
```

The second command maps clients to local accounts of the same name. Use Ksetup with no arguments to see the current settings.

Kerberos Configuration Examples

The configurations that follow include examples of the specifics, such as keytab file creation, required to implement Kerberos authentication in a SiteMinder environment. Note that additional configuration is required when the KDC is deployed in a UNIX operating environment and the Policy Server, or web server, or workstation is in a Windows operating environment.

KDC Configuration on Windows 2003 Example

The steps listed following exemplify how to configure a Windows 2003 domain controller to support SiteMinder Kerberos authentication.

1. Promote a Windows 2003 SP 1 Server to a domain controller (named test.com in this example) using Windows dcpromo utility.
2. Raise the domain functional level:
 1. Open the Active Directory users and computers dialog from Administrative tools.
 2. Right-click the test.com drop-down on the left side of dialog.
 3. Click Raise domain functional level.
 4. Raise the domain functional level of Active directory to Windows Server 2003.
Important! This step is irreversible.
3. Create a user account (for example, testkrb). Provide a password for this account. Clear the option, User must change password at next logon. Add this account to the domain administrators group so that the user has permissions to login. The Windows workstation uses this account to log in to test.com.
4. Create a service account for the web server (for example, wasrvwin2k3iis6). Create a password for this account. Clear the option, User must change password at next logon. Add this account to the domain administrators group so that the user has permissions to login. The Windows 2003 IIS web server host (win2k3iis6) uses this account to log in to test.com.
5. Create a service account for the Policy Server (polsrvwin2kps). Provide a password for this account. Clear the option, User must change password at next logon. Add this account to the domain administrators group so that the user has permissions to login. The Win2k3 Policy Server host (win2kps) uses this account to log in to test.com.
6. Join the web server (win2k3iis6) and the Policy Server (win2kps) hosts to the test.com domain using their service accounts created in Steps 4 and 5.

7. Associate the web server account (wasrvwin2k3iis6) with a web server principal name (HTTP/win2k3iis6.test.com@TEST.COM) and create a keytab file using the Ktpass utility. The syntax differs depending on whether the Policy Server is on Windows or on UNIX.

Note: The Ktpass command tool utility is a Windows support tool. You can install it from MSDN download or an installation CD. Always verify the version of support tools. The default encryption type must always be RC4-HMAC. The encryption type can be confirmed by running ktpass /? at the command prompt.

When the Policy Server is on Windows:

```
ktpass -out c:\wasrvwin2k3iis6.keytab -princ HTTP/win2k3iis6.test.com@TEST.COM  
-ptype KRB5_NT_PRINCIPAL -mapuser wasrvwin2k3iis6 -pass <<password>>
```

Targeting domain controller: winkdc.Test.com

Using legacy password setting method

Successfully mapped HTTP/win2k3iis6.test.com to wasrvwin2k3iis6.

Key created.

Output keytab to c:\wasrvwin2k3iis6.keytab:

Keytab version: 0x502

keysize 67 HTTP/win2k3iis6.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 2

etype 0x17 (RC4-HMAC) keylength 16 (0xfd77a26f1f5d61d1fafd67a2d88784c7)

The password is the same as the one used for creating the service account for the web server.

When the Policy Server is on UNIX:

```
ktpass -out d:\sol8sunone_host.keytab -princ host/sol8sunone.test.com@TEST.COM  
-pass <<password>> -mapuser sol8sunone -crypto DES-CBC-MD5 +DesOnly -ptype  
KRB5_NT_PRINCIPAL -kvno 3
```

Targeting domain controller: winkdc.test.com

Successfully mapped host/sol8sunone.test.com to sol8sunone.

Key created.

Output keytab to d:\sol8sunone_host.keytab:

Keytab version: 0x502

keysize 52 host/sol8sunone.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype

0x3 (DES-CBC-MD5) keylength 8 (0xb5a87ab5070e7f4a)

Account sol8sunone has been set for DES-only encryption.

8. Associate the Policy Server account (polstrwin2kps) with a Policy Server principal name (smpls/win2kps.test.com@TEST.COM) and create another keytab file destined for the Policy Server host (win2kps).

When the Policy Server is on Windows

```
Ktpass -out c:\polstrwin2kps.keytab -princ smpls/win2kps.test.com@TEST.COM  
-ptype KRB5_NT_PRINCIPAL -mapuser polstrwin2kps -pass <<password>>
```

Targeting domain controller: winkdc.Test.com

Using legacy password setting method

Successfully mapped smpls/win2kps.test.com to polstrwin2kps.

Key created.

Output keytab to c:\polstrwin2kps.keytab:

Keytab version: 0x502

keysize 72 smpls/win2kps.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 2 etype
0x17 (RC4-HMAC) keylength 16 (0xfd77a26f1f5d61d1fafd67a2d88784c7)

The password is same as the one used for creating the service account for Policy Server.

When the Policy Server is on UNIX:

```
ktpass -out d:\sol8polstrv.keytab -princ host/sol8sunone.test.com@TEST.COM -pass  
<<password>> -mapuser sol8sunone -crypto DES-CBC-MD5 +DesOnly -ptype  
KRB5_NT_PRINCIPAL -kvno 3
```

Targeting domain controller: winkdc.test.com

Successfully mapped host/sol8sunone.test.com to sol8sunone.

Key created.

Output keytab to d:\sol8polstrv.keytab:

Keytab version: 0x502

keysize 52 host/sol8sunone.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype
0x3 (DES-CBC-MD5) keylength 8 (0xb5a87ab5070e7f4a)

Account sol8sunone has been set for DES-only encryption.

9. Specify that the web server and Policy Server service accounts are Trusted for Delegation as follows:
 1. Right-click the service account (polstrwin2kps/wasrvwin2k3iis6) properties.
 2. Select the Delegation tab.
 3. Select the second option, Trust this user for delegation to any service (Kerberos only)

Or, select the third option, Trust this user for delegation to specified service. Select the Use Kerberos only option button, and add the corresponding service principal name.

The domain controller is ready for SiteMinder Kerberos authentication.

KDC Configuration on UNIX Example

The process listed following exemplifies how to configure a KDC Kerberos Realm on a UNIX host to support SiteMinder Kerberos authentication.

1. Install MIT Kerberos, if necessary.
2. Use the `kdb5_util` command to create the Kerberos database and an optional stash file. The stash file is used to authenticate the KDC to itself automatically before starting the `kadmind` and `krb5kdc` daemons as part of the host auto-boot sequence.

Both the stash file and the keytab file are potential point-of-entry for a break-in. If you install a stash file, it must be readable only by root, must not be backed up, and must exist only on the KDC local disk. If you do not want a stash file, run the `kdb5_util` without the `-s` option.

This example generates the following five database files in the directory specified in `kdc.conf` file:

- Two Kerberos database files: `principal.db` and `principal.ok`
- One Kerberos administrative database file: `principal.kadm5`
- One administrative database lock file: `principal.kadm5.lock`
- One stash file: `.k5stash`

```
[root@rhasmit init.d]# kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/kerberos/krb5kdc/principal' for realm
'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

3. Create a user principal (`testkrb`).
4. Create a user principal (for example, `testwakrb`), a host principal (`host/win2k3iis6.example.com@EXAMPLE.COM`), and a service principal (`HTTP/win2k3iis6.example.com@EXAMPLE.COM`) for the web server host. The password used for creating host account must be same as the password specified when using the `ksetup` utility on the web server host.
5. Create a user principal (`testpskrb`), host principal (`host/win2kps.example.com@EXAMPLE.COM`) and service principal (`smps/win2kps.example.com@EXAMPLE.COM`) for the Policy Server host. The password used for creating host account must be same as the password specified when using the `ksetup` utility on the Policy Server host.

6. Create a keytab file for the web server service principal as follows:

```
ktadd -k /tmp/win2k3iis6.keytab HTTP/win2k3iis6.example.com
```

7. Create keytab for Policy Server service principal as follows:

```
ktadd -k /tmp/win2kps.keytab smps/win2kps.example.com
```

The Kerberos Realm is configured for SiteMinder on a UNIX host.

Kerberos Configuration at the Policy Server on Windows Example

The process listed following exemplifies how to configure a Policy Server on Windows to support SiteMinder Kerberos authentication.

Note: If the Policy Server is installed on Windows and the KDC is deployed on UNIX, be sure to perform additional required configuration on the Policy Server host using the [Ksetup utility](#) (see page 1001).

1. Install and configure SiteMinder Policy Server.
2. Install and configure policy store directory services.
3. Log in to the Policy Server host with the service account (for example, polsrvwin2kps) created in Active Directory on the Windows domain controller.
4. Add a Host Configuration Object referencing the Policy Server.
5. Create an Agent Configuration Object and add these three new parameters:

Parameter	Value
KCCExt	.kcc
HttpServicePrincipal	Specifies the web server principal name. Example: HTTP/win2k3iis6.test.com@TEST.COM
SmpsServicePrincipal	Specifies the Policy Server principal name. Example: smps/win2kps.test.com@TEST.COM

6. Create a user directory.
7. Create a user, for example, testkrb, in the user directory.

8. Configure a new Authentication Scheme using the SiteMinder Admin UI:
 1. Create the scheme using the custom template.
 2. Specify the SiteMinder Kerberos Authentication Scheme library.
 3. Select the parameter field and specify the following three values in same order delimited by a semicolon:
 - Server name and target fields.
 - Policy Server principal name from the Windows 2003 Kerberos realm.
 - Mapping between the user principal and an LDAP search filter.

Sample parameter field:

```
http://win2k3iis6.test.com/siteminderagent/Kerberos/creds.kcc;smpls/win2kps.test.com@TES.COM;(uid=%{UID})
```

9. Configure a policy domain.
10. Add a realm to protect a resource using the Authentication Scheme.
11. Add Rules and Policies to allow access for the user, testkrb.
12. Configure a Kerberos configuration file (krb5.ini) and place krb5.ini in Windows system root path:
 - Configure the KDC for the Windows 2003 Kerberos realm (domain) to use the Windows 2003 domain controller.
 - Configure krb5.ini to use the Windows 2003 KDC keytab file containing the Policy Server principal credentials.

See the sample krb5.ini listed following.

```
[libdefaults]
default_realm = TEST.COM
default_keytab_name = C:\WINDOWS\krb5.keytab
default_tkt_enctypes = rc4-hmac des-cbc-md5
default_tgs_enctypes = rc4-hmac des-cbc-md5
[realms]
TEST.COM = {
kdc = winkdc.test.com:88
default_domain = test.com
}
[domain_realm]
.test.com = TEST.COM
```

13. Deploy the Windows KDC keytab file containing the Policy Server principal credentials to a secure location on the Policy Server.

The Policy Server on a Windows host is configured for Kerberos authentication.

Kerberos Configuration at the Policy Server on UNIX Example

The process listed following exemplifies how to configure a Policy Server on a UNIX host to support SiteMinder Kerberos authentication.

1. Create a user, for example, sol8psuser, with the same password used for creating service account for the Policy Server host (sol8ps) in Active Directory.
2. Add the host to test.com domain and login to host with user sol8psuser.
3. Install and configure SiteMinder Policy Server.
4. Install and configure policy store directory services.
5. Add a Host Configuration Object referencing the Solaris Policy Server.
6. Add an Agent Configuration Object and add the following three new parameters:

Parameter	Value
KCCExt	.kcc
HttpServicePrincipal	Specify the web server principal name. Example: HTTP/win2k3is6.test.com@TEST.COM
SmpsServicePrincipal	Specify the Policy Server principal name. Example: smps/win2kps.test.com@TEST.COM

7. Create a user directory.
8. Create a new user, for example, testkrb, in the user directory.
9. Configure a new Authentication Scheme using the SiteMinder Admin UI:
 1. Create the scheme using the custom template.
 2. Specify the SiteMinder Kerberos Authentication Scheme library.
 3. Select the parameter field and mention the following three values in same order delimited by a semicolon:
 - Server name and target fields.
 - Policy Server's principal name from the Windows 2003 Kerberos realm.
 - Mapping between the user's principal and an LDAP search filter.Sample parameter field:
`http://sol8sunone.test.com/siteminderagent/Kerberos/creds.kcc;smps/sol8ps.test.com@TEST.COM;(uid=%{UID})`
10. Configure a policy domain.
11. Add a realm to protect a resource using the Authentication Scheme.
12. Add Rules and Policies to allow access for the user, testkrb.

13. Configure a Kerberos configuration file (krb5.ini) and place krb5.ini in /etc/krb5 system path

- Configure the KDC for the Windows 2003 Kerberos realm (domain) to use the Windows 2003 domain controller.
- Configure krb5.ini to use the Windows 2003 KDC keytab file containing the policy server principal's credentials.

See the sample krb5.ini listed following:

```
[libdefaults]
ticket_lifetime = 24000
default_realm=TEST.COM
default_tgs_enctypes = des-cbc-md5
default_tkt_enctypes = des-cbc-md5
default_keytab_name = FILE:/etc/krb5.keytab
dns_lookup_realm = false
dns_lookup_kdc = false
forwardable = true
proxiable = true
[realms]
TEST.COM = {
    kdc = winkdc.test.com:88
    admin_server = winkdc.test.com:749
    default_domain = test.com
}
[domain_realm]
.test.com=TEST.COM
test.com=TEST.COM
```

14. Use the ktutil program to merge the keytab files (sol8ps_smps.keytab & sol8ps_host.keytab) containing the host principal and service principal names for the policy Server host in /etc/krb5.keytab file:

```
ktutil: rkt sol8ps_host.keytab
ktutil: wkt /etc/krb5.keytab
ktutil: q
ktutil: rkt sol8ps_smps.keytab
ktutil: wkt /etc/krb5.keytab
ktutil: q
```

15. Verify the created krb5.keytab as follows:

```
klist -k
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
----
```

```
-----
3 host/sol8ps.test.com@TEST.COM
3 smps/sol8ps.test.com@TEST.COM
```

16. Deploy the Windows 2003 KDC keytab file containing the host and Policy Server principal credentials to a secure location on the Policy Server.
17. Verify that the following environment variable is set before starting the Policy Server:

```
KRB5_CONFIG=/etc/krb5/krb5.conf
```

The Policy Server on a UNIX host is configured for Kerberos authentication.

Verify that a Resource is Protected

You can verify that a resource in your Kerberos domain is protected by creating a SiteMinder policy using the Kerberos Authentication scheme and attempting to access the resource with a user principle.

To verify that a resource is protected by Kerberos authentication

1. Log into the Policy Server User Interface.
Note: When you create or modify a Policy Server object in the Policy Server User Interface, use ASCII characters. Object creation or modification with non-ASCII characters is not supported.
2. Configure a policy domain.
3. Configure a realm using the Kerberos Authentication scheme.
4. Configure a rule to protect a specific resource within your Kerberos domain.
5. Configure a policy to protect the Kerberos resource, and add a test user to the policy.
6. Log in to the Kerberos domain as a user principle and attempt to access the protected resource.

SiteMinder authenticates the user using the KDC security token.

Troubleshooting SiteMinder Kerberos Authentication

Be aware of the following issues when working with SiteMinder Kerberos authentication:

- Be sure that the SiteMinder system clocks are synchronized (within 2 minutes) to the KDC system clock. Otherwise, Kerberos authentication fails because of clock skew errors.
- All the hosts must have suitable entries for IP Address, fully qualified domain name (FQDN) and hostname in the `/etc/hosts` file. The order of these entries also matters in some cases. Also, include at least single space between these entries:

```
IP Address  FQDN  hostname
```

- Use the Windows resource tool, `kerbtray`, to see the Kerberos tickets description. The GUI shows the various tickets received by KDC client, its flags, encryption types, and time stamp.
- Use `klist` (UNIX KDC clients) to list tickets present in the credential cache. The flag `-f` provides information about tickets flags. These tickets are forwarded to complete the Kerberos authentication process.

For example, see sample `klist` output listed following.

```
bash-2.05$ klist
Ticket cache: /tmp/krb5cc_1002
Default principal: HTTP/sol8sunone.test.com@TEST.COM
```

Valid starting	Expires	Service principal
Wed Dec 26 15:00:03 2007	Wed Dec 26 21:40:03 2007	krbtgt/TEST.COM@TEST.COM

- Always use the `kinit` program for the user and service principal at the MIT KDC host and UNIX KDC client.

The `kinit` syntax for user principal is:

```
kinit principalname
```

This command prompts for the password used while creating the user principal.

The `kinit` syntax (it can vary from host to host) for service principal is:

```
kinit -k [-t keytab location containing service principal] principalname
```

The `kinit` command does not prompt for a password, because it uses a keytab file to authenticate the service principal.

- Install any network packet trace utility on the workstation to see the Kerberos tokens exchanged between the browser and web server. The token starting with `TIR` indicates NTLM tokens, and tokens starting with `YII` denote the Kerberos tokens.
- Log off the workstation host after any change in encryption type at the KDC. Also, restart the Policy Server and web server services after any changes are made to the Kerberos configuration or the keytab files.

- The KDC does not support post-dated tickets.
- Only DES-CBC-MD5 and DES-CBC-CRC encryption types are available for MIT interoperability.
- Always enable the Policy Server and the Web Agent logs to record the authentication error messages.
- Verify that the name and location of keytab file for any host matches the ones specified in krb5.conf file.
- The SPN is always case-sensitive. The krb5.keytab files for the Policy Server and the web server host must contain the host and service principals for the Policy Server and web server hosts.
- Always verify the version of ktpass utility. The ktpass command tool utility is included in the Windows support tools and can be installed from MSDN download or Windows 2003 SP1 CD.
- Always confirm the encryption type used while creating the keytab file. Windows by default supports RC4-HMAC encryption.
- Confirm the Kerberos version number (kvno). The kvno of the service principal must match the kvno of its keytab file. The kvno number of any keytab is displayed when it is created.

- To determine the kvno of a service account:

For Windows Active Directory:

The kvno number of any service principal can be confirmed using ADSI Edit as follows:

1. Run adsiedit.msc from command prompt.
2. Go to service account, under CN=Users, DC=domain, DC=com under drop-down Domain (fqdn_ADhost) at the left.
3. Right-click the service account and click properties.
4. Verify the value of the msDs-KeyVersionNumber attribute, which matches the one shown while creating the keytab file for the service account.

The kvno for any user account changes every time its password is changed. If the version numbers do not match, create an account and keytab, or change the password to match the kvno to the kvno of the keytab.

For UNIX MIT KDC:

Use the kvno utility. The syntax of this command is:

Kvno principalname

- On Windows:

You can verify if your keytab file is valid by installing windows support tools on both Policy Server and Web Agent running the following command:

```
Kinit -k -t <keytab file location> <respective spn>
```

For example:

```
kinit -k -t C:\Windows\webserver.keytab HTTP/websvriis6.test.com@KRISH.COM
```

This command returns no error when the keytab file is valid.

- On Solaris:

Verify both keytab files validity by running the following commands:

```
Kinit -k -t <keytab file> <SPN>
```

For example:

```
kinit -k -t host.keytab host/<fqdn>@DOMAIN.COM
```

```
kinit -k -t smps.keytab smps/<fqdn>@DOMAIN.COM
```

If you get no errors, keytab files are fine, and the krb.conf file has valid values.

If you get an error, check whether the SPN is valid in KDC using the following command:

```
Kinit host/<fqdn>@DOMAIN.COM
```

This command usually asks for password. If you provide a valid password, you do not get an error message. If this command does not ask for password, the SPN was not identified. Check the property of that object, and in Account tab your SPN can be visible, for example, *host/fqdn*. If it is visible, make sure no other object has the same entry set as SPN.

Appendix B: CA Arcot Integration

This section contains the following topics:

[CA Arcot WebFort and RiskFort](#) (see page 1015)

[CA Arcot A-OK](#) (see page 1023)

CA Arcot WebFort and RiskFort

You use the CA Arcot Adapter™ (Adapter) to integrate SiteMinder with an on-premise implementation of the CA Arcot WebFort strong authentication solution and the CA Arcot RiskFort adaptive authentication solution.

Consider the following before you begin:

- The integration requires a minimum version of the Adapter and CA Arcot RiskFort.
- The integration requires a minimum version of CA Arcot WebFort.

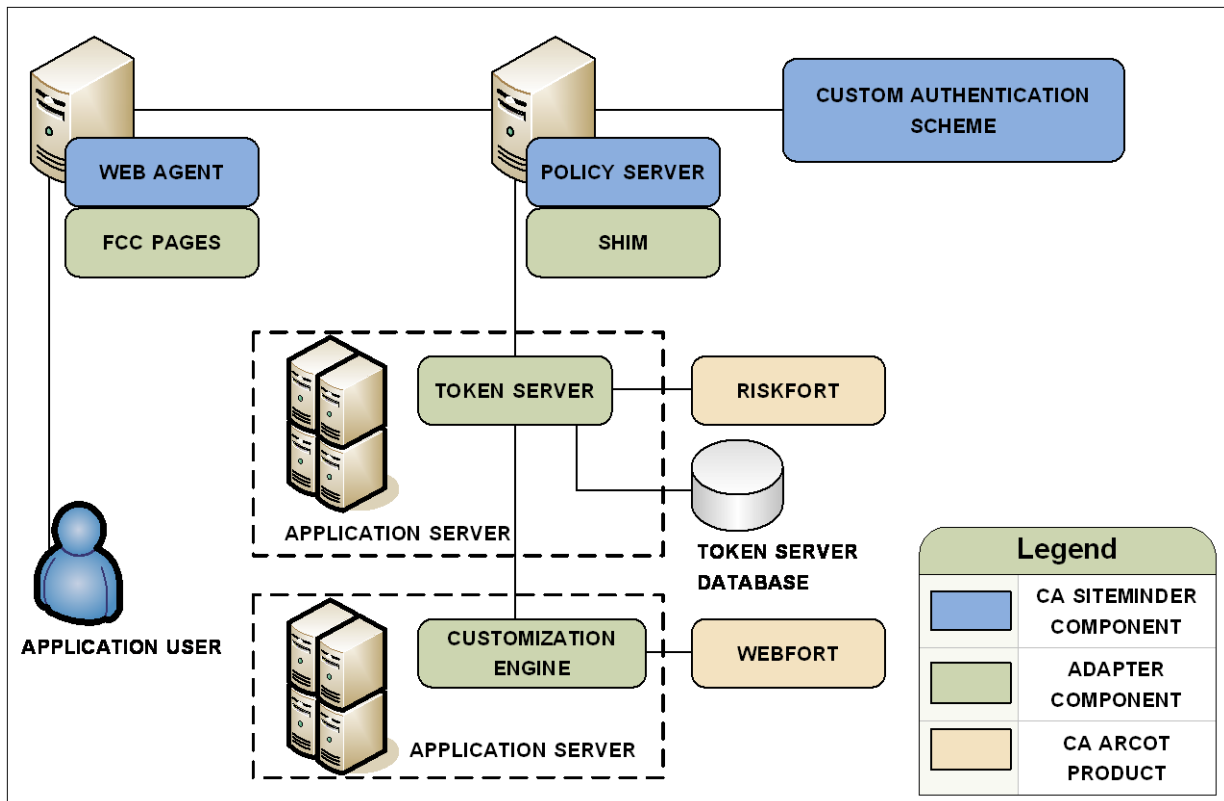
Note: For more information about the supported versions, see the r6.0 SP6 SiteMinder Platform Support Matrix.

The purpose of the following diagram is to:

- Illustrate how the Adapter and its components, CA Arcot RiskFort, and CA Arcot WebFort integrate in a SiteMinder environment.
- Detail the major components and their general relationships. This is not a workflow diagram.

Note: For more information about installing and configuring all CA Arcot components, see the CA Arcot documentation.

Figure 1: CA SiteMinder and CA Arcot integration architecture



Authentication in an On-Premise Arcot Integration

CA Arcot assumes authentication services in an integrated environment by guiding users through the authentication (CA Arcot WebFort) and risk evaluation (CA Arcot RiskFort) processes. During the authentication process:

- CA Arcot WebFort provides the strong authentication, which helps to ensure that the identities of the users requesting the SiteMinder-protected resources are legitimate.

Note: For more information about strong authentication, see the *CA Arcot WebFort Installation and Deployment Guide*. For more information about configuring the supported authentication methods, see the *CA Arcot WebFort Administration Guide*.

- CA Arcot RiskFort collects a range of data to complete a risk evaluation, which determines the level of risk associated with each transaction.

Note: For more information about risk evaluation and risk scores, see the *CA Arcot RiskFort Installation and Deployment Guide*. For more information about configuring risk scoring, see the *CA Arcot RiskFort Administration Guide*.

The result of the risk evaluation is a risk score and corresponding advice, which is a recommend action, such as allow or deny the authentication.

CA Arcot forwards the advice to the Policy Server, which if necessary, continues with its authorization services.

Note: For more information about the Adapter workflow and the role of each CA Arcot component during authentication, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

Confidence Levels and SiteMinder Authorization

The Policy Server maintains authorization services in an integrated environment and can apply the risk score to authorization decisions. The risk score is created during the [authentication process](#) (see page 1016).

The Policy Server applies the risk score as a SiteMinder confidence level (confidence level). A confidence level is based on a risk score, and as such, is also an integer that represents the likelihood that the transaction is safe.

The following example workflow details the relationship between both values and explains how the Policy Server applies a confidence level to authorization decisions:

1. After the user is successfully authenticated, the Adapter converts the risk score to a confidence level using the following algebraic formula:
$$(100 - \text{risk score}) * 10 = \text{confidence level}$$
2. The Adapter inserts the confidence level into the SiteMinder session ticket.
3. As the user requests protected resources, the Policy Server compares the confidence level in the session to ticket to the confidence level configured in the policy.
4. The following can occur:
 - If the policy rule is configured to allow access and the user's confidence level is equal to or greater than the confidence level configured in the policy, the policy rule is triggered.

Note: If the user's confidence level is less than the confidence level configured in the policy, SiteMinder denies access.

- If the policy rule is configured to reject access and the confidence level is less than the value configured in the policy, the policy rule is triggered.

More information:

[Add a Confidence Level to a Policy](#) (see page 389)

Risk Scores and Confidence Levels Compared

Although a risk score and a confidence level both help to ensure that the transaction is safe, there are differences between both values. Consider the following differences when planning for authorization decisions:

CA Arcot Risk Score	SiteMinder Confidence Level
A risk score is represented by a numeric scale 0–100.	A confidence level is represented by a numeric scale 0 -1000.
The lower the risk score, the greater the chance that the transaction is safe.	The higher the confidence level, the greater the chance that the transaction is safe. Note: A value of zero (0) represents no confidence and results in SiteMinder denying access to the requested resource.

The following example workflow details the inverse relationship between a risk score and a confidence level:

1. A user requests a SiteMinder protected resource and is forwarded to CA Arcot for authentication.
2. The Adapter guides the user through authentication and risk analysis. Based on the CA Arcot evaluation and scoring rules, the user is authenticated with a risk score of 30. The lower risk score is representative of a safe transaction.

Note: For more information about risk evaluation and scoring rules, see the *CA Arcot RiskFort Administration Guide*.

3. The Adapter:
 - a. Forwards the authentication decision to the Policy Server
 - b. Converts the risk score to a confidence level using the following algebraic formula:
$$(100 - \text{risk score}) * 10 = \text{confidence level}$$
In this example, the Adapter converts the risk score to a confidence level using the following:
$$(100 - 30) * 10 = 700$$
The higher confidence level is representative of a safe transaction.
4. The Adapter inserts the confidence level into the user session ticket.
5. The user requests a resource protected by a policy that requires a confidence level of at least 700.
6. The Policy Server grants access to the resource.

CA Arcot Integration Use Cases

The following use cases detail how you can integrate SiteMinder with CA Arcot strong authentication and risk evaluation. The use cases begin with a simple integration and progress into more complex scenarios.

CA Arcot Authentication and Risk Analysis

The simplest deployment includes integrating the Adapter and all related components with SiteMinder.

The Adapter guides users through the authentication (CA Arcot WebFort) and risk evaluation (CA Arcot RiskFort) processes to apply a [risk score during authentication](#) (see page 1016).

To deploy Arcot authentication and risk analysis

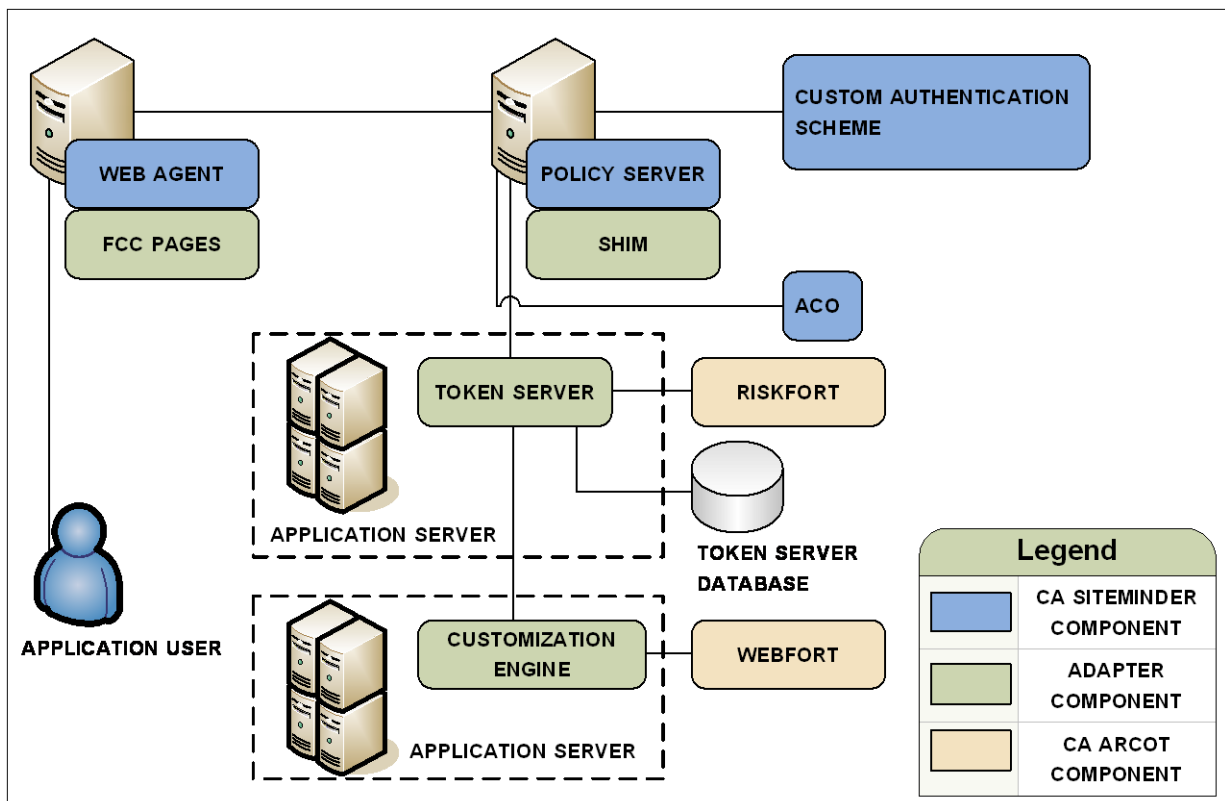
1. Be sure that CA Arcot RiskFort and CA Arcot WebFort are installed and configured.
Note: For more information, see the respective CA Arcot installation and deployment guide.
2. Install and deploy the CA Arcot Adapter and all related components. These components include a set of Forms Credential Collector files. These files let you use the Adapter HTML forms authentication scheme to gather user credentials.
Note: For more information about installing and configuring the Adapter and all related components, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

3. Do the following:
 - a. Configure a SiteMinder Custom authentication scheme to call the Adapter library.
 - b. Determine which Web Agents are included in the CA Arcot integration and configure the respective Agent Configuration Objects (ACO) to support the integration.

Note: For more information about the required custom authentication scheme and ACO settings, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

The following diagram illustrates this deployment scenario:

Figure 2: CA Arcot authentication and risk analysis



SiteMinder Authentication and CA Arcot Risk Analysis

You can configure the Adapter for risk evaluation only by integrating a SiteMinder authentication scheme. A SiteMinder authentication scheme that is part of the integration is known as backing authentication.

If you use a SiteMinder authentication scheme as backing authentication, the Shim acts as an interface between SiteMinder and the SiteMinder authentication scheme.

Note: For more information about backing authentication, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*. Not all SiteMinder authentication schemes are supported for backing authentication. For more information, see the r6.0 SP6 SiteMinder Platform Support Matrix.

To deploy SiteMinder authentication and Arcot risk analysis

1. Complete the steps listed in [CA Arcot Authentication and Risk Analysis](#) (see page 1019).

Important! The integration requires that a SiteMinder Custom authentication scheme is configured. The SiteMinder Custom authentication scheme calls the required Adapter library. This library is required even if you are deploying backing authentication.

2. Be sure that you configure the SiteMinder Custom authentication scheme with a valid CA Arcot parameter. This parameter must represent a user flow that supports the SiteMinder authentication scheme that is functioning as backing authentication. You enter this value in in the Parameter field.

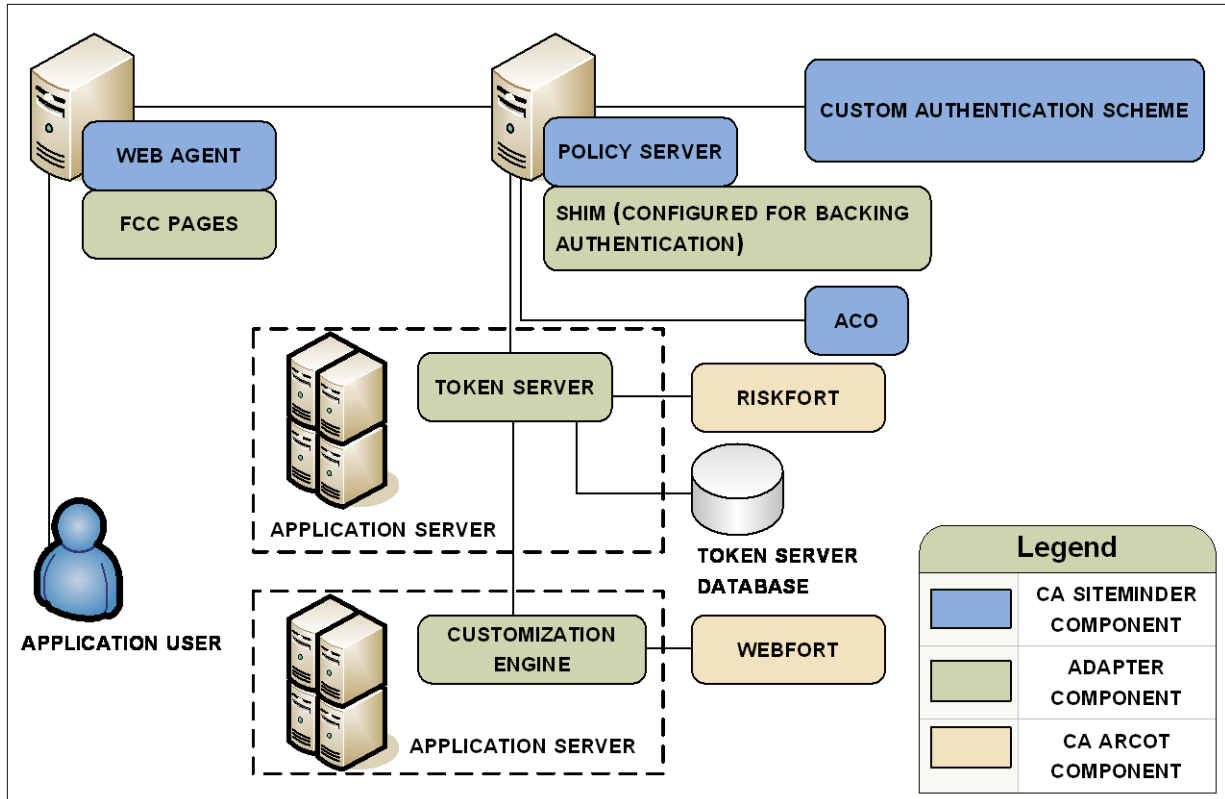
Note: For more information about user flows and the corresponding parameter values, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

3. Configure the Shim to use the SiteMinder authentication scheme as a backing authentication.

Note: For more information about configuring a backing authentication scheme, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

The following diagram illustrates this deployment scenario:

Figure 3: CA SiteMinder authentication and CA Arcot risk analysis



SiteMinder Authorization and Confidence Levels

You can extend the Policy Server authorization services by adding a [confidence level](#) (see page 1017) to policies. You add a confidence level to policies in the context of an active policy expression.

Adding a confidence level to policies lets you apply the CA Arcot risk analysis results to authorization decisions.

To add a confidence level to policies

1. Complete the steps in [CA Arcot Authentication and Risk Analysis](#) (see page 1019) or [SiteMinder Authentication and CA Arcot Risk Analysis](#) (see page 1020).
2. Create or modify a policy to include an active policy expression that references the following.
 - The CA Arcot risk analysis library (`smriskactiveexpr`).
 - A confidence level value of 1–1000.

More information:

[Add a Confidence Level to a Policy](#) (see page 389)

User Store Consideration

All SiteMinder users to which the integration applies must be made available to the CA Arcot WebFort database.

Contact CA Arcot Support for assistance.

Note: For contact information, see the *CA Arcot Adapter for CA SiteMinder Installation and Configuration Guide*.

CA Arcot A-OK

You use the CA Arcot A–OK Adapter™ (A–OK Adapter) to integrate SiteMinder with the hosted CA Arcot A–OK service.

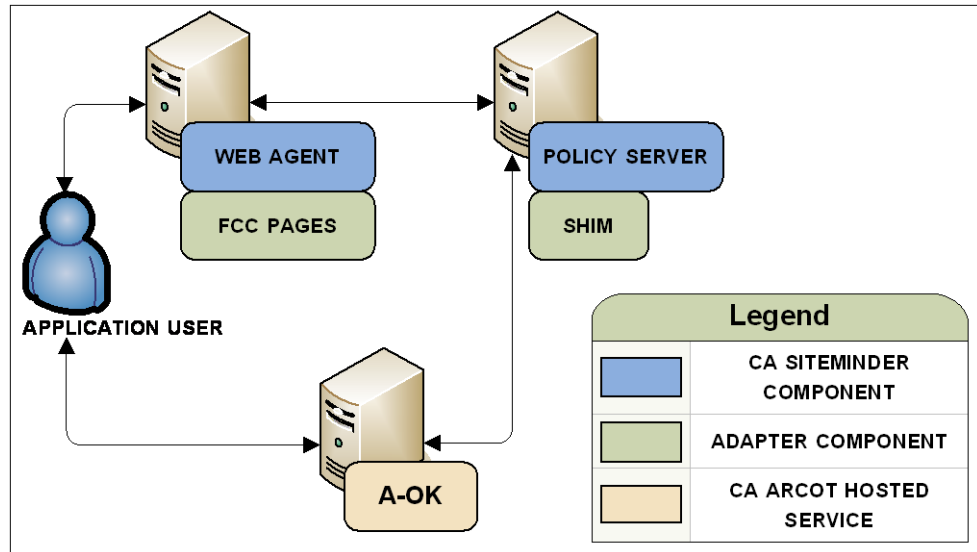
Note: The integration requires a minimum version of the A–OK Adapter. For more information about the supported version, see the r6.0 SP6 SiteMinder Platform Support Matrix.

The purpose of the following diagram is to:

- Illustrate how the A–OK Adapter and its components integrate in a SiteMinder environment.
- Detail the major components and their general relationships. This is not a workflow diagram.

Note: For more information about installing and configuring the A-OK Adapter, see the *CA Arcot A-OK Adapter for CA SiteMinder Installation and Configuration Guide*.

Figure 4: CA SiteMinder and CA Arcot A-OK integration architecture



Authentication in a Hosted CA Arcot Integration

CA Arcot A-OK assumes authentication services in an integrated environment by guiding users through the authentication and risk evaluation processes. CA Arcot A-OK uses a series of SAML requests and responses to step through the authentication workflow.

Note: For more information about the authentication workflow, see the *CA Arcot A-OK Adapter for CA SiteMinder Installation and Configuration Guide*.

The result of the risk evaluation is a risk score and corresponding advice, which is a recommend action, such as allow or deny the authentication.

CA Arcot A-OK forwards the advice to the Policy Server, which if necessary, continues with authorization services.

Note: For more information about managing user credentials and configuring the rules associated with the risk evaluation process, see the *CA Arcot A-OK User Administration Guide*.

Confidence Levels and SiteMinder Authorization

The Policy Server maintains authorization services in an integrated environment and can apply the risk score to authorization decisions. The risk score is created during the [authentication process](#) (see page 1024).

The Policy Server applies the risk score as a SiteMinder confidence level. A confidence level is based on a risk score, and as such, is also an integer that represents the likelihood that the transaction is safe.

The following example workflow details the relationship between both values and explains how the Policy Server applies a confidence level to authorization decisions:

1. After the user is successfully authenticated, the A-OK Adapter converts the risk score to a confidence level using the following algebraic formula:

$$(100 - \text{risk score}) * 10 = \text{confidence level}$$
2. The A-OK Adapter inserts the confidence level into the SiteMinder session ticket.
3. As the user requests protected resources, the Policy Server compares the confidence level in the session to ticket to the confidence level configured in the policy.
4. The following can occur:
 - If the policy rule is configured to allow access and the user's confidence level is equal to or greater than the confidence level configured in the policy, the policy rule is triggered.

Note: If the user's confidence level is less than the confidence level configured in the policy, SiteMinder denies access.

 - If the policy rule is configured to reject access and the confidence level is less than the value configured in the policy, the policy rule is triggered.

More information:

[Add a Confidence Level to a Policy](#) (see page 389)

Risk Scores and Confidence Levels Compared

Although a risk score and a confidence level both help to ensure that the transaction is safe, there are differences between both values. Consider the following differences when planning for authorization decisions:

CA Arcot Risk Score	SiteMinder Confidence Level
A risk score is represented by a numeric scale 0–100.	A confidence level is represented by a numeric scale 0–1000.

CA Arcot Risk Score	SiteMinder Confidence Level
The lower the risk score, the greater the chance that the transaction is safe.	The higher the confidence level, the greater the chance that the transaction is safe. Note: A value of zero (0) represents no confidence and results in SiteMinder denying access to the requested resource.

The following example workflow details the inverse relationship between a risk score and a confidence level:

1. A user requests a SiteMinder protected resource and is forwarded to CA Arcot A-OK for authentication.
2. The A-OK Adapter guides the user through authentication and risk analysis. Based on the CA Arcot A-OK evaluation and scoring rules, the user is authenticated with a risk score of 30. The lower risk score is representative of a safe transaction.

Note: For more information about managing user credentials and configuring the rules associated with the risk evaluation process, see the *CA Arcot A-OK User Administration Guide*.

3. The A-OK Adapter:
 - a. Forwards the authentication decision to the Policy Server
 - b. Converts the risk score to a confidence level using the following algebraic formula:
$$(100 - \text{risk score}) * 10 = \text{confidence level}$$
In this example, the A-OK Adapter converts the risk score to a confidence level using the following:
$$(100 - 30) * 10 = 700$$
The higher confidence level is representative of a safe transaction.
4. The A-OK Adapter inserts the confidence level into the user session ticket.
5. The user requests a resource protected by a policy that requires a confidence level of at least 700.
6. The Policy Server grants access to the resource.

CA Arcot A-OK Integration Use Cases

The following use cases detail how you can integrate SiteMinder with CA Arcot A-OK strong authentication and risk evaluation. The use cases begin with a simple integration and progress into more complex scenarios.

CA Arcot A-OK Authentication and Risk Analysis

The simplest deployment includes integrating the A-OK Adapter and all related components with SiteMinder.

The A-OK Adapter guides users through the authentication and risk evaluation processes to apply a risk score during the [authentication process](#) (see page 1024).

To deploy CA Arcot A-OK authentication and risk analysis

1. Be sure that the CA Arcot A-OK service is available.
2. Install and deploy the A-OK Adapter and all related components. These components include a set of Forms Credential Collector files. These files let you use the A-OK Adapter HTML forms authentication scheme to gather user credentials.

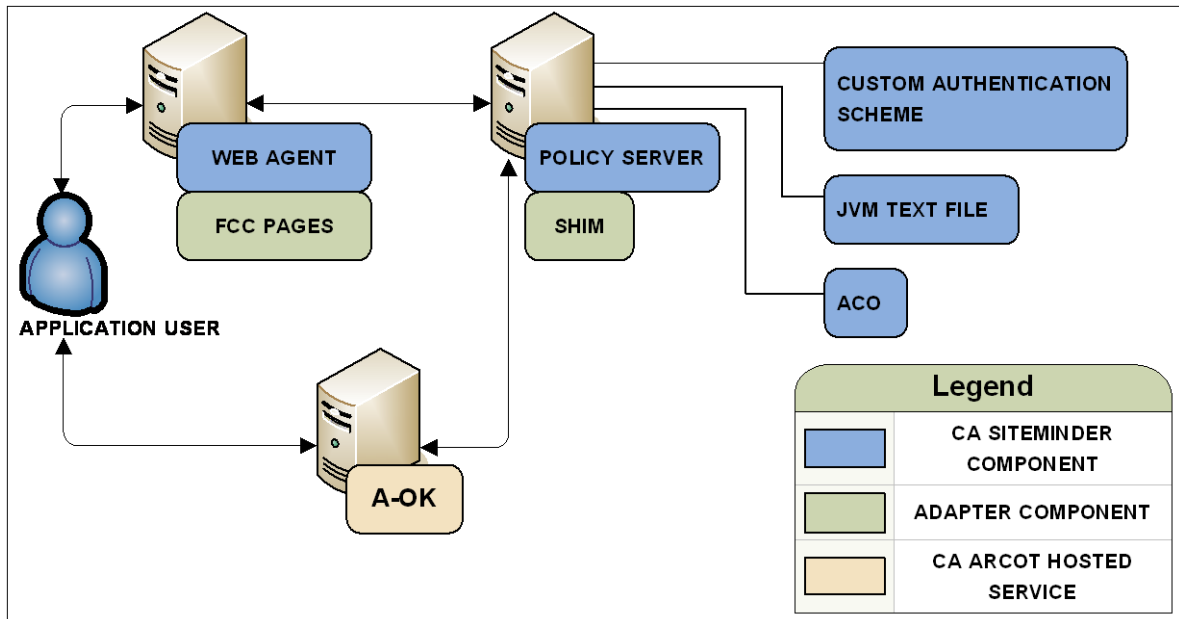
Note: For more information about installing and configuring the A-OK Adapter and all related components, see the *CA Arcot A-OK Adapter for CA SiteMinder Installation and Configuration Guide*.

3. Do the following:
 - a. Configure a SiteMinder Custom authentication scheme to call the A-OK Adapter library.
 - b. Determine which Web Agents are included in the CA Arcot A-OK integration and configure the respective Agent Configuration Objects (ACO) to support the integration.
 - c. Add the A-OK Adapter JAR files, certificates, and properties files to the Policy Server's Java Virtual Machine (JVM) file (JVMOptions.txt).

Note: For more information about the required custom authentication scheme, ACO settings, and edits to the Policy Server JVM file, see the *CA Arcot A-OK Adapter for CA SiteMinder Installation and Configuration Guide*.

The following diagram illustrates this deployment scenario:

Figure 5: CA Arcot A-OK authentication and risk analysis



SiteMinder Authorization and Confidence Levels

You can extend the Policy Server authorization services by adding a [confidence level](#) (see page 1025) to policies. You add a confidence level to policies in the context of an active policy expression.

Adding a confidence level to policies lets you apply the CA Arcot A-OK risk analysis results to authorization decisions.

To add a confidence level to policies

1. Complete the steps in [CA Arcot A-OK Authentication and Risk Analysis](#) (see page 1027).
2. Create or modify a policy to include an active policy expression that references the following.
 - The CA Arcot A-OK risk analysis library (smriskactiveexpr).
 - A confidence level value of 1–1000.

More information:

[Add a Confidence Level to a Policy](#) (see page 389)

User Store Consideration

All SiteMinder users to which the integration applies must be made available to the CA Arcot A-OK hosted service.

Contact CA Arcot Support for assistance.

Note: For contact information, see the *CA Arcot A-OK Adapter for CA SiteMinder Installation and Configuration Guide*.

Index

#

- # Filters field, customizing • 950
- # Records Per Page field, customizing • 957

A

- About Access Control Lists • 48
- About LanMan User Directories • 933
- About SiteMinder Security Policies • 49
- access control lists
 - described • 47
 - limitations • 48
- access control, requirements • 54, 58
- access roles
 - and SiteMinder • 59
 - description • 59
 - providing application entitlements • 59
- Access Roles in Identity Manager • 59
- Access the SiteMinder SQL Query Scheme Dialog • 209
- access, write • 577
- accessing the Policy Server User Interface • 64
- accessProperties.getVisible • 966
- Act As Part of Operating System system privilege • 558
- Action Group Box • 695, 753
- actions
 - authentication events • 331
 - authorization events • 333
 - introduced • 329
 - Web Agent • 330
- active
 - policies, configuring • 395, 910
 - response, described • 359
 - rules, configuring • 346
 - rules, configuring for global rules • 445
- Active Directory and NetBIOS Names • 98
- Active Directory Overview • 155
- Active Directory Users Cannot Change Passwords • 514
- active expression • 413
- active policies
 - configuring for global policies • 454
- Active Policy Editor dialog • 910
- Active Policy Editor Dialog • 910
- Active Policy Editor Fields and Controls • 911
- Active Policy Editor Prerequisites • 910
- Active Responses Created for Custom Agents • 372
- Active Rule Editor dialog • 909
- Active Rule Editor Dialog • 909
- Active Rule Editor Dialog Fields and Controls • 910
- Active Rule Editor Dialog Prerequisites • 909
- Active Rule Tab • 698
- AD namespace • 567
- Add a Change Password Link to the Password Services CGI • 492
- Add a Change Password Link to the Password Services Servlet • 494
- Add a Confidence Level to a Policy • 389
- Add a Data Source Group Box • 575
- Add a Host Name • 392
- Add a Host Name for a Global Policy • 452
- Add a Range of IP Addresses • 394
- Add a Range of IP Addresses for a Global Policy • 453
- Add a Rule To or Remove a Rule From a Rule Group • 350
- Add a Subnet Mask • 393
- Add a Subnet Mask for a Global Policy • 453
- Add Additional Servers to the Setup for Load Balancing and Failover • 181
- Add Agents to an Agent Group • 140
- Add an Expression to a Policy • 414
- Add an IP Address dialog • 916
- Add an IP Address Dialog • 916
- Add an IP Address Dialog Fields and Controls • 917
- Add an IP Address Dialog Prerequisites • 916
- Add an IP Address or Name Group Box • 572
- Add and Remove Global Policy Time Restrictions • 454
- Add and Remove Time Restrictions for Policies • 394
- Add Attributes to Properties Files • 964
- Add AuthValidate Directory Mapping Dialog • 596
- Add AuthValidate Directory Mapping Dialog Fields and Controls • 596
- Add Directory Mapping dialog • 592
- Add Directory Mapping Dialog • 592
- Add Directory Mapping Fields and Controls • 592
- Add Fields to Collector Pages • 966
- Add Global Rules and Global Responses to a Global Policy • 449

- Add Identity Manager Environments to a Policy Domain • 312
- Add LDAP Expressions to Policies • 387
- Add Multiple Policy Servers to the Host Configuration Object • 114
- Add New Strings • 505
- Add Roles to a Policy • 387
- Add Rules to a Policy • 380
- Add Tasks to the Task Selector Page • 960
- Add the Root Certificate Authority to the Certificate Database • 191
- Add the Server Certificate to the Certificate Database • 193
- Add Time Restrictions for Rules • 344
- Add Users by Manual Entry • 385
- Add Users to a Policy • 379
- adding
 - Agents to an Agent group • 140
 - range of IP addresses to a policy • 394, 453
 - rules to a global policy • 449
 - rules to a policy • 380, 381
 - subnet mask to a policy • 392, 393, 452, 453
 - user directories to a policy domain • 310
- administration
 - services, described • 35
 - window, using • 65
- administrative roles • 45
- Administrative Roles • 45
- administrator
 - accounts • 63, 223
 - assigning to a policy domain • 311
 - configuring • 226
 - deleting • 229
 - flushing realm from resource cache • 326
 - introduced • 39
 - password authentication • 600
 - privileges • 223, 225
 - requiring credentials • 559
 - roles in presentation properties files • 943
 - scope, configuring • 227, 602
 - super user • 63
 - system • 227
 - tasks, configuring • 602
 - user lookup • 600
- Administrator and Related Dialogs Reference • 599
- Administrator Concepts • 223
- Administrator Credentials Group Box • 551, 555, 560, 569
- Administrator Lookup Group Box • 600
- Administrator Properties Dialog • 599
- Administrator Properties Dialog General Group Box • 600
- Administrator Scope and Tasks • 225, 227
- administrators
 - domain • 227, 228
- Administrators Tab • 677
- advanced
 - menu • 69
 - rule options • 344, 697, 754
- Advanced Group box • 697
- Advanced Group Box • 754
- Advanced Policy Options • 391
- Advanced Rule Options • 334, 344
- Advanced Tab • 362, 448
- Advantages of Centrally Configuring Web Agents • 105
- Affiliate Agent
 - response attributes • 357
- Affiliate Agent Actions • 331
- Affiliate Agent Response Attributes • 357
- Affiliate Agent Responses • 355
- Affiliate Attribute Editor Dialog • 794
- Affiliate Attribute Editor Dialog Fields and Controls • 795
- Affiliate Attribute Editor Dialog--Advanced Tab • 796
- Affiliate Attribute Editor Dialog--Attribute Setup Tab • 795
- Affiliate Attribute Editor Prerequisites • 794
- Affiliate Dialog • 787
- Affiliate Dialog Fields and Controls • 788
- Affiliate Dialog Prerequisites • 790
- Affiliate Dialog--Advanced Tab • 798
- Affiliate Dialog--Assertions Tab • 791
- Affiliate Dialog--Attributes Tab • 793
- Affiliate Dialog--IP Addresses Tab • 797
- Affiliate Dialog--Session Tab • 793
- Affiliate Dialog--Time Tab • 797
- Affiliate Dialog--Users Tab • 791
- affiliate domains
 - introduced • 39
- Affiliate Domains • 313
- Affiliates Tab • 678
- affiliates, SAML
 - introduced • 41
- After Certificate Prompt, Authentication Failure Received • 931
- After Following Previous Procedure, Still No Certificate Prompt • 928

Agent

- adding to an Agent group • 140
- Agent and Related Dialogs Reference • 535
- Agent API Support • 285, 288
- Agent Components • 106
- Agent Configuration Dialog Fields and Controls • 541
- Agent Configuration Object dialog • 540
- Agent Configuration Object Dialog • 540
- Agent Configuration Object Prerequisites • 540
- Agent Configuration objects
 - creating • 129
 - description • 106
- Agent Configuration Objects
 - introduced • 39
- Agent Dialog • 535
- Agent Dialog Fields and Controls • 536
- Agent Dialog—RADIUS Radio Button Selected • 537
- Agent Dialog—Support 4.x Agents Checked • 536
- Agent group dialog • 541
- Agent Group Dialog • 541
- Agent Group Dialog Fields and Controls • 542
- Agent Group Items dialog • 544
- Agent Group Items Dialog • 544
- Agent Group Object Prerequisites • 541
- Agent Group Usage • 138
- Agent groups
 - adding Agents • 140
 - configuring • 139
 - introduced • 39
- Agent keys
 - rollovers for user sessions • 93
- Agent Object Association with an Agent Configuration Object or File • 124
- Agent Object Prerequisites • 535
- Agent Type Group Box • 542, 700, 706, 713, 755
- Agent Types
 - introduced • 39
- Agents
 - introduced • 39
- Agents and Agent Groups • 99
- Agents Overview • 99
- allow access rule • 330
- Allow Nested Groups in Policies • 382
- Allow Specific Users to Change Their Passwords in CGI • 493
- Allow Specific Users to Change Their Passwords in Servlet • 494
- Allow/Deny and Enable/Disable Group Box • 697, 752
- AND Users/Groups Check Box • 382
- Anonymous authentication scheme
 - configuring • 642
- anonymous authentication schemes
 - configuring • 288
 - described • 236, 642
 - prerequisites • 289
- Anonymous Authentication Schemes • 236, 288
- Anonymous ID field • 564
- Anonymous Scheme Prerequisites • 289
- Apache Web Server Prerequisites • 482
- Application Server Agents • 103
- applications
 - entitlements • 59
- Assertion Consumer Services Dialog • 827
- Assign Administrator Privileges to the SiteMinder User • 178
- Assign an Administrator • 311
- Assign an Authorization Directory to a Realm • 216
- Assign User Directories • 310
- Associate a Rule with a Global Response • 381
- Associate a Rule with a Response or Response Group • 380
- Assumptions for Novell eDirectory Configuration • 166
- Asynchronous Call Support Configuration • 203
- Asynchronous Call Support During Failover and Connection Pooling • 202
- Attribute Authority and Query Reference • 889
- Attribute Caching Group Box • 709, 758
- Attribute Fields Group Box • 710, 759
- Attribute Kind Group Box • 709, 758
- Attribute List Group Box • 707, 714, 756
- Attribute Svc Dialog • 889
- Attribute Svc Dialog Fields and Controls • 890
- Attribute Svc Dialog Prerequisites • 889
- attribute-based delegation
 - specifying managed organization attributes • 944
- attributes
 - collecting new attributes • 963
 - directory, user directories, See also user directory • 552, 556, 570
 - extending object classes • 963
 - specifying in object properties files • 944
 - user, See also user • 577
- Attributes Dialog for Attribute Queries • 892
- Attributes Dialog Prerequisites • 891
- attributes, response • 356, 359, 446
- auditing services • 35

Auth. Scheme Properties--WS-Federation--Advanced Tab • 886

Auth. Scheme Properties--WS-Federation--Signout Tab • 885

Auth. Scheme Properties--WS-Federation--SSO Tab • 883

Auth. Scheme Properties--WS-Federation--Users Tab • 882

Authenticate Group Box • 601

authenticated user's security context running in • 551, 560, 569

authentication

- directory • 213
- events • 331
- services • 35

Authentication Events • 331

Authentication in a Hosted CA Arcot Integration • 1024

Authentication in an On-Premise Arcot Integration • 1016

Authentication over SSL • 236

Authentication Processing for Hierarchical Policies • 410

Authentication Scheme and Related Dialogs Reference • 607

Authentication Scheme Dialog • 607

Authentication Scheme Dialog -- SAML 2.0 Template • 658, 841

Authentication Scheme Dialog Fields and Controls • 609

Authentication Scheme Dialog Prerequisites • 799

Authentication Scheme Dialog—Anonymous Template • 642

Authentication Scheme Dialog—Anonymous Template—Advanced Tab • 643

Authentication Scheme Dialog—Anonymous Template—Scheme Setup • 643

Authentication Scheme Dialog—Basic over SSLTemplate • 611

Authentication Scheme Dialog—Basic over SSLTemplate—Advanced Tab • 612

Authentication Scheme Dialog—Basic over SSLTemplate—Scheme Setup Tab • 611

Authentication Scheme Dialog—Basic Template • 610

Authentication Scheme Dialog—Basic Template—Advanced Tab • 611

Authentication Scheme Dialog—Basic Template—Scheme Setup Tab • 610

Authentication Scheme Dialog—Custom Template • 644

Authentication Scheme Dialog—Custom Template for Federation • 660

Authentication Scheme Dialog—Custom Template for Federation —Scheme Setup Tab • 661

Authentication Scheme Dialog—Custom Template—Advanced Tab • 645

Authentication Scheme Dialog—Custom Template—Scheme Setup Tab • 644

Authentication Scheme Dialog—HTML Form Template • 612

Authentication Scheme Dialog—HTML Form Template—Advanced Tab • 615

Authentication Scheme Dialog—HTML Form Template—Scheme Setup Tab • 613

Authentication Scheme Dialog—Impersonation Template • 662

Authentication Scheme Dialog—Impersonation Template—Advanced Tab • 664

Authentication Scheme Dialog—Impersonation Template—Scheme Setup Tab • 662

Authentication Scheme Dialog—MS Passport Template • 619

Authentication Scheme Dialog—MS Passport Template—Advanced Tab • 622

Authentication Scheme Dialog—MS Passport Template—Scheme Setup Tab • 621

Authentication Scheme Dialog—RADIUS CHAP/PAP Template • 622

Authentication Scheme Dialog—RADIUS CHAP/PAP—Advanced Tab • 622

Authentication Scheme Dialog—RADIUS CHAP/PAP—Scheme Setup Tab • 622

Authentication Scheme Dialog—RADIUS Server Template • 623

Authentication Scheme Dialog—RADIUS Server Template—Advanced Tab • 624

Authentication Scheme Dialog—RADIUS Server Template—Scheme Setup Tab • 623

Authentication Scheme Dialog—SafeWord HTML Form Template • 625

Authentication Scheme Dialog—SafeWord HTML Form Template—Advanced Tab • 627

Authentication Scheme Dialog—SafeWord HTML Form Template—Scheme Setup Tab • 626

Authentication Scheme Dialog—SafeWord Template • 624

Authentication Scheme Dialog—SafeWord Template—Advanced Tab • 625

Authentication Scheme Dialog—SafeWord Template—Scheme Setup Tab • 624

Authentication Scheme Dialog--SAML Artifact Template • 645, 800

Authentication Scheme Dialog--SAML Artifact Template--Advanced Tab • 652, 808

Authentication Scheme Dialog--SAML Artifact Template--Scheme Setup Tab • 645, 801

Authentication Scheme Dialog--SAML POST Template • 652, 808

Authentication Scheme Dialog--SAML POST Template--Advanced Tab • 658, 814

Authentication Scheme Dialog--SAML POST Template--Scheme Setup Tab • 653, 809

Authentication Scheme Dialog—SecurID HTML Form Template • 628

Authentication Scheme Dialog—SecurID HTML Form Template—Advanced Tab • 630

Authentication Scheme Dialog—SecurID HTML Form Template—Scheme Setup Tab • 629

Authentication Scheme Dialog—SecurID Template • 627

Authentication Scheme Dialog—SecurID Template—Advanced Tab • 628

Authentication Scheme Dialog—SecurID Template—Scheme Setup Tab • 627

Authentication Scheme Dialog—Windows Template • 615

Authentication Scheme Dialog—X509 Client Cert and Basic Template • 632

Authentication Scheme Dialog—X509 Client Cert and Basic Template—Advanced Tab • 634

Authentication Scheme Dialog—X509 Client Cert and Basic Template—Scheme Setup Tab • 633

Authentication Scheme Dialog—X509 Client Cert and Form Template • 637

Authentication Scheme Dialog—X509 Client Cert and Form Template—Advanced Tab • 639

Authentication Scheme Dialog—X509 Client Cert and Form Template—Scheme Setup Tab • 637

Authentication Scheme Dialog—X509 Client Cert or Basic Template • 634

Authentication Scheme Dialog—X509 Client Cert or Basic Template—Advanced Tab • 636

Authentication Scheme Dialog—X509 Client Cert or Basic Template—Scheme Setup Tab • 635

Authentication Scheme Dialog—X509 Client Cert or HTML Form Template—Advanced Tab • 642

Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template • 639

Authentication Scheme Dialog—X509 Client Cert or HTML Forms Template—Scheme Setup Tab • 640

Authentication Scheme Dialog—X509 Client Cert Template • 631

Authentication Scheme Dialog—X509 Client Cert Template—Advanced Tab • 632

Authentication Scheme Dialog—X509 Client Cert Template—Scheme Setup Tab • 631

Authentication Scheme Group Box • 683

Authentication Scheme Object Prerequisites • 608

Authentication Scheme Properties Dialog--WS-Federation Template • 877

authentication schemes

- anonymous • 236, 288, 642
- anonymous prerequisites • 289
- basic • 234, 241
- basic over SSL • 243
- basic over SSL prerequisites • 243
- basic, configuring • 610
- basic, prerequisites • 242, 259
- certificate mapping • 293
- configuration order • 240
- configuring • 607
- configuring custom • 644
- configuring RADIUS CHAP/PAP • 622
- configuring RADIUS server • 623
- configuring SafeWord Server • 624, 625
- configuring SecurID • 627
- configuring X.509 client certificate • 631
- configuring X.509 client certificate and basic • 632
- configuring X.509 client certificate or basic • 634
- custom • 236, 290, 644
- custom prerequisites • 290
- described • 231
- digest • 236
- forms-based authentication, installation • 247, 269, 275
- HTML forms • 234, 244, 612
- HTML forms prerequisites • 247, 292
- HTML forms templates • 248
- introduced • 39
- multiple instances of a scheme • 241
- NTLM • 234
- over SSL • 236

- overview • 232, 233
- policy server • 240
- protection levels • 237
- proxy • 235
- proxy authentication • 235
- RADIUS CHAP/PAP • 265
- RADIUS CHAP/PAP prerequisites • 266
- RADIUS server • 267, 623
- RADIUS server prerequisites • 267
- SafeWord Server • 268, 624
- SafeWord Server prerequisites • 268
- SAML 2.0 Template • 291
- SAML Artifact Template • 291
- SAML POST Template • 291
- SAML Session Ticket • 291
- SecurID • 270
- SecurID prerequisites • 274, 275
- SecurID with HTML forms support • 270
- setting up • 240
- specifying localized FCCs • 973
- Web server • 240
- WS-Security • 291
- X.509 client certificate • 235, 277
- X.509 client certificate and basic • 279, 283
- X.509 client certificate and basic prerequisites • 280
- X.509 client certificate or basic • 281, 286
- X.509 client certificate or basic prerequisites • 282, 287
- X.509 client certificate prerequisites • 278
- XML Digital Signature • 291
- XML Document Credential Collector • 291
- Authentication Schemes • 231
- Authentication Schemes Overview • 231
- Authentication Schemes Requiring Additional Attribute Information • 507
- Authentication Scheme—Windows Template—Advanced Tab • 619
- Authentication Scheme—Windows Template—Scheme Setup Tab • 616
- authorization
 - directory • 213
 - directory, assigning • 216
 - events • 333
- Authorization Events • 333
- Authorization Processing for Hierarchical Policies • 412
- AuthValidate Directory Mappings Dialog • 595

- AuthValidate Directory Mappings Dialog Fields and Controls • 596
- AuthValidate Directory Mappings Dialog Prerequisites • 595
- Availability of SiteMinder-generated Response Attributes • 370

B

- Basic .fcc Requirements for Impersonation • 467
- basic authentication schemes
 - described • 234, 241
 - prerequisites • 242, 259
- Basic Authentication Schemes • 234, 241
- basic over SSL authentication schemes
 - described • 243
 - prerequisites • 243
- Basic Over SSL Authentication Schemes • 243
- Basic over SSL Scheme Prerequisites • 243
- Basic Scheme Prerequisites • 242
- Before You Configure a Connection over SSL • 189
- Bind Policies to Group Attributes • 399
- Bind Policies to Organization Attributes • 400
- Bind Policies to Organization Units • 400
- Bind Policies to Organizational Roles • 399
- Bind Policies to Organizations • 400
- Bind Policies to SQL Queries • 404
- Bind Policies to User Attributes • 399, 404
- Bind Policies to User Groups • 399, 404
- Bind Policies to Users with the Manual Entry Field • 398, 401, 403
- Bind Policies to Users with the Search Feature • 398, 401, 403
- Binding Policies to Custom Object Classes • 400
- Boolean logic
 - used by eTelligent Rules • 421

C

- CA Arcot A-OK • 1023
- CA Arcot A-OK Authentication and Risk Analysis • 1027
- CA Arcot A-OK Integration Use Cases • 1026
- CA Arcot Authentication and Risk Analysis • 1019
- CA Arcot Integration • 1015
- CA Arcot Integration Use Cases • 1019
- CA Arcot WebFort and RiskFort • 1015
- CA Directory Variable Requirements • 422
- CA SSO/WAC Integration • 977
- CA Technologies Product References • iii

- cache
 - flushing realms from • 326
- Calculate an Average Elapsed Time • 527
- Cascading Style Sheet
 - See CSS • 937
- Cascading Style Sheet Customization • 940
- Certificate Attributes that Require Custom Mappings • 531
- Certificate Authorities and Web Services Variables • 415
- Certificate Map Test Dialog • 672
- Certificate Map Test Dialog Fields and Controls • 673
- Certificate Map Test Dialog Prerequisites • 672
- Certificate Map Test Results • 296
- certificate mapping
 - CRL checking • 296
 - custom mapping expressions • 300
 - introduced • 293
 - multiple attributes • 304
 - OCSP checking • 296
 - testing • 295
- Certificate Mapping • 293
- Certificate Mapping Properties Dialog • 667
- Certificate Mapping Properties Dialog Fields and Controls • 667
- Certificate Mapping Properties Dialog Prerequisites • 667
- Certificate Mappings Dialog • 665
- Certificate Mappings Dialog Fields and Controls • 666
- Certificate Mappings Dialog Prerequisites • 665
- Certificate Revocation List (CRL) Checking Group Box • 670
- Certificate Revocation List See CRL checking • 296
- Certificate-based Authentication Tests • 531
- certificates
 - Apache, enabling the Web server to trust • 927
 - IIS, enabling the Web server to trust • 926
 - Netscape, enabling the Web server to trust • 925
 - verifying browser • 930
- CGI-based password services • 479
- CGI-based Password Services HTML Form Customization • 497
- CGI-based Password Services HTML Form Templates • 497
- Challenge/Response attribute • 564
- Challenge-Handshake Authentication Protocol. See CHAP • 265
- Change Required Group Box • 782
- Change Tabs • 68
- Change the Character Encoding in The JSPs • 974
- Change the Default Redirect URL • 483
- Change the Search Attributes • 953
- Change the Tasks Displayed For Administrators • 959
- Change The Text of Links • 958
- changing tabs • 68
- CHAP Overview • 265
- CHAP-based authentication • 265
- character encoding • 970
 - changing in JSPs • 974
 - localizing DMS • 970
 - specifying in FCCs • 973
- characters in object names, unsupported • 75
- Check the Web Server's Certificate Expiration • 930
- checkForEmptyValues method • 966
- class
 - in CSS • 940
 - name • 577
 - of an object • 577
- Class Definitions • 940
- Cluster Setup dialog • 544
- Cluster Setup Dialog • 544
- collapsing objects • 68
- Collect Additional Attributes • 254, 963
- collector pages
 - adding new attributes • 966
 - description • 949
 - manage_data.jsp • 949
 - user_data_collector.jsp • 949
- color
 - background color style in CSS • 937
 - border color style in CSS • 937
- COM (Component Object Model) • 92
- Combined Central and Local Configuration • 122
- Component Object Model See COM • 92
- components, SiteMinder • 33
- Condition Group Box • 740, 747
- Confidence Levels and SiteMinder Authorization • 1017, 1025
- Confidence Levels in Policies • 378**
- configuration files
 - SmHost.conf • 106
 - WebAgent.conf • 106
- configuration objects
 - Agent • 129
- Configuration of Novell eDirectory LDAP Server Connections • 166
- Configuration Order • 44

Configuration Overview • 96

Configure a Basic Authentication Scheme • 242

Configure a Basic Over SSL Authentication Scheme • 244

Configure a Certificate Mapping • 294

Configure a Custom Agent Type • 136

Configure a Custom Authentication Scheme • 290

Configure a Directory Mapping • 215

Configure a Domain Administrator • 228

Configure a Global Active Policy • 454

Configure a Global Response • 445

Configure a Global Web Agent Response Attribute • 447

Configure a Kerberos Authentication Scheme • 999

Configure a LanMan Directory Connection • 934

Configure a LanMan User Directory Connection • 935

Configure a Multi-Value Parameter • 116, 132

Configure a Nested Realm • 324

Configure a New SQL Query Scheme • 197

Configure a Novell eDirectory LDAP Connection in Policy Server User Interface • 169

Configure a Policy • 378

Configure a Policy Domain • 309

Configure a Policy Server Object • 75

Configure a RADIUS Agent • 134

Configure a RADIUS CHAP/PAP Authentication Scheme • 266

Configure a RADIUS Response Attribute • 363

Configure a RADIUS Server Authentication Scheme • 267

Configure a Realm • 320

Configure a Realm for a 4.x Affiliate Agent • 323

Configure a Realm Protected by a RADIUS Agent • 322

Configure a Realm Protected by a SiteMinder Web Agent • 321

Configure a Response • 357

Configure a Response Attribute that Contain Variables • 365

Configure a Response Group • 373

Configure a Rule for an Affiliate Agent • 340

Configure a Rule for Authentication Event Actions • 336

Configure a Rule for Authorization Event Actions • 337

Configure a Rule for Impersonation Event Actions • 339

Configure a Rule for Web Agent Actions • 334

Configure a SafeWord Server and HTML Forms Authentication Scheme • 270

Configure a SafeWord Server Authentication Scheme • 268

Configure a SecurID and HTML Forms Authentication Scheme • 275

Configure a SecurID Authentication Scheme • 274

Configure a System Administrator • 228

Configure a Web Agent • 120

Configure a Web Agent Response Attribute • 360

Configure a Windows Authentication Scheme • 260

Configure a Windows Directory Connection • 205

Configure Active Directory Connections • 171

Configure Administrator Identity and Authentication Details • 226

Configure Advanced Password Options • 491

Configure Agent Identities for Non-Web Agents • 133

Configure an Active Policy • 395

Configure an Active Rule • 346

Configure an Active Rule as part of a Global Rule • 445

Configure an ADAM Directory Server • 177

Configure an Administrator • 226

Configure an Affiliate Agent Response Attribute • 362

Configure an Agent Configuration Object for a Web Agent • 126

Configure an Agent Group • 139

Configure an Agent Object for a 4.x Web Agent Identity • 124

Configure an Agent to Use startimp.fcc • 466

Configure an Anonymous Authentication Scheme • 289

Configure an Encrypted Parameter • 116, 131

Configure an HTML Form Authentication Scheme • 257

Configure an Impersonation Authentication Scheme • 293, 466

Configure an MS Passport Authentication Scheme • 264

Configure an smauthetsso Custom Authentication Scheme • 992

Configure an smetssockie Web Agent Active Response Attribute • 990

Configure an X.509 Certificate and Basic Authentication Scheme • 280

Configure an X.509 Certificate and HTML Forms Authentication Scheme • 285

Configure an X.509 Certificate Authentication Scheme • 279

Configure an X.509 Certificate or Basic Authentication Scheme • 283

Configure an X.509 Certificate or HTML Forms Authentication Scheme • 288

Configure Anonymous LDAP Access on Novell eDirectory • 167

Configure Authentication and Validation Directory Mapping • 217

Configure Both Load Balancing and Failover for User Directories • 183

Configure CA, Oracle, and IBM Directory Server LDAP Directory Connections • 160

Configure Certificate Revocation List Checking • 297

Configure Custom Directory Connections • 209

Configure Domino Directory Connections • 163

Configure Global Policies • 440

Configure Global Policy Objects • 449

Configure Kerberos External Realm on Windows Host • 1001

Configure NetWare • 167

Configure ODBC Data Sources for Failover • 184

Configure ODBC Directory Connections • 183

Configure Online Certificate Status Protocol Checking • 299

Configure Oracle Internet Directory Connections • 164

Configure Password Composition • 487

Configure Password Expiration • 485

Configure Password Restrictions • 490

Configure Policies for Impersonation • 471

Configure Policy Server Clusters for a Host Configuration Object • 117

Configure Registration Services for Different Languages • 971

Configure Registration Services To Use The Appropriate FCC Files • 973

Configure Registry Keys for a LanMan Directory Connection • 934

Configure Regular Expression Matching • 489

Configure Response Attribute Caching • 363

Configure Single Sign-On from CA SSO Client to SiteMinder • 987

Configure Single Sign-On from CA SSO to SiteMinder • 988

Configure Single Sign-On from SiteMinder to CA SSO • 984

Configure SQL Query Schemes • 197

Configure SQL Query Schemes for Authentication via Stored Procedures • 200

Configure the ADAM User Store in the Policy Server • 179

Configure the IIS Web Server to use SSL • 926

Configure the Netscape Web Server to use SSL • 925

Configure the Scheme Setup Tab • 630

Configure the User Directory Connection for SSL • 195

Configure the Web Service Variable Resolver • 432

Configure Time Restrictions for Global Rules • 444

Configure User Directory Connections • 156

Configure Web Agents Centrally • 121

Configure Web Agents Locally • 121

Configure Your Test Environment Agent • 518

configuring

- active policies • 395, 910
- active policies for global policies • 454
- active rules • 346
- active rules for global rules • 445
- administrative user lookup • 600
- administrator scope • 227, 602
- administrator tasks • 602
- administrators • 226, 326
- advanced password policy options • 491
- advanced policy options • 391
- Affiliate Agent response attribute • 362
- Agent groups • 139
- authentication schemes • 607
- authorization event rule • 337
- authorization event rule for global policies • 443
- basic authentication schemes • 610
- custom authentication schemes • 644
- directories for failover, ODBC, See also ODBC • 574
- directory mapping • 215
- eTrust SSO
 - with SiteMinder • 984
- eTrust SSO client
 - with SiteMinder • 987
- global responses • 445
- LanMan user directories • 934
- nested realms • 324
- ODBC directories, failover, See also failover • 574
- ODBC user directories • 183, 204, 209
- password authentication • 600
- password composition requirements • 487
- password regular expression matching • 488
- password restrictions • 490

- policies • 378
- policy domains • 309
- RADIUS Agent response attribute • 363
- RADIUS Agents • 134
- RADIUS CHAP/PAP authentication schemes • 622
- RADIUS server authentication schemes • 623
- realms • 320
- response attributes • 359
- response attributes for global responses • 446
- response groups • 373
- responses • 357
- rule actions for Web Agents • 334
- rule for Affiliate Agent • 331, 340
- rule, authentication events • 336
- rule, authentication events for global policies • 442
- SafeWord Server authentication schemes • 624, 625
- SecurID authentication schemes • 627
- session
- properties • 321, 322
 - SiteMinder
- with eTrust SSO • 984
- with eTrust SSO client • 987
 - smauthetsso custom authentication scheme • 992
- time restrictions • 344
- time restrictions for global policies • 444
- user directories, WinNT, See also WinNT • 557
- Web Agent response attribute • 360
- Web Agent response attribute for global responses • 447
- Web Agent rule • 334
- Web Agents • 123, 124
- WinNT user directories • 557
- X.509 client certificate and basic authentication • 632, 637
- X.509 client certificate authentication schemes • 631
- X.509 client certificate or basic authentication scheme • 634, 639, 642
- Connect the Policy Server to an Active Directory Global Catalog • 176
- Contact CA Technologies • iii
- containers
 - specifying • 946
- Content Minimums Group Box • 778
- cookie domains
 - configuring HTML forms auth for • 613

- configuring SafeWord HTML forms auth for • 626
- supporting • 90
- cookies
 - session, to store user credentials • 91, 236, 244, 252, 613, 642, 686
- Create a Custom Agent Object for the Agent Identity • 137
- Create a Global Rule for Authorization Event Actions • 443
- Create a Host Configuration Object • 112
- Create a Host Configuration Object Without Using an Existing Object • 113
- Create a Novell eDirectory User Account for SiteMinder Administration • 169
- Create a Realm • 312
- Create a Rule Group • 350
- Create a SiteMinder User For Connecting to the ADAM User Store • 178
- Create a Variable • 433
- Create an ADAM Instance and User Store Connection • 177
- Create an Agent Configuration Object • 128
- Create an Agent Configuration Object without Using an Existing One • 129
- Create an Agent Object to Establish a Web Agent Identity • 123
- Create and Manage the Key Database Using Smkeytool • 417
- Create and Use a Localized Password Services Properties Files • 501
- Create Custom Agents • 136
- Create Custom Certificate Mappings • 533
- Create Password Policies • 483
- Create the Certificate Database Files • 190
- credentials
 - LDAP, See also LDAP • 550, 555, 569
 - user directory, WinNT, See also WinNT • 559
- CRL checking
 - described • 296
- CSS
 - darkBackground style • 937
 - definition • 937
 - definition of classes • 940
 - FieldSmallTagFont style • 937
 - FieldText style • 937
 - FormTextFont style style • 937
 - FormTitleFont style • 937
 - lightBackground style • 937
 - PageTitle style • 937

- RequiredFieldFont style • 937
- SectionHeader style • 937
- SmallLinkFont style • 937
- style samples • 937
- using for pages • 941
- WarningFont style • 937
- Current Setup Group Box • 573, 575
- Custom Attribute Mappings for Testing • 532
- Custom Authentication Scheme Library Writing and Installation • 248
- custom authentication schemes • 236, 290, 644
 - configuring • 644
 - prerequisites • 290
- Custom Authentication Schemes • 236, 290
- Custom Certificate Mapping for Multiple Attributes of the Same Type • 304
- Custom Directory Overview • 156
- custom mapping expressions • 300
- Custom Mapping Expressions • 300
- Custom Namespace Group Box • 563
- Custom Scheme Prerequisites • 290
- Customize Session Validity Duration • 838
- Customize the Results Per Page • 957
- Customize the Search Results • 954
- Customizing Registration Services • 937
- Customizing Registration Services Overview • 937

D

- darkBackground style • 937
- data source name • 184
- Default Resource Protection Group Box • 684
- Default Super User Administrator Account • 63
- Define Access Control Requirements • 58
- Define Implementation Requirements • 58
- Define Task-Assessment Requirements • 57
- Define the Same User Directory Connection in Multiple Policy Stores • 157
- delegating sessions • 92
- Delete a Directory Mapping • 217
- Delete a Host Configuration Object • 119
- Delete a Policy Server Object • 81
- Delete an Agent Configuration Object • 132
- Delete Dialog • 594
- Delete Multiple Objects • 82
- Delete Trusted Host Objects • 111
- deleting
 - administrators • 229
 - global rule • 445

- multiple objects • 82
- objects • 81
- policies • 405
- policy domains • 313
- realms • 325
- response groups • 374
- rule groups • 351
- rules • 347
- deny access rule • 330
- description
 - smetssockie Web Agent active response • 990
- Determine SSL Connection Ability • 923
- dialog boxes
 - specifying titles for • 942
- dialog reference
 - Active Policy Editor • 910
 - Active Rule Editor • 909
 - Add an IP Address • 916
 - Add Directory Mapping • 592
 - Agent Configuration Object • 540
 - Agent Group • 541
 - Agent Group Items • 544
 - Cluster Setup • 544
 - Directory List • 582
 - Directory Mappings • 590
 - Domain • 675
 - Edit Directory Mapping • 593
 - Expression Editor • 738
 - Global Policy • 761
 - Global Response • 755
 - Global Response Attribute • 757
 - Global Rule • 751
 - Host Configuration Object • 538
 - LDAP Setup • 571
 - LDAP/AD • 584
 - ODBC Setup • 574
 - Password Policy • 769
 - Policy Dialog • 733
 - Policy Expression • 745
 - POST Variable Editor • 725
 - Realm • 681
 - Response • 705
 - Response Attribute Editor • 707
 - Rule • 691
 - Rule Group Items • 702
 - Rule Group Properties • 698
 - Rule Items • 742, 765
 - Search Database • 586
 - Search WinNT Directory • 587

-
- Set Global Response • 767
 - Set Response • 744
 - SQL Query Scheme • 576
 - Static Variable Editor • 717
 - Time • 905
 - User Context Variable Editor • 721
 - User directory • 545
 - User Lookup • 589
 - User/Groups • 912
 - Web Services Variable Editor • 727
 - Dictionary Group Box • 784
 - digest authentication schemes • 236
 - Digest Authentication Schemes • 236
 - directory
 - attributes • 552, 556, 570
 - setup, LDAP, See also LDAP • 548
 - setup, ODBC, See also ODBC • 554
 - setup, WinNT, See also WinNT • 558, 562
 - directory attributes • 158
 - Directory List dialog • 582
 - Directory List Dialog • 582
 - Directory List Dialog Fields and Controls • 583
 - Directory List Dialog Prerequisites • 583
 - directory mapping • 213, 357
 - authorization
 - directory, assigning • 216
 - configuring • 215
 - examples • 219
 - responses • 222
 - Directory Mapping • 213
 - Directory Mapping and Responses • 222
 - Directory Mapping by Universal ID • 221
 - Directory Mapping Case Sensitivity • 221
 - Directory Mapping Examples • 219
 - Directory Mapping for Hierarchical Policies • 413
 - Directory Mapping Group Box • 688
 - Directory Mapping Overview • 213
 - Directory Mappings dialog • 590
 - Directory Mappings Dialog • 590
 - Directory Mappings Dialog Fields and Controls • 591
 - Directory Mappings Dialog Prerequisites • 591
 - directory namespace • 723
 - directory path • 723
 - directory server • 723
 - definition • 482
 - password policies, effect on password service • 482
 - types • 482
 - Directory Setup Group Box • 548, 554, 562, 567
 - Directory Topology and LDAP Referrals • 150
 - Disable Global Policy Processing for a Domain • 310
 - disabling
 - global rules • 441
 - rules • 343
 - distinguished name. See DN • 288
 - DMS
 - configuration wizard • 69
 - customizable components • 937
 - properties files • 941
 - using the CSS • 941
 - dms_en_US.properties • 942
 - adding new attributes • 965
 - adding tasks • 960
 - customizing links • 958
 - dms_style.css • 940
 - dmserrorpage.jsp • 970
 - modifying encoding • 974
 - DN
 - attributes • 359
 - creating Anonymous authentication scheme • 288
 - user context variable • 723
 - DN Attribute • 589
 - domain administrator • 227, 228
 - Domain Deletion • 313
 - Domain dialog • 675
 - Domain Dialog • 675
 - Domain Dialog Fields and Controls • 676
 - Domain Dialog Prerequisites • 675
 - Domain Dialog Reference • 675
 - Domain Editing • 313
 - Domain Manager For These Domains Dialog • 604
 - Domain Manager For These Domains Dialog Fields and Controls • 604
 - Domain Manager For These Domains Dialog Prerequisites • 604
 - Domain Type Group Box • 676
 - Domains • 307
 - Domains and User Membership • 308
 - Domains tab, described • 65
 - Domino
 - LDAP • 163, 174
 - Domino Directory Connections Prerequisites • 163
 - Duplicate a Host Configuration Object • 112
 - Duplicate an Agent Configuration Object • 129
 - Dynamic Host Configuration Object (HCO) Updates • 118
-

E

- Edit a Host Configuration Object • 115
- Edit a Plain Parameter • 130
- Edit an Existing Policy Server Object • 78
- Edit an Existing SQL Query Scheme • 199
- Edit AuthValidate Directory Mapping Dialog • 597
- Edit AuthValidate Mapping Directory Dialog Fields and Controls • 597
- Edit Directory Mapping dialog • 593
- Edit Directory Mapping Dialog • 593
- Edit Directory Mapping Fields and Controls • 594
- Edit Multiple Objects • 79
- Edit Parameter Dialog • 543
- Edit Parameter Dialog—Encrypted Radio Button Selected • 543
- Edit Parameter Dialog—Multi-value Radio Button Selected • 543
- Edit Parameter Dialog—Plain Radio Button Selected • 543
- editing
 - multiple objects • 79
 - objects • 78
 - policy domains • 313
 - realms • 325
 - response groups • 374
 - responses • 366
 - rule groups • 351
- Effective Starting Date Group Box • 907
- Effects of Authentication Scheme Protection Levels • 460
- Effects of FCCCompatMode • 470
- EJB • 92
- Employee Accesses an Engineering Realm Resource • 220
- Enable a Web Agent • 133
- Enable and Disable Global Policies • 450
- Enable and Disable Global Rules • 441
- Enable and Disable Policies • 390
- Enable and Disable Rules • 343
- Enable Caching for a CA Directory User Store • 162
- Enable Dynamic Host Configuration Object (HCO) Updates • 119
- Enable Impersonation through an .fcc File • 466
- Enable LegacyCertMapping Registry Key • 302
- Enable Non-browser Client Support • 258
- Enable the Web Server to Trust Client Certificates in Apache • 927
- Enable the Web Server to Trust Client Certificates in Netscape • 925
- Enable the Web Server to Trust Client Certificates in Windows • 926
- Enable User Store DSA Parameters • 161
- EnableCustomExprOnly Registry Key • 301
- enabling
 - global rules • 441
 - rules • 343
 - user-initiated password changes • 492
- encoding
 - changing in JSPs • 974
 - specifying in FCCs • 973
- encoding, character • 970
- Enhance Policy Server's LDAP Authorization Performance • 386
- Enhanced Active Directory Integration Considerations • 510
- Enhanced Active Directory Integration with Password Services • 510
- Enhanced LDAP Referral Handling • 152
- Enterprise Java Beans. See EJB • 92
- entitlements • 411
 - access control • 353
 - from nested realms • 412
 - in access roles • 59
 - to resources • 38, 405
- error messages
 - in locale properties file • 942
- Error Not Found Message Received • 932
- Establish Trust for the Netscape Certificate Authority • 926
- establishing an SSL connection
 - X.509 client certificate authentication • 278
- eTelligent Rules • 421
 - Boolean logic, using • 421
 - introduced • 421
 - reducing policies • 421
- eTelligent Rules Configuration • 422
- eTelligent Rules Properties Files • 422
- eTrust SSO
 - configuring with SiteMinder
- procedures • 984
 - integrating with SiteMinder • 984
- overview • 977
 - SiteMinder integration examples • 978
- eTrust SSO client
 - configuring with SiteMinder
- procedures • 987

- integrating with SiteMinder • 987
- eTrust SSO Web Agent
 - configuring with SiteMinder
- procedures • 988
 - integrating with SiteMinder • 988
- events
 - authentication • 331
 - authorization • 337
 - authorization for global rule • 443
- Events Group Box • 688
- Example
 - Adding a Register New Users Link • 962
 - Collecting the Drink Attribute • 968
- Example .fcc Files for Impersonation • 470
- Example 1 • 302
 - User Accesses SiteMinder-Protected Resource Before CA SSO • 979
- Example 2 • 303
 - Authenticated CA SSO Client User Accesses SiteMinder Resource • 980
- Example 3 • 303
 - User Accesses WAC-Protected Resource Before SiteMinder • 982
- Example of Using Central and Local Configuration • 123
- Examples • 320
- Exclude a User or Group from a Policy • 381
- Exclude Roles in a Policy • 388
- Expand and Collapse Objects • 68
- expanding objects • 68
- Expiration Date Group Box • 907
- Expression Editor • 738
- Expression Editor dialog • 738
- Expression Editor Fields and Controls • 739
- Expression Editor Prerequisites • 738
- expressions
 - regular • 693, 752
 - variable • 427
- Expressions in Policies • 413
- Extend the Object Class • 963

F

- failover
 - configuring ODBC directories • 184, 574
 - enabling • 180
 - WinNT user directories • 208
- Failover for Windows User Directories • 935
- Failover for WinNT User Directories • 208

- FCC
 - described • 249
 - localizing DMS • 973
 - specifying in authentication schemes • 973
 - specifying language encoding • 973
- FCC Directives for Impersonation • 468
- features, Policy Server • 34
- Federation Attribute Variable Editor Reference • 896
- Federation Authentication Scheme Dialog--Scheme Setup Tab • 879
- Federation Security Services Authentication Schemes • 291
- fields
 - adding to collect new attributes • 966
 - modifying in search • 950
 - specifying in locale properties file • 942
 - specifying style for • 937
- FieldSmallTagFont • 937
- FieldText style • 937
- Flush a Realm from the Resource Cache • 326
- flushing realms from resource cache • 326
- font
 - field names • 937
 - linking CSS to pages • 941
 - links • 937
 - medium headers • 937
 - page titles • 937
 - required fields • 937
 - section headers • 937
 - warning messages • 937
- forgotpassword.jsp • 949
- forgotten password
 - newpassword.jsp • 949
 - submitusername.jsp • 949
- form post variable
 - Description • 726
 - Form Field Name • 726
 - Name • 726
- Form Post Variables • 427
- Forms Credential Collector. See FCC • 249
- Forms for Initiating the Impersonation Process • 475
- Forms Support for Activating New User Accounts • 276
- Forms Support for Re-activating and Verifying SecurID Users • 276
- FormTextFont style • 937
- FormTitleFont style • 937
- Functionality Test Results • 525
- functionality tests, description • 523

G

- General Information About LDAP • 143
- Get rule action • 330, 695
- Global Objects • 42
- global policies
 - adding responses to • 449
 - adding rules to • 449
 - configuring an active policy • 454
 - enabling and disabling • 450
 - introduced • 42
- Global Policies • 435
- Global Policies, Rules, and Responses • 435
- Global Policies, Rules, and Responses Dialogs
 - Reference • 751
- Global Policy dialog • 761
- Global Policy Dialog • 761
- Global Policy Dialog Fields and Controls • 761
- Global Policy Dialog—Advanced Tab • 764
- Global Policy Dialog—IP Addresses Tab • 763
- Global Policy Dialog—Rules Tab • 762
- Global Policy Dialog—Time Tab • 764
- Global Policy Dialog—Users Tab • 762
- Global Policy Object Characteristics • 436
- Global Policy Object Prerequisites • 761
- Global Policy Processing • 440
- Global Response Attribute Editor dialog • 757
- Global Response Attribute Editor Dialog • 757
- Global Response Attribute Editor Dialog - Advanced Tab • 760
- Global Response Attribute Editor Dialog - Attribute Setup Tab • 758
- Global Response Attribute Editor Dialog
 - Prerequisites • 757
- Global Response Attribute Types • 446
- Global Response dialog • 755
- Global Response Dialog • 755
- Global Response Dialog Fields and Controls • 755
- Global Response Object Prerequisites • 755
- Global Response Objects • 445
- global responses
 - configuring • 445
 - configuring attributes • 446
 - introduced • 42
- Global Rule Deletion • 445
- Global Rule dialog • 751
- Global Rule Dialog • 751
- Global Rule Dialog Fields and Controls • 751
- Global Rule Object Prerequisites • 751

- Global Rule Objects • 440
- global rules
 - configuring active rules • 445
 - deleting • 445
 - disabling • 441
 - enabling • 441
 - enabling and disabling • 441
 - introduced • 42
 - time restrictions, configuring for • 444
- Global Rules for Authentication Event Actions • 442
- Global Unique Identifier. See GUID • 288
- global.properties file
 - adding tasks • 960
 - customizing tasks • 959
 - description (object) • 954
 - description (presentation) • 943
 - setting search limits • 954
- GMT, configuring time relative to • 83
- Greenwich Mean Time. See GMT • 83
- Group Members Group Box • 701
- group name • 577
- group.js • 966
- group.properties
 - description, object properties • 944
- Grouping Resources in Realms • 315
- groups
 - collecting new attributes • 963
 - specifying attributes • 944
 - specifying container None • 946
- GUID
 - described • 288

H

- Help menu • 69
- host configuration file • 106
- host configuration object
 - description • 106
 - introduced • 39
- Host Configuration Object dialog • 538
- Host Configuration Object Dialog • 538
- Host Configuration Object Dialog Fields and Controls • 538
- Host Configuration Object Dialog—Clusters Tab • 539
- Host Configuration Object Dialog—General Tab • 539
- Host Configuration Object Prerequisites • 538
- Hourly Restrictions Group Box • 908

- How a Change Password Link is Added • 492
- How a Host Configuration Object is Configured for a Trusted Host • 111
- How a Security Model Is Established • 54
- How a Security Model Requirements Table Is Created • 55
- How a User Session Begins • 89
- How a User Session Ends • 94
- How a User Session Is Validated • 91
- How a Windows User Security Context Is Obtained • 94
- How Agent Key Management and Session Timeouts are Coordinated • 93
- How an Impersonation Session is Initiated • 463
- How Containers are Specified for Users • 946
- How Content Is Personalized • 51
- How Name/Value Pairs are Generated in FCC Files • 252
- How Organization and Resource Requirements Are Established • 55
- How Organization Security Requirements Are Defined • 56
- How Password Services Work • 480
- How Persistent Sessions for User Security Contexts Are Maintained • 95
- How Privileges Are Established • 51
- How Resources Are Identified by Agent, Realm, and Rule • 316
- How Role-Based Access Control Is Implemented • 59
- How Rule and Policy Time Restrictions Interact • 395, 454
- How Rules Work as Part of a Policy • 328
- How Session Information Is Delegated • 92
- How Sessions Across Multiple Cookie Domains Are Maintained • 90
- How Sessions Across Realms Are Maintained • 90
- How Sessions Are Managed • 52
- How Sessions Are Revalidated • 95
- How SiteMinder Confirms the Validity of Certificates • 296
- How SiteMinder Establishes an SSL Connection for Certificates • 278
- How SiteMinder Identifies Users Based on Certificates • 278
- How SiteMinder Is Configured to Provide a Windows User Security Context • 96
- How SiteMinder Manages User Sessions • 87
- How SiteMinder Processes Responses • 354
- How SiteMinder Uses UIDs • 212

- How the End-user Experience Is Managed • 51
- How the Policy Server Binds to an LDAP User Store • 153
- How the Policy Server Processes Rules • 328
- How the Policy Server Processes Variables • 428
- How the Policy Server Processes Variables Contained in Policy Expressions • 428
- How the Policy Server Processes Variables contained in Responses • 430
- How the Web Agent and Policy Server Calculate Time • 83
- How to Configure an LDAP User Directory Connection over SSL • 188
- How To Configure SiteMinder Kerberos Authentication • 996
- How to Work with the Object Pane • 68
- HTML forms
 - authentication schemes • 234
 - authentication schemes, described • 244, 612
 - forms-based authentication
- installation components • 247, 269, 275
- prerequisites, authentication schemes • 247, 292
- writing and installing authentication scheme libraries • 248
- HTML Forms Authentication Schemes • 244
- HTML Forms Authentication Templates • 248
- HTML Forms Scheme Prerequisites • 247
- HTML Forms-based Authentication Schemes • 234
- HTML templates, password services • 497
- HTTP Header Dialog • 730
- HTTP Header Dialog Fields and Controls • 731
- HTTP Header Dialog Prerequisites • 731

I

- identifiers
 - in presentation properties files • 943
- Identify Resources and Roles • 57
- identifying a resource • 316
- identity management
 - access roles • 59
- Identity Manager Environment Objects • 60
- IdentityMinder
 - access roles • 59
 - and SiteMinder • 59
- IdentityMinder Environments
 - introduced • 39
- idle timeout
 - about • 52

- session • 684
- Impersonation • 457
- Impersonation Authentication Schemes • 292
- Impersonation Event Configuration • 471
- Impersonation Events • 333
- Impersonation Logoff Forms • 477
- Impersonation Overview • 457
- Impersonation Process • 458
- Impersonation Realms and Events • 461
- Impersonation Result Forms • 476
- Impersonation Scheme Prerequisites • 292
- Implement an Operation Mode • 109
- implementation
 - defining security requirements • 54
 - personalization requirements • 58
- Implementing Policy-based Security • 47
- Important Considerations • 482
- inc_header.jsp
 - including the CSS • 941
 - modifying encoding • 974
 - specifying character encoding • 970
- Include Header Values in the FCC Namespace • 255
- Incorrect Password Group Box • 775
- Incorrect Password Message Does Not Appear • 515
- Infix Notation Group Box • 741, 748
- initiator pages
 - description • 949
 - launch.jsp • 949
 - manage_data.jsp • 949
 - newpassword.jsp • 949
 - submitusername.jsp • 949
 - table of • 949
 - welcome.jsp • 949
- Install the IIS Web Server Certificate • 927
- Install the Netscape Web Server Certificate • 924
- Install the NSS Utility • 190
- Installing the Apache Web Server Certificate • 928
- Is field
 - customizing • 950
- ISO-8859-1 • 974
- Issuer DN Mapping • 532

J

- JavaScript
 - defining a new task • 960
- JavaServer Pages
 - See JSPs • 949
- JavaServer Pages (JSPs) • 949

JSPs

- changing encoding • 974
- collector pages, description • 949
- forgotpassword.jsp • 949
- initiator pages list • 949
- initiator pages, description • 949
- locale properties file • 942
- manage_data_submitter.jsp • 949
- object profile pages, description • 949
- object table pages, description • 949
- overview • 949
- password services forms • 479
- style formatting • 937
- submitter pages • 949
- user_data_collector.jsp • 949
- user_data_submitter.jsp • 949

JVMOptions.txt file

- for eTelligent Rules • 422

JVMOptions.txt File • 423

K

- KDC Configuration on UNIX Example • 1005

- KDC Configuration on Windows 2003 Example • 1002

- Kerberos Authentication Configuration at the Policy Server • 997

- Kerberos Authentication Configuration at the Web Server • 998

- Kerberos Authentication Configuration at the Windows Workstation • 999

- Kerberos Configuration at the Policy Server on UNIX Example • 1008

- Kerberos Configuration at the Policy Server on Windows Example • 1006

- Kerberos Configuration Examples • 1002

- Kerberos KDC Configuration at the Domain Controller • 996

- Kerberos Overview • 995

- Keys and Certificates in the Key Database • 415

L

- language encoding

- changing in JSPs • 974

- specifying in FCCs • 973

- languages

- configuring • 971

- LanMan Directory Connection Prerequisites • 933

- LanMan user directories

-
- configuring • 934
 - described • 207
 - prerequisites • 933
 - LanMan User Directories • 207, 933
 - LanMan User Directory Search Criteria • 936
 - launch application
 - welcome.jsp • 949
 - launch.jsp
 - adding tasks • 960
 - customizing links • 958
 - description • 949
 - LDAP
 - attributes, specifying in object properties files • 944
 - credentials • 550, 555, 569
 - directory setup • 548
 - Domino • 163, 174
 - referrals • 150
 - search filters • 143
 - user directories, collecting new attributes • 963
 - user directories, extending object classes • 963
 - user directories, storing profiles without containers • 946
 - user directories, user disambiguation • 143
 - user directories, user lookup via DN • 143
 - user directories, user lookup via search expression • 143
 - user directories, working with • 142
 - user directory searches • 584
 - user directory searches, See also user directory searches • 584
 - User DN Lookup Group Box • 550
 - LDAP Directory Attribute Search Group Box • 585
 - LDAP Overview • 142
 - LDAP Referrals • 150
 - LDAP Search Group Box • 549, 567
 - LDAP Setup dialog • 571
 - LDAP Setup Dialog • 571
 - LDAP Setup Dialog Fields and Controls • 572
 - LDAP Setup Dialog Prerequisites • 571
 - LDAP User DN Lookup Group Box • 550, 568
 - LDAP Users Do Not Disable • 514
 - LDAP/AD dialog • 584
 - lightBackground style • 937
 - Lightweight Data Access Protocol. See LDAP • 142
 - Limit Policy Server Search to One User Store during Authentication • 233
 - links
 - customizing • 958
 - in locale properties file • 942
 - specifying style • 937
 - Links Between The Cascading Style Sheet to Registration Services Pages • 941
 - List pane
 - described • 65
 - menu bar • 69
 - using • 68
 - List the Certificates in the Certificate Database • 194
 - load balancing
 - configuring for LDAP directories • 180
 - round robin • 519
 - Load Balancing/Failover Configuration for LDAP Directory Connections • 180
 - Load Users into the ADAM User Store • 179
 - Locale Properties File • 965
 - locale properties files
 - adding new attributes • 965
 - adding tasks • 960
 - customizing links • 958
 - mapping • 947
 - Locale Properties Files • 942
 - Localize CGI-based Password Services • 501
 - Localize Registration Services • 970
 - Localize Servlet-based Password Services • 505
 - localizing DMS
 - locale properties file • 942
 - modifying FCCs • 973
 - overview • 970
 - specifying the FCC • 973
 - localizing password services • 501
 - LoggerConfig.properties
 - for eTelligent Rules • 422
 - sample file • 424
 - logging
 - in Policy Server • 64
 - LoggingOn configuration setting • 424
 - loginandregister-dms-i.fcc
 - specifying SMLOCALE • 973
 - loginandregisterwithforgottenpassword-dms-i.fcc
 - specifying SML • 973
 - login-dms-i.fcc
 - adding SMLOCALE • 973
- ## M
- manage_data.jsp
 - customizing search, overview • 950
 - description • 949

- displaying search results in separate page • 955
- manage_data_submitter.jsp
 - description • 949
- managedorg.properties
 - description, object properties • 944
- managing users
 - controlling idle timeouts • 52
 - ending sessions • 52
 - end-user experience • 51
 - maximum user timeouts • 52
 - using access control lists • 47
 - using security policies • 47
- Manual Entry • 577
- Map from Issuer DN in Certificate to Directory Type Group Box • 668
- Map Properties Files • 947
- Map Search Specifications for Passport Authentication • 263
- Map to Non-Required Attributes • 305
- Map to the Certificate Serial Number or IssuerDN • 304
- Mapped DNs Group Box • 593, 594
- Mapping Group Box • 669
- mapping properties files
 - overview • 947
- Max
 - Results • 549, 563, 567
 - Time • 549, 563, 567
- max sockets per port, setting (IIS) • 109
- maximum timeout, session • 684
- menu
 - Advanced • 69
 - Edit • 69
 - Help • 69
 - Session • 69
 - Tools • 69
 - View • 69
- menu bar
 - described • 65
- minimum sockets per port
 - setting (IIS) • 109
- Modify an Agent Configuration Object • 130
- Modify CGI-based Password Services HTML Templates • 499
- Modify Servlet-based Password Services JSP Forms • 504
- Modify Strings Displayed in Forms • 505
- Modify the FCC Files • 973
- Modify the Links • 958

- Modify the LoggerConfig.properties File • 424
- Modify the Properties Files • 972
- Modify the Search Fields • 950
- Modify the Search Results Page • 955
- MS Passport Authentication Schemes • 261
- Multiple Cookie Domain Support • 473
- Multiple Instances of a Single Authentication Scheme Configuration • 241

N

- N_MAX_ITEMS • 957
- name
 - form post variable • 726
 - group • 577
 - user, See also user • 577
- Name IDs Dialog Fields for Attribute Queries • 894
- Name, NT Domain • 558
- namespace
 - described • 143
- Namespace, Custom • 562
- NAS client devices • 103
- Navigate to the Active Policy Editor • 911
- Navigate to the Active Rule Editor Dialog • 909
- Navigate to the Add an IP Address Dialog • 916
- Navigate to the Add AuthValidate Directory Mapping Dialog • 596
- Navigate to the Add Directory Mapping Dialog • 592
- Navigate to the Administrator Properties Dialog • 599
- Navigate to the Affiliate Attribute Editor Dialog • 794
- Navigate to the Affiliates Dialog • 787
- Navigate to the Agent Configuration Object Dialog • 540
- Navigate to the Agent Dialog • 536
- Navigate to the Agent Group Dialog • 542
- Navigate to the Attribute Svc Dialog • 889
- Navigate to the Attributes Dialog • 892
- Navigate to the Auth. Scheme Properties Dialog--WS Federation Template • 878
- Navigate to the Authentication Scheme Dialog • 608, 799
- Navigate to the Authentication Scheme Dialog for an Existing Authentication Scheme • 608
- Navigate to the AuthValidate Directory Mappings Dialog • 595
- Navigate to the Certificate Map Test Dialog • 672
- Navigate to the Certificate Mapping Properties Dialog • 667

Navigate to the Certificate Mappings Dialog • 666
Navigate to the Directory List Dialog • 583
Navigate to the Directory Mappings Dialog • 591
Navigate to the Domain Dialog • 675
Navigate to the Domain Dialog for an Existing Policy or Object • 675
Navigate to the Domain Manager For These Domains Dialog • 604
Navigate to the Edit AuthValidate Directory Mapping Dialog • 597
Navigate to the Edit Directory Mapping Dialog • 593
Navigate to the Expression Editor • 739
Navigate to the Global Policy Dialog • 761
Navigate to the Global Response Attribute Editor Dialog • 757
Navigate to the Global Responses Dialog • 755
Navigate to the Global Rule Dialog • 751
Navigate to the Host Configuration Object Dialog • 538
Navigate to the HTTP Header Dialog • 731
Navigate to the LDAP Setup Dialog • 572
Navigate to the ODBC Setup Dialog • 575
Navigate to the Password Policy Dialog to Create a New Policy • 769
Navigate to the Password Policy Dialog to Modify an Existing Policy • 770
Navigate to the Policy Dialog • 734
Navigate to the Policy Expression Dialog • 746
Navigate to the Post Variable Editor Dialog • 726
Navigate to the Realm Dialog to Create a New Realm • 681
Navigate to the Realms Dialog to Edit an Existing Realm • 682
Navigate to the Request Context Variable Editor Dialog • 720
Navigate to the Resource Partner Properties Dialog • 862
Navigate to the Response Attribute Editor Dialog to Create an Attribute • 708
Navigate to the Response Attribute Editor Dialog to Edit an Attribute • 708
Navigate to the Response Dialog • 705
Navigate to the Response Group Items Dialog • 715
Navigate to the Response Group Properties Dialog • 713
Navigate to the Restrictions Dialog • 827
Navigate to the Rule Dialog to Create a Rule • 691
Navigate to the Rule Dialog to Edit an Existing Rule • 692

Navigate to the Rule Group Items Dialog • 702
Navigate to the Rule Group Properties Dialog to Create a Rule Group • 699
Navigate to the Rule Group Properties Dialog to Edit an Existing Rule • 699
Navigate to the Rule Items Dialog • 742, 765
Navigate to the SAML 2.0 Auth Scheme Properties Dialog • 844
Navigate to the SAML Affiliation Dialog • 897
Navigate to the SAML Service Provider Dialog • 815
Navigate to the Search Database Dialog • 586
Navigate to the Search LDAP/AD Dialog • 584
Navigate to the Search WinNT Dialog • 588
Navigate to the Set Global Response Dialog • 767
Navigate to the Set Response Dialog • 744
Navigate to the SQL Query Dialog for an Existing Query Scheme • 577
Navigate to the SQL Query Scheme Dialog • 576
Navigate to the Static Variable Editor Dialog • 718
Navigate to the Time Dialog • 906
Navigate to the User Context Variable Editor Dialog • 722
Navigate to the User Directory Dialog • 546
Navigate to the User Directory Dialog for an Existing User Directory Connection • 547
Navigate to the User Lookup Dialog • 589
Navigate to the Users/Groups Dialog • 913
Navigate to the Web Services Variable Editor Dialog • 727
Navigating to the SAML Service Provider Attribute Editor Dialog • 829
nested realms
 configuring • 324
 rules and • 329
 understanding • 318
Nested Realms • 318
Nested Realms and Resources • 407
Netegrity
 Policy Server User Interface • 63
Netscape Web Agent
 sockets per port, modifying • 109
Network Access Server. See NAS • 103
new socket step
 setting (IIS) • 109
New User Passwords are Rejected • 513
neworg.jsp • 966
newpassword.jsp
 description • 949
newuserjs.jsp • 966

-
- n-level radio button
 - customizing with searchNoScope • 950
 - Non-Persistent and Persistent Cookies • 85
 - Non-Persistent and Persistent Sessions • 85
 - non-persistent sessions
 - configuring • 686
 - NT Domain Name • 558
 - NT LAN Manager. See NTLM • 234
 - NT1 • 557
 - NTLM authentication schemes • 234
 - number of items • 73
- ## O
- object class
 - extending • 963
 - object names, unsupported characters in • 75
 - object pane
 - described • 65
 - Domains tab • 65
 - System tab • 65
 - working with the • 68
 - object profile pages
 - description • 949
 - object properties files
 - adding new attributes • 965
 - mapping • 947
 - sample entry • 944
 - Object Properties Files • 944, 965
 - object table pages • 949
 - objects
 - configuring • 75
 - deleting • 75, 81
 - deleting multiple objects • 82
 - editing • 75, 78
 - editing multiple objects • 79
 - expanding and collapsing • 68
 - icons, Policy Server • 74
 - Policy Server, introduced • 38
 - system • 39
 - viewing • 75, 76
 - viewing multiple • 77
 - Obtain the Session Specification using an FCC • 469
 - OCSP checking • 296
 - ODBC • 554
 - configuring directories for failover • 184, 574
 - configuring user directories • 183, 204, 209
 - Data Source • 575
 - directory setup • 554
 - supported types • 153
 - user directories
 - policy bindings • 402
 - user directories, working with • 153
 - user directory searches • 586
 - user directory searches, See also user directory searches • 586
 - ODBC Connection Pooling • 203
 - ODBC Database Overview • 153
 - ODBC Setup dialog • 574
 - ODBC Setup Dialog • 574
 - ODBC Setup Dialog Fields and Controls • 575
 - ODBC Setup Dialog Prerequisites • 574
 - ODBC1 • 553
 - OnAccessAccept authorization event
 - described • 333
 - fire rule on • 695, 753
 - OnAccessReject authorization event
 - described • 333
 - fire rule on • 695, 753
 - OnAuthAccept authentication event
 - described • 331
 - fire rule on • 411, 695, 753
 - OnAuthAttempt authentication event
 - described • 331
 - fire rule on • 695, 753
 - OnAuthChallenge authentication event
 - described • 331
 - fire rule on • 695, 753
 - OnAuthReject authentication event
 - described • 331
 - fire rule on • 695, 753
 - OnAuthUserNotFound authentication event
 - described • 331
 - Online Certificate Status Protocol See OCSP checking • 296
 - Operation Mode • 108
 - org.properties
 - description, object properties • 944
 - Organization Administrators
 - presentation privileges • 943
 - organizations
 - collecting new attributes • 963
 - specifying profile attributes • 944
 - ou attribute
 - ou=Groups • 946
 - ou=People • 946
 - ou=Roles • 946
 - ou=Groups • 946

- ou=People • 946
- ou=Roles • 946
- overview • 33
 - authentication schemes • 232, 233
 - password services • 479
 - Policy Server • 35
- Overview • 33, 47, 315, 415, 919, 977
- Overview of Authentication Scheme Processing • 232
- Overview of Servlet-based Password Services JSP Forms • 502

P

- PageTitle style • 937
- PAP • 265
- PAP Overview • 265
- Parameter • 563
- Part of Operating System, Act As • 558
- Passport Authentication Prerequisites • 263
- Passport Authentication Support in the Policy Server • 262
- password
 - newpassword.jsp • 949
 - specifying attributes in user.properties • 944
- Password
 - Attribute • 552, 556, 564, 570
 - Data • 552, 556, 564, 570
- Password Authentication Protocol. See PAP • 265
- Password CGI-based Services Properties Files • 500
- Password Changes are Forced • 514
- Password Expires from Inactivity Group Box • 776
- Password Expires if Not Changed Group Box • 774
- Password Group Box • 601
- Password Length Group Box • 777
- password policies
 - advanced options • 491
 - composition • 487
 - expiration • 485
 - interaction with Directory Server password policies • 482
 - introduced • 39
 - regular expression matching • 488
 - restrictions • 490
- Password Policies • 479
- Password Policy dialog • 769
- Password Policy Dialog • 769
- Password Policy Dialog Box • 769
 - Advanced tab • 491, 770, 785
 - Composition tab • 487, 770, 777
 - Expiration tab • 485, 770, 774
 - General tab • 770, 771
 - Regular Expressions tab • 488, 770, 779
 - Restrictions tab • 490, 770, 782
- Password Policy Dialog Fields and Controls • 770
- Password Policy Dialog Prerequisites • 769
- Password Policy Dialog—Advanced Tab • 785
- Password Policy Dialog—Composition Tab • 777
- Password Policy Dialog—Expiration Tab • 774
- Password Policy Dialog—General Tab • 771
- Password Policy Dialog—Regular Expressions Tab • 779
- Password Policy Dialog—Restrictions Tab • 782
- Password Policy Enabled State Group Box • 772
- Password Policy Priority Group Box • 786
- Password Policy Troubleshooting • 512
- Password Pre-Processing Group Box • 785
- Password Regular Expressions • 488
- Password Self-Changes • 496
- password services • 479
 - CGI-based • 479
 - HTML forms • 479
 - HTML templates • 497
 - JSP forms • 479
 - localizing • 501, 507
 - properties files • 500
 - servlet-based • 479
 - templates • 502
 - time differential effect on • 482
 - user-initiated password changes • 492
- Password Services Dialog Reference • 769
- Password Services Overview • 479
- Password Services with Active Directory Integration • 512
- Passwords MUST Match These Expressions Group Box • 779
- Passwords MUST NOT Match These Expressions Group Box • 781
- Perform a Regression Test • 527
- Persistent Session Group Box • 686
- Point the Policy Server to the Certificate Database • 196
- policies • 375, 733
 - adding a range of IP addresses • 394, 453
 - adding a single IP address • 392, 452
 - adding a subnet mask • 392, 393, 452, 453
 - adding and removing time restrictions • 394
 - adding rules and responses to • 380

-
- adding rules to • 380, 381
 - adding users to a • 379
 - configuring • 378
 - configuring advanced options • 391
 - configuring advanced policy options • 391
 - configuring an active policy • 395, 910
 - deleting • 405
 - enabling and disabling • 390
 - establishing policy bindings • 396
 - excluding users from • 381
 - introduced • 41
 - policy bindings • 377
 - policy processing • 405
 - reducing instances • 421
 - roles-based • 59
 - specifying a host name • 392, 452
 - specifying a range of IP addresses • 394, 453
 - specifying a single IP address • 392, 452
 - specifying a subnet mask • 393, 453
 - specifying IP addresses • 391
 - unprotected • 318
 - Policies • 375
 - Policies and Responses • 408
 - Policies Explanation • 377
 - Policy and Related Dialogs Reference • 733
 - Policy Binding Establishment • 396
 - policy bindings
 - establishing • 396
 - LDAP user directories • 396
 - ODBC user directories • 402
 - WinNT user directories • 400
 - Policy Bindings • 377
 - Policy Bindings for LDAP Directories • 396
 - Policy Bindings for Microsoft SQL Server and Oracle User Directories • 402
 - Policy Bindings for WinNT User Directories • 400
 - Policy Considerations When Configuring Global OnAccessReject Rules • 444
 - Policy Considerations When Configuring OnAccessReject Rules • 338
 - Policy Deletion • 405
 - Policy dialog • 733
 - Policy Dialog • 733
 - Policy Dialog Fields and Controls • 734
 - Policy Dialog Prerequisites • 733
 - Policy Dialog—Advanced Tab • 738
 - Policy Dialog—Expression Tab • 737
 - Policy Dialog—IP Addresses Tab • 736
 - Policy Dialog—Rules Tab • 735
 - Policy Dialog—Time Tab • 737
 - Policy Dialog—Users Tab • 735
 - Policy Domain Objects • 41
 - Policy Domain Overview • 307
 - policy domains • 307
 - assigning administrators • 311
 - assigning user directories • 310
 - configuring • 309
 - creating realms in • 312
 - deleting • 313
 - editing properties • 313
 - introduced • 39
 - understanding • 308
 - user membership in a directory • 308
 - Policy Expression dialog • 745
 - Policy Expression Dialog • 745
 - Policy Expression Dialog Fields and Controls • 746
 - Policy Expression Dialog Prerequisites • 746
 - Policy Overview • 375
 - Policy Processing • 405
 - Policy Processing for Authorized Users • 412
 - Policy Processing for Unauthorized Users • 412
 - Policy Server • 240
 - administrator accounts • 63
 - features • 34
 - menu bar • 69
 - object icons • 74
 - objects <Italics>See objects • 38
 - overview • 33, 35
 - starting Policy Server User Interface • 64
 - time • 83
 - User Interface • 63
 - starting • 64
 - using the SiteMinder Administration window • 65
 - Policy Server Administrator Accounts • 63
 - Policy Server Administrators • 223
 - Policy Server Components • 107
 - Policy Server Does Not Connect to LDAP Stores Using SSL with AKI • 922
 - Policy Server Hangs When Web Agent Connection Is Idle • 921
 - Policy Server Identification • 519
 - Policy Server Information for Version 5 Agents • 520
 - Policy Server Management • 37
 - Policy Server Object Icons • 74
 - Policy Server Object Management Overview • 38
 - Policy Server Object Overview • 39
 - Policy Server Object Types • 39
 - Policy Server Objects for Impersonation • 461
-

- Policy Server Overview • 35
- Policy Server System Management Overview • 37
- Policy Server User Interface • 63
- Policy Server User Interface Overview • 63
- Port Numbers • 181
- Post rule action • 330, 695
- POST Variable Editor dialog • 725
- POST Variable Editor Dialog • 725
- Post Variable Editor Dialog Fields and Controls • 726
- Post Variable Editor Dialog Prerequisites • 725
- Precautions on Deleting or Editing an Administrator • 229
- Prerequisites Before Integrating SiteMinder and CASO • 983
- presentation properties files
 - accessing privilege information • 966
 - adding new attributes • 965
 - adding tasks • 960
 - customizing displayed tasks • 959
 - mapping • 947
 - sample entry • 943
- Presentation Properties Files • 943, 965
- ProcessSOAP rule action • 330, 695
- ProcessXML rule action • 330, 695
- Profile Attributes Group Box • 783
- profile pages
 - adding new attributes • 966
- properties files
 - adding new attributes • 964
 - description • 941
 - locale, overview • 942
 - mapping, overview • 947
 - password services • 500
 - presentation, overview • 943
- Properties Files • 506, 941
- properties.getValue
 - adding a new field • 966
- Property • 723
- property, value of a • 577
- Pros and Cons of Choosing an AD Namespace for an Active Directory Connection • 173
- Pros and Cons of Choosing an LDAP Namespace for an Active Directory Connection • 174
- protection levels • 237
- Protection Levels • 237
- proxy authentication • 235
- proxy authentication schemes • 235
- Proxy Authentication Schemes • 235
- Put rule action • 330, 695

Q

- query
 - schemes, SQL, See also SQL • 555, 576
 - to fetch group names • 577

R

- Radio Button, IP Address • 572
- RADIUS Agent Response Attributes • 357
- RADIUS Agents • 103
 - configuring • 134
 - response attributes • 357
- RADIUS CHAP/PAP authentication schemes • 265
 - configuring • 622
 - prerequisites • 266
- RADIUS CHAP/PAP Authentication Schemes • 265
- RADIUS CHAP/PAP Scheme Overview • 266
- RADIUS CHAP/PAP Scheme Prerequisites • 266
- RADIUS Responses • 356
- RADIUS server authentication schemes • 267, 623
 - configuring • 623
 - prerequisites • 267
- RADIUS Server Authentication Schemes • 267
- RADIUS Server Scheme Prerequisites • 267
- Realm and Resources Group Box • 693
- Realm Deletion • 325
- Realm dialog • 681
- Realm Dialog • 681
- Realm Dialog Fields and Controls • 682
- Realm Dialog Reference • 681
- Realm Dialog—Advanced Tab • 687
- Realm Dialog—Resource Tab • 682
- Realm Dialog—Session Tab • 684
- Realm Editing • 325
- Realm Editing and Deletion • 325
- Realm Object Prerequisites • 681
- realms • 315
 - configuring • 320
 - configuring a nested realm • 324
 - configuring realms for an Affiliate Agent • 323
 - configuring realms protected by a RADIUS Agent • 322
 - configuring realms protected by a Web Agent • 321
 - creating in a policy domain • 312
 - deleting • 325
 - editing • 325
 - introduced • 41
 - maintaining sessions • 90

-
- maximum timeout • 684
 - nested realms, understanding • 318
 - protected • 684
 - selecting a directory mapping • 688
 - selecting a registration scheme • 687
 - session idle timeout • 684
 - session validation period • 686
 - unprotected • 318, 684
 - using to define resources • 57
 - Realms • 315
 - Realms in Request Processing • 320
 - Realms Tab • 678
 - Redirect Group Box • 772
 - referrals, LDAP • 150
 - Register a Trusted Host with the Policy Server • 107
 - Registration Dialog Reference • 903
 - Registration Group Box • 687
 - registration scheme • 65, 225, 321, 687
 - introduced • 39
 - registration services • 34
 - Registration Space Selector page
 - welcome.jsp • 949
 - Registration Spaces
 - mapping to properties files • 947
 - regression tests
 - description • 527
 - performing • 527
 - regspace_locale.properties
 - description • 947
 - regspace_presentation.properties
 - description • 947
 - regspace_tables.properties
 - description • 947
 - regular expressions • 334, 342, 693, 752
 - matching in passwords • 488
 - Regular Expressions for Resource Matching • 342
 - Regular Expressions Syntax • 488
 - Rejected Authentication Attempts • 411
 - Remote Authentication Dial-In User Service. See RADIUS • 103
 - Remote URL group box • 728
 - Remove the Login ID When Redirecting for Password Services • 496
 - Repeating Characters Group Box • 777
 - Report Viewing • 530
 - request context variable
 - action • 721
 - description • 721
 - name • 721
 - property • 721
 - Resource • 721
 - server • 721
 - Request Context Variable Editor Dialog • 719
 - Request Context Variable Editor Dialog Fields and Controls • 721
 - Request Context Variable Editor Dialog Prerequisites • 720
 - Request Context Variables • 426
 - Request Timeout • 108
 - Require Credentials • 563
 - Required Agent Configuration Object Parameters • 127
 - required fields • 75
 - Required Fields • 75
 - RequiredFieldFont style • 937
 - requirements
 - access control • 54, 58
 - defining implementation requirements • 58
 - implementation • 54
 - resource • 54
 - task assessment • 54, 57
 - resource
 - cache • 326
 - defining in realms • 57
 - defining in rules • 57
 - filter • 682, 752
 - identifying • 316
 - matching • 341
 - Resource Group Box • 682, 752
 - Resource Matching and Regular Expressions • 341
 - Resource Partner Attribute Dialog Prerequisites • 871
 - Resource Partner Attribute Dialog--Advanced Tab • 873
 - Resource Partner Attribute Dialog--Attribute Setup Tab • 872
 - Resource Partner Properties Dialog • 861
 - Resource Partner Properties Dialog Fields and Controls • 863
 - Resource Partner Properties Dialog Prerequisites • 861
 - Resource Partner Properties Dialog--Advanced Tab • 874
 - Resource Partner Properties Dialog--Attributes Tab • 871
 - Resource Partner Properties Dialog--General Tab • 868

- Resource Partner Properties Dialog--Name IDs Tab • 866
- Resource Partner Properties Dialog--Signout Tab • 873
- Resource Partner Properties Dialog--SSO Tab • 869
- Resource Partner Properties Dialog--Users Tab • 865
- Resource Protection with a SiteMinder Agent • 138
- Response and Related Dialogs Reference • 705
- response attribute
 - user attribute • 359
- Response Attribute Editor dialog • 707
- Response Attribute Editor Dialog • 707
- Response Attribute Editor Dialog - Advanced Tab • 712
- Response Attribute Editor Dialog - Attribute Setup Tab • 709
- Response Attribute Editor Dialog Prerequisites • 708
- Response Attribute Types • 359
- response attributes • 39, 90, 94, 134, 353, 355, 356, 357, 537
 - Affiliate Agent • 357
 - configuring • 359
 - configuring Affiliate Agent • 362
 - configuring RADIUS Agent • 363
 - configuring Web Agent • 360
 - configuring web agent for global policies • 447
 - DN Attribute • 359
 - RADIUS Agent • 357
 - static • 359
 - Web Agent • 356
- Response Attributes • 356, 359
- Response Attributes for Global Responses • 446
- Response Deletion • 372
- Response dialog • 705
- Response Dialog • 705
- Response Dialog Fields and Controls • 705
- Response Group Deletion • 374
- Response Group Items Dialog • 715
- Response Group Items Dialog Fields and Controls • 715
- Response Group Items Dialog Prerequisites • 715
- Response Group Properties Dialog • 712
- Response Group Properties Dialog Fields and Controls • 713
- Response Group Properties Dialog Prerequisites • 713
- Response Group Property Editing • 374
- response groups • 372
 - configuring • 373
 - deleting • 374
 - editing properties • 374
 - introduced • 41
- Response Groups • 372
- Response Object Prerequisites • 705
- Response Property Editing • 366
- responses • 59, 353
 - active response • 359
 - Affiliate Agent • 355
 - cache value in Agents • 363, 709, 758
 - configuring • 357
 - directory mapping and • 222
 - editing properties • 366
 - introduced • 41
 - RADIUS Agent • 356
 - response attributes • 356
 - types • 355
 - Web Agent • 355
- Responses • 353
- Responses and Directory Mappings • 357
- Responses and Response Groups • 353
- Restricting Impersonation • 460
- restrictions
 - password • 490
- Restrictions Dialog Field and Controls • 870
- Restrictions Dialog Fields and Controls • 828
- Restrictions Dialog for a Resource Partner • 870
- Restrictions Dialog for an SP • 827
- Restrictions Dialog Prerequisites • 828, 870
- Resulting Notation Group Box • 748
- Return Query group box • 729
- return type • 718, 723
- Reuse Group Box • 782
- revoking user sessions • 52
- Risk Scores and Confidence Levels Compared • 1018, 1025
- roles
 - access • 59
 - administrative • 45
 - displaying searchable attributes based on roles • 953
 - specifying presentation privileges • 943
- roles-based access control • 59
- role-specific Task Selector page
 - welcome.jsp • 949
- root, search • 548
- rule actions
 - OnAuthAccept • 331
 - OnAuthAttempt • 331

- OnAuthChallenge • 331
- OnAuthReject • 331
- OnAuthUserNotFound • 331
- Rule Actions • 329
- Rule and Rule Group Dialogs Reference • 691
- Rule Deletion • 347
- Rule dialog • 691
- Rule Dialog • 691
- Rule Dialog Fields and Controls • 692
- Rule Dialog Prerequisites • 691
- Rule Group Deletion • 351
- Rule Group Items dialog • 702
- Rule Group Items Dialog • 702
- Rule Group Items Dialog Fields and Controls • 702
- Rule Group Items Dialog Prerequisites • 702
- Rule Group Overview • 349
- Rule Group Properties dialog • 698
- Rule Group Properties Dialog • 698
- Rule Group Properties Dialog Fields and Controls • 700
- Rule Group Properties Dialog Prerequisites • 699
- Rule Group Property Editing • 351
- rule groups • 349
 - adding or removing a rule • 350
 - deleting • 351
 - editing properties • 351
 - introduced • 41
- Rule Groups • 349
- Rule Items dialog • 742, 765
- Rule Items Dialog • 742, 765
- Rule Items Dialog Fields and Controls • 743, 766
- Rule Items Dialog Prerequisites • 742, 765
- rules • 327, 328
 - actions • 329
 - advanced rule options • 344, 697, 754
 - allow access • 330
 - allow and deny access • 330
 - authentication events • 331
 - authorization events • 333
 - configuring • 334, 343
 - configuring active rules • 346
 - configuring authentication events • 336
 - configuring authentication events for global policies • 442
 - configuring authorization events • 337
 - configuring authorization events for global policies • 443
 - configuring for an Affiliate Agent • 331, 340
 - configuring Web Agent rule actions • 334

- deleting • 347
- disabling • 343
- enable and disable • 343
- enabling • 343
- introduced • 41
- time restrictions, configuring • 344
- understanding processing of • 328
- unprotected • 318
- using to define resources • 57
- Rules • 327
- Rules and Nested Realms • 329
- Rules Overview • 327
- Run a Functionality Test • 523
- Run in Authenticated User's Security Context • 551, 560, 569
- Running Certificate or Basic but Cannot Enter Basic credentials. • 932

S

- SafeWord Server and HTML Forms Authentication Schemes • 269
- SafeWord Server and HTML Forms Scheme Prerequisites • 269
- SafeWord Server authentication scheme
 - configuring • 624, 625
 - prerequisites • 268
- SafeWord Server Authentication Schemes • 268
- SafeWord Server Scheme Prerequisites • 268
- SAML 1.x Affiliate Dialog Reference • 787
- SAML 1.x Authentication Scheme Properties • 799
- SAML 2.0 Attribute Authority Reference • 889
- SAML 2.0 Attribute Query Reference • 891
- SAML 2.0 Auth Scheme Properties Dialog • 844
- SAML 2.0 Auth Scheme Properties Dialog Fields and Controls • 845
- SAML 2.0 Auth Scheme Properties Dialog Prerequisites • 844
- SAML 2.0 Auth Scheme Properties Dialog--Attributes Tab • 854
- SAML 2.0 Auth Scheme Properties Dialog--Backchannel Tab • 854
- SAML 2.0 Auth Scheme Properties Dialog--Encryption Tab • 853
- SAML 2.0 Auth Scheme Properties Dialog--Name IDs Tab • 854
- SAML 2.0 Auth Scheme Properties Dialog--SLO Tab • 852

SAML 2.0 Auth Scheme Properties Dialog--SSO Tab • 847

SAML 2.0 Auth Scheme Properties Dialog--Users Tab • 846

SAML 2.0 Auth. Scheme Properties Dialog--Advanced Tab • 856

SAML 2.0 Authentication Properties--Advanced Tab • 660, 844

SAML 2.0 Authentication Properties--Scheme Setup Tab • 659, 842

SAML 2.0 authentication scheme • 291

SAML 2.0 Authentication Scheme Properties Reference • 841

SAML 2.0 Service Provider Reference • 815

SAML 2.0--Additional URL Configuration • 858

SAML 2.0--Additional URL Configuration Dialog • 836

SAML Affiliate Agent and Federation Security Services Configuration • 102

SAML Affiliate Agents • 101

SAML Affiliation Dialog Fields and Controls • 898

SAML Affiliation Dialog Prerequisites • 897

SAML Affiliation Dialog--Name IDs Tab • 899

SAML Affiliation Dialog--Users Tab • 900

SAML Affiliation Properties Dialog • 897

SAML affiliations

- introduced • 39

SAML Affiliations Reference • 897

SAML Artifact authentication scheme • 291

SAML Auth Schemes Tab • 902

SAML POST profile authentication scheme • 291

SAML Service Provider Attribute Editor Dialog Prerequisites • 829

SAML Service Provider Attribute Editor Dialog--Advanced Tab • 832

SAML Service Provider Attribute Editor Dialog--Attribute Setup Tab • 830

SAML Service Provider Dialog • 815

SAML Service Provider Dialog Prerequisites • 816

SAML Service Provider Dialog--Advanced Tab • 835

SAML Service Provider Dialog--Attributes Tab • 829

SAML Service Provider Dialog--Backchannel Properties • 823

SAML Service Provider Dialog--Encryption Tab • 834

SAML Service Provider Dialog--General Tab • 821

SAML Service Provider Dialog--IPD Tab • 833

SAML Service Provider Dialog--Name IDs Tab • 819

SAML Service Provider Dialog--SLO Tab • 832

SAML Service Provider Dialog--SSO Tab • 824

SAML Service Provider Dialog--Users Tab • 819

SAML Service Provider Properties Dialog Fields and Controls • 816

SAML Service Providers Tab • 679, 902

SAML Session Ticket authentication scheme • 291

Sample Forms for Impersonation • 475

Sample Impersonation Implementation Assessment • 474

Sample Implementation of Impersonation • 473

Sample SiteMinder Configuration with Nested Realms • 405

Sample WebServiceConfig.properties Configuration File • 433

Save and Load Configurations • 523

Scheme Common Setup Group Box • 609

Scope Group Box • 602

Scroll Between Groups of Items • 73

scrolling between groups of items • 73

search

- customizing results • 954
- customizing results per page • 957
- customizing, overview • 950
- default scope • 950
- headings • 937
- manage_data.jsp • 949
- manage_data_submitter.jsp • 949
- searchFixedFilterLines parameters • 950
- searchFixedRoot parameter • 950
- searchNoFilter parameter • 950
- searchNoFilterLines • 950
- searchNoScope parameter • 950
- searchObject parameter • 950
- searchRoot parameter • 950
- searchScope parameter • 950
- specifying limits • 954
- specifying searchable attributes • 953
- user directories • 211
- user directory, WinNT. See also WinNT • 587

Search Customization • 950

Search Database dialog • 586

Search Database Dialog • 586

Search Database Dialog Fields and Controls • 587

Search Database Dialog Prerequisites • 586

Search Directories • 211

Search LDAP/AD Dialog • 584

Search LDAP/AD Dialog Fields and Controls • 584

Search LDAP/AD Dialog Prerequisites • 584

search limits

- customizing • 954

search results

- customizing number • 957
- search root • 169, 548
- Search root field
 - customizing searchFixedRoot parameter • 950
- Search WinNT Dialog Fields and Controls • 588
- Search WinNT Dialog Prerequisites • 587
- Search WinNT Directory Dialog • 587
- Search with the SiteMinder Administration Window
 - 73
- search.properties
 - customizing searchable attributes based on r • 953
 - description, presentation properties • 943
- SEARCH_LIMIT • 954
- SEARCH_TIMEOUT • 954
- searchFixedFilterLines parameter • 950
- searchFixedRoot parameter • 950
- searchNoFilter parameter • 950
- searchNoFilterLines parameter • 950
- searchNoScope parameter • 950
- searchObject parameter • 950
- searchRoot parameter • 950
- searchScope parameter • 950
- SectionHeader style • 937
- secure connection
 - requiring • 551, 563, 569
- secure sockets layer
 - authentication over • 236
- Secure Sockets Layer. See SSL • 236
- SecurID authentication scheme
 - configuring • 627
 - introduced • 270
 - prerequisites • 274, 275
- SecurID authentication scheme with HTML forms support • 270
- SecurID Authentication Schemes • 270
- SecurID Scheme Prerequisites • 274
- SecurID with HTML Forms Support Scheme Prerequisites • 275
- Security Considerations for Impersonation • 459
- security policies
 - benefits of • 49
 - introduced • 47
- Security Requirements When Resolving Web Services Variables • 432
- Select a Test Mode • 520
- Select a Variable Using Variable Lookup • 365
- Select an SQL Query Scheme for an ODBC User Directory Connection • 199

- servers
 - directory types • 482
- services
 - administration • 35
 - auditing • 35
 - authentication • 35
 - authorization • 35
- servlet-based password services • 479
- Servlet-based Password Services JSP Form Customization • 502
- Session Idle Timeouts • 460
- session management
 - Agent key rollovers • 93
 - disabling users • 94
 - for single sign-on • 90
 - overview • 85
 - revoking • 94
 - session creation • 89
 - session delegation • 92
 - session validation • 91
 - timeouts • 92
- Session menu • 69
- Session Tickets • 86
- session timeouts
 - configuring • 684
- Session Timeouts • 92
- Session Timeouts Group Box • 684
- sessions
 - configuring properties • 321, 322
 - configuring properties for realm • 321, 322
 - configuring type • 321, 322
 - idle timeout • 684
 - maximum timeout • 684
 - timeouts • 321, 322
 - validation period • 686
- Set a Language Other Than English as the Default Language • 507
- Set Agent/Group dialog • 682, 752
- Set Expiration Date Group Box • 908
- Set Global Response dialog • 767
- Set Global Response Dialog • 767
- Set Global Response Dialog Fields and Controls • 768
- Set Global Response Dialog Prerequisites • 767
- Set Protection Levels for Passport Authentication • 263
- Set Response dialog • 744
- Set Response Dialog • 744
- Set Response Dialog Fields and Controls • 745
- Set Response Dialog Prerequisites • 744

- Set Starting Date Group Box • 907
- Set the Configuration Parameters in the Agent Configuration Object • 126
- Set the Encoding Spec • 522
- Set Up an Authentication Scheme Object in the Policy Server User Interface • 240
- Set Up the Policy Server for Version 4 Agents and RADIUS Agent Simulations • 519
- Shared Dialog Reference • 905
- Shared Dialog Reference Overview • 905
- shared secret
 - not required for centrally configured Agents • 105
 - specify for 4.x Agent • 124
 - specifying for 4.x Agent • 536
 - specifying for custom Agent • 137
 - specifying for RADIUS Agent • 134, 537
- single sign-on • 83
 - and persistent sessions • 95
 - and protection levels • 90, 237
 - and SiteMinder • 33
 - session management • 90
 - using forms-based authentication • 94
- Single Sign-on in Security Context • 98
- sitemap
 - using to identify resources • 57
- SiteMinder
 - access roles • 59
 - Administration window, search feature • 73
 - components • 33
 - configuring with eTrust SSO
- procedures • 984
 - configuring with eTrust SSO client
- procedures • 987
 - configuring with eTrust Web Agent • 988
 - enabling access roles • 59
 - eTrust SSO integration examples • 978
 - integrating with eTrust SSO • 984
- overview • 977
 - integrating with eTrust SSO client • 987
 - integrating with eTrust SSO Web Agent • 988
- SiteMinder Administrator Properties Prerequisites • 599
- SiteMinder and CA SSO Integration Architectural Examples • 978
- SiteMinder Authentication and CA Arcot Risk Analysis • 1020
- SiteMinder Authorization and Confidence Levels • 1022, 1028
- SiteMinder Components • 33
- SiteMinder eTelligent Rules Benefits • 421
- SiteMinder FCC Files • 249
- SiteMinder Features • 34
- SiteMinder Generated User Attributes • 366
- SiteMinder Global Policy Concept • 438
- SiteMinder Global Response Attribute Editor Dialog Fields and Controls • 757
- SiteMinder Kerberos Authentication • 995
- SiteMinder List Pane • 68
- SiteMinder Menu Bar • 69
- SiteMinder objects
 - configuration order • 44
- SiteMinder Overview • 33
- SiteMinder Policy Should Allow Access, but SSL-Authentication Failed Message Received • 932
- SiteMinder Response Attribute Editor Dialog Fields and Controls • 708
- SiteMinder Test Tool • 517
- SiteMinder Tool Bar • 72
- SmallLinkFont style • 937
- smauthetsso custom authentication scheme
 - configuring • 992
- smetssockcookie Web Agent active response
 - description • 990
- SmHost.conf file • 106
- SmHost.conf path • 518
- SMLOCALE
 - specifying in FCCs • 973
- Special Access for the SiteMinder Administrator • 169
- Special Name/Value Pairs • 252
- Specify a Single IP Address • 392
- Specify a Single IP Address for a Global Policy • 452
- Specify Allowable IP Addresses for Policies • 391
- Specify AND/OR Relationships between Users/Groups • 384
- Specify Attributes in the Search Results • 956
- Specify Directory Attributes • 158
- Specify IP Addresses for Global Policies • 451
- Specify Resource Information • 521
- Specify Search Limits • 954
- Specify the Agent Name • 128
- Specify the Domino Users • 128
- Specify the IIS (Proxy) User • 127
- Specify User Credentials • 522
- specifying
 - a host name • 392, 452
 - a range of IP addresses • 394, 453

- a single IP address • 392, 452
- a subnet mask • 393, 453
- IP addresses • 391
- SQL
 - Query Schemes dialog, accessing • 209
- SQL Query Scheme dialog • 576
- SQL Query Scheme Dialog • 576
- SQL Query Scheme Dialog Fields and Controls • 577
- SQL Query Scheme Dialog Prerequisites • 576
- SQL Query Scheme Group Box • 555
- SQL query schemes • 197
 - introduced • 39
- SQL Server User Store Case Insensitivity and Extra Trailing Spaces Password Issues • 185
- SSL
 - basic over SSL authentication schemes • 243
 - described • 236
 - establishing a connection for an X.509 client certificate • 278
 - Web server, configuring • 924
- SSL Configuration • 924
- SSL Troubleshooting • 928
- Standard Resource Matching • 341
- Start the Policy Server User Interface • 64
- static
 - response attribute • 359
- static variable • 426
 - description • 718
 - return Type • 718
 - value • 718
- Static Variable Editor dialog • 717
- Static Variable Editor Dialog • 717
- Static Variable Editor Dialog Fields and Controls • 718
- Static Variable Editor Dialog Prerequisites • 717
- Static Variables • 426
- static variable
 - name • 718
- Strategies for Managing Security and Users • 47
- stress tests
 - description • 528
 - thread control file • 528
- Stress Tests • 528
- style
 - linking the CSS to pages • 941
- submitter pages
 - description • 949
 - manage_data_submitter.jsp • 949
- submitusername.jsp • 949

- Successful Authentications • 411
- Super Administrators
 - presentation privileges • 943
- super user • 63
- Support a New Language • 506
- Supported Authentication Schemes • 97, 238
- Supported Authentication Schemes and Password Policies • 239
- synchronous auditing • 687
- Synchronous Auditing Group Box • 687
- system administrator • 227
- system objects • 39
- System Objects • 39
- system privileges
 - Act As Part Of Operating System • 558
- System tab
 - described • 65

T

- table_profile_manage.jsp
 - customizing search results • 956
- tabs, changing • 68
- task assessment requirements
 - establishing • 54
- Task Selector page
 - adding tasks • 960
 - customizing links • 958
 - customizing tasks displayed for administrators • 959
 - customizing, overview • 958
 - launch.jsp • 949
 - sample New Users link • 962
- Task Selector Page Customization • 958
- task-assessment requirements
 - about • 57
- Tasks Group Box • 602
- Tasks Related to Attributes and Attribute Queries • 896
- Tasks Related to the Active Policy Editor • 912
- Tasks Related to the Active Rule Editor Dialog • 910
- Tasks Related to the Add an IP Address Dialog • 918
- Tasks Related to the Add Directory Mapping Dialog • 593
- Tasks Related to the Administrator Properties Dialog • 603
- Tasks Related to the Affiliate Dialog • 798
- Tasks Related to the Agent Configuration Object Dialog • 541

Tasks Related to the Agent Dialog • 537
Tasks Related to the Agent Group Dialog • 543
Tasks Related to the Authentication Scheme Dialog • 664
Tasks Related to the AuthValidate Directory Mappings Dialog • 596
Tasks Related to the Certificate Map Test Dialog • 673
Tasks Related to the Certificate Mapping Properties Dialog • 672
Tasks Related to the Certificate Mappings Dialog • 666
Tasks Related to the Delete Dialog • 595
Tasks Related to the Directory List Dialog • 583
Tasks Related to the Directory Mappings Dialog • 592
Tasks Related to the Domain Manager For These Domains Dialog • 605
Tasks Related to the Edit Directory Mapping Dialog • 594
Tasks Related to the Expression Editor • 742
Tasks Related to the Global Policy Dialog • 764
Tasks Related to the Global Response Attribute Editor Dialog • 761
Tasks Related to the Global Response Dialog • 756
Tasks Related to the Global Rule Dialog • 754
Tasks Related to the Host Configuration Object Dialog • 540
Tasks Related to the LDAP Setup Dialog • 574
Tasks Related to the ODBC Setup Dialog • 576
Tasks Related to the Password Policy Dialog • 786
Tasks Related to the Policy Dialog • 738
Tasks Related to the Policy Expression Dialog • 749
Tasks Related to the Post Variable Editor Dialog • 726
Tasks Related to the Realm Dialog • 689
Tasks Related to the Request Context Variable Editor Dialog • 721
Tasks Related to the Resource Partner Properties Dialog • 875
Tasks Related to the Response Attribute Editor Dialog • 712
Tasks Related to the Response Dialog • 707
Tasks Related to the Response Group Properties Dialog • 714
Tasks Related to the Restrictions Dialog • 828
Tasks Related to the Rule Dialog • 698
Tasks Related to the Rule Group Items Dialog • 703

Tasks Related to the Rule Group Properties Dialog • 701
Tasks Related to the Rule Items Dialog • 743, 766
Tasks Related to the SAML 1.x Authentication Scheme Dialog • 814
Tasks Related to the SAML 2.0 Auth Scheme Properties Dialog • 859
Tasks Related to the SAML Service Provider Attribute Editor Dialog • 839
Tasks Related to the Search Database Dialog • 587
Tasks Related to the Search LDAP/AD Dialog • 585
Tasks Related to the Search WinNT Dialog • 588
Tasks Related to the Set Global Response Dialog • 768
Tasks Related to the Set Response Dialog • 745
Tasks Related to the SQL Query Scheme Dialog • 582
Tasks Related to the Static Variable Editor Dialog • 719
Tasks Related to the Time Dialog • 909
Tasks Related to the User Context Variable Editor Dialog • 725
Tasks Related to the User Directory Dialog • 547
Tasks Related to the User Directory Dialog—AD Namespace • 571
Tasks Related to the User Directory Dialog—Custom Namespace • 566
Tasks Related to the User Directory Dialog—LDAP Namespace • 553
Tasks Related to the User Directory Dialog—ODBC Namespace • 557
Tasks Related to the User Directory Dialog—WinNT Namespace • 561
Tasks Related to the User Lookup Dialog • 590
Tasks Related to the Users/Groups Dialog • 916
Tasks Related to the Web Services Variable Editor Dialog • 730
Tasks Related to the WS-Federation Authentication Scheme Properties Dialog • 887
Tasks Related With the Domain Dialog • 679
TCP/IP Connections • 109
Tell Users Why Login Failed • 256
Template String Usage • 304
templates
 HTML forms authentication schemes • 248
 password services • 497, 502
 password services, customizing • 499, 504
Temporary Employee Accesses a Quality Assurance Realm Resource • 220
terminating user sessions • 94

- Test a Certificate Mapping • 295
- Test Tool
 - description • 517
 - loading configurations • 523
 - performing functionality tests • 523
 - performing regression tests • 527
 - performing stress test • 528
 - saving configurations • 523
 - scripting language commands • 528
- Test Tool Overview • 517
- testing certificate mapping • 295
- The Cascading Style Sheet • 937
- The SiteMinder Web Agent Model • 104
- The smkeydatabase.properties File • 416
- There Was No Prompt for a Certificate • 928
- thread control file
 - description • 528
- Thread Control Files • 528
- Time dialog • 905
- Time Dialog • 905
- Time Dialog Fields and Controls • 906
- Time Dialog Prerequisites • 906
- time differential effect on password services • 482
- time restrictions
 - how policy and rule restrictions interact • 395, 454
 - policies • 394
- Time Restrictions Tab • 697
- time, how Web Agent and Policy Server calculate • 83
- timeouts
 - idle • 52
 - session • 321, 322
 - user sessions • 92
- tool bar
 - described • 65
 - using • 72
 - working with the • 72
- tools
 - Test Tool • 517
- Tools menu • 69
- TransactionMinder Authentication Schemes • 291
- TransactionMinder XML Agents • 104
- TransactionMinder-related User Interface Elements • 919
- Troubleshooting • 921
- Troubleshooting SiteMinder Kerberos Authentication • 1011
- Troubleshooting SSL Authentication Schemes • 923

- Trusted Host Configuration Parameters • 108
- Trusted Host Configuration Settings • 108
- trusted hosts
 - introduced • 39
 - operation mode • 108
- Trusted Hosts • 107
- Types of Authentication Schemes • 233
- Types of Responses • 355

U

- unauth page • 775
- understanding
 - nested realms • 318
 - rules processing • 328
 - user directories • 141
 - variable processing • 428
- universal IDs
 - for user sessions • 92
 - introduced • 212
- Universal IDs • 212
- Unprotected Realms, Rules, and Policies • 318
- Unsuccessful Authentication Attempts • 411
- unsupported characters in object names • 75
- URLs
 - and Registration Space mapping • 947
- Use the Policy Server User Interface • 65
- Use Variable Objects in Responses • 364
- user
 - attributes • 359, 577
 - attributes, WinNT, See also WinNT • 561, 564
 - directory contents, viewing, See also viewing • 582
- User Account Management with Active Directory Integration • 511
- User Accounts are Mistakenly Disabled • 513
- User Accounts are Prematurely Disabled • 513
- user context variable
 - Buffer Size • 723
 - Directory Entry Property • 723
 - Directory Namespace • 723
 - Directory Path • 723
 - Directory Server • 723
 - DN • 723
 - Is User Context Initialized • 723
 - Item • 723
 - Name • 723
 - Property • 723
 - Return Type • 723

-
- Session ID • 723
 - User Path • 723
 - User Property • 723
 - Username • 723
 - User Context Variable Editor dialog • 721
 - User Context Variable Editor Dialog • 721
 - User Context Variable Editor Dialog Fields and Controls • 723
 - User Context Variable Editor Dialog Prerequisites • 722
 - User Context Variables • 426
 - user directories
 - attributes, directory • 158, 552, 556, 570
 - configuring, LanMan • 934
 - configuring, ODBC • 183, 204, 209
 - configuring, WinNT • 557
 - directory mapping • 213
 - introduced • 39
 - LanMan, described • 207
 - LanMan, prerequisites • 933
 - LDAP, credentials • 550, 555, 569
 - LDAP, described • 142
 - LDAP, policy bindings • 396
 - LDAP, search filters • 143
 - LDAP, user disambiguation • 143
 - LDAP, user lookup via DN • 143
 - LDAP, user lookup via search expression • 143
 - ODBC, policy bindings • 402
 - ODBC, query schemes • 555
 - ODBC, working with • 153
 - opening the dialog box • 179
 - searches • 211
 - SQL, query schemes • 197, 576
 - understanding • 141
 - User directory dialog • 545
 - viewing directory contents • 210, 582
 - WinNT, failover • 208
 - WinNT, policy bindings • 400
 - WinNT, using attributes in configuring responses • 208
 - WinNT, working with • 154, 204
 - User Directories • 141
 - User Directories Tab • 677
 - User Directory • 406
 - User Directory and Search Dialog Reference • 545
 - User Directory Connections Overview • 141
 - User Directory dialog • 545
 - User Directory Dialog • 545
 - User Directory Dialog—AD Namespace • 566
 - User Directory Dialog—AD Namespace—Credentials and Connection Tab • 569
 - User Directory Dialog—AD Namespace—Directory Setup Tab • 567
 - User Directory Dialog—AD Namespace—User Attributes Tab • 570
 - User Directory Dialog—Custom Namespace • 561
 - User Directory Dialog—Custom Namespace—Credentials and Connection Tab • 563
 - User Directory Dialog—Custom Namespace—Directory Setup Tab • 562
 - User Directory Dialog—Custom Namespace—User Attributes Tab • 564
 - User Directory Dialog—LDAP Namespace • 547
 - User Directory Dialog—LDAP Namespace—Credentials and Connection Tab • 550
 - User Directory Dialog—LDAP Namespace—Directory Setup Tab • 548
 - User Directory Dialog—LDAP Namespace—User Attributes Tab • 552
 - User Directory Dialog—ODBC Namespace • 553
 - User Directory Dialog—ODBC Namespace—Credentials and Connection Tab • 555
 - User Directory Dialog—ODBC Namespace—Directory Setup Tab • 554
 - User Directory Dialog—ODBC Namespace—User Attributes Tab • 556
 - User Directory Dialog—WinNT Namespace • 557
 - User Directory Dialog—WinNT Namespace—Directory Credentials and Connection Tab • 559
 - User Directory Dialog—WinNT Namespace—Directory Setup Tab • 558
 - User Directory Dialog—WinNT Namespace—User Attributes Tab • 561
 - User Directory Information Group Box • 771
 - User Directory Object Prerequisites • 546
 - user directory searches
 - LDAP • 584
 - LDAP, See also LDAP • 584
 - ODBC • 586
 - ODBC, See also ODBC • 586
 - WinNT • 587
 - user disambiguation • 143
 - User Disambiguation in an LDAP Directory • 143
-

- User DN Lookup Formats for Windows
 - Authentication Schemes • 617
- user interface
 - Policy Server • 63, 64
- User Lookup dialog • 589
- User Lookup Dialog • 589
- User Lookup Dialog Fields and Controls • 590
- User Lookup Dialog Prerequisites • 589
- user management
 - access roles • 59
- User Session Overview • 85
- user sessions
 - creating • 89
 - maintaining sessions • 85
 - terminating • 94
 - validating • 91
- User Sessions • 85
- User Store Consideration • 1023, 1029
- user.properties
 - description, object properties • 944
 - description, presentation properties • 943
- User/Groups dialog • 912
- user_data_collector.jsp
 - description • 949
- user_data_submitter.jsp • 949
 - description • 949
- user-initiated password changes
 - enabling • 492
- User-initiated Password Changes • 492
- users
 - collecting new attributes • 963
 - displaying attributes in search results • 956
 - role-based presentation • 943
 - specifying container None • 946
 - specifying profile attributes • 944
 - user_data_collector.jsp • 949
 - user_data_submitter.jsp • 949
- Users/Groups Dialog • 912
- Users/Groups Dialog Fields and Controls • 913
- Users/Groups Dialog Prerequisites • 912
- using
 - List pane • 68
 - menu bar • 69
 - object pane • 68
 - SiteMinder Administration window • 65
 - tool bar • 72

V

- validating user sessions • 91
- validation period, session • 686
- value of a property • 577
- values, string • 577
- Variable Editor • 433
- Variable Lookup Dialog • 731
- Variable Types • 426
- Variable Use in Policies • 427
- Variable Use in Responses • 428
- Variable-related Dialogs Reference • 717
- variables
 - expressions • 427
 - introduced • 41
 - understanding processing • 428
- Variables • 421
- Variables Overview • 425
- Verify Browser Certificate Validity • 930
- Verify Correct Policy Server and Web Agent Configuration • 931
- Verify that a Resource is Protected • 1010
- Verify That All Web Servers Are Configured to Use SSL and Require Certificates • 929
- Verify the CA Directory Cache Configuration • 162
- Verify the Following Settings for each SiteMinder Virtual Directory • 929
- Verify the SSL Connection • 196
- View menu • 69
- View Multiple Objects • 77
- View the Properties of a Policy Server Object • 76
- View Trusted Host Objects • 110
- View User Directory Contents • 210
- viewing
 - directory contents, user directories, See also user directory • 582
 - multiple objects • 77
 - objects • 76
 - user directory contents • 210, 582

W

- WarningFont style • 937
- Web Agent
 - Configuration File • 106
 - configuring • 123, 124
 - response attributes • 356
 - rule actions • 330
- Web Agent Actions • 330
- Web Agent Configuration Options • 104

-
- Web Agent Parameter Descriptions • 121
 - Web Agent Response Attributes • 356
 - Web Agent Responses • 355
 - Web Agents • 100
 - Web Server • 240
 - Web Service Credentials group box • 729
 - Web Service Variables • 430
 - Web Services Variable Editor dialog • 727
 - Web Services Variable Editor Dialog • 727
 - Web Services Variable Editor Dialog Fields and Control - HTTP Headers Tab • 730
 - Web Services Variable Editor Dialog Fields and Controls—General Tab • 728
 - Web Services Variable Editor Dialog Fields and Controls—SOAP Document Tab • 730
 - Web Services Variable Editor Dialog Prerequisites • 727
 - Web Services Variables • 427
 - WebAgent.conf file • 106
 - WebAgent-SAML-Session-Ticket-Variable response attribute • 356
 - WebAgent-WSSec-Variable response attribute • 356
 - welcome.jsp • 949
 - Where field
 - customizing • 950
 - displaying searchable attributes based on role • 953
 - Where to Search Group Box • 740
 - Windows
 - authentication schemes • 234, 258
 - Windows Authentication Scheme • 234
 - Windows Authentication Scheme Considerations • 260
 - Windows Authentication Scheme Prerequisites • 259
 - Windows Authentication Schemes • 258
 - Windows Directory Connections • 204
 - Windows Directory Overview • 154
 - WinNT
 - configuring user directories • 557
 - connections, user directories • 154, 204
 - credentials, user directory • 559
 - directory setup • 558, 562
 - failover, user directory • 208
 - policy bindings for user directories • 400
 - searches, user directory • 587
 - Selection WinNT Configuration, user directories • 208
 - user attributes • 561, 564
 - user directories, configuring, See also configuring • 557
 - user directory searches, See also user directory searches • 587
 - WinNT Attributes in Responses • 208
 - WinNT Domain Connection Requirements • 204
 - Work with Policy Server Objects • 75
 - WS-Federation Auth. Scheme Fields and Controls • 881
 - WS-Federation Auth. Scheme Properties Dialog Prerequisites • 878
 - WS-Federation Authentication Scheme Dialog--Advanced Tab • 880
 - WS-Federation Authentication Scheme Reference • 877
 - WS-Federation Resource Partner Properties Reference • 861
 - WS-Security authentication scheme • 291
- ## X
- X.509 Certificate or Basic Authentication Schemes • 281
 - X.509 Client Certificate and Basic authentication scheme
 - configuring • 632, 637
 - description • 279
 - prerequisites • 280
 - X.509 Client Certificate and Basic Authentication Schemes • 279
 - X.509 Client Certificate and Basic Scheme Prerequisites • 280
 - X.509 Client Certificate and HTML Forms authentication scheme
 - X.509 Client Certificate and HTML Forms authentication scheme • 283
 - X.509 Client Certificate and HTML Forms Authentication Schemes • 283
 - X.509 Client Certificate and HTML Forms Scheme Prerequisites • 284
 - X.509 client certificate authentication scheme configuring • 631
 - X.509 Client Certificate authentication scheme configuring • 281
 - establishing an • 278
 - prerequisites • 278
 - X.509 client certificate authentication schemes described • 277
 - introduced • 235
-

- X.509 Client Certificate Authentication Schemes •
235, 277
- X.509 Client Certificate or Basic authentication
scheme
 - configuring • 634
 - prereq • 282
- X.509 Client Certificate or Basic Scheme
 - Prerequisites • 282
- X.509 Client Certificate or HTML Forms
authentication scheme
 - configuring • 639
 - prerequisites • 287
- X.509 Client Certificate or HTML Forms
authentication scheme • 286
- X.509 Client Certificate or HTML Forms
Authentication Schemes • 286
- X.509 Client Certificate or HTML Forms Scheme
 - Prerequisites • 287
- X.509 Client Certificate Scheme Prerequisites • 278
- X.509 client certificates
 - certificate mapping • 293
- XML Agent Actions • 330
- XML Digital Signature authentication scheme • 291
- XML Document Credential Collector authentication
scheme • 291
- XML Key Database
 - properties file • 416
- XML Signing and Validation Service
 - described • 415