

CA SiteMinder®

ポリシー サーバ設定ガイド

r12.0 SP3



このドキュメント(組み込みヘルプ システムおよび電子的に配布される資料を含む、以下「本ドキュメント」)は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社(以下「CA」)により随時、変更または撤回されることがあります。

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複製、譲渡、開示、変更、複本することはできません。本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、(i) 本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

上記にかかわらず、本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害(直接損害か間接損害かを問いません)が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本ドキュメントの制作者は CA です。

「制限された権利」のもとでの提供:アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2010 CA. All rights reserved. 本書に記載された全ての製品名、サービス名、商号およびロゴは各社のそれぞれの商標またはサービスマークです。

CA Technologies 製品リファレンス

このマニュアルでは、以下の CA Technologies 製品に言及しています。

- CA Single Sign-On
- CA SiteMinder®
- CA SiteMinder® Federation セキュリティ サービス
- CA Enterprise Log Manager

CA への連絡先

テクニカル サポートの詳細については、弊社テクニカル サポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。

目次

第 1 章: SiteMinder の概要	29
SiteMinder のコンポーネント	29
ポリシー サーバの概要	29
ポリシー サーバ管理コンソールの概要	32
 第 2 章: ポリシー サーバオブジェクトの概要	 33
ポリシー サーバのオブジェクトタイプ	33
インフラストラクチャ オブジェクト	34
ポリシードメインのオブジェクト	37
グローバル オブジェクト	38
設定順	40
 第 3 章: ポリシーベースのセキュリティの実装	 43
ポリシー ベース セキュリティの概要	43
セキュリティとユーザを管理するための計画	44
アクセス制御リスト	44
SiteMinder セキュリティ ポリシー	45
エンド ユーザ エクスペリエンスの管理	47
セキュリティ モデルの実装	51
セキュリティ モデル要件の整理	52
組織とリソースの要件に関する考慮事項	52
タスク評価要件の定義	55
実行要件の定義	56
SiteMinder アプリケーション ロール	56
 第 4 章: 管理ユーザ インタフェースの管理	 59
SiteMinder 管理者ユーザ インターフェース	59
管理 UI の概要	60
管理 UI の起動	60
ポリシー サーバ オブジェクトの管理	61
ポリシー サーバ オブジェクトの複製	62

ポリシー サーバ オブジェクト プロパティの表示	63
既存のポリシー サーバ オブジェクトの変更	64
ポリシー サーバ オブジェクトの削除	64
タスク永続性データベースの管理	65
サブミット済みタスクのクリーンアップ	66
反復タスクの削除	68
SiteMinder 管理者	68
デフォルトの管理者アカウント	68
管理者のタイプ	69
管理者ストア オプション	73
レガシー管理者を作成する方法	73
外部管理者ストアを設定する方法	78
外部管理者ストア クレデンシャルの更新	90
外部管理者ストア接続の変更	94
管理者を作成する方法	94
管理者の無効化	101
レガシー管理者の無効化	101
管理者アクセスの復元	103
Web エージェントとポリシー サーバの時間の計算方法	103

第 5 章: ユーザ セッション 105

ユーザ セッションの概要	105
非永続 Cookie と永続 Cookie	105
非永続セッションと永続セッション	106
セッションチケット	106
SiteMinder によるユーザセッションの管理	107
ユーザ セッションを開始する方法	109
複数のレルムにまたがるセッションを維持する方法	110
複数の cookie ドメインにまたがるセッションを維持する方法	111
ユーザ セッションを検証する方法	111
セッション情報を委任する方法	112
セッション タイムアウト	113
エージェント キー管理とセッション タイムアウトを整合させる方法	113
ユーザ セッションを終了する方法	114
Windows ユーザ セキュリティ コンテキスト	115
ユーザ セキュリティ コンテキストの永続セッションのメンテナンス方法	116

Windows ユーザ セキュリティ コンテキストの要件	117
第 6 章: エージェントおよびエージェントグループ	121
Web エージェントに対するトラステッド ホスト	121
ポリシー サーバへのトラステッド ホストの登録	121
トラステッド ホスト設定	122
トラステッド ホスト オブジェクトの削除	125
トラステッド ホストホストのホスト設定オブジェクト	126
ホスト設定オブジェクトのコピー	126
ホスト設定オブジェクトへの複数のポリシー サーバの追加	127
ホスト設定オブジェクトのポリシー サーバ クラスタの構成	129
SiteMinder エージェントの概要	130
Web エージェント	131
SAML アフィリエイトエージェント	132
RADIUS エージェント	135
アプリケーション サーバ エージェント	135
CA SOA Security Manager SOA エージェント	136
Web エージェント設定の概要	137
Web エージェントを中央で設定した場合の利点	137
Web エージェント コンポーネント	138
Web エージェントに関連するポリシー サーバ オブジェクト	139
Web エージェントの設定方法	140
Web エージェントの中央設定	141
ホスト設定オブジェクトの作成	142
ローカルでの Web エージェントの設定	143
中央設定とローカル設定の組み合わせ	144
エージェント オブジェクトの作成による Web エージェント ID の確立	145
4.x Web エージェント識別情報を作成するためのエージェントオブジェクトの設定	146
エージェント設定オブジェクトでの設定パラメータの設定	148
エージェント設定オブジェクトの概要	148
エージェント設定オブジェクトのコピー	149
エージェント設定オブジェクトの作成	150
エージェント設定オブジェクトの必須パラメータ	152
エージェント設定パラメータの変更	153
Web エージェントの有効化	155
非 Web エージェントのエージェント識別情報の設定	155

RADIUS エージェントの設定	157
エージェント グループ	159
エージェント グループの設定	160
エージェント グループへのエージェントの追加	161
カスタム エージェント	162
カスタム エージェント タイプの設定	162
エージェント 識別情報用のカスタム エージェント オブジェクトの作成	163
SiteMinder エージェントによるリソースの保護	164

第 7 章: ユーザ ディレクトリ 165

ユーザ ディレクトリ接続の概要	166
LDAP の概要	166
ODBC データベースの概要	179
Windows ディレクトリの概要	180
Active Directory の概要	181
カスタム ディレクトリの概要	182
ディレクトリ属性の概要	183
CA Directory ユーザ ディレクトリ接続を設定する方法	184
ユーザ ストア システムに Ping を発行します。	185
CA Directory ユーザ ディレクトリ接続の設定	185
ユーザ ストア DSA パラメータの有効化	186
CA Directory ユーザ ストアのキャッシュの有効化	186
CA ディレクトリ キャッシュ設定の確認	187
Sun Java System ユーザ ディレクトリ接続を設定する方法	188
ユーザ ストア システムに Ping を発行します。	188
Sun Java System ユーザ ディレクトリ接続の設定	188
IBM Directory Server ユーザ ディレクトリ接続を設定する方法	189
ユーザ ストア システムに Ping を発行します。	189
IBM Directory Server ユーザ ディレクトリ接続の設定	190
Domino ユーザ ディレクトリをユーザ ストアとして設定する方法	191
Domino のユーザ ディレクトリがポリシー サーバの要件を満たすことを確認	191
ユーザ ストア システムに Ping を発行します。	191
Domino ディレクトリ接続の設定	192
Novell eDirectory LDAP ディレクトリ接続を設定する方法	193
NetWare の設定	194
Novell eDirectory における匿名 LDAP アクセスの設定	194

SiteMinder 管理者による特別なアクセス.....	196
SiteMinder 管理用の Novell eDirectory ユーザ アカウントの作成	196
ユーザ ストア システムに Ping を発行します。	197
Novell eDirectory の LDAP ディレクトリ接続の設定	197
ADAM ユーザ ディレクトリ接続を設定する方法.....	198
ADAM インスタンスおよびユーザ ストア接続の作成.....	199
ADAM ユーザ ストアに接続するための SiteMinder ユーザの作成	200
SiteMinder ユーザへの管理者権限の割り当て	200
ADAM ユーザ ストアへのユーザの読み込み	201
ADAM ユーザ ストア ディレクトリ接続の設定	202
Active Directory ディレクトリ接続を設定する方法	203
Active Directory に関する考慮事項	203
Active Directory 接続のための LDAP ネームスペース	206
Active Directory 接続のための AD ネームスペース	208
ユーザ ストア システムに Ping を発行します。	208
Active Directory 接続の設定	209
Active Directory グローバル カタログ ユーザ ディレクトリ接続を設定する方法	210
ユーザ ストア システムに Ping を発行します。	211
Active Directory グローバル カタログ ディレクトリ接続の設定	211
Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法	212
Oracle Internet Directory ユーザ ディレクトリの LDAP リフェラル制限	212
ユーザ ストア システムに Ping を発行します。	213
OID ディレクトリ用の組織単位を作成します。.....	213
Oracle インターネット ディレクトリ接続の設定	214
ODBC ユーザ ディレクトリ接続を設定する方法	215
ユーザ ストア システムに Ping を発行します。	215
ODBC ディレクトリ接続の設定	215
SQL Server ユーザ ストアの大文字/小文字の区別なしに関する問題および末尾に余分な空 白のあるパスワードに関する問題	216
Windows ディレクトリ接続を設定する方法	220
WinNT ドメイン接続の要件	221
ユーザ ストア システムに Ping を発行します。	222
Windows ディレクトリ接続の設定	222
カスタム ユーザ ディレクトリ接続を設定する方法	225
ユーザ ストア システムに Ping を発行します。	226
カスタム ディレクトリ接続を設定します	226
SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法	227

SSL 接続を設定する前に	228
NSS ユーティリティのインストール	229
証明書データベースファイルの作成	230
証明書データベースへのルート証明機関の追加	231
証明書データベースへのサーバ証明書の追加	233
証明書データベース内の証明書の一覧表示	235
ユーザ ディレクトリの SSL 接続の設定	236
ポリシー サーバから証明書データベースへの参照の設定	237
SSL 接続の検証	238
LDAP ロード バランシングおよびフェイルオーバー	238
ポート番号に関する考慮事項	239
フェイルオーバーの設定	239
ロード バランシングの設定	240
ロード バランシングとフェイルオーバーの設定	241
ユース ケース - ロード バランシングとフェイルオーバー	243
ODBC データ ソース フェイルオーバーの設定	244
SQL クエリ方式	244
SQL クエリ方式の設定	245
ODBC ユーザ ディレクトリ接続への SQL クエリ方式の追加	246
ストアード プロシージャを介して認証するように SQL クエリ方式を設定する方法	247
フェイルオーバーおよび接続プーリング時の非同期呼び出しの対応	250
複数ポリシー ストアの同じユーザ ディレクトリ接続の定義	253
ユーザ ディレクトリの内容の表示	254
ユーザ ディレクトリの検索	255
ユニバーサル ID	255
SiteMinder での UID の使用方法	256
名前付き式	256
名前付き式のメリット	257
名前付き式の定義	258
ユーザ属性マッピング	273
ユーザ属性マッピングの概要	273
属性マッピングのしくみ	275
属性マッピングの定義	277
ユーザ属性マッピングの適用	293

第 8 章: ディレクトリマッピング 299

ディレクトリ マッピングの概要	299
ディレクトリ マッピングの要件	301
サポートされているディレクトリ マッピング	301
認証と認可のディレクトリ マッピングを設定する方法	301
ディレクトリ マッピングの設定	302
許可ディレクトリのレルムへの割り当て	302
AuthValidate ディレクトリ マッピングを設定する方法	303
認証/検証ディレクトリ マッピングの設定	304
既存のレルムへの認証/検証ディレクトリ マッピングの割り当て	304
ディレクトリ マッピングの例	305
従業員がエンジニアリングレルムのリソースにアクセスする	306
臨時従業員が品質保証レルムのリソースにアクセスする場合	306
ユニバーサル ID によるディレクトリ マッピング	307
ディレクトリ マッピングの大文字/小文字の区別	308
ディレクトリ マッピングとレスポンス	308

第 9 章: 認証方式 309

認証方式の概要	309
サポートされる認証方式とパスワード ポリシー	310
認証時のポリシー サーバ検索を 1 つのユーザストアに限定	311
認証方式処理	312
認証方式タイプ	313
保護レベル	317
認証方式とクレデンシャル要件	318
ポリシー サーバユーザ インターフェースでの認証方式オブジェクトのセットアップ	319
単一の認証方式の設定の複数のインスタンス	320
基本認証方式	320
基本認証方式の前提条件	321
基本認証方式の設定	321
SSL を介した基本認証方式	322
SSL を介した基本認証方式の前提条件	323
SSL を介した基本認証方式の設定	323
HTML フォーム認証方式	324
HTML フォーム認証方式の前提条件	327

HTML フォーム認証テンプレート	328
HTML フォーム認証方式の設定	340
Windows 認証方式	342
Kerberos のサポート	342
Windows 認証方式の前提条件	343
Windows 認証方式に関する考慮事項	345
Windows 認証方式の設定	345
Information Card 認証方式	346
情報カード (Information Card) の概要	346
アイデンティティセレクトアの概要	346
SiteMinder Information Card 認証方式 (ICAS)	347
MS Passport 認証方式	361
ポリシー サーバでの Passport 認証のサポート	362
Passport 認証の保護レベルの設定	362
Passport 認証方式の前提条件	363
MS Passport 認証方式の設定	363
Passport 認証の検索仕様のマッピング	364
RADIUS CHAP/PAP 認証方式	365
PAP の概要	365
CHAP の概要	365
RADIUS CHAP/PAP 方式の概要	366
RADIUS CHAP/PAP 認証方式の前提条件	366
RADIUS CHAP/PAP 認証方式の設定	366
RADIUS サーバ認証方式	367
RADIUS サーバ認証方式の前提条件	367
RADIUS サーバ認証方式の設定	368
SafeWord サーバ認証方式	368
SafeWord サーバ認証方式の前提条件	369
SafeWord サーバ認証方式の設定	369
SafeWord サーバ認証方式と HTML フォーム認証方式	370
SafeWord サーバ認証方式と HTML フォーム認証方式の前提条件	370
SafeWord サーバ認証方式と HTML フォーム認証方式の設定	371
SecurID 認証方式	372
SecurID 認証方式の前提条件	376
SecurID 認証方式の設定	377
HTML フォームに対応した SecurID 認証方式の前提条件	378

SecurID 認証方式と HTML フォーム認証方式の設定	379
SecurID ユーザの再アクティブ化および確認に使用するフォームのサポート	380
新規ユーザアカウントをアクティブにするためのフォーム	380
X.509 クライアント証明書認証方式	381
証明書認証の証明書の抽出	382
SiteMinder でユーザの識別に証明書データを使用する方法	383
X.509 クライアント証明書認証方式の前提条件	383
X.509 証明書認証方式の設定	384
X.509 クライアント証明書および基本認証方式	385
X.509 クライアント証明書および基本認証方式の前提条件	386
X.509 証明書および基本認証方式の設定	386
X.509 クライアント証明書または基本認証方式	387
X.509 クライアント証明書または基本認証方式の前提条件	389
X.509 証明書または基本認証方式の設定	389
X.509 クライアント証明書および HTML フォーム認証方式	390
X.509 クライアント証明書および HTML フォームの認証方式の前提条件	391
エージェント API のサポート	392
X.509 証明書および HTML フォーム認証方式の設定	392
X.509 クライアント証明書または HTML フォーム認証方式	393
X.509 クライアント証明書または HTML フォームの認証方式の前提条件	395
エージェント API のサポート	396
X.509 証明書または HTML フォーム認証方式の設定	396
匿名認証方式	397
匿名認証方式の前提条件	398
匿名認証方式の設定	398
カスタム認証方式	399
カスタム認証方式の前提条件	399
カスタム認証方式の設定	399
Federation セキュリティ サービス認証方式	400
SOA Security Manager 認証方式	400
偽装認証方式	401
インパーソネーション認証方式の前提条件	401
インパーソネーション認証方式の設定	402
 第 10 章: X.509 証明書の証明書マッピングおよび有効性チェック	 405
X.509 クライアント認証方式の証明書マッピング	405

証明書マッピングの設定	405
証明書マッピングのテスト	407
カスタム マッピング式	407
同一タイプの複数属性でのカスタム証明書マッピング	412
証明書の有効性チェック(任意)	413
有効性チェックを実行するための前提条件	414
証明書破棄リスト(CRL)のチェック	414
オンライン証明書状態プロトコル チェック(OCSP)	422
OCSP と CRL 間のフェイルオーバー	435
証明書検証のトラブルシューティング	438

第 11 章: 強い認証 441

クレデンシアル セレクタの概要	441
クレデンシアル セレクタのユースケース	442
パスワード/証明書の認証によるアクセスのリクエスト	442
Windows 認証によるアクセスのリクエスト	443
SecurID 認証によるアクセスのリクエスト	444
SafeWord 認証によるアクセスのリクエスト	444
このユースケースのクレデンシアル セレクタ ソリューション	444
フロントエンド認証方式の確立	445
フォーム認証情報コレクタ(FCC)の使用	446
フロントエンド認証のための selectlogin.fcc ファイルの設定	447
フロントエンド認証方式の設定	453
失敗した認証の管理	455
バックエンド処理のセットアップ	456
バックエンド処理で使用する認証方式のセットアップ	456
バックエンド処理のためのポリシードメインのセットアップ	456
バックエンド ポリシードメインのレルムの設定	457
バックエンド ポリシードメインのルールを作成	459
バックエンド ポリシードメインの AuthContext レスポンスの設定	459
バックエンド クレデンシアル選択のポリシーの設定	461
サンプル アプリケーションの保護	464
サンプル アプリケーションのレルムとルール	465
サンプル アプリケーションを保護するポリシーの設定	467
サンプル アプリケーションのレスポンスの設定	468
クレデンシアル セレクタ ソリューションのテスト	469

第 12 章: ドメイン	471
ポリシードメインの概要	471
ドメインとユーザ メンバシップ	473
ポリシードメインの設定方法	473
ポリシードメインの設定	474
ユーザ ディレクトリの割り当て	474
レルムの作成	475
ドメインのグローバル ポリシー処理の無効化	478
アフィリエイトドメイン	479
ドメインの変更	479
ドメインの削除	479
 第 13 章: レルム	 481
レルムの概要	481
エージェント、レルム、およびルールによるリソースの識別	482
保護されていないレルム、ルールおよびポリシー	484
ネストされたレルム	485
リクエスト処理におけるレルム	487
例	487
レルムの設定	487
SiteMinder Web エージェントで保護されたレルムの設定	488
RADIUS エージェントにより保護されたレルムの設定	489
4.x アフィリエイト エージェントのレルムの設定	490
レルムの変更	491
レルムの削除	491
ネストされたレルムの設定	492
リソース キャッシュからのレルムのクリア	493
 第 14 章: ルール	 495
ルールの概要	495
ルールがポリシーの一部として動作する仕組み	496
ポリシー サーバがルールを処理する方法	497
ルールとネストされたレルム	498
ルール アクション	498
高度なルール オプション	503

Web エージェント アクションのルールの設定	504
認証イベントアクション用のルールの設定	505
許可イベント アクションのルールの設定	506
OnAccessReject ルールのポリシーに関する考慮事項	507
インパーソネーション イベント アクションのルールの設定	508
アフィリエイト エージェントのルールの設定	509
リソース照合と正規表現	511
標準的なリソース照合	511
リソース照合のための正規表現	511
ルールの有効化および無効化	512
高度なルール オプション	513
ルールへの時間制限の追加	513
アクティブ ルールの設定	514
ルールの削除	515

第 15 章: ルール グループ 517

ルール グループの概要	517
ルール グループの作成	518
ルール グループへのルールの追加	519
ルール グループの変更	520
ルール グループの削除	520

第 16 章: レスポンスとレスポンス グループ 521

レスポンス	521
SiteMinder がレスポンスを処理する方法	522
レスポンス タイプ	523
レスポンス属性	524
レスポンスとディレクトリ マッピング	527
レスポンスの設定	528
レスポンス属性	528
レスポンスにおける変数オブジェクト	533
レスポンス属性キャッシングの設定	537
レスポンスの編集	537
レスポンスの削除	538
SiteMinder が生成するユーザ属性	538

レスポンス グループ	544
レスポンス グループの設定	544
レスポンス グループへのレスポンスの追加	546
レスポンス グループの変更	546
レスポンス グループの削除	547

第 17 章: ポリシー 549

ポリシーの概要	549
ポリシーの説明	551
ポリシー バインド	551
ポリシーにおける式	552
ポリシーの設定方法	553
ポリシーの作成	553
ポリシーにユーザを追加します。	555
ポリシーへのルールを追加	556
レスポンスまたはレスポンス グループへのルールの関連付け	556
グローバル レスポンスへのルールの関連付け	557
ポリシーへの式の追加	558
ユーザまたはグループをポリシーから除外	559
ネストされたグループをポリシーに許可	560
[AND ユーザ/グループ] チェック ボックス	561
ユーザ/グループ間の AND/OR 関係の指定	563
手動設定によるユーザの追加	564
ポリシー サーバの LDAP 許可パフォーマンスの向上	566
ポリシーへの LDAP 式の追加	567
ポリシーの有効化および無効化	568
ポリシーの詳細オプション	569
ポリシーが使用できる IP アドレス	570
ポリシーの時間制限	572
アクティブなポリシーの設定	574
ポリシー バインドの確立	574
LDAP ディレクトリのポリシー バインド	575
ポリシーのバインド (Windows NT ユーザディレクトリ用)	582
Microsoft SQL Server または Oracle ユーザ ディレクトリのポリシー バインド	584
ポリシーの削除	587
SQL クエリへのポリシーのバインド	587

ポリシーの処理.....	588
ネストされたレルムを持つ SiteMinder を設定するときの例.....	588
階層構造のポリシーに対する認証処理.....	594
階層構造のポリシーに対する許可処理.....	596
階層構造のポリシーでのディレクトリ マッピング.....	598

第 18 章: エンタープライズ ポリシー管理 (EPM) 599

EPM によるアプリケーションの保護.....	599
アプリケーション セキュリティ ポリシーを作成する方法.....	600
アプリケーション セキュリティ ポリシーを作成するための管理権限.....	601
アプリケーションを保護する場合の EPM ユース ケース.....	602
Web ポータルを保護するアプリケーション セキュリティ ポリシー.....	602
ロールに基づくアプリケーション セキュリティ ポリシー.....	607
ユーザ マッピングと名前付き式を使用したアプリケーション セキュリティ ポリシー.....	616
アプリケーションの詳細なポリシー コンポーネント.....	625

第 19 章: 変数 627

eTelligent ルール.....	627
eTelligent ルールのコンポーネント要件.....	628
SiteMinder eTelligent ルールのメリット.....	628
eTelligent ルールの設定.....	629
変数の概要.....	632
変数タイプ.....	632
ポリシーでの変数の使用.....	634
レスポンスでの変数の使用.....	634
ポリシー サーバが変数进行处理する方法.....	635
Web サービス変数.....	637
Web サービス変数のコンポーネント要件.....	639
Web サービス変数を解決する場合のセキュリティ要件.....	639
Web サービスの変数リゾルバの設定.....	640
WebServiceConfig.properties のサンプル構成ファイル.....	641
認証機関と Web サービス変数.....	641
変数の作成.....	642
スタティック変数の作成.....	643
コンテキスト要求変数の作成.....	644

ユーザ コンテキスト変数の作成	645
フォーム ポスト変数の作成	646
Web サービス変数の作成	647
第 20 章: SiteMinder キー データベース管理	649
キー データベースの概要	649
Smkeydatabase のエイリアス	650
キー データベースのプロパティファイル	650
DBLocation の設定	651
NativeDBName の設定	651
XMLDocumentOpsImplementation の設定	651
AffiliateIXMLSignatureImplementation の設定	651
IXMLSignatureImplementation の設定	652
EncryptedPassword の設定	652
IXMLEncryptDecryptImplementation の設定	652
DBUpdateFrequencyMinutes の設定	652
Smkeytool によるキー データベースの作成および管理	653
smkeytool を使用したキー データベースの変更	655
Smkeytool コマンドの構文およびオプション	657
UNIX プラットフォームでの smkeytool の使用例	666
Windows プラットフォームでの smkeytool の使用例	667
第 21 章: グローバル ポリシー、ルール、レスポンス	669
グローバル ポリシー	669
グローバル ポリシー オブジェクトの特性	670
SiteMinder グローバル ポリシーの概念	672
グローバル ポリシー処理	673
グローバル ポリシーを設定する方法	674
グローバル ルール	674
グローバル レスポンス オブジェクト	681
グローバル レスポンスのレスポンス属性	682
グローバル ポリシー オブジェクトを設定する方法	684
グローバル ポリシーの有効と無効	686
グローバル アクティブ ポリシーの設定	686
グローバル ポリシーで許容される IP アドレス	687

グローバル ポリシーに対する単一 IP アドレスの指定	688
グローバル ポリシーのホスト名の追加	688
グローバル ポリシーのサブネット マスクの追加	689
グローバル ポリシーの IP アドレスの範囲の追加	690
グローバル ポリシーの時間制限の追加と削除	690

第 22 章: インパーソネーション(偽装) 693

インパーソネーションの概要	693
インパーソネーション処理	694
インパーソネーションでのセキュリティに関する考慮事項	696
認証方式保護レベルの有効性	697
セッションアイドルタイムアウト	697
インパーソネーションの制限	697
インパーソネーションレルムおよびイベント	698
インパーソネーション用のポリシー サーバ オブジェクト	699
インパーソネーションセッションの開始方法	700
startimp.fcc を使用するエージェントの設定	704
インパーソネーション認証方式の設定	704
.fcc ファイルによるインパーソネーションの有効化	705
インパーソネーション イベントの設定	709
インパーソネーション用のポリシーの設定	709
複数 cookie ドメインのサポート	712
インパーソネーションのサンプル実装	712
サンプル インパーソネーション実装の評価	714
インパーソネーションのサンプルフォーム	714

第 23 章: パスワード ポリシー 717

パスワード サービスの概要	717
パスワード サービスの機能	719
Apache Web サーバの前提条件	721
パスワード サービスの注意事項	721
CGI またはサーブレットの場合のカスタム パスワード サービス ディレクトリの注意事項	722
デフォルトの CGI リダイレクト URL の変更	723
パスワード ポリシーの作成	724
パスワードの有効期限の設定	725

パスワード構成の設定	726
パスワード正規表現	726
正規表現の構文	727
正規表現照合の設定	728
パスワードの制限の設定	729
高度なパスワード オプションの設定	730
ユーザが開始するパスワード変更	731
[パスワードの変更]リンクの追加	731
ユーザによるパスワードの変更	735
パスワード サービスでリダイレクトされる際のログイン ID の削除	735
CGI ベースのパスワード サービスの HTML フォームのカスタマイズ	736
CGI ベースのパスワードサービス HTML フォームテンプレート	736
CGI ベースのパスワード サービス HTML テンプレートの変更	738
CGI ベースのパスワード サービスのプロパティファイル	739
サーブレットベースパスワード サービスの JSP 形式のカスタマイズ	742
サーブレットベースのパスワードサービス JSP フォームの概要	742
サーブレットベースのパスワード サービス JSP フォームの変更	744
サーブレットベースのパスワード サービスのローカライズ	745
追加属性情報を必要とする認証方式	748
パスワード ポリシーのトラブルシューティング	750
新しいユーザ パスワードが拒否される	750
ユーザ アカウントが誤って無効にされる	751
ユーザ アカウントが途中で無効になる	751
パスワードの変更が強制される	751
LDAP ユーザが無効にならない	752
Active Directory ユーザがパスワードを変更できない	752
パスワードが誤っていることを示すメッセージが表示されない	753

第 24 章: SiteMinder テストツール 755

テスト ツールの概要	755
テスト環境エージェントの設定	756
ポリシー サーバの識別	757
リソース情報の指定	760
ユーザ クレデンシャルの指定	760
エンコーディング仕様の設定	761
設定の保存とロード	762

機能テストの実行	762
機能テストの結果	765
平均経過時間の計算	767
リグレーション テストの実行	767
ストレス テスト	768
スレッド制御ファイル	768
レポートの表示	770
証明書ベースの認証テスト	771
カスタム マッピングを必要とする証明書属性	771
テスト用のカスタム属性マッピング	771

付録 A: SSL 認証方式のトラブルシューティング 775

概要	775
SSL 接続機能の決定	775
SSL 構成	776
Netscape Web サーバ証明書のインストール	776
SSL を使用する Netscape Web サーバの設定	777
Netscape のクライアント証明書に対する Web サーバの信頼	778
Windows のクライアント証明書に対する Web サーバの信頼	779
Apache のクライアント証明書に対する Web サーバの信頼	780
SSL のトラブルシューティング	781
証明書用のプロンプトはありませんでした	781
前の手順に従っても、証明書のプロンプトが表示されない	782
証明書のプロンプトの表示後に認証失敗を受け取る	785
SiteMinder ポリシーによってアクセスが許可されるはずですが、SSL 認証失敗のメッセージを受信しました	786
「見つかりません」エラー メッセージの表示	786
証明書認証方式または基本認証方式を実行しているが、基本クレデンシャルを入力できません。	787

付録 B: LanMan ユーザ ディレクトリ 789

LanMan ユーザ ディレクトリの概要	789
LanMan ディレクトリ接続の前提条件	790
LanMan ディレクトリ接続の設定	790
LanMan ディレクトリ接続用のレジストリ キーの設定	790
LanMan ユーザ ディレクトリ接続の設定	791

Windows ユーザ ディレクトリのフェイルオーバー	792
LanMan ユーザディレクトリの検索条件	792

付録 C: CA SSO/WAC の統合 793

概要	793
統合された SiteMinder および CA SSO のアーキテクチャ例	794
CA SSO 前の SiteMinder で保護されたリソースへのユーザ アクセス	795
認証済み CA SSO クライアント ユーザの SiteMinder リソースへのアクセス	797
SiteMinder にアクセスする前の eTrust WAC で保護されたリソースへのユーザ アクセス	799
SiteMinder と CA SSO の統合の前提条件	801
SiteMinder から CA SSO へのシングル サインオンの設定	802
CA SSO クライアントから SiteMinder へのシングル サインオンの設定	805
CA SSO から SiteMinder へのシングル サインオンの設定	806
smetssock Web エージェント アクティブ レスポンス属性の設定	808
smauthsso カスタム認証方式の設定	810

付録 D: Enterprise Log Manager 統合 813

付録 E: RADIUS サーバとしてのポリシー サーバの使用 815

RADIUS サーバとしてのポリシー サーバの使用	815
RADIUS クライアント/サーバ アーキテクチャ	816
ポリシー サーバを使用した RADIUS 認証の動作	816
RADIUS 環境のポリシー	818
RADIUS リソースと非 RADIUS リソース	820
レルム ヒントの使用	821
RADIUS ポリシー ドメインのレスポンス	823
レスポンスの機能	823
属性タイプ	824
常に RADIUS 属性を返す SiteMinder の設定	826
エージェント タイプの属性の作成	827
RADIUS 環境への SiteMinder の展開	832
RADIUS デバイスの保護に関するガイドライン	833
同機種 RADIUS 環境でユーザを認証する方法	834
ユーザ ディレクトリのセットアップ	835
ポリシー ドメインのセットアップ	835

認証方式の作成	836
1 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザの認証	836
1 つのディレクトリを持つ異機種環境でユーザが認証される仕組み	837
システムとポリシー ドメインの設定	838
1 つのディレクトリを持つ異機種環境のエージェントの定義	840
ユーザ ディレクトリの設定	840
ポリシー ドメインの作成	841
2 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法	841
システムとポリシー ドメインを設定する方法	844
2 つのディレクトリを持つ異機種環境のエージェントの定義	845
ユーザ ディレクトリのセットアップ	845
2 つのポリシー ドメインの作成	846
RADIUS エージェント グループの概要	846
RADIUS エージェント グループのセットアップ	847
RADIUS レスポンスのグループ化	848
RADIUS のトラブルシューティングとテスト	849
アカウンティングとデバッグ用の RADIUS ログの生成	850
smreadclog による RADIUS ログ ファイルの読み込み	850
SiteMinder テスト ツールを使用してテストする方法	852

付録 F: 属性および式のリファレンス 855

データ型	855
式の構文の概要	858
貼り付け	860
オペレータ	862
等価演算子	863
不等価演算子	864
「より小さい」演算子	864
「より大きい」演算子	865
「以下」演算子	866
「以上」演算子	866
先頭演算子	867
末尾演算子	867
包含演算子	868
セット包含演算子	868
パターン一致演算子	869

セット共通演算子	873
セット結合演算子	873
NOT 演算子	874
AND 演算子	874
OR 演算子	874
排他的 OR 演算子	875
文字列連結演算子	875
算術加算演算子	876
算術減算演算子	876
算術乗算演算子	877
算術除算演算子	877
条件付き演算子	878
インデックス演算子	878
式の中に使用できる関数	879
ABOVE 関数 -- 指定した LDAP DN より上にあるユーザ	882
ABS 関数 -- 絶対値の取得	883
AFTER 関数 -- 文字列の取得	884
ALL 関数 -- 設定されたすべてのビット	885
ANDBITS 関数 -- ビット単位の AND 演算の実行	886
ANY 関数 -- 設定されたいずれかのビット	887
AT 関数 -- 指定された LDAP DN にあるユーザ	888
BEFORE 関数 -- 文字列の取得	888
BELOW 関数 -- 指定した LDAP DN より下にあるユーザ	890
BOOLEAN 関数 -- 「TRUE」または「FALSE」への切り替え	891
CHAR 関数 -- ASCII 値の変換	892
CENTER 関数 -- ソース文字列の長さの調整	893
COMMONDN 関数 -- 共通ルートの取得	894
COUNT 関数 -- セット内の要素の数	895
DATE 関数 -- 午前 0 時(形式 1) への設定	896
DATE 関数 -- 年、月、日、時、分、および秒(形式 2) の変換	897
DATEFROMSTRING 関数 -- 数値への文字列の変換	898
DATETOSTRING 関数 -- 文字列への数値の変換	899
DAY 関数 -- 月の日付の取得	901
DOW 関数 -- 曜日の取得	902
DOY 関数 -- 年内の日付の取得	903
ENUMERATE 関数 -- セット要素のチェック	903

ERROR 関数 -- コンソール ログへのエラー メッセージの書き込み	905
EVALUATE 関数 -- 式の評価	905
EXISTS 関数 -- ファイル名の検索	906
EXPLODEDN 関数 -- セットへの LDAP DN の変換	907
FILTER 関数 -- セット要素のチェック	908
FIND 関数 -- 文字列内の位置の取得	909
GET 関数 -- ユーザ ディレクトリでの属性の検索	911
HEX 関数 -- 16 進数への変換	912
HOURL 関数 -- 時間への変換	913
HOUR24 関数 -- 時間への変換	913
INFO 関数 -- コンソール ログへの INFO メッセージの書き込み	914
KEY 関数 -- キーの検索	915
LCASE 関数 -- 小文字への変換	916
LEFT 関数 -- 文字列の一部を取得	917
LEN 関数 -- 文字列の長さの取得	917
LOG 関数 -- ファイルへの文字列の書き込み	918
LOOP 関数 -- ループでの仮想属性の呼び出し	919
LPAD 関数 -- ソース文字列の左側での長さの調整	920
LTRIM 関数 -- 文字列の先頭にあるスペースの削除	921
MAX 関数 -- 2 つのうち、より大きい値の特定	922
MAYBE 関数 -- 不確定結果のレポート	922
MID 関数 -- 文字列の一部を取得	924
MIN 関数 -- 2 つのうち、より小さい値の特定	925
MINUTE 関数 -- 日付の「分」要素の取得	926
MOD 関数 -- 除算の剰余の取得	926
MONTH 関数 -- 日付の「月」要素の取得	927
NOTBITS 関数 -- ビット単位の NOT の実行	928
NOW 関数 -- 秒単位での現在の時間の取得	928
NOWGMT 関数 -- 秒単位での現在の時間の取得	929
NUMBER 関数 -- 数値への変換	930
ORBITS 関数 -- ビット単位の OR 演算の実行	930
PARENTDN 関数 -- LDAP ツリー内の親の取得	931
PCASE 関数 -- 文字列の大文字への変換	932
QS 関数 -- クエリ文字列からの項目の取得	933
RDN 関数 -- LDAP DN の最初の要素の取得	934
RELATIONDN 関数 -- 2 つの識別名の比較	935

RIGHT 関数 -- 文字列からの文字の取得	937
RPAD 関数 -- 右側の文字列の長さの調整	938
RPT 関数 -- 文字列の反復	939
RTRIM 関数 -- 文字列の後続スペースの削除	940
SECOND 関数 -- 日付の秒数の取得	941
SET 関数 -- 属性値の設定	941
SIGN 関数 -- 数値の符号の取得	942
SORT 関数 -- セットの並べ替え	943
SPACE 関数 -- 一連のスペースの取得	944
STRING 関数 -- 文字列への変換	945
THROW 関数 -- 処理の停止およびカスタム エラーのレポート	945
TRACE 関数 -- コンソール ログへのトレース エントリの書き込み	946
TRANSLATE 関数 -- 文字列値の置換	947
UCASE 関数 -- 大文字への変換	948
URL 関数 -- URL 文字列の要素の取得	949
URLDECODE 関数 -- URL 文字列のデコード	952
URLENCODE 関数 -- 文字列のエンコード	952
VEXIST 関数 -- パラメータ定義の判定	953
TRACE 関数 -- コンソール ログへのトレース エントリの書き込み	955
XORBITS 関数 -- ビット単位の XOR 演算の実行	955
YEAR 関数 -- 日付の数値の「年」要素の取得	956
YEAR4 関数 -- 日付の「年」要素の取得 (4 桁)	957

第 1 章: SiteMinder の概要

このセクションには、以下のトピックが含まれています。

[SiteMinder のコンポーネント](#) (P. 29)

[ポリシー サーバの概要](#) (P. 29)

[ポリシー サーバ管理コンソールの概要](#) (P. 32)

SiteMinder のコンポーネント

SiteMinder は、以下の 2 つの中心的なコンポーネントで構成されています。

ポリシー サーバ

ポリシー サーバには、ポリシー管理、認証、許可、およびアカウントिंगの各機能が用意されています。

SiteMinder エージェント

標準の Web サーバまたはアプリケーション サーバと SiteMinder エージェントを統合することにより、SiteMinder では、あらかじめ定義されたセキュリティポリシーに従って、Web アプリケーションやコンテンツへのアクセスを管理することができます。そのほかに、SiteMinder エージェントには、SiteMinder が Web 以外のエンティティへのアクセスを制御することを可能にするタイプもあります。たとえば、SiteMinder RADIUS エージェントは RADIUS デバイスへのアクセスを管理し、SiteMinder アフィリエイト エージェントはポータル サイトからアフィリエイトの Web サイトに渡された情報を管理します。

ポリシー サーバの概要

通常、ポリシー サーバは別個の Windows または Solaris システム上で稼働して、SiteMinder の重要なセキュリティ処理を実行します。ポリシー サーバは、以下の機能を提供します。

認証

ポリシー サーバは、さまざまな認証方式をサポートしています。ポリシー サーバは、ユーザ名とパスワード、トークン、フォームベースの認証、あるいは公開キー証明書などに基づいてユーザを認証します。

許可

ポリシー サーバは、ポリシー サーバ管理者が確立したアクセス制御ルールの管理および適用を行います。これらのルールは、保護された各リソースに対して許可された操作が定義されています。

管理

ポリシー サーバは、**CA SiteMinder 管理 UI** を使用して設定できます。管理 UI を使用して設定情報をポリシー ストアに記録することを可能にしているのが、ポリシー ストアの管理サービスです。

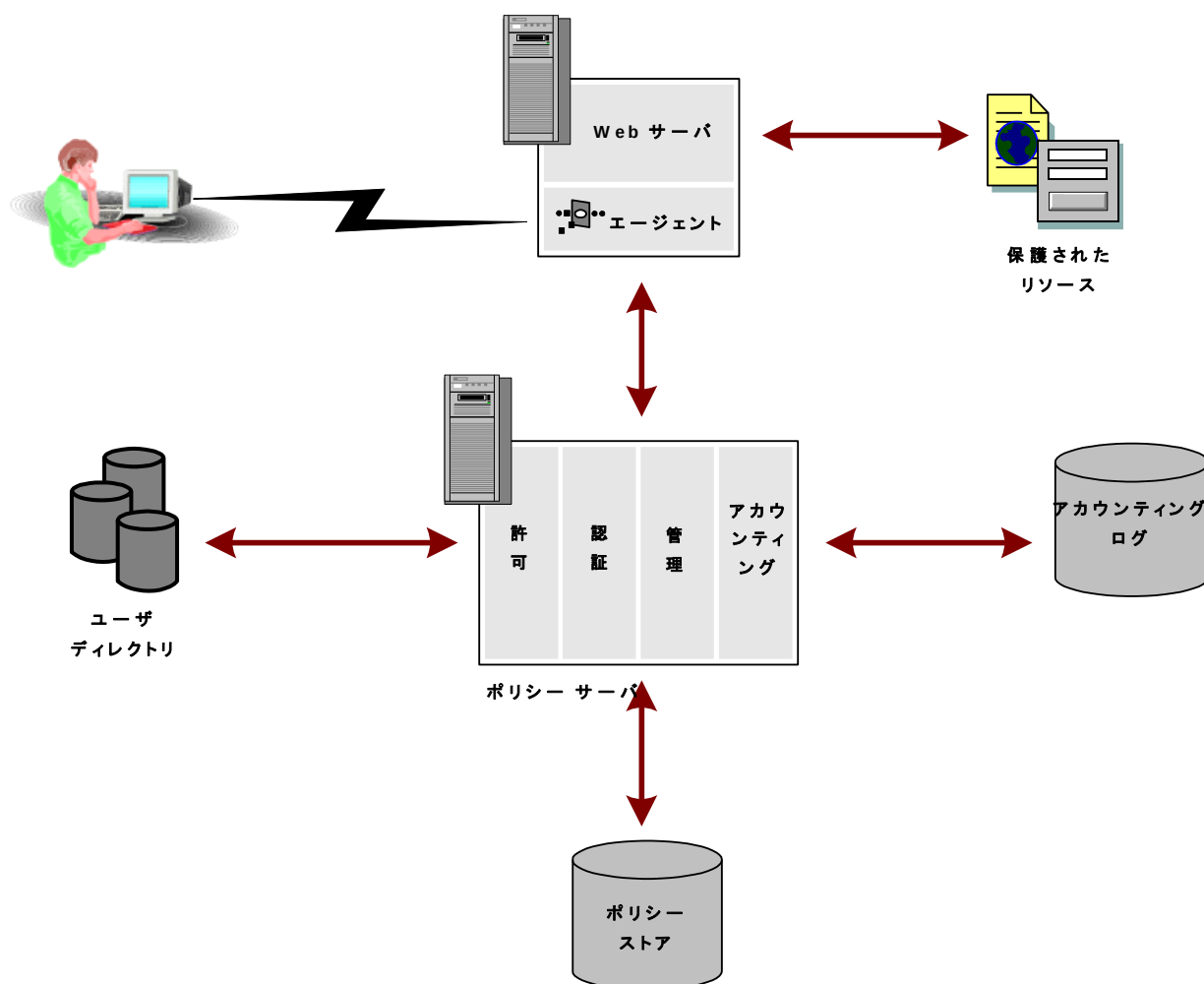
アカウントिंग

ポリシー サーバは、システム内で発生するイベントの監査情報を記録するログ ファイルを生成します。これらのログは、あらかじめ定義されたレポート形式で印刷して、セキュリティイベントや異常の分析に使用できます。

正常性監視

ポリシー サーバには、**SiteMinder** 展開全体でアクティビティを監視する機能があります。

以下の図に、単純な SiteMinder 環境を示します。



実際の Web 環境では、ユーザはブラウザを介してリソースをリクエストします。このリクエストは、Web サーバが受信し、SiteMinder Web エージェントがインターセプトします。Web エージェントは、リソースが保護されているかどうかを判別し、保護されている場合は、ユーザのクレデンシャルを収集してポリシー サーバに渡します。ポリシー サーバは、ユーザ固有のディレクトリに照らし合わせてユーザを認証します。次に、ポリシーストア内のルールとポリシーに基づいて、リクエストしたリソースへのアクセスが認証されたユーザに許可されているかどうかを確認します。ユーザが認証され、許可されると、ポリシー サーバは保護されたリソースへのアクセス権を付与し、権限と権限付与に関する情報を配信します。

注: その他のタイプのエージェントは、エージェント API を使用して作成できます。

ポリシー サーバ管理コンソールの概要

ポリシー サーバ設定タスクの大半は、管理 UI を使用して実行します。ただし、ポリシー サーバ設定タスクの中には、ポリシー サーバ管理コンソールを使用して実行するものもあります。

ポリシー サーバ管理コンソールによって制御する管理タスクを、以下に示します。

- ポリシー サーバ プロセスの起動と終了
- ポリシー サーバ エグゼクティブの設定
- キャッシュ管理
- キー管理
- グローバル設定
- ユーザの管理

注: ポリシー サーバ管理コンソールの詳細については、「ポリシー サーバ管理ガイド」を参照してください。

第 2 章: ポリシー サーバ オブジェクトの概要

このセクションには、以下のトピックが含まれています。

[ポリシー サーバのオブジェクトタイプ](#) (P. 33)

[設定順](#) (P. 40)

ポリシー サーバのオブジェクト タイプ

ポリシー サーバが使用するオブジェクトのカテゴリは、主に 3 つです。

- [インフラストラクチャ オブジェクト](#) (P. 34)
- [ポリシードメインのオブジェクト](#) (P. 37)
- [グローバル オブジェクト](#) (P. 38)

インフラストラクチャ オブジェクト

インフラストラクチャ オブジェクトは、**SiteMinder** を展開するにあたって全般的に使用されます。これらオブジェクトには、既存のユーザ ディレクトリへの接続、管理者、エージェント、認証方式、登録方式、およびパスワード ポリシーなどがあります。

インフラストラクチャ オブジェクトには次のものがあります。

エージェント

エージェントは、**Web** サーバやアプリケーション サーバなどのネットワーク エンティティ上にインストールされ、リソースへのアクセスを安全に保ちます。サーバにインストールしたエージェントは、管理 UI で **SiteMinder** オブジェクトとして設定する必要があります。

エージェント グループ

エージェント グループは、エージェントのグループを指す **SiteMinder** オブジェクトです。グループ内のエージェントを別のサーバ上にインストールすることができますが、そのすべてのエージェントは同じリソースを保護します。通常 **SiteMinder** では、使用頻度の高いリソース セットにアクセスする際の作業負荷を分散させるサーバ グループに対してエージェント グループを設定します。

エージェント設定オブジェクト

1 つ以上の **Web** エージェントの設定パラメータが保持されます。

ホスト設定オブジェクト

トラステッドホストの設定パラメータが保持されます。

ユーザ ディレクトリ

SiteMinder のユーザ ディレクトリは、**SiteMinder** の外部にある既存のユーザ ディレクトリに接続するための詳細情報が設定されているオブジェクトです。ユーザ ディレクトリ接続によって、**SiteMinder** 内のユーザ情報をレプリケートする代わりに、既存のユーザ ディレクトリへの接続を設定できます。

ポリシードメイン

ポリシードメインは、1 つ以上のユーザディレクトリ、管理者、およびレルムに関連したリソースの論理グループです。このポリシー サーバ オブジェクトは、権限付与に関するデータの基礎となります。ポリシードメインを作成することで、管理者は、特定のリソースグループ(レルム)やそのリソースにアクセスできるユーザ、および権限を設定する管理者を含む権限付与のためのコンテナを作成することができます。

アフィリエイトドメイン

アフィリエイトドメインは、1 つ以上のユーザディレクトリと管理者に関連付けられた SAML アフィリエイトの論理グループです。

注: アフィリエイトドメインは、**Federation** セキュリティサービス管理ユーザインターフェースを使用して作成する必要があります。アフィリエイトドメインの詳細については、「*Federation Security Services Guide*」を参照してください。

管理者

管理者は、**SiteMinder** 管理者アカウントのプロファイル情報が設定されているオブジェクトです。**SiteMinder** にログインするユーザはすべて管理者としてみなされます。管理者アカウントの権限と動作は、管理ロールによって異なります。

認証方式

認証方式は、ユーザが保護されたリソースにアクセスするために必要な証明を決定するポリシー サーバ オブジェクトです。認証方式はレルムに割り当てられます。ユーザがレルム内のリソースにアクセスしようとすると、そのリソースへのアクセスに必要な証明がレルムの認証方式により決定されます。

登録方式

登録方式は、ユーザがネットワーク上のリソースグループにアクセスできるように、自らユーザ登録したり、管理者が登録ユーザを管理できるようにしたりするポリシー サーバ オブジェクトです。登録方式により、大規模なユーザデータベースの管理作業が簡素化されます。

エージェントタイプ

エージェントタイプは、**Web** やアフィリエイト、**RADIUS**、カスタムなどのエージェントタイプによってサポートされる、アクションおよびレスポンス属性を定義するポリシー サーバ オブジェクトです。

SQL クエリ方式

SQL クエリ方式は、SiteMinder SQL クエリを格納するオブジェクトです。これらのクエリは、SiteMinder ユーザ ディレクトリとして使用するリレーショナル データベースから、ユーザ グループのリストなどの情報を取得する際に使用します。

パスワード ポリシー

パスワード ポリシーは、有効期限、制約、構成要件などのパスワードに関するルールが設定されているポリシー サーバ オブジェクトです。

SAML アフィリエーション

SAML アフィリエーションは、単一のプリンシパルの名前識別子を共有する SAML 2.0 エンティティのグループです。

注: SAML アフィリエーションは、Federation セキュリティ サービス管理ユーザ インターフェースを使用して作成する必要があります。SAML アフィリエーションの詳細については、「*Federation Security Services Guide*」を参照してください。

トラステッド ホスト

ポリシー サーバに接続するクライアント コンポーネントを表します。

ポリシードメインのオブジェクト

ポリシードメインとは、リソースの特定ドメインを処理するオブジェクトのグループです。たとえば、企業でネットワークリソースを業務単位によって分割し、マーケティング用のポリシードメインを作成したり、エンジニアリング用に別のポリシードメインを作成したりする場合があります。ポリシードメインのオブジェクトは、特定のポリシードメインに属しています。これらのオブジェクトには、リソースへのアクセスを制御するルールやポリシーが含まれます。

ポリシードメイン オブジェクトには以下の項目が含まれます。

レルム

レルムはリソースのグループを識別するポリシー サーバ オブジェクトです。通常レルムは、ディレクトリまたはフォルダと、必要に応じてそのサブディレクトリを定義します。

ルール

ルールは、リソースと、そのリソースに対して許可または拒否されるアクションを定義するポリシー サーバ オブジェクトです。またルールには、特定のイベントに関連したアクション（たとえば、証明の要求時にユーザが認証に失敗した場合のアクションなど）も含まれます。

ルール グループ

ルール グループは、複数のルールが設定されているポリシー サーバ オブジェクトです。ルール グループは、1 つのポリシーで使用される別々のルールを結び付けるために使用されます。

レスポンス

レスポンスは、ルールに対する反応を決定するポリシー サーバ オブジェクトです。レスポンスはポリシーの一部であり、ルールが起動されると発生します。

レスポンス グループ

レスポンス グループは、レスポンスの論理グループが含まれているポリシー サーバ オブジェクトです。レスポンス グループが最もよく使用されるのは、1 つのポリシーに多くのレスポンスが含まれるケースです。

ポリシー

ポリシーは、ユーザ、ルール、レスポンスを結び付けたり、必要に応じて時間制限や IP アドレス制限を結び付けたりするポリシー サーバ オブジェクトです。ポリシーによって **SiteMinder** で保護されたエンティティに対する権限の付与が実際に行われます。ユーザがリソースにアクセスしようすると、**SiteMinder** は最終的にポリシーを使用して、そのリクエストを判断します。

変数

変数は、1 つの値に解決して、その値をリクエストの許可段階に組み込むことができるオブジェクトです。変数オブジェクトの値は、動的データの結果であり、実行時に評価されます。

アフィリエイト

アフィリエイト オブジェクトは、ユーザを結び付けたり、必要に応じて時間制限や IP アドレス制限を結び付けたりするオブジェクトです。ユーザの認証後にアフィリエイトに渡す設定データやユーザ権限付与属性もこのオブジェクトに含まれています。

注: アフィリエイトについての詳細は、「*Federation Security Services Guide*」を参照してください。

グローバル オブジェクト

ドメイン内の特定リソースにポリシーを設定するだけでなく、すべてのリソースに適用されるグローバル ポリシー オブジェクトを設定することもできます。

グローバル オブジェクトには以下のものが含まれます。

グローバル ルール

グローバル ルールは、多数のリソースの集合に対してグローバル ポリシーを適用するためのフィルタを指定するポリシー サーバ オブジェクトです。

グローバル レスポンス

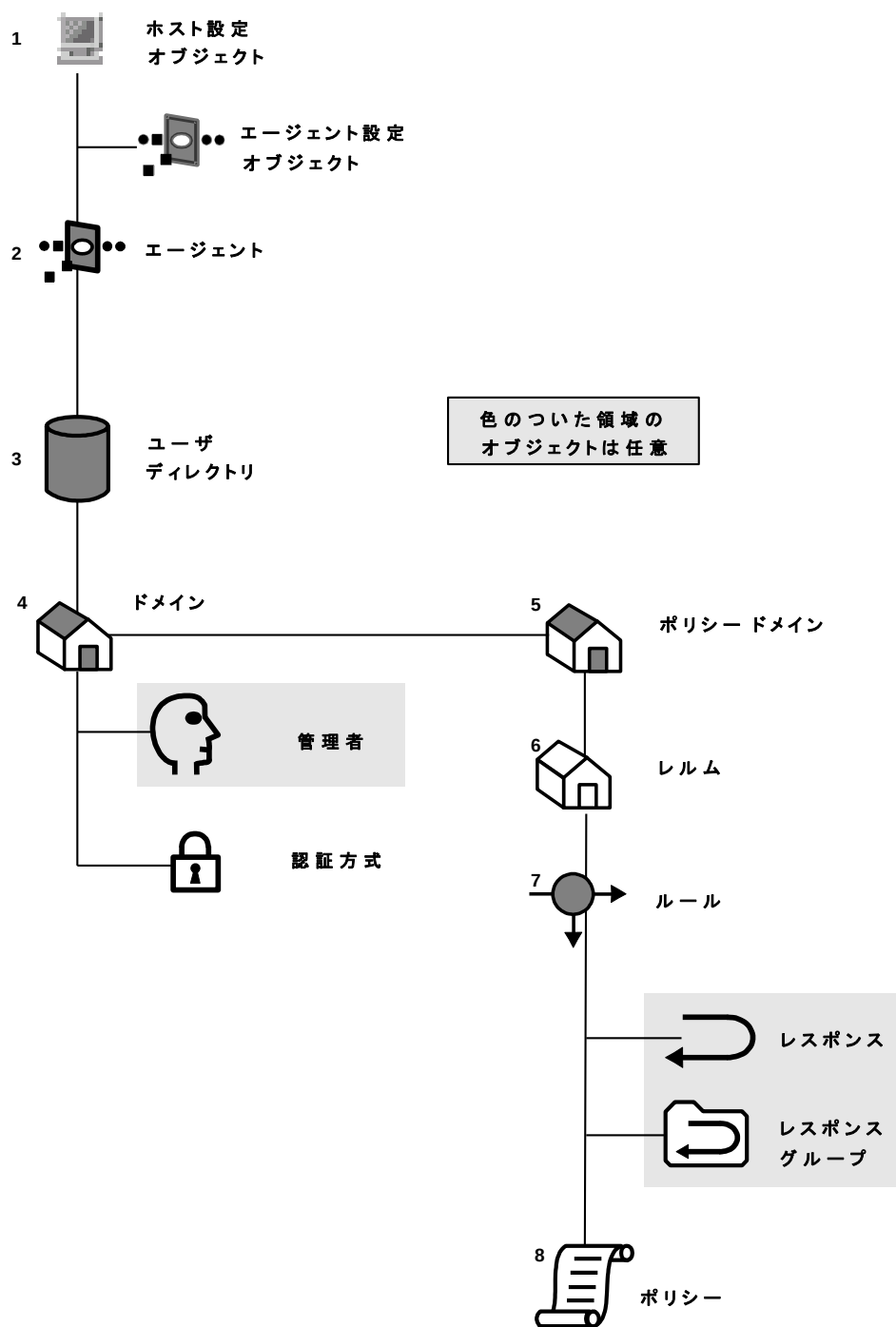
グローバル レスポンスは、グローバル ルールに対する反応を決定するポリシー サーバ オブジェクトです。グローバル レスポンスはグローバル ポリシーの一部であり、グローバル ルールが起動されると発生します。

グローバル ポリシー

グローバル ポリシーは、ユーザ、グローバル ルール、グローバル レスポンスを結び付けたり、必要に応じて時間制限や IP アドレス制限を結び付けたりするポリシー サーバ オブジェクトです。ユーザがリソースにアクセスしようとする、**SiteMinder** は最終的にグローバル ポリシーを使用して、そのリクエストを判断します。

設定順

SiteMinder の展開では、特定の順序でコア SiteMinder オブジェクトを設定する必要があります。以下の図は、コア SiteMinder オブジェクトの設定指示を示します。



第 3 章: ポリシーベースのセキュリティの実装

このセクションには、以下のトピックが含まれています。

[ポリシー ベース セキュリティの概要](#) (P. 43)

[セキュリティとユーザを管理するための計画](#) (P. 44)

[セキュリティモデルの実装](#) (P. 51)

[SiteMinder アプリケーション ロール](#) (P. 56)

ポリシー ベース セキュリティの概要

ユーザを管理し、リソースへの安全なアクセスを確保することは、アプリケーションや Web サイト、ポータル運用にとって重要な課題です。ユーザと保護されたリソースを効率的に管理するには、次の条件を満たす必要があります。

- ユーザが目的の情報に簡単かつすばやくアクセスできるようにする。
- ユーザが安心できるセキュリティを提供する。
- アクセス権のないユーザからリソースを保護する。

ユーザの管理とリソースの保護を効率よく実行できないと、会社と顧客に、以下のようなマイナスの影響が及ぶことがあります。

- セキュリティが不十分な場合には、リソースが危険に晒される。
- サイトのセキュリティが信頼できない場合、ユーザがサイトの利用を避けるようになり、ビジネスチャンスを逃す。
- リソースにアクセスするための手順が面倒な場合、ユーザが不満を感じる。
- ユーザの動向を的確に把握していないと、消費者データを失う。

Web サイトまたは Web アプリケーションを構築するときには、リソースの保護と、ユーザベースの管理の両方を配慮した計画を作成し、実行することが必須条件となります。

セキュリティとユーザを管理するための計画

次のような疑問に答えられることが、完全なセキュリティソリューションの必須条件です。

- どの部分にどの程度のセキュリティが必要か。
- どんな種類のタスクを実行するか。セキュリティに関連したタスクを実行する必要があるのか。
- 保護する必要があるリソースはどれで、どの程度の保護が必要か。
- リソースへのアクセスが許可されるのはどのユーザで、そのリソースに対してどんな操作を実行できるか。
- 保護されたリソースに対するユーザ アクセスをどのように制御するか。

ユーザ管理およびリソース保護に使用される最も一般的な方法のうちの 2 つが、アクセス制御リスト (ACL) とセキュリティ ポリシーです。

セキュリティ ポリシーは、ユーザまたはユーザグループが持つリソースに対するアクセス権限のタイプだけでなく、そのユーザまたはユーザグループがリソースにアクセスした場合にどのような処理が行われるかを定義することもできる、最も完全なセキュリティソリューションです。セキュリティ ポリシーを使用すれば、ACL のように単にアクセスを管理するだけでなく、ユーザに対して行われる処理を管理することも可能になります。SiteMinder の許可モデルは、セキュリティ ポリシーに基づきます。

アクセス制御リスト

アクセス制御リスト (ACL) は、リソースに関連付けられたオブジェクトで、個々のユーザまたはユーザグループのアクセス権限を定義するものです。ACL をリソースに関連付けることで次のことを定義できます。

- リソースへのアクセスが許可されるユーザ
- そのユーザに付与されるアクセス権限のタイプ

ACL は、ユーザまたはユーザグループを指定して、リソースへのアクセスを許可または拒否するかを定義する簡単な方法です。以下に例を示します。

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

このアクセス制御リストは、管理者グループに属するユーザに完全なアクセス権限、事務職グループに属するユーザには読み込み専用アクセス権限をそれぞれ付与し、その他のすべてのグループに属するユーザに対してはリソースへのアクセスを拒否します。

ACL には次のような短所があります。

- いったんリソースへのアクセスが許可され、認証されると、ユーザとリソースとの対話を制御するのが難しい。たとえば、ACL を使用してセッションタイムアウトを定義することはできません。
- 多数のリソースの ACL を管理するのは時間と労力、コストを要する。ACL の管理は管理者にとって大きな負担となります。
- ユーザ情報のプロフィール作成が難しい。ACL を使用する場合、ACL にユーザを追加してリソースに関連付けるという方法をとるため、あるユーザがアクセスを許可されているすべてのリソースを特定するのが難しくなります。
- アクセス権限に条件を設定するのが難しい。時間や場所に基づいてアクセスを制限するのはほぼ不可能です。
- 内容のパーソナライズやレスポンスの定義が難しい。ユーザにリソースへのアクセスを許可しても、その内容をパーソナライズしたり、ユーザがアクセスを許可または拒否された場合にどのような処理が行われるかを定義したりするのは難しくなります。たとえば、ACL を使用した場合、ユーザがリソースにアクセスしたときに特定のボタンを表示したり、非表示にしたりするといった制御はできません。

ACL は、リソースを保護するには効率的な方法と言えますが、ユーザに対して行われる処理までをも管理するには不十分です。

SiteMinder セキュリティポリシー

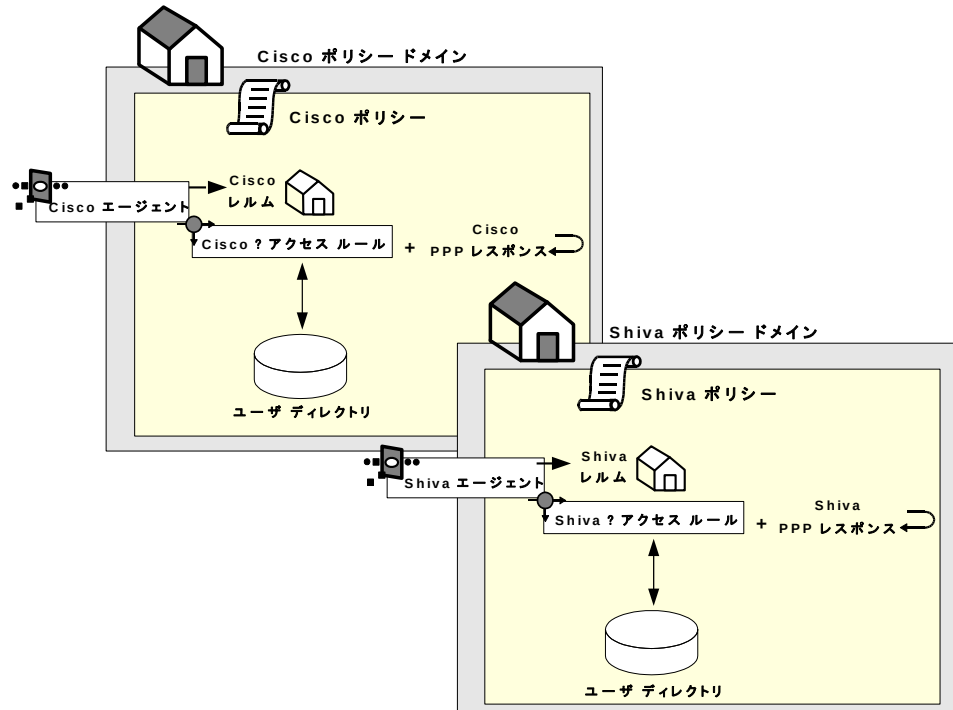
ACL とは異なり、ポリシーはセキュリティを提供するだけでなく、ユーザに対する処理を管理するという 2 つの役割を果たします。ポリシーはユーザ主体型です。つまり、ポリシーはリソースではなく、ユーザグループを主体として作成します。

ポリシーを使用する場合には、ルールやレスポンス、時間/場所による制約を使用してアクセス許可を定義し、このポリシーをユーザまたはユーザグループに関連付けて、次のような条件を確立します。

- リソースの場所
- リソースへのアクセスが許可されるユーザ

- そのユーザに付与されるアクセス権限のタイプ
- リソースへのアクセスが許可される場合
- ユーザがリソースにアクセスしたときに行われる処理
- ユーザがリソースにアクセスできないときに行われる処理

以下の図は、SiteMinder セキュリティ ポリシーの定義の仕組みを示します。



ポリシーを使用すれば、ユーザ管理やリソース保護を効率的に行うことができます。これには、次のような理由があります。

- ポリシーを使用すれば、よりきめ細かな管理が可能になり、内容をパーソナライズすることができます。ユーザがアクセスを許可されたときにどのグラフィックを表示するか、ユーザがアクセスを拒否されたときにどこにユーザをリダイレクトするかなど、ユーザがアクセスを許可または拒否された場合にレスポンスを通じてどのような処理が行われるかを定義できます。
- ポリシーは管理も容易です。ユーザの変更、グループへの追加、またはグループからの削除などの変更が行われると、そのユーザが含まれるすべてのポリシーが自動的に更新されます。
- ポリシーはきめ細かなセキュリティを提供します。ポリシー定義には、時間的制約や場所 (I.P) による制約を含めることができます。

このようにポリシーは強力かつ柔軟であるため、セキュリティポリシーに基づく認証モデルは ACL に基づくモデルよりも効率的で効果的です。

エンド ユーザ エクスペリエンスの管理

SiteMinder のポリシーでは、リソースを保護するだけでなく、以下によってエンド ユーザ エクスペリエンスを管理することもできます。

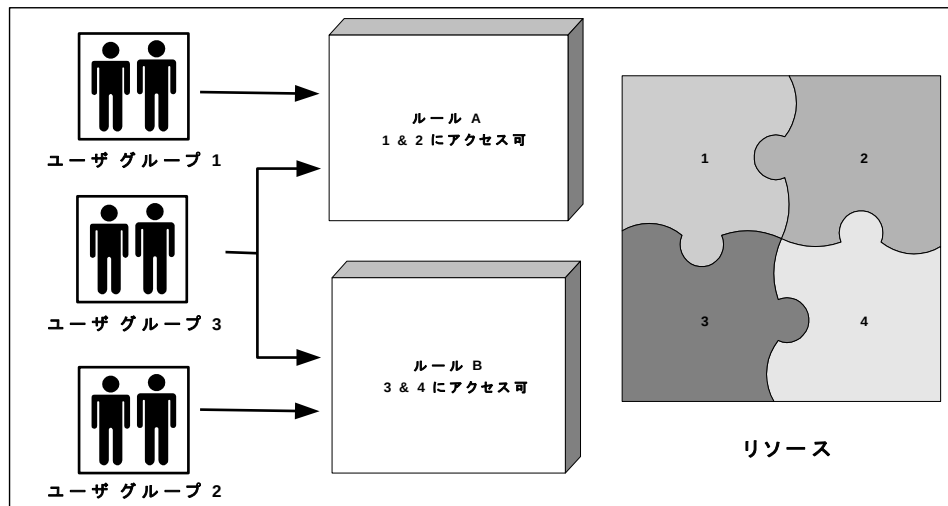
- [権限を確立する方法](#) (P. 47)
- [コンテンツをパーソナライズする方法](#) (P. 48)
- [セッションを管理する方法](#) (P. 49)

権限を確立する方法

ポリシーでは、ポリシーにルールを追加してリソースへのアクセス権限を確立します。ルールとは、特定のリソースを識別して、そのリソースに対するアクセスを許可または拒否するものです。多数のユーザについて 1 つのポリシーで権限を確立できます。つまり、ポリシーには個々のユーザ、ユーザのグループ、またはユーザディレクトリ全体のメンバーを関連付けることができます。

たとえば、以下の図のような例ではアクセス権限は次のようになります。

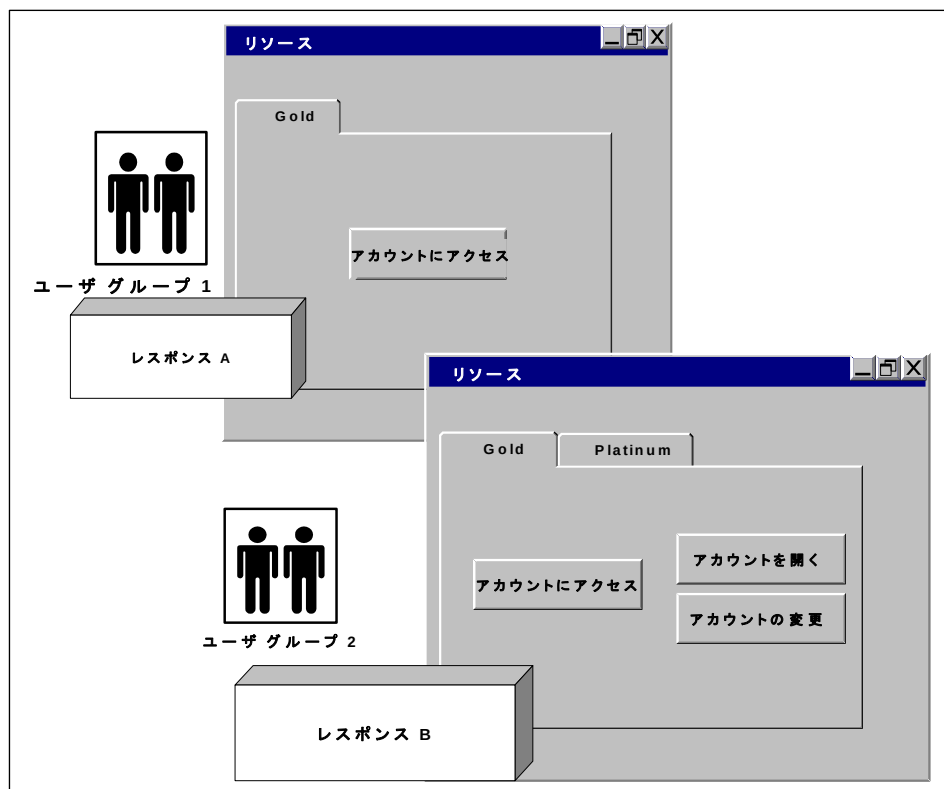
- グループ 1 には、ルール A が割り当てられており、リソース 1 と 2 へのアクセスが許可されます。
- グループ 2 には、ルール B が割り当てられており、リソース 3 と 4 へのアクセスが許可されます。
- グループ 3 には、ルール A とルール B の両方が割り当てられており、すべてのリソースへのアクセスが許可されます。



コンテンツをパーソナライズする方法

リソースへのアクセス権限を付与されたユーザについては、ポリシーを使ってそのユーザのリソースのビューをパーソナライズできます。ポリシーでは、レスポンスを使用することでリソースのビューをカスタマイズできます。レスポンスはルールと対になっており、ルールが起動されると発生します。

たとえば、以下の図の場合、グループ 1 とグループ 2 は[リソース]ダイアログボックスにアクセスできます。ただし、各グループに表示されるダイアログボックスのビューは異なります。グループ 1 のポリシーでは、レスポンス A が使用されているため、レスポンス B で表示される 2 つのボタン([アカウントを開く]および[アカウントの変更])や[Platinum]タブは表示されません。



セッションを管理する方法

セッションを管理することで、アクセス権限を持つ認証されたユーザがリソースにアクセスできる時間を制御できます。セッションを制御するには、次の条件を指定します。

- ユーザがリソースにアクセスせず、アイドル状態でいられる制限時間を指定します。

アイドル タイムアウトを設定すれば、セッションが使用されないままアクティブ状態を保持できる時間を制限することで、リソースの不正使用を防ぐことができます。アイドル タイムアウトは、ユーザがセッションをログアウトしないでコンピュータの前を離れた場合に特に便利です。アイドル タイムアウトの制限時間になると、セッションは自動的に終了します。

- ユーザが再認証を必要とせずにリソースにアクセスできる最長時間を指定します。

最大タイムアウトを設定し、一度認証されたユーザでも一定の時間が経過すると強制的に再度認証を受けるようにすることで、リソースの不正使用を防止できます。この安全策を講じることで、認証されたユーザがログアウトせずにコンピュータを離れたすきに誰かが開いているセッションを利用して、一定の時間が経過するとセッションは自動的に終了します。ユーザが戻ってきても、再度認証を受けないとリソースの使用を続けることはできません。

- いつでもリソースへのアクセスを禁止できます。

セッションがアクティブ状態を保てる最長時間を制限するだけでなく、リソースの正当性が損われたと疑われる場合にはセッションをただちに終了させることもできます。ユーザ セッションが取り消されると、管理 UI を使用して再度有効にするまで、そのユーザはユーザ ディレクトリで無効となります。

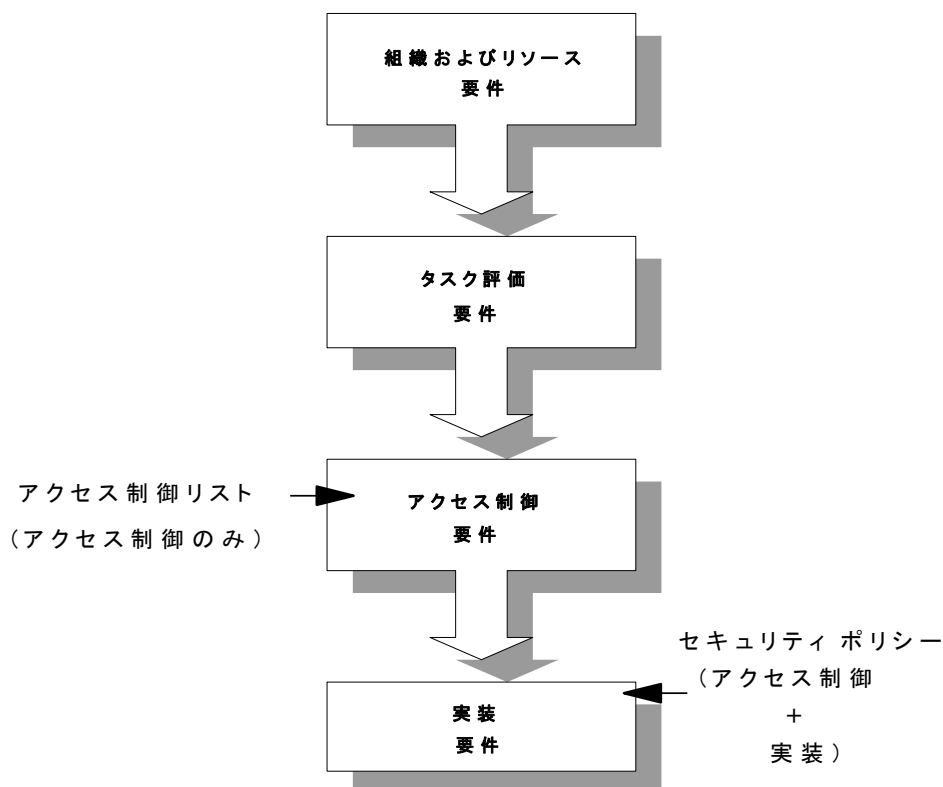
注: セッション管理の詳細については、「管理ガイド」を参照してください。

- 永続セッションを有効にします。

永続セッションを有効にすれば、**Windows** のセキュリティコンテキスト機能と統合 **Web** サービスのサポートを提供できます。

セキュリティモデルの実装

組織のニーズに最適なセキュリティモデルを実現するには、次のフローチャートに示すように、設計段階で収集した情報をもとにセキュリティポリシーを作成します。



1. 組織とリソースの要件 - セキュリティモデルの基本的な目標を設定し、リソースを特定します。
2. タスク評価の要件 - ユーザとロールを特定し、タスクにロールを関連付けます。

3. アクセス制御の要件 - ユーザ ロールの要件に基づいて、ユーザのアクセス要件を確立します。

アクセス制御リスト (ACL) のみに基づく認証モデルの場合は、作業はここまです。

4. 実装の要件 - アクセスを実装する方法 (ユーザを追跡する方法とユーザに合わせてコンテンツをパーソナライズする方法) とユーザ セッションを管理する方法を定義します。

SiteMinder セキュリティ ポリシーに基づく許可モデルには、アクセス制御モデルとアクセス実装モデルの両方が組み込まれます。

セキュリティモデル要件の整理

セキュリティモデルの設計では、各段階で次に示すような表を使って情報を整理します。この表の欄に情報を入力した後、その情報をもとに SiteMinder セキュリティ ポリシーを作成できます。

リソース	ロー ル	タスク	アクセス	実装

組織とリソースの要件に関する考慮事項

セキュリティモデルを作成するには、必要な組織とリソースを特定する必要があります。一般的な考慮事項は次のとおりです。

- 従うべきセキュリティに関するガイドライン、規定または法規があるか
- セキュリティの要件とビジネス要件が競合した場合、どちらが優先されるか
- セキュリティが不十分なために組織の評価に影響が出ているか
- セキュリティ上の問題により損害を被ったり、サイトが停止した経験がこれまでにあるか
- セキュリティに関する全体的な要件と課題を特定したら、以下で説明するように具体的なセキュリティの詳細を定義します。

組織セキュリティの要件を定義する方法

より詳細なセキュリティの要件を判断するには、次の問題を考慮する必要があります。

組織の要件	影響するタスク
■ リソースにアクセスする必要があるユーザは誰か ■ 必要なアクセス権限はどの程度か ■ アクセス要件が同じユーザをグループに分類できるか	ユーザ ディレクトリ接続の設定
■ 保護する必要があるリソースはどれか ■ リソースに応じて異なる保護レベルを設定する必要があるか	ポリシードメインとレルムの作成
■ 情報の機密性と重要性はどの程度か ■ ユーザの信頼度はどの程度か ■ ユーザは、ローカルユーザまたはリモートユーザか ■ ユーザが期待するセキュリティのタイプは何か ■ 期待を満たす水準のセキュリティを実装しないと、顧客を失う可能性があるか	認証テンプレートを使用した認証方式の作成
■ 従うべきセキュリティに関するガイドライン、規定または法規があるか ■ オブジェクトに応じてきめ細かな保護やパーソナライズを設定する必要があるか ■ どんなタイプのアクションを管理の対象とするか	ルールの定義
■ ユーザや顧客はどのような種類のセキュリティおよび管理を期待しているのか ■ ユーザグループに応じて異なるリソースビューを提供する必要があるか ■ ユーザの認証時または許可時にどのようなイベントを発生させるのか	レスポンスの定義
■ どのような方法で要件を実現するのか	ポリシーの定義

リソースとロールの識別

組織に関する要件を確立するための次の段階は、リソースを特定し、それらのリソースをロールに対応付けることです。

この手順の目的は、リソースとロールを関連付けることにあります。これらの 2 つのコンポーネントを関連付けることで、何に対してどのようなタイプの保護を適用する必要があるかを明確に把握できます。

リソースを識別するには、次の手順に従います。

- まだ存在しないが今後購入を計画しているリソースを含め、すべての既知のリソースを識別します。今後追加する予定のリソースを含め、すべてのリソースをここで特定してセキュリティ計画に含めておけば、時間と手間を節約できます。
- Web サイトのサイトマップを作成して、リソースの構造を把握します。

ポリシーに適用する方法は、以下のとおりです。

リソースはレルムとルールに定義します。ユーザのロールは、そのユーザが所属するユーザグループまたはユーザ属性を表します。たとえば、航空会社の場合には、パイロットユーザグループに属するユーザはパイロットロールに関連付けられたタスクを実行できます。

リソースとロールを識別する方法

1. この章の前出のセキュリティモデル表のような表やチャートを使用して、「リソース」欄にリソースをリストします。
2. 1 つのリソースについてすべての細目を識別してください。たとえば、`/bidding` という名前のディレクトリの下に `/bidding/flights` と `/bidding/standby` という 2 つの下位ディレクトリがある場合には、これらの下位ディレクトリもリソースとして入力します。これらの下位ディレクトリをそれぞれ別個のリソースとして扱うことで、各リソースに個別にセキュリティが必要になるかどうかを判断するのが容易になります。
3. 各リソースの隣に、そのリソースにアクセスする必要があるロールを入力します。

タスク評価要件の定義

タスク評価の要件は、ロールの要件とアクセス制御の要件の間にあるギャップを埋める役割を果たします。タスク評価の要件を識別するときには、各ロールがリソースを使用して実行するタスクを定義します。

ポリシーに適用する方法は、以下のとおりです。

タスクは特にポリシーには定義されませんが、次の項ではこの情報をもとに、リソースとロールのペアごとにアクセス権限を割り当てます。

タスク評価要件を定義する方法

1. ロールごとに、そのロールがリソースを使用して実行する必要があるタスクを識別します。
2. 関連するリソースとロールの横にある [タスク] 欄にタスクを入力します。

アクセス制御要件の定義

アクセス制御の要件は、ロールがリソースにアクセスする必要があるかどうか、アクセスする必要がある場合にはどんなタイプのアクセス権限が必要かを判断します。同じリソースにアクセスするロールでも、必要なアクセス権限は異なる場合があります。

たとえば、2 つのユーザグループが同じディレクトリでタスクを実行するとします。グループ A はリソースの表示と変更を実行するのに対して、グループ B のメンバーはリソースを表示するだけです。このように、各ユーザグループではリソースの利用法が異なるため、割り当てられるアクセス権限も次のように異なります。

- グループ A: Get、Post
- グループ B: Get

ポリシーに適用する方法は、以下のとおりです。

アクセス権限はルールに定義します。

アクセス制御要件を定義する方法

1. タスクごとに、関連するタスクを実行するために必要なアクセス権限のタイプを識別します。
2. 「アクセス」列に必要なアクセス権限を入力します。

実行要件の定義

実装要件は、リソースの使用方法を定義します。リソースがどのように使用されるかは、次のような多数の条件によって決まります。

- パーソナライズ
- リソースを使用できる時間的な制約
- リダイレクト

ポリシーに適用する方法は、以下のとおりです。

実装要件は、レスポンスに定義します。

実装要件を定義する方法

1. タスクごとに、アクセス権限をどのように適用するかを識別します。
2. 「実装」列に要件を入力します。

セキュリティ情報の入力が完了したら、サイトのニーズに応じたポリシーの作成に着手します。

SiteMinder アプリケーション ロール

アプリケーション ロールでは、組織のビジネス要件に基づいてアクセス制御を行うためにユーザのグループを指定することができます。

SiteMinder 管理者がロールを作成して割り当てます。このロールによって、保護されたアプリケーションにアクセスできるかどうかが決まります。たとえば、「会計士」の資格を持った従業員のみが会計アプリケーションを使用することを要件とするビジネスルールがあるとしてします。SiteMinder 管理者は、「会計士」の資格を持った従業員がメンバシップであるロールを作成し、そのロールにそれらの従業員を組み込みます。次に、管理者は、アプリケーションを保護するためのアプリケーション セキュリティ ポリシーを作成して、ロールをそのポリシーに関連付けます。ポリシーにより、会計アプリケーションが保護され、「会計士」の資格を持つユーザのみが許可されます。

ポリシーにユーザやユーザ グループを追加する場合と異なり、ロールの範囲は、1 つのディレクトリに限定されることも、特定のディレクトリタイプにも制限されることもありません。SiteMinder 管理者は、メンバシップの式を作成することで、管理 UI でビジネス要件を表します。メンバシップの式は、特定の LDAP および ODBC のユーザ ディレクトリ属性にマッピングされます。SiteMinder 管理者は、メンバシップの式を使用してロールを定義します。その結果、ロールはユーザ ディレクトリ固有の属性に依存せず、複数のユーザ ディレクトリにまたがることができます。

注: アプリケーション ロールの詳細については、「エンタープライズ ポリシー管理」を参照してください。

第 4 章：管理ユーザ インタフェースの管理

このセクションには、以下のトピックが含まれています。

[SiteMinder 管理者ユーザ インタフェース \(P. 59\)](#)

[管理 UI の概要 \(P. 60\)](#)

[管理 UI の起動 \(P. 60\)](#)

[ポリシー サーバ オブジェクトの管理 \(P. 61\)](#)

[タスク永続性データベースの管理 \(P. 65\)](#)

[SiteMinder 管理者 \(P. 68\)](#)

SiteMinder 管理者ユーザ インタフェース

SiteMinder r12 SP1 以降、SiteMinder ポリシー オブジェクトを設定するグラフィカル ユーザ インタフェース (UI) としては以下の 2 つがあります。

管理 UI

管理 UI はポリシー サーバとは別にインストールされる Web ベースの管理コンソールです。管理 UI では、Federation セキュリティ サービスに関連していないすべてのポリシー サーバ オブジェクトの表示、変更、削除が可能です。フェデレーション関連のすべての設定タスクは、FSS 管理 UI を使用して処理する必要があります。

Federation セキュリティ サービスの管理 UI (FSS 管理 UI)

FSS 管理 UI はポリシー サーバと一緒にインストールされるアプレット ベースのアプリケーションです。Federation 固有の UI オブジェクトは、パートナー間のフェデレーション通信をサポートするために設定する、アフィリエイト (コンシューマ、サービスプロバイダ、リソース パートナ) と SAML 認証方式で構成されています。

FSS 管理 UI は、SiteMinder Federation セキュリティ サービスを管理するために用意されています。以前のバージョンの SiteMinder ポリシー サーバのユーザ インタフェースと異なり、すべての SiteMinder オブジェクトが FSS 管理 UI に表示されることがわかります。表示されないオブジェクトは、EPM (Enterprise Policy Management) およびレポートに関連するオブジェクトだけです。FSS 管理 UI を使用すると、SiteMinder オブジェクトを管理できます。FSS 管理 UI を使用しているときに情報が必要となった場合は、FSS 管理 UI オンライン ヘルプ システムを調べます。

重要: 各 UI はポリシー サーバに登録する必要があります。FSS 管理 UI をポリシー サーバに登録することで、コンポーネント間の通信が必ず FIPS で暗号化 (AES 暗号) されるようになります。UI の登録の詳細については、「ポリシー サーバインストール ガイド」を参照してください。

管理 UI の概要

ポリシー サーバは、グラフィカル ユーザ インタフェースを使用して管理します。このインターフェースは、ユーザの管理権限に基づいて動的に生成されます。この章では、管理 UI にログインする方法と、ポリシー サーバのオブジェクトを設定および管理する場合に使用する一般的な手順について説明します。

管理 UI には、以下の 2 つのペインがあります。

- メニュー ペイン - タスクのメニュー (左側)
- タスク ペイン - 現在のタスク (右側)

左側のタスクのメニューは、開いたり閉じたりすることができます。メニューが閉じている場合は、右向きの矢印をクリックして開くことができます。同様に、メニューが開いている場合は、左向きの矢印をクリックして閉じることができます。

重要: 右側のタスク ペインを使用しているときは、左側のメニュー ペインを開いたり閉じたりする前、または別のタスクに移動する前に、必ず変更内容を保存してください。

管理 UI の起動

管理 UI を使用するには、有効な管理者としてログインする必要があります。

管理 UI を起動する方法

1. Web ブラウザを開きます。
2. 以下の URL を入力します。

`http://host.domain/iam/siteminder/adminui`

`host`

管理 UI ホスト システムの名前を指定します。

domain

管理 UI ホスト システムの完全修飾ドメイン名を指定します。

例: `http://siteminder.security.com:81/iam/siteminder/adminui`

注: URL では、完全修飾ホスト名を使用してください。ホストシステム上の Web サーバが標準の HTTP ポート(81)で実行していない場合は、ドメインの最後にコロンとポート番号を追加する必要があります。

ログイン画面が表示されます。

3. それぞれのフィールドに、有効なユーザ名とパスワードを入力します。
4. [ログイン]をクリックします。

システムは、管理者の権限に関連するタブを表示します。このウィンドウの表示内容は、UI のログイン時に使用した管理者アカウントの権限によって変わります。

ポリシー サーバ オブジェクトの管理

管理 UI では、ポリシー サーバ オブジェクトを表示、変更、および削除できます。各タスクの詳細はオブジェクトによって異なりますが、全般的な方法は似ています。たとえば、エージェントを削除する手順は、レスポンスを削除する手順と似ています。

以下のセクションでは、ポリシー サーバ オブジェクトの表示/変更/削除を行う場合の一般的なタスクを説明します。このガイド内の他の章は、リソースを管理および保護するのに必要なポリシー サーバ オブジェクトを作成する方法を説明します。

ポリシー サーバ オブジェクトの複製

新しいポリシー サーバオブジェクトを作成する最も容易な方法は、既存のオブジェクトをコピーして、そのプロパティを変更することです。新しいオブジェクトで異なる情報のみを変更して、既存のオブジェクトのプロパティをテンプレートとして使用できます。

注: 以下のオブジェクトはコピーできません。

- エージェントのタイプ
- AuthAz ディレクトリ マッピング
- AuthValidate ディレクトリ マッピング
- 証明書マッピング
- ユーザディレクトリ
- アプリケーション
- アプリケーション リソース
- ドメイン
- ポリシー
- レルム
- レスポンス
- レスポンス属性
- ルール
- グローバル ポリシー
- グローバル レスポンス
- グローバル ルール
- パスワード ポリシー
- 管理者

注: ユーザの管理者権限によってアクセスできるオブジェクトが決定されます。

既存のオブジェクトをコピーおよび変更して新規オブジェクトを作成する方法

1. [<タブ>]-[<ポリシー サーバ カテゴリ>]をクリックします。

例: [インフラストラクチャ]-[エージェント]をクリックします。

2. [<ポリシー サーバ オブジェクト>]-[<ポリシー サーバ オブジェクト> の作成] をクリックします。

オブジェクトの作成ウィンドウが開きます。

例:

[エージェント]-[エージェントの作成]をクリックします。

3. [オブジェクトのコピーの作成]を選択して検索条件を指定し、[検索]をクリックします。

検索条件と一致するオブジェクトのリストが開きます。

4. リストからオブジェクトを選択して[OK]をクリックします。

[<オブジェクト: 名前> の作成]ウィンドウが開きます。

5. [一般]グループ ボックス上のフィールドに、新しい名前および説明を入力します。

6. 新規オブジェクトで異なるプロパティを変更し、[サブミット]をクリックします。
オブジェクトの作成タスクを処理できるよう送信します。

ポリシー サーバ オブジェクト プロパティの表示

ポリシー サーバ オブジェクトのプロパティを表示できます。

注: ユーザの管理者権限によってアクセスできるオブジェクトが決定されます。

オブジェクトのプロパティを表示する方法

1. [<タブ>]-[<ポリシー サーバ カテゴリ>]をクリックします。

例: [ポリシー]-[ドメイン]をクリックします。

2. [<ポリシー サーバ オブジェクト>]-[<ポリシー サーバ オブジェクト> の表示] を選択します。

オブジェクトの表示ペインが開きます。

例:

[ドメイン]-[ドメインの表示]をクリックします。

[ドメインの表示]ペインが開きます。

3. 検索条件を指定して[検索]をクリックします。

検索条件と一致するオブジェクトのリストが開きます。

4. リストからオブジェクトを選択し、[選択]をクリックします。

オブジェクトの表示ペインが開きます。

注: 別のポリシー サーバ オブジェクトを表示するには、[検索に戻る]をクリックします。ペインを閉じるには、[閉じる]をクリックします。

既存のポリシー サーバ オブジェクトの変更

管理 UI を使用して、既存のポリシー サーバ オブジェクトのプロパティを変更できます。

注: ユーザの管理者権限によってアクセスできるオブジェクトが決定されます。

既存のオブジェクトのプロパティを変更する方法

1. [<タブ>]-[<ポリシー サーバ カテゴリ>]をクリックします。

例: [ポリシー]-[ドメイン]をクリックします。

2. [<ポリシー サーバ オブジェクト>]-[<ポリシー サーバ オブジェクト>の変更]を選択します。

オブジェクトの変更ペインが開きます。

例:

[レルム]-[レルムの変更]をクリックします。

[レルムの変更]ペインが開きます。

3. 検索条件を指定して[検索]をクリックします。

検索条件と一致するオブジェクトのリストが開きます。

4. リストからオブジェクトを選択し、[選択]をクリックします。

[<オブジェクト: 名前>の変更]ペインが開きます。

5. オブジェクトのプロパティを変更し、[サブミット]をクリックします。

オブジェクトの変更タスクが、処理のためにサブミットされます。

ポリシー サーバ オブジェクトの削除

必要でなくなったポリシー サーバ オブジェクトを削除できます。

注: ユーザの管理者権限によってアクセスできるオブジェクトが決定されます。

オブジェクトをの削除方法

1. [<タブ>]-[<ポリシー サーバ カテゴリ>]をクリックします。
例: [インフラストラクチャ]-[認証]をクリックします。
2. [<ポリシー サーバ オブジェクト>]-[<ポリシー サーバ オブジェクト> の削除]をクリックします。
[オブジェクトの削除]ウィンドウが開きます。
例:
[認証方式]-[認証方式の削除]をクリックします。
[認証方式の削除]ウィンドウが開きます。
3. 検索条件を指定して[検索]をクリックします。
検索条件と一致するオブジェクトのリストが開きます。
4. リストからオブジェクトを選択し、[選択]をクリックします。
確認ウィンドウが表示されます。
注: 一度に複数のオブジェクトを選択できます。
5. [はい]をクリックします。
オブジェクトの削除タスクを処理できるよう送信します。

タスク永続性データベースの管理

すべての 管理 UI タスクは、タスク永続性データベースに無期限に、または SiteMinder 管理者が削除するまで格納されます。クリーンアップ タスクをスケジュールすれば、データベースからタスクを削除し、ディスク空き容量を増やすことができます。クリーンアップ タスクでは、タスク永続性データベースのサイズを管理できるほか、ランタイムのパフォーマンスを改善できます。

どのタスクも、以下のいずれかの状態でタスク永続性データベースに格納されています。

■ 監査状態

監査状態のタスクは 管理 UI で開始されていますが、サブミットされていません。たとえば、表示タスクは 管理 UI で開始されますが、サブミットされることはありません。

- サブミット済み状態

サブミットされたタスクは、管理 UI で処理のためにサブミットされていますが、まだ完了していません。

- 完了状態

完了したタスクは処理を完了しているサブミットされたタスクです。完了したタスクには、正常に処理を完了したタスク、および処理は正常に完了していないが、とりあえず完了しているタスクが含まれます。

クリーンアップ タスクでは、監査状態と完了状態のタスクをタスク永続性データベースから削除できます。ただし、保留中のサブミットされたタスクは削除できません。

管理 UI の[管理]タブにある[管理 UI]メニューでは、以下の 2 つのオプションを使ってクリーンアップ タスクをスケジュール、変更、および削除できます。

サブミット済みタスクのクリーンアップ

新しいクリーンアップ タスクのスケジュール、または既存スケジュールの変更を行うには、このオプションを使用します。

反復タスクの削除

スケジュールされたクリーンアップ タスクを削除するには、このオプションを使用します。

サブミット済みタスクのクリーンアップ

クリーンアップ タスクをスケジュールすることで、タスク永続性データベースのサイズを管理し、ランタイムのパフォーマンスを改善できます。クリーンアップ タスクを設定して、データベースから完了状態および監査状態のタスクを削除できます。また、クリーンアップ タスク自体にも制限を設定できます。

サブミット済みタスクのクリーンアップ方法

1. [管理]、[管理 UI]、[サブミット済みタスクのクリーンアップ]をクリックします。

[反復]ペインが表示されます。

2. 以下のオプション ボタンのいずれかを選択します。

- 今すぐ実行

スケジュール手順をスキップし、直接[サブミット済みタスクのクリーンアップ]ペインを表示するには、このオプションを選択して[次へ]をクリックします。

- 新規ジョブのスケジュール

[スケジュールリング]グループ ボックスが表示されます。

- 既存ジョブの変更

[スケジュールリング]グループ ボックスが表示されます。

3. [スケジュールリング]グループ ボックスの[ジョブ名]フィールドにクリーンアップタスクの名前を指定し、スケジュールのタイプ、スケジュールの詳細も指定して、[次へ]をクリックします。

[サブミット済みタスクのクリーンアップ]ペインが表示されます。

4. [サブミット済みタスクのクリーンアップ]ペインの以下のフィールドに入力します。

最小期間

タスク永続性データベースから削除する完了したタスクの最小期間を月、週、日、時間、分で指定します。

注: タスクの期間は、タスクが完了した時点から測定します。

監査タイムアウト

(オプション)タスクをタスクの永続性データベース内で監査状態に維持する最大日数を指定します。

制限: 1 以上

デフォルト: 1

時間制限

(オプション)クリーンアップ タスクの時間制限を分単位で指定します。

タスク制限

(オプション)クリーンアップ タスクのタスク制限を指定します。

5. [完了]をクリックします。

クリーンアップ タスクは処理のためにサブミットされます。

反復タスクの削除

必要なくなったスケジュール済みクリーンアップ タスクを削除できます。

反復タスクを削除する方法

1. [管理]、[管理 UI]、[反復タスクの削除]をクリックします。
[反復タスクの削除]ペインが表示されます。
2. 削除する 1 つ以上のスケジュール済みクリーンアップ タスクを選択し、[サブミット]をクリックします。
タスクの削除は、処理のためにサブミットされます。

SiteMinder 管理者

ポリシー サーバ オブジェクトとツールにアクセスできるユーザはすべて SiteMinder 管理者です。組織内の各ロールに応じて、SiteMinder 管理者がアクセスできるリソースと利用できる機能は異なり、責任を負うタスクも異なります。

SiteMinder 管理モデルでは、きめ細かい管理者権限を実装できるため、組織の少数または多数のユーザを対象にしてポリシー サーバ オブジェクトと SiteMinder ツールの管理を整理できます。

デフォルトの管理者アカウント

ポリシー ストアを設定すると、デフォルトの SiteMinder スーパーユーザ アカウントが作成されます。このアカウントには最大のシステム権限があり、以下で使用します。

- ポリシー サーバへの 管理 UI の登録
- 他の管理者アカウントを含む、その他のタイプのポリシー サーバ オブジェクトの作成
- ポリシー サーバ ツールの使用
- トラストッド ホスト管理者としての機能
- ポリシー管理 API の管理

スーパーユーザ アカウントには以下のクレデンシャルがあります。

ユーザ名

siteminder

パスワード

ポリシー ストアを設定するときに指定したパスワード

注: ポリシー ストアの設定の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。管理 UI の登録の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

管理者のタイプ

以下のタイプの管理者を作成できます。

- 管理者
- レガシー管理者

いずれのタイプの管理者も 管理 UI を使用して **SiteMinder** オブジェクトを管理できますが、実行する機能はそれぞれ異なります。

管理者

管理者は以下を実行することができます。

- 管理 UI の管理
- XPSImport や XPSEExport などのポリシー サーバ ツールの管理

管理者を設定する際、委任権限によって管理者が使用できる 管理 UI タスクをコントロールします。きめ細かい管理モデルを使用することで、作成した管理者に組織内での管理ロールに最適な権限を割り当てることができます。

注: アクセス方法、ポリシー サーバ ツールの使用、セキュリティ カテゴリの詳細については、管理 UI オンライン ヘルプを参照してください。

レガシー管理者

レガシー管理者は以下を実行することができます。

- 管理 UI の管理
- FSS 管理 UI の管理
- smobjimport と smobjexport などのポリシー サーバツールの管理
- トラストド ホスト管理者としての機能トラストド ホスト管理者には、SiteMinder エージェント ホストシステムからホスト登録プロセスを実行する権限があります。登録プロセスにより、エージェントとポリシー サーバが通信できるようになります。
- ポリシー管理 API の管理

注: 環境でポリシー管理 API を使用するスクリプトやプログラムが使われている場合は、レガシー管理者が必要です。ポリシー管理 API から機能を実行できる認証権限を持つレガシー管理者を作成します。

レガシー管理者の権限

レガシー管理者は、管理 UI と FSS 管理 UI で権限を持つことができます。以下の点について考慮してください。

- 管理モデルは FSS 管理 UI と 管理 UI で異なります。
大規模なシステムやドメインレベルのタスクは FSS 管理 UI、さらに詳細なユーザ インターフェース アクセス方法、ポリシー サーバツールの使用、セキュリティカテゴリは、管理 UI に該当します。
- 両方のユーザ インターフェースへの権限の委任は、2 段階のプロセスで行います。

FSS 管理 UI 権限

レガシー管理者が FSS 管理 UI を使用して SiteMinder オブジェクトを管理する場合、管理者を作成してからシステムおよびドメイン レベルの権限を委任します。

以下の表は、システム レベル タスクに関連付けられる権限について説明します。

システム タスク	管理権限
システムとドメイン オブジェクトの管理	■ エージェント、エージェント設定オブジェクト、エージェントグループ、エージェントタイプ、ホスト設定オブジェクト、ユーザ ディレクトリ、ポリシードメイン、認証方式、ディレクトリ マッピング、証明書マッピング、登録方式、SQL クエリ方式の作成、編集、および削除 ■ すべてのドメイン オブジェクトの作成、編集、削除 ■ すべてのドメインでの親レルムの作成と削除 ■ 管理者の作成、編集、および削除 ■ キャッシュされたリソースを含むすべてのキャッシュのクリア ■ グローバル設定の変更 ■ トラストド ホストの削除 注: この権限では、トラストド ホスト オブジェクトを作成、または編集できません。削除のみ可能です。トラストド ホストを登録するには、トラストド ホストの登録権限が必要です。
ユーザの管理	■ すべてのユーザセッションキャッシュのクリア、または全ディレクトリのあらゆる個別ユーザのユーザセッションキャッシュクリア ■ 全ディレクトリのユーザの有効化と無効化 ■ 全ディレクトリのすべてのユーザに対するパスワード変更の強制
キーとパスワードポリシーの管理	■ パスワードポリシーの作成、編集、および削除 ■ キーの管理
トラストドホストの登録	■ トラストドホストの登録

以下の表は、ドメイン レベル タスクに関連付けられる権限について説明します。

ドメイン タスク	管理権限
ドメインオブジェクトの管理	■ 管理対象ドメインでの、ルール、ルールグループ、レスポンス、レスポンスグループ、ポリシーの作成、編集、および削除 ■ 管理対象ドメイン内の最上位レルムの編集(リソース フィルタを除く) ■ 管理対象ドメイン内のネストされたレルムの作成、編集、および削除 ■ リソース キャッシュからの特定のレルムのクリア、およびキャッシュからのすべてのリソース(権限を付与されたドメイン内)のクリア
ユーザの管理	■ 管理対象ドメインに紐付けられたディレクトリ内の個別ユーザのユーザ セッション キャッシュのクリア ■ 管理対象ドメインに紐付けられたディレクトリ内のユーザの有効化と無効化 ■ 管理対象ドメインに紐付けられたディレクトリ内のユーザに対するパスワード変更の強制
パスワード ポリシーの管理	■ 管理対象ドメインに紐付けられたユーザに対するパスワードポリシーの作成、編集、および削除

管理 UI 権限

レガシー管理者が 管理 UI を使用して SiteMinder オブジェクトを管理する場合は、管理者を作成してから詳細なレベルの権限を委任します。

注: アクセス方法、ポリシー サーバ ツールの使用、セキュリティ カテゴリの詳細については、**管理 UI** オンライン ヘルプ システムを参照してください。

管理者ストア オプション

デフォルトでは、管理 UI はポリシー ストアを管理者アイデンティティのソースとして使用します。このデフォルト設定では、管理 UI のインストール直後から環境を管理することができます。ただし、企業ディレクトリなどの外部管理者ユーザーストアを使用して、管理者 ID を格納することもできます。

管理者 ID を格納する場所の判断では、以下を考慮します。

- 作成する管理者のタイプは、管理者 ID を格納する場所によって決まります。
 - ポリシー ストアを使用している場合は、レガシー管理者を作成してポリシー ストアに管理者レコードを作成します。次に管理者アカウントを作成し、作成した管理者レコードを探して 管理 UI 権限を委任します。
 - 外部ストアを使用している場合は、管理者を作成してから外部ストアで管理者レコードを探し、管理 UI 権限を委任します。
- 管理 UI に設定しているポリシー サーバが 1 つの場合は、ポリシー ストアを管理者 ID の格納に使用できます。
- 管理 UI に設定しているポリシー サーバが複数の場合は、外部管理者ストアが必要になります。
- 外部管理者ストアを使用することで、複数の 管理 UI インスタンスが同じ一連の管理者を共有できるようになります。デフォルトで、管理 UI は登録済みポリシー サーバが設定されているポリシー ストアを使用します。

注: 管理 UI のインストール、および追加のポリシー サーバの接続設定の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

レガシー管理者を作成する方法

レガシー管理者を作成するには、以下の手順を完了します。

1. レガシー管理者に関する注意事項を確認します。
2. 管理者レコードを作成します。
3. レガシー管理者が 管理 UI にアクセスする必要がある場合は、管理 UI 権限を委任します。

レガシー管理者に関する注意事項

この処理は、以下の条件の 1 つ以上に当てはまると適用されます。

- レガシー管理者が以下を使用して SiteMinder オブジェクトを管理する予定の場合。
 - FSS 管理 UI
 - ポリシー管理 API
- レガシー管理者がトラステッド ホスト管理者として機能する予定の場合。
- 管理 UI でポリシー ストアを管理者 ID のソースとして使用する場合。管理 UI がポリシー ストアを使用している場合は、管理 UI 権限を委任する前に以下を検討してください。
 - 各管理者がそれぞれのジョブを実行するために必要な権限。これら権限の特定は、管理 UI 権限を委任する際、レガシー管理者に適切なセキュリティ カテゴリを割り当ての役に立ちます。
 - 管理 UI タスクとポリシー サーバ ツールを配布する必要があるレガシー管理者の数。
 - アプリケーション セキュリティ ポリシーを任せる必要があるレガシー管理者が 1 人か、複数か。

重要: 管理 UI が管理者 ID のソースとして外部ストアを使用している場合、レガシー管理者は、SiteMinder オブジェクトの管理に 管理 UI を使用できません。レガシー管理者が 管理 UI を使用する必要がある場合は、このユーザが外部ストアに存在していること、さらに管理者として設定されていることを確認してください。

管理者レコードの作成

管理者レコードを作成し、レガシー管理者の識別情報をポリシー ストアに格納します。

管理者レコードを作成する方法

1. [管理]、[管理者]、[レガシー管理者]、[レガシー管理者の作成]をクリックします。
[レガシー管理者]画面が表示されます。

2. [OK]をクリックします。

レガシー管理者に固有のパラメータが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスで以下を実行します。

a. [名前]フィールドに一意の ID を入力します。

会社の基準に従った一意の ID を入力するようにしてください。会社の基準に従うことで、以下の利点が得られます。

- 管理者が新しいクレデンシャルのセットを覚えておく必要がありません。
- レガシー管理者 ID が外部ストアの一意の ID と一致するため、外部管理者ストアへの移行がスムーズに進みます。

b. [説明]フィールドにユーザのフル ネームを入力します。

この値は、ユーザが 管理 UI にログインするときに表示名として表示されます。表示名により、ログインしているユーザを識別できます。

例: Joe Smith と入力する場合、管理 UI の表示名は以下のように表示されます。

ログイン名: Joe Smith

4. SiteMinder データベース オプション ボタンは選択したままにします。

5. それぞれのフィールドに管理者パスワードを入力します。

6. 以下のいずれかの操作を行います。

■ 以下のみを実行するレガシー管理者を作成している場合は、[サブミット]をクリックします。

- 管理 UI の使用
- ポリシー サーバ ツールの使用

管理者レコードの作成

■ 上記タスク、および以下のタスクのいずれかを実行するレガシー管理者を作成している場合は、次の手順に進みます。

- FSS 管理 UI の使用
- ポリシー管理 API の管理
- トラステッド ホスト管理者としての機能

7. [管理者の権限]グループ ボックスから[システムまたはドメイン]オプション ボタンを選択します。

注: これらの権限は FSS 管理 UI に適用されます。レガシー管理者を作成した後に、権限を 管理 UI に委任します。

システム

管理者はシステム権限を使用し、FSS 管理 UI のすべてのポリシードメインにアクセスします。[システム]を選択すると、[タスク]グループ ボックスが表示されます。

ドメイン

管理者はドメイン権限を使用し、FSS 管理 UI の一部のポリシードメインにアクセスします。[ドメイン]を選択すると、[タスク]と[範囲]グループ ボックスが表示されます。[タスク]グループ ボックスには、実行できる管理タスクがリスト表示されます。[範囲]グループ ボックスには、管理できるドメインがリスト表示されます。

8. 管理者が実行できるタスクを選択します。
 - システム管理者の場合、以下の 1 つ、または複数のタスクを選択します。
 - システムとドメイン オブジェクトの管理
 - ユーザの管理
 - キーとパスワードポリシーの管理
 - トラストドホストの登録
 - ドメイン管理者の場合は、以下を実行します。
 - a. 以下のタスクの 1 つまたは複数を選択する。
 - ドメイン オブジェクトの管理
 - ユーザの管理
 - パスワード ポリシーの管理
 - b. 管理者が管理する必要があるドメインを[範囲]グループ ボックスで選択する。
9. [サブミット]をクリックします。

これで、レガシー管理者が作成されました。

管理 UI 権限の委任

管理 UI 権限を委任し、レガシー管理者が特定の 管理 UI タスクやポリシー サーバツールにアクセスできるようにします。

1. [管理]、[管理者]、[管理者]、[管理者の作成]をクリックします。

[管理者]ダイアログが表示されます。

2. [OK]をクリックします。

管理者に固有のパラメータが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [詳細]グループ ボックスの[検索]をクリックします。

[ユーザの選択]ダイアログ ボックスが表示されます。

4. デフォルトの検索条件のままにして[検索]をクリックします。

ポリシー ストアのレガシー管理者が表示されます。

5. 目的のレガシー管理者を選択し、[選択]をクリックします。

[名前]フィールドにユーザのフル ネームが表示されます。ユーザの一意の ID が[ユーザ パス]フィールドに表示されます。

6. 以下のいずれかの操作を行います。

- すべての権限を委任するには、[スーパーユーザ]を選択して[サブミット]をクリックします。
- きめ細かく権限を委任するには、次の手順に進みます。

7. 管理者に許可するポリシー サーバとの通信方法を、[アクセス方法]グループ ボックスで指定します。管理者がタスクを実行するのに必要な方法をすべて選択します。

例: 管理者が XPSImport と XPSExport ツールを使用する場合は、インポートおよびエクスポートを可能にするオプションを選択する必要があります。

8. [権限]グループ ボックスの[作成]をクリックします。

[権限の作成]ダイアログが表示されます。

9. 管理者に管理させるセキュリティ カテゴリを選択し、[次へ]をクリックします。

注: セキュリティ カテゴリは 1 つ、または複数のタスクで構成されています。これらのタスクは特定の SiteMinder オブジェクトに相当します。詳細については、*管理 UI* オンライン ヘルプ システムを参照してください。一度に選択できるカテゴリは 1 つのみです。ただし、権限の最初のセットを作成した後に別のカテゴリを選択することはできます。

[範囲の選択]ダイアログが表示されます。

10. 以下のいずれかの操作を行います。

- ポリシー管理者、またはアプリケーション管理者を選択した場合は、管理者が管理できるドメインを選択して[次へ]をクリックします。
- 他のセキュリティ カテゴリを選択した場合は、[すべて]を選択して[次へ]をクリックします。

[権限の選択]ダイアログが表示されます。

11. セキュリティ カテゴリのタスクに適用する権限を選択し、[完了]をクリックします。

セキュリティ カテゴリが[権限]グループ ボックスに表示されます。

12. 以下のいずれかの操作を行います。

- a. 別のセキュリティ カテゴリを委任するには、[作成]をクリックします。
- b. [サブミット]をクリックします。

レガシー管理者アカウントに 管理 UI 権限が付与されます。レガシー管理者はそれぞれの 管理 UI およびポリシー サーバ ツールの権限を使って、SiteMinder オブジェクトの管理を開始できます。

外部管理者ストアを設定する方法

外部管理者ストアへの接続を設定するには、以下の手順を完了します。

1. 外部管理者ストアに関する注意事項を確認します。
2. SSL の注意事項を確認します。
3. ストア タイプに合わせて、以下を実行します。
 - (LDAP)ディレクトリ サーバ接続情報を収集します。
 - (RDB)データベース接続情報を収集します。

- (RDB) Java Database Connectivity (JDBC) データソースをアプリケーション サーバに展開します。
 - スタンドアロン オプションを使用して 管理 UI をインストールした場合は、`smjdbcsetup` ユーティリティでデータソースを設定して展開します。
 - 既存のアプリケーション サーバ インフラストラクチャに 管理 UI をインストールした場合は、データソースの設定および展開について各ベンダーのマニュアルを参照してください。
- 4. 外部管理者ストアへの接続を設定します。
- 5. (オプション)レガシー管理者の 管理 UI 権限を移行します。

外部管理者ストアの考慮事項

外部管理者ストア接続を設定する場合は、以下の点を考慮してください。

- **重要:** ポリシー ストアを管理者 ID のソースとして使用することをいったん中止すると、元に戻すことはできません。これは、外部ストアを使用するように設定されている 管理 UI にのみ影響します。外部ストアを使用するように設定されていない他の 管理 UI は、そのままポリシー ストアを使用して管理者を識別します。
- デフォルトの SiteMinder スーパーユーザ アカウントなどのレガシー管理者は、継続して以下を実行できます。
 - FSS 管理 UI での SiteMinder オブジェクトの管理
 - ポリシー管理 API の管理
 - トラストド ホスト管理者としての機能
 - `smobjimport` と `smobjexport` などのポリシー サーバ ツールの使用
- デフォルトの SiteMinder スーパーユーザ アカウントなどのレガシー管理者は、管理 UI で SiteMinder オブジェクトを管理できなくなりました。

- レガシー管理者に引き続き 管理 UI を使用させる必要がある場合は、各ベンダーが提供するツールを使用してこれらのユーザを外部ストアに追加してください。ユーザ ID が外部ストアにいったん作成されれば、ポリシー ストアから外部ストアに既存のユーザ パスをマップして、これらの権限を回復させることができます。

重要: 外部管理者認証の場合、同じレガシー管理者アカウントが 管理 UI、FSS 管理 UI、ポリシー管理 API に対する権限、およびトラステッド ホスト権限を同時に保持することはできません。レガシー管理者がこれらのロールで機能を継続する必要がある場合、レガシー管理者を変更しないでください。ユーザが外部ストアに存在していることを確認し、その外部ユーザの身元を使用して新しい管理者を別に設定します。

- 外部ストアへの接続を設定するときに特定されたスーパーユーザが、デフォルトの SiteMinder スーパーユーザ アカウントに取って代わります。外部ユーザはスーパーユーザになり、管理 UI で最大限の権限を持ち、すべてのポリシー サーバツールにアクセスできます。

外部スーパーユーザを使用して、新しい管理者に権限を委任します。

- 複数の 管理 UI インスタンスが同じ管理者認証ストアを使用する場合は、同じネットワーク識別子を使用して各接続を設定する必要があります。同じ外部管理者認証ストアに対する複数の 管理 UI 接続で異なるネットワーク識別子を使用することはサポートされていません。

例: 最初の接続で 172.16.0.0 を設定した場合は、後続の接続の作成でも 172.16.0.0 を使用します。最初の接続で comp001@example.com を設定した場合は、後続の接続の作成でも comp001@example.com を使用します。

SSL の考慮点

SSL を使った外部管理者ストア接続を設定する場合は、以下について考慮してください。

- ディレクトリ サーバは、SSL で通信するように設定する必要があります。

注: SSL を使用するディレクトリ サーバの設定の詳細については、各ベンダーのマニュアルを参照してください。

- スタンドアロン オプションを使用して 管理 UI をインストールした場合、組み込みの証明書データベースもインストールされています。
- 既存のアプリケーション サーバ インフラストラクチャに 管理 UI をインストールした場合は、アプリケーション サーバによって、必要に応じて証明書データベースが実装されます。

注: 証明書データベースの実装の詳細については、各ベンダーのマニュアルを参照してください。

- ディレクトリ接続情報を入力する際には、必ず SSL 対応ポートを入力してください。SSL 対応ポートを入力しないと、管理認証ウィザードは応答しなくなります。

ディレクトリ サーバ情報の収集

ディレクトリ サーバへの接続を設定している場合は、以下の情報を収集します。

- **ホスト名** -- ディレクトリ サーバ ホスト システムの IP アドレスまたは完全修飾ドメイン名。
- **ポート** - ディレクトリ サーバがリスニングするポート。
- **ディレクトリ サーバ ユーザのクレデンシャル** - ディレクトリ サーバに読み取り/書き込み権限を持つアカウントのユーザ名とパスワード。
- **検索ルート** - ディレクトリ サーバのベース DN。
- **SSL 証明書** - ディレクトリ サーバが SSL 接続を使用して通信するように設定されている場合、SSL 証明書を取得する。

データベース情報の収集

データベースへの接続を設定している場合は、以下の情報を収集します。

- **ホスト名** - データベース ホスト システムの名前。
- **ポート** - データベースがリスニングするポート。
- **(Microsoft SQL Server) データベース名** - データベースの名前。
- **(Oracle) サービス名** - データベースのサービス名。
- **データベース ユーザのクレデンシャル** - データベースに読み取り/書き込み権限を持つユーザ アカウントのクレデンシャル。

重要: Oracle への接続を設定している場合は、このユーザに必ずデフォルトスキーマを設定してください。デフォルトスキーマは、管理ユーザを含むテーブルに関連付けられているスキーマである必要があります。このユーザにデフォルトスキーマを設定しないと、管理認証ウィザードはデータベースでユーザを見つけられません。

JDBC データソースの展開

リレーショナル データベースへの接続を設定する場合、管理 UI では JDBC データソースと管理者ストアが通信する必要があります。データソースを作成するユーティリティが必要です。スタンドアロン オプションを使用して 管理 UI をインストールした場合は、smjdbcsetup ユーティリティを使用することができます。

注: 既存のアプリケーション サーバに 管理 UI をインストールした場合は、JDBC データソースの展開について各ベンダーのマニュアルを参照してください。

smjdbcsetup ユーティリティを使用して、データソースを展開する方法

1. 管理 UI ホスト システムにログインします。
2. (UNIX) SiteMinder の 管理 UI サービスを停止します。

注: サービスの停止の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

3. `administrative_ui_home\CA\SiteMinder\adminui\bin` に移動します。

`administrative_ui_home`

管理 UI インストール パスを指定します。

4. 以下のいずれかのコマンドを実行します。

- (Windows)

`smjdbcsetup.bat`

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

- (UNIX)

`smjdbcsetup.sh`

ユーティリティは一意の識別子を入力するよう要求します。ユーティリティはデータソースにその識別子を追加します。

5. 値を入力し、**Enter** キーを押します。

ユーティリティはデータベース ドライバのタイプを入力するように指示します。ドライバのタイプには、先頭に数値が付いています。

6. 数値を入力してドライバのタイプを選択し、**Enter** キーを押します。

ユーティリティはデータベース ホストシステムの名前を入力するように指示します。

7. データベース ホストの名前を入力し、**Enter** キーを押します。

ユーティリティはデータベースのリスニング ポートを入力するように指示します。

8. データベースのポートを入力し、**Enter** キーを押します。

- Microsoft SQL Server への接続を設定している場合、ユーティリティはデータベース名を入力するようにユーザに指示します。

- Oracle への接続を設定している場合、ユーティリティはサービス名を入力するようにユーザに指示します。

9. データベース名またはサービス名を入力し、**Enter** キーを押します。

ユーティリティはデータベース ユーザ アカウント名を入力するように指示します。

10. データベースユーザ アカウント名を入力し、**Enter** キーを押します。

注: このユーザ アカウントはデータベースに対して読み取り/書き込み権限を持っている必要があります。

ユーティリティはデータベース ユーザのパスワードを入力するように指示します。

11. パスワードを入力し、**Enter** キーを押します。

接続詳細が表示されます。

12. 詳細を確認し、以下のいずれかを実行します。

- データソースを設定して展開するには、「y」を入力して **Enter** キーを押します。

ユーティリティは、

`admin_ui_home\CA\SiteMinder\adminui\server\default\deploy` にデータソースを展開し、SiteMinder の 管理 UI サービスを再起動するように指示します。

`admin_ui_home`

管理 UI インストール パスを指定します。

注: データソースを使用して接続を作成できるようにするには、まず、SiteMinder の 管理 UI サービスを再起動する必要があります。

- 展開をキャンセルするには、「n」を入力して **Enter** キーを押します。

13. 以下のいずれかの操作を行います。

- サービスを自動的に開始するには、以下のいずれかを実行します。
 - (Windows) 「y」を入力し、**Enter** キーを押します。
 - (UNIX) 手動でサービスを開始する必要があります。サービスの開始の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。
- サービスを手動で開始するには、「n」を入力して **Enter** キーを押します。

データソースが設定されて、ユーティリティが終了します。

LDAP 管理者ストア接続の設定

ポリシー ストアから外部ストアに管理者 ID のソースを変更する接続の設定

外部ストア接続を設定する方法

1. [管理]、[管理 UI]、[管理認証の設定]をクリックします。

- 初めて外部管理者ストアを設定する場合は、管理認証の設定ウィザードが表示されます。
- 管理 UI に外部管理者ストアが設定される場合、接続詳細が表示されます。[OK]をクリックし、管理認証の設定ウィザードを開始します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [ディレクトリタイプ]リストからディレクトリ サーバのベンダーを選択し、[次へ]をクリックします。

ウィザードは接続詳細を指定するように指示します。

3. 以下の手順を実行します。

- a. [ホスト]フィールドに、ディレクトリ サーバ ホスト システムの IP アドレスまたは完全修飾ドメイン名を入力します。

重要: 複数の 管理 UI インスタンスが同じ管理者認証ストアを使用する場合は、入力したネットワーク識別子を記録しておきます。同じ外部管理者認証ストアに対する複数の 管理 UI 接続で異なるネットワーク識別子を使用することはサポートされていません。

例: 最初の接続で 172.16.0.0 を設定した場合は、後続の接続の作成でも 172.16.0.0 を使用します。最初の接続で comp001@example.com を設定した場合は、後続の接続の作成でも comp001@example.com を使用します。

- b. [ポート]フィールドに、ディレクトリ サーバのリスニング ポートを入力します。

重要: SSL を使った接続を設定する場合は、必ず SSL 対応ポートを入力してください。SSL 対応ポートを入力しないと、[次へ]をクリックしても管理認証ウィザードは応答しなくなります。

- c. (オプション) [SSL を使用] チェック ボックスを選択し、認証機関 (CA) の証明書をアップロードして 管理 UI と管理者ストアの間の SSL 通信を有効にします。

注: ディレクトリ サーバが、SSL を介して通信するよう設定されている必要があります。SSL を使用するディレクトリ サーバの設定の詳細については、各ベンダーのマニュアルを参照してください。

- d. それぞれのフィールドに、ディレクトリ サーバ ユーザの共通名とパスワードを入力します。

注: このユーザはディレクトリ サーバに対して読み取り/書き込み権限を持っている必要があります。

- e. [次へ] をクリックします。

ウィザードはオブジェクト クラス情報を入力するように指示します。

- 4. 以下の手順を実行します。

- a. [検索ルート] フィールドに、ディレクトリ サーバの検索ルートを入力します。

- b. シャトル コントロールを使用して、SiteMinder 管理者に合わせてオブジェクト クラスを追加および削除します。

- c. [次へ] をクリックします。

ウィザードは、管理者ユーザへのマップに必要な属性を個別に指定するように指示します。ディレクトリ サーバの属性の一覧が自動的に作成されます。これにより、各属性が識別できます。

- 5. 必要な各属性にマップするニーモニック属性文字列を選択し、[次へ] をクリックします。

ウィザードはユーザを検索するように指示します。

- 6. キーワード フィールドにユーザ名のすべて、または一部を入力します。

検索条件と一致するユーザが表示されます。

- 7. ユーザを選択し、[次へ] をクリックします。

注: 選択できるユーザは 1 人だけです。選択したユーザは、接続の設定時にスーパーユーザになります。

サマリ画面が表示されます。

- 8. 接続詳細を確認し、[完了] をクリックします。

外部ストアへの接続が設定されます。

RDB 管理者ストア接続の設定

ポリシー ストアから外部ストアに管理者 ID のソースを変更する接続の設定

外部ストア接続を設定する方法

1. [管理]、[管理 UI]、[管理認証の設定]をクリックします。
 - 初めて外部管理者ストアを設定する場合は、管理認証の設定ウィザードが表示されます。
 - 管理 UI に外部管理者ストアが設定される場合、接続詳細が表示されます。[OK]をクリックし、管理認証の設定ウィザードを開始します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. ディレクトリタイプリストから以下のいずれかを選択します。
 - ユーザ スキーマにユーザのフル ネームを識別する列が含まれている場合は、以下のテンプレートを選択します。
リレーショナル データベース([フルネーム]列あり)
 - ユーザのフル ネームを計算する必要がある場合は、以下のテンプレートを選択します。
リレーショナル データベース([フルネーム]列なし)

3. [次へ] をクリックします。

ウィザードはデータ ソースを選択するように指示します。

注: データ ソースが表示されない場合は、[キャンセル]をクリックし、アプリケーション サーバに JDBC データ ソースを展開します。データ ソースを展開せずに接続を作成することはできません。

4. データ ソースを選択し、[次へ]をクリックします。

ウィザードは、SiteMinder 管理者を含むユーザ テーブルを選択するように指示します。

5. ユーザ テーブルを選択し、[次へ]をクリックします。

ウィザードは、管理者ユーザへのマップに必要な属性を個別に指定するように指示します。データベースの列名の一覧が自動的に作成されます。これにより、各属性が識別できます。

6. 以下のいずれかの操作を行います。
 - リレーショナル データベース テンプレートを選択した場合は、必要なユーザ属性のそれぞれにマップする列名を選択し、[次へ]をクリックします。

ウィザードはユーザを検索するように指示します。
 - リレーショナル データベース (フル ネームなし) テンプレートを選択した場合は、以下のようにします。
 - a. 必要なユーザ属性のそれぞれにマップする列名を選択します。
 - b. ##FIRST_NAME、##LAST_NAME、および ##PRIMARY_KEY を各ユーザ属性にマップする列名で置き換えて、フル ネームの取得クエリを構築します。

注: クエリの最後には疑問符(?)を残します。

例: `select SmUser.FirstName + ' ' + SmUser.LastName from SmUser where SmUser.UserID = ?`
 - c. [次へ]をクリックします。

ウィザードはユーザを検索するように指示します。
7. [ユーザ キーワード]フィールドにユーザ名のすべて、または一部を入力します。

検索条件と一致するユーザが、[検索結果]グループ ボックスに表示されます。
8. ユーザを選択し、[次へ]をクリックします。

注: 選択するユーザは、接続の設定時にスーパーユーザになります。

サマリ画面が表示されます。
9. 接続詳細を確認し、[完了]をクリックします。

外部ストアへの接続が設定されます。

レガシー管理者の権限の移行

外部管理者ストアへの接続を設定した後も、レガシー管理者が 管理 UI またはポリシー サーバツールを継続して使用する必要がある場合は、権限を移行します。

重要： 外部管理者認証の場合、同じレガシー管理者アカウントが 管理 UI、ポリシー サーバツール、FSS 管理 UI、ポリシー管理 API に対する権限、およびトラステッド ホスト権限を同時に保持することはできません。レガシー管理者がこれらの 1 つ以上のロールで機能を継続する必要がある場合、レガシー管理者は変更しないでください。ユーザが外部ストアに存在していることを確認し、その外部ユーザの身元を使用して新しい管理者を別に設定します。

権限を移行する方法

1. 管理者が外部ストアに存在していることを確認します。
2. 外部スーパーユーザを使用して、管理 UI にログインします。
3. [管理]、[管理者]、[管理者]、[管理者の変更]をクリックします。
[検索]ダイアログが表示されます。
4. ユーザのフル ネームを使用して検索条件を指定し、[検索]をクリックします。
検索条件と一致するユーザが表示されます。
5. 目的のレガシー管理者を選択し、[選択]をクリックします。
ユーザ パスはポリシー ストアを指しています。
6. [詳細]グループ ボックスの[検索]をクリックします。
[ユーザの選択]ダイアログ ボックスが表示されます。
7. 検索条件を指定し、[検索]をクリックします。
指定された条件と一致するユーザが表示されます。
8. 目的のユーザを選択し、[選択]をクリックします。
ユーザ パスは外部ストアを指すように更新されます。
9. [サブミット]をクリックします。

管理 UI は外部ストアを使用して管理者を認証します。ポリシー ストアが管理者 ID の格納に使われていた場合、管理者のアクセスレベルは 管理 UI に対するレベルと同じになります。

外部管理者ストア クレデンシャルの更新

外部管理者ストアへの接続で 管理 UI が使用するクレデンシャルを変更する場合は、新しいクレデンシャルを 管理 UI に送信してください。送信しない場合、SiteMinder の管理者認証が失敗します。

スタンドアロン オプションを使って 管理 UI をインストールした場合は、以下の 2 つのユーティリティを使うことができます。

- (LDAP) **smjndissetup** ユーティリティ。ディレクトリ サーバ ユーザ アカウントのクレデンシャルを更新します。

注: ディレクトリ サーバ ホストのシステム名またはポート情報を更新する必要がある場合は、管理 UI を使用して外部管理者ストアへの接続を作成し直します。**smjndissetup** ユーティリティでは、ホストやポート情報を更新できません。

- (RDB) **smjdbcsetup** ユーティリティ。データベース ユーザ アカウントのクレデンシャルを更新します。

注: データベース ホストのシステム名またはポート情報を更新する必要がある場合は、**smjdbcsetup** ユーティリティを使用して JNDI データ ソースを展開し直します。

既存のアプリケーション サーバ インフラストラクチャに 管理 UI をインストールした場合は、以下に注意してください。

- (LDAP) クレデンシャルの更新に関する詳細を SiteMinder サポートに問い合わせます。
- (RDB) データベース固有のツールを使用してデータ ソースを更新します。

ディレクトリ サーバのクレデンシャルの更新

ディレクトリ管理者クレデンシャルを更新するには、`smjndissetup` ユーティリティを使用します。

注: `smjndissetup` ユーティリティは、管理 UI を使用して設定された接続詳細のみを更新できます。`smjndissetup` ユーティリティを使用して接続クレデンシャルを作成することはできません。

ディレクトリ サーバ クレデンシャルを更新する方法

1. 管理 UI ホスト システムにログインします。
2. (UNIX) SiteMinder の 管理 UI サービスを停止します。

注: SiteMinder の 管理 UI サービスの停止の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

3. `administrative_ui_home\CA\SiteMinder\adminui\bin` に移動します。

`administrative_ui_home`

管理 UI インストール パスを指定します。

4. 以下のいずれかのコマンドを実行します。

- (Windows)

```
smjndissetup.bat --reset-password
```

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

- (UNIX)

```
smjndissetup.sh --reset-password
```

ユーティリティはユーザ名を入力するように指示します。

5. 以下のいずれかの操作を行います。
 - 新しいディレクトリ ユーザを入力して **Enter** キーを押します。
 - **Enter** キーを押してデフォルト ユーザ名を受け入れます。
ユーティリティはユーザのパスワードを入力するように指示します。

6. 新しいパスワードを入力して **Enter** キーを押します。
ユーティリティはクレデンシャルを確認し、ディレクトリ接続クレデンシャルを更新するように指示します。
7. 「y」を入力し、**Enter** キーを押します。
 - このユーティリティを **Windows** で実行すると、ユーティリティは SiteMinder の 管理 UI サービスを再起動し、新しいディレクトリ接続詳細を更新します。
 - このユーティリティを **UNIX** で実行すると、ユーティリティは 管理 UI サービスを開始し、新しいディレクトリ接続詳細を更新します。

注: 管理 UI サービスの開始の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

データベース クレデンシャルの更新

JNDI データソースのデータベース ユーザ クレデンシャルを更新するには、smjdbcsetup ユーティリティを使用します。

データベース クレデンシャルを更新する方法

1. 管理 UI ホスト システムにログインします。
2. (UNIX) SiteMinder の 管理 UI サービスを停止します。

注: SiteMinder の 管理 UI サービスの停止の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。
3. `administrative_ui_home\CA\SiteMinder\adminui\bin` に移動します。
`administrative_ui_home`
管理 UI インストール パスを指定します。

4. 以下のいずれかのコマンドを実行します。

- (Windows)

```
smjdbcsetup.bat --reset-password
```

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドラインウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

- (UNIX)

```
smjdbcsetup.sh --reset-password
```

ユーティリティは、一意の識別子を入力するように指示します。

5. 展開されているデータソースの名前を入力します。

注: データソース名がわからない場合、展開されているすべてのデータソースは `administrative_ui_home\SiteMinder\adminui\server\default\deploy` にあります。

`administrative_ui_home`

管理 UI インストールパスを指定します。

ユーティリティはデータベース ユーザ名を入力するように指示します。

6. ユーザ名を入力して Enter キーを押します。

ユーティリティはユーザ パスワードを入力するように指示します。

7. パスワードを入力し、Enter キーを押します。

ユーティリティは、新しいデータソース クレデンシャルを確認し、それらが更新できることを確かめるように指示します。

8. 「y」を入力し、Enter キーを押して新しいデータソース クレデンシャルを確認します。

ユーティリティはデータソースを更新します。

- (Windows) ユーティリティは、SiteMinder の 管理 UI サービスを再起動するようにユーザに指示します。

- (UNIX) SiteMinder の 管理 UI サービスを手動で開始する必要があることを説明するメッセージが表示されます。

9. 以下のいずれかを実行します。

- (Windows) 「y」を入力して Enter キーを押し、ユーティリティで SiteMinder の 管理 UI サービスを再起動し、更新されたデータソースを展開します。
- (Windows) 「n」を入力して Enter キーを押し、手動で SiteMinder の 管理 UI サービスを開始します。サービスが開始されるまで、データソースは展開されません。
- (UNIX) SiteMinder の 管理 UI サービスを手動で開始し、データソースを展開します。

注: SiteMinder の 管理 UI サービスの開始の詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

外部管理者ストア接続の変更

管理者認証のために 管理 UI が接続する外部ストアを変更する場合は、管理認証ウィザードを再度実行します。

管理者を作成する方法

管理者を作成するには、以下の手順を完了します。

1. 管理者に関する注意事項を確認します。
2. 管理者を作成します。

管理者に関する注意事項

管理者を設定する際には、以下の点に注意します。

- このプロセスは、管理 UI で管理者 ID のソースとして外部ストアが使用されている場合に適用される。管理者 ID のソースとしてポリシー ストアを使用している場合は、レガシー管理者を作成する。
- 各管理者がそれぞれのジョブを実行するために必要な権限は何か。これら権限の特定は、管理者に適切なセキュリティカテゴリを委任する際に役立ちます。
- 管理 UI とポリシー サーバ ツール タスクを配布する管理者の数。
- アプリケーション セキュリティ ポリシーを任せる管理者を 1 人とするか、複数とするか。

重要: 管理者は別の管理者を作成できますが、その際に付与できる権限は自分の権限以下になります。たとえば、GUI とレポート権限を持つ管理者が、GUI、レポート権限、ローカル API 権限を持つ別の管理者を作成することはできません。

管理者の作成

管理者を作成する方法

1. 少なくとも新しい管理者が必要とするのと同じ権限を持つ管理者として 管理 UI にログインします。

2. [管理]、[管理者]、[管理者]、[管理者の作成]をクリックします。

[管理者]ダイアログが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [詳細]グループ ボックスの[検索]をクリックします。

[ユーザの選択]ダイアログ ボックスが表示されます。

4. 検索条件を指定し、[検索]をクリックします。

指定された条件と一致するユーザが表示されます。

5. 目的の管理者を選択し、[選択]をクリックします。

[名前]フィールドにユーザのフル ネームが表示されます。外部ストアのユーザへの URL が[ユーザ パス]フィールドに表示されます。

6. 以下のいずれかの操作を行います。
 - すべての権限を委任するには、[スーパーユーザ]を選択して[サブミット]をクリックします。
 - きめ細かく権限を委任するには、次の手順に進みます。
7. 管理者に許可するポリシー サーバとの通信方法を、[アクセス方法]グループ ボックスで指定します。管理者がタスクを実行するのに必要な方法をすべて選択します。

例: 管理者が XPSImport と XPSExport ツールを使用する場合は、インポートおよびエクスポートを可能にするオプションを選択する必要があります。
8. [権限]グループ ボックスの[作成]をクリックします。

[権限の作成]ダイアログが表示されます。
9. 管理者に管理させるセキュリティ カテゴリを選択し、[次へ]をクリックします。

以下の点について考慮してください。

 - セキュリティ カテゴリは 1 つ、または複数のタスクで構成されています。これらのタスクは特定の SiteMinder オブジェクトに相当します。詳細については、*管理 UI オンライン ヘルプ* システムを参照してください。
 - 一度に選択できるカテゴリは 1 つのみです。ただし、権限の最初のセットを作成した後に別のカテゴリを選択することはできます。

[範囲の選択]ダイアログが表示されます。
10. 以下のいずれかの操作を行います。
 - a. ポリシー管理者、またはアプリケーション管理者を選択した場合は、管理者が管理できるドメインを選択して[次へ]をクリックします。
 - b. 他のセキュリティ カテゴリを選択した場合は、[すべて]を選択して[次へ]をクリックします。

[権限の選択]ダイアログが表示されます。
11. セキュリティ カテゴリのタスクに適用する権限を選択し、[完了]をクリックします。

セキュリティ カテゴリが[権限]グループ ボックスに表示されます。
12. 以下のいずれかの操作を行います。
 - a. 別のセキュリティ カテゴリを委任するには、[作成]をクリックします。
 - b. [サブミット]をクリックします。

管理者が作成されます。

管理者ユース ケース

この管理者ユース ケースは以下について示します。

- 管理者を作成する方法
- 既存の管理者が新しい管理者を作成して権限を与える方法

このシナリオでは、3 人の管理者を使います。

- スーパーユーザ
- 上級管理者
- 下級管理者

これら 3 人の管理者を使用して、以下のシナリオを説明します。

- スーパーユーザが上級管理者を作成する
- 上級管理者が下級管理者を作成する

このユース ケースで使われる用語の説明は以下のとおりです。

アクセス方法

管理者がポリシー サーバと通信する方法を指定します。

セキュリティ カテゴリ

管理者が、ポリシー サーバのオブジェクト、またはツールを管理するタスクを実行できる機能領域を指定します。

スコープ

管理者の権限がすべてのドメインやアプリケーションに有効なのか、特定のドメインやアプリケーションにのみ有効なのかを示します。

アクセス権

管理者がセキュリティ カテゴリに関連するタスクの読み取り、管理、伝達、実行が可能かどうか判断します。

権利

セキュリティ カテゴリ、スコープおよびアクセス権のグループに基づいて、管理者の完全権限セットを定義します。

スーパーユーザによる上級管理者の作成

スーパーユーザは、外部管理者ユーザストアへの接続が設定されたときにシステム権限が委任された管理者です。スーパーユーザは、すべてのカテゴリ、権限、スコープを他の管理者に割り当てることができます。

管理 UI から、スーパーユーザは上級管理者という名前の管理者を作成します。上級管理者は、スーパーユーザが割り当てるまで最初は何も権限を持っていません。

スーパーユーザは、上級管理者に以下を割り当てます。

アクセス方法

GUI を許可

権利

セキュリティ カテゴリ	スコープ	権限 *
保守管理	すべて	V, M
エージェント管理	すべて	V, M
アプリケーション管理	すべて	V, M, P
ポリシー管理	ドメイン 1	V, M, P

* 権限: 表示、管理、伝達、eXecute (レポート実行の場合のみ)

重要: 伝達権限では、ある管理者が別の管理者にカテゴリを割り当てることができます。

この段階で、スーパーユーザは既存のセキュリティ カテゴリの権限を変更できます。

上級管理者の追加権限

スーパーユーザは、上級管理者に権限を追加して割り当てることができます。スーパーユーザが追加する際に選択できるセキュリティ カテゴリのリストは、上級管理者にすでに割り当てられているカテゴリによって若干異なります。上級管理者に割り当て済みのエージェントと管理者管理カテゴリを除くすべてのカテゴリが表示されます。また、これらにはスコープを割り当てることができないため、何も変更できません。管理者管理カテゴリに割り当てられているスコープは ALL であり何も変更できないため、このカテゴリは表示されません。

元のカテゴリ セットで利用できるカテゴリは、ポリシー管理のみです。このカテゴリはスコープを割り当てることができる、つまり、特定のドメインやアプリケーションに限って権限を適用できるため使用できます。スーパーユーザがポリシー管理カテゴリを選択すると、スコープのダイアログには、**ALL** という選択肢を含むリストと、**Domain1** を除くドメインの全リスが表示されます。上級管理者はすでに **Domain1** が割り当てられています。

注: アプリケーション管理者は、ポリシー管理者と同じようにスコープを設定できるカテゴリです。ただし、このカテゴリのスコープは **ALL** に定義されているため、このカテゴリを選択肢として再度表示する必要はありません。

スーパーユーザは **Domain2** を選択し、上級管理者の権限を第 2 ドメインに拡張します。

上級管理者の権限は以下のとおりです。

セキュリティ カテゴリ	スコープ	権限 *
保守管理	すべて	V, M
エージェント管理	すべて	V, M
アプリケーション管理	すべて	V, M, P
ポリシー管理	Domain1	V, M, P
ポリシー管理	Domain2	V, M, P

* 権限: 表示、管理、伝達、eXecute (レポート実行の場合のみ)

上級管理者による下級管理者の作成

このユース ケースの後半では、特定の権限セットを持った管理者が別の管理者に権限を割り当てる方法について示します。

上級管理者が、下級管理者という名前の管理者を作成します。上級管理者が下級管理者のセキュリティ カテゴリを選択する際の選択肢は、以下の 2 つのみです。

- アプリケーション管理
- ポリシー管理

上級管理者には、これら 2 つのカテゴリに関する P (伝達) 権限しかないため、下級管理者にはこれら 2 つのカテゴリしか割り当てることができません。

重要: 伝達権限では、ある管理者が別の管理者にカテゴリを割り当てることができます。

上級管理者は下級管理者に、アクセス方法と権限を以下のように割り当てることができます。

アクセス方法

GUI を許可

権利

- 上級管理者はアプリケーション管理カテゴリを選択します。上級管理者のアプリケーション管理スコープに対する設定は ALL であるため、上級管理者が下級管理者に割り当てを行うためにこのカテゴリを選択すると、システムで使用可能なすべてのアプリケーションのリストが表示されます。ここで、上級管理者は下級管理者のスコープに ALL を割り当てますが、権限はアプリケーションを表示できるのみにします。
- 上級管理者はポリシー管理カテゴリを選択します。上級管理者のスコープ ダイアログに表示される選択肢は、Domain1 と Domain2 のみです。上級管理者は下級管理者に Domain1 を選択し、表示権限と管理権限のみを与えます。

注: カテゴリの選択時、上級管理者の選択肢はポリシー管理カテゴリのみです。これは、もう 1 つの選択肢だったアプリケーション管理カテゴリを上級管理者がすでに割り当てていたためです。

上級管理者が後から Domain2 を割り当てる場合、Domain1 がすでに割り当てられているため、Domain2 はスコープリストで選択可能なただ一つのドメインです。

下級管理者の最終的な権限は以下のとおりです。

セキュリティ カテゴリ	スコープ	権限 *
アプリケーション管理	すべて	V
ポリシー管理	Domain1	V, M
ポリシー管理	Domain2	V, M

* 権限: 表示、管理、伝達、eXecute (レポート実行の場合のみ)

管理者の無効化

アカウントを削除せずに、管理者を一時的に無効にできます。アカウントを無効にするだけであれば、権限を回復する際にアカウントを再度作成する必要がありません。

管理者を無効にする方法

1. [管理]、[管理者]、[管理者]、[管理者の変更]をクリックします。
[検索]ダイアログが表示されます。
2. 検索条件を指定し、[検索]をクリックします。
検索条件と一致するユーザが表示されます。
3. 無効にする管理者を選択し、[選択]をクリックします。
管理者の詳細が表示されます。
4. [詳細]グループ ボックスで[無効]を選択します。
5. [サブミット]をクリックします。

管理者は無効になります。

この手順を繰り返し、[無効]チェック ボックスの選択を解除して変更をサブミットすれば、いつでも管理者を有効にできます。

レガシー管理者の無効化

アカウントを削除せずに、一時的にレガシー管理者を無効にできます。アカウントを無効にするだけであれば、権限を回復する際にアカウントを再度作成する必要がありません。

レガシー管理者を無効にする方法

1. [管理]、[管理者]、[レガシー管理者]、[レガシー管理者の変更]をクリックします。
[検索]ダイアログが表示されます。
2. 検索条件を指定し、[検索]をクリックします。
検索条件と一致するユーザが表示されます。

3. 無効にするユーザを選択し、[選択]をクリックします。

システムまたはドメイン タスクが[管理者の権限]グループ ボックスに表示されます。

4. [タスク]グループ ボックスでタスクをクリアし、[サブミット]をクリックします。

これでこのレガシー管理者は、FSS 管理 UI やトラステッド ホスト管理者などの機能にアクセスできなくなりました。

5. (オプション)レガシー管理者が 管理 UI で権限を持っている場合は、以下を実行します。

- a. [管理]、[管理者]、[管理者の変更]をクリックします。

[検索]ダイアログが表示されます。

- b. 検索条件を入力し、[検索]をクリックします。

検索条件と一致するユーザが表示されます。

- c. FSS 管理 UI 権限を削除したレガシー管理者を選択し、[選択]をクリックします。

管理者の詳細が表示されます。

- d. [詳細]グループ ボックスで[無効]を選択し、[サブミット]をクリックします。

レガシー管理者は無効になります。

無効にしたレガシー管理者は、そのレガシー管理者アカウントをシステムまたはドメイン固有のタスクを含めるように変更すれば、また、該当する場合はそれぞれの管理者レコードの[無効]チェック ボックスの選択を解除すれば有効にできます。

管理者アクセスの復元

管理者アカウントが誤って変更、または削除された場合、ポリシー サーバへのフル アクセス権限を持つ管理者は権限を復元できます。これらの権限を持つ管理者は、以下の 2 タイプです。

- デフォルトの SiteMinder スーパーユーザ アカウント(siteminder)。ポリシー ストアを使用して管理者 ID を格納している場合、権限の復元には siteminder アカウントを使用します。
- 外部管理者ストア接続の作成時に設定したスーパーユーザ アカウント。外部ストアを使用して管理者 ID を格納している場合、権限の復元には外部スーパーユーザを使用します。

デフォルトの管理者アカウントにアクセスできない場合は、以下を実行します。

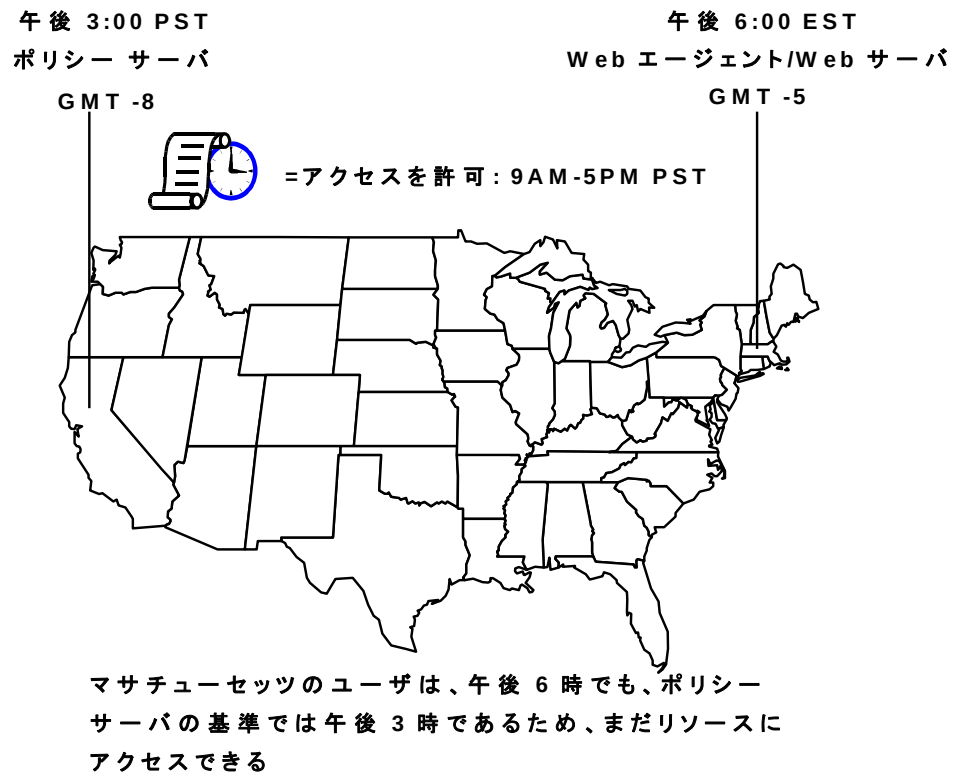
- smreg ユーティリティを使用して、デフォルトの SiteMinder スーパーユーザ アカウントのパスワードを変更します。
- XPSSecurity ユーティリティを使用して、外部管理者ストアに存在しているユーザをスーパーユーザにします。

注: これらユーティリティの使い方の詳細については、「ポリシー サーバ管理ガイド」を参照してください。

Web エージェントとポリシー サーバの時間の計算方法

ポリシー サーバまたは Web エージェントがインストールされているシステムは、それぞれのシステム クロックをシステムの地理的な場所に対応したタイムゾーンに設定する必要があります。ポリシー サーバと Web エージェントは、このタイムゾーンを使用して、GMT (グリニッジ標準時) を基準にした時間を計算します。

次の図は、ポリシー サーバでポリシーを実行するときの時間の関係を示しています。リソースは、マサチューセッツにある Web サーバに格納され、カリフォルニアにあるポリシー サーバによって保護されています。ポリシーによって、午前 9 時から午後 5 時までの間、リソースへのアクセスが許可されています。この場合、マサチューセッツのユーザは、午後 6 時でもリソースにアクセスできます。その理由は、ポリシーはポリシー サーバに設定された PST(太平洋標準時)のタイムゾーンを基準としているため、Web エージェントに設定されている EST(東部標準時)より 3 時間遅れているからです。



注: Windows システムの場合、コントロール パネルの[日付と時刻]で設定する[タイムゾーン]と[日付と時刻]は一致している必要があります。たとえば、米国にあるシステムを東部標準時から太平洋標準時にリセットする場合は、システム クロックを 3 時間戻して、時間を太平洋標準時に変更します。この 2 つの設定が一致していないと、複数のドメインのシングル サインオンやエージェントキー管理機能が正しく動作しません。

第 5 章: ユーザ セッション

このセクションには、以下のトピックが含まれています。

[ユーザ セッションの概要](#) (P. 105)

[ユーザ セッションを開始する方法](#) (P. 109)

[複数のレルムにまたがるセッションを維持する方法](#) (P. 110)

[複数の cookie ドメインにまたがるセッションを維持する方法](#) (P. 111)

[ユーザ セッションを検証する方法](#) (P. 111)

[セッション情報を委任する方法](#) (P. 112)

[セッション タイムアウト](#) (P. 113)

[エージェント キー管理とセッション タイムアウトを整合させる方法](#) (P. 113)

[ユーザ セッションを終了する方法](#) (P. 114)

[Windows ユーザ セキュリティ コンテキスト](#) (P. 115)

ユーザ セッションの概要

SiteMinder は、多層アプリケーション間で永続的なユーザ セッションを保持し、管理します。世界中のどこからでもアプリケーションの利用やインターネットを介した取引が可能になり、環境やアプリケーション技術、セキュリティの条件が多様化している今日、ユーザ セッション情報を保持することの重要性はますます高まっています。

非永続 Cookie と永続 Cookie

標準の SiteMinder セッション cookie は非永続的です。非永続 cookie は、Web ブラウザのメモリでのみ保持されるものです。ユーザがブラウザを閉じると、セッション cookie は破棄され、ユーザは有効にログアウトされます。

Web ブラウザ メモリに cookie を保持するだけでなく、ハード ディスクに cookie が書き込まれるよう SiteMinder を設定できます。Web ブラウザ内およびハード ディスク上で保持される SiteMinder cookie は永続 cookie と言います。永続 cookie を使用する場合、ユーザがブラウザを閉じて再度開いてもログインしたままになります。

注: 永続 cookie の設定の詳細については、「Web エージェント設定ガイド」を参照してください。

非永続セッションと永続セッション

SiteMinder では、永続セッションを設定することで、Windows のセキュリティコンテキスト機能や Federated Web サービスが提供されます。永続セッションでは、SiteMinder cookie が、SiteMinder セッション ストア、Web ブラウザのメモリ、および任意でハード ディスクに保持されます。

永続セッションを実装する前に、以下の点を考慮してください。

- 永続セッションはレルム単位で設定します。
- 永続セッションは必要な場合に限って使用する必要があります。セッション サービスを使用してセッションを保持すると、システムのパフォーマンスに影響を及ぼします。

注: セッション サービスの有効化および設定の詳細については、「ポリシー サーバ管理ガイド」を参照してください。

セッションチケット

SiteMinder では、セッション チケットを使用してセッション管理を実行します。セッション チケットには、ユーザに関する基本情報とそのユーザの認証情報が含まれています。シングル サインオン SiteMinder 環境内のすべてのサイトで、このセッション チケットを使用してユーザのセッションが識別されます。セッション チケットは暗号化され、ポリシー サーバのみが解読と確認を行うことができます。SiteMinder Web エージェントは、セッション チケットを使用してユーザを識別し、ポリシー サーバにセッション情報を提供します。

セッションチケットの扱いは、セッションが永続か非永続かによって異なります。

注: 非永続および永続 Cookie は、ユーザの SiteMinder セッションが永続的な非永続的なとは関係がありません。

非永続セッション

Web エージェントは、cookie にセッション チケットを格納します。cookie には、ユーザ セッション データが含まれます。ユーザ固有のデータは cookie 自体には保持されません。Web エージェントは、cookie を確認し、セッション タイムアウトを実行します。

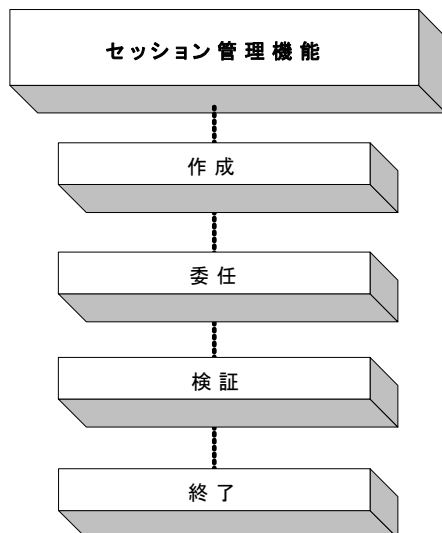
永続セッション

Web エージェントはセッション サーバ データベースにセッション チケットを格納します。可能な場合は、クライアント上のオプション **cookie** にもセッション チケットを格納します。

セッション チケット データは、Web エージェント キャッシュへのインデックスとして使用されます。このキャッシュにはユーザ セッション データが含まれます。**cookie** が書き込まれた場合、ユーザ固有のデータは **cookie** 自体には保持されません。Web エージェントは、セッションの正当性を確認し、セッションタイムアウトを実行します。

SiteMinder によるユーザセッションの管理

通常は、以下のように、SiteMinder ではユーザ セッションを自動的に管理し、ユーザ セッションのライフ サイクルにおいて数多くのセッション管理機能を実行します。



セッション作成

ユーザがアプリケーションへのログインに成功すると、セッションが確立されます。ユーザ認証が失敗すると、セッションは確立されません。

セッション委任

アプリケーション環境全体でセッション情報を渡します。1 つのアプリケーションのロジックが複数のアプリケーション階層にまたがる場合は、セッション情報の委任が必要です。

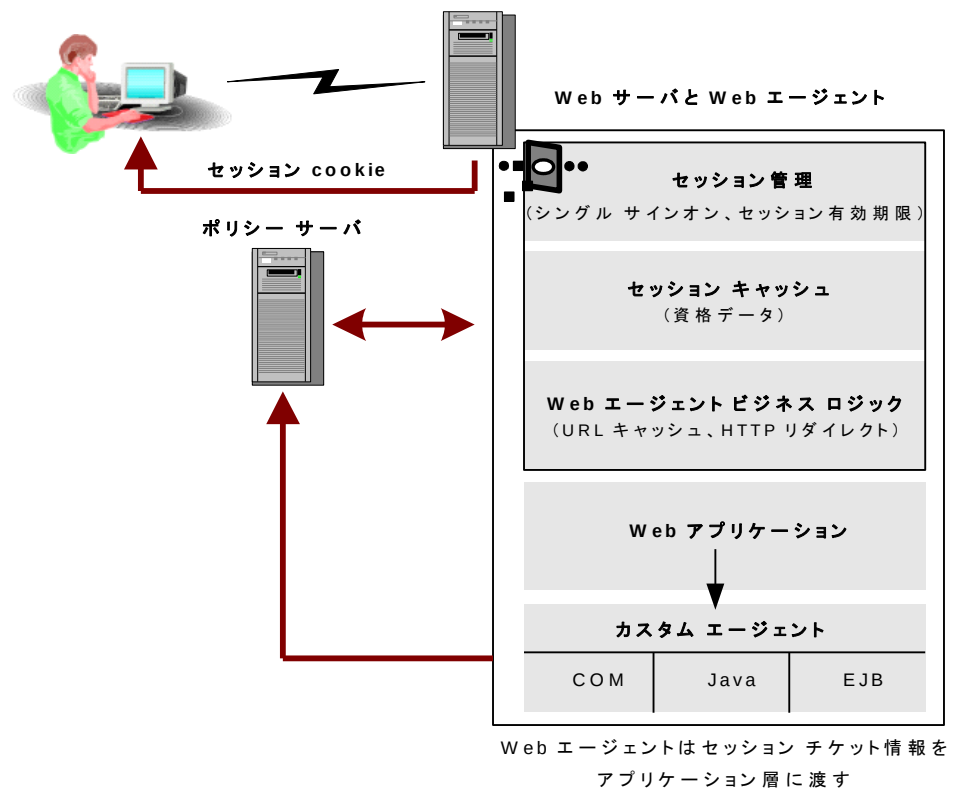
セッション検証

ユーザ セッションがまだアクティブであること、つまりセッションが期限切れであったり、終了したりしていないことをセッション チケットで確認します。

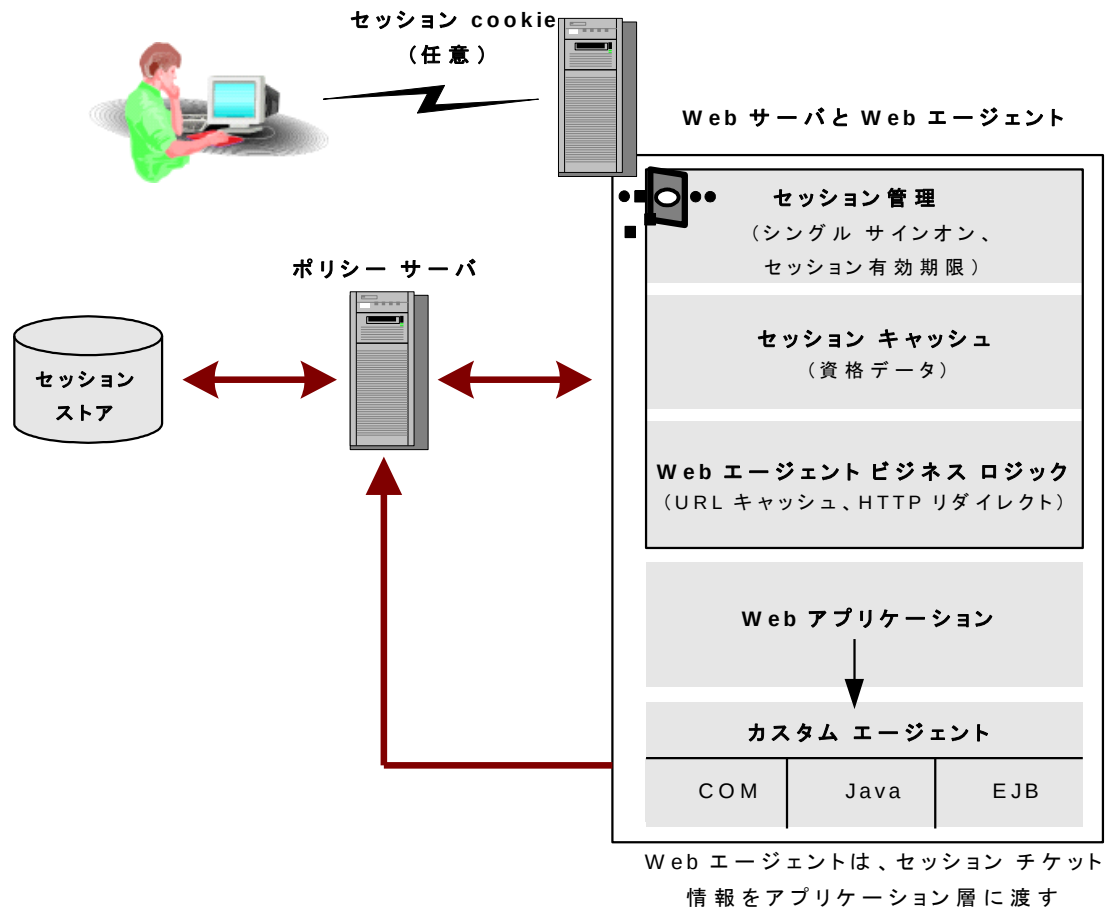
セッション終了

ユーザがログアウトしたか、設定済みのセッション タイムアウトを過ぎたか、または SiteMinder システム マネージャによってユーザが手動で無効にされると、ユーザ セッションは終了します。ログアウトした場合、またはユーザ セッションが失効した場合には、ユーザは再度ログインして新しいセッションを作成する必要があります。手動で無効化された場合、そのユーザはセッションを再開できません。

以下の図は、SiteMinder が非永続的セッションを管理する仕組みを示します。



以下の図は、SiteMinder が永続的セッションを管理する仕組みを示します。



ユーザ セッションを開始する方法

ユーザ セッションは、ユーザがログインして SiteMinder によって認証されると開始されます。ポリシー サーバがセッション チケットを発行します。このチケットは、そのユーザ セッションでの後続のすべてのエージェントリクエストに必要です。ユーザが、より高い保護レベルのために、セッション中に再認証を強制された場合は、既存のセッションが維持されます。

セッションは、ユーザがログオフするか、セッションが失効するまで、または管理者がユーザを無効にしてセッションが終了されるまで、アクティブな状態を保持します。

複数のレルムにまたがるセッションを維持する方法

ユーザがリソースへのアクセスを要求すると、そのリソースが含まれるレルムのコンテキスト内にユーザのセッションが作成されます。レルムには認証方式も関連付けられています。この認証方式によって、ユーザがリソースにアクセスするために必要な証明のタイプが決まります。

この認証コンテキストは、使用する認証方式、ユーザの認証に使用されるネームスペース、その他の関連情報など、コンポーネントを定義する **SiteMinder** のデフォルトの **HTTP** ヘッダを介して、**SiteMinder** インストール環境内のすべての **Web** サーバで使用可能になります。デフォルトのヘッダに加え、ポリシーにレスポンス属性を設定することで、誕生日や電話番号など、ユーザをさらに詳細に識別するための情報を伝達することができます。

シングルサインオンに対応するように **SiteMinder** インストール環境が設定されている場合は、管理者によって、認証方式に保護レベルが割り当てられていることがあります。保護レベルは **1** ～ **1000** の数値で設定されます。**1** は安全性が最も低いレベルで、**1000** は最も高いレベルです。保護レベルを設定することで、管理者はシングルサインオン環境のセキュリティを強化し、柔軟性を高めることができます。

1 つのレルムで認証されたユーザは、別のレルムでも認証を受けることなく、同じセッションを継続して使用できます。ただし、別のレルムの認証方式に設定されている保護レベルが元のレルムの保護レベルよりも低いか、同等であることが条件になります。元のレルムの保護レベルよりも低いか、同等であれば、別のレルムで再度認証を受ける必要はありません。つまり、ユーザセッションは引き続き有効になります。より高い保護レベルが設定された認証方式で保護されているリソースにアクセスしようとする、**SiteMinder** はユーザにクレデンシャルを再入力することを求めるプロンプトを表示します。この場合、既存のユーザセッションは終了し、新しいセッションが作成されます。

シングルサインオン環境の保護レベルを設定するには、「[認証方式](#) (P. 309)」を参照してください。

複数の cookie ドメインにまたがるセッションを維持する方法

SiteMinder は、各種の Web サーバ プラットフォームで構成される異機種環境で複数の cookie ドメインにまたがるシングル サインオンをサポートします。ユーザが `companyA.com` にアクセスした後で `companyB.com` にアクセスした場合、そのユーザのセッション情報は保持されます。クッキードメイン境界を越えてセッション情報を保持するには、シングルサインオンに対応するように Web エージェントを設定する必要があります。シングル サインオンを設定すると、シングルサインオン環境内のすべてのエージェントとサーバでセッション情報が含まれる cookie を使用できるようになります。

cookie ドメインの境界を越えてセッション情報と識別情報を引き渡すことが可能な場合、あるサイトの 1 つの cookie ドメインでいったん認証を受けたユーザは、再度情報の入力を求められることなく、そのサイト内の別の cookie ドメインにアクセスできるようになります。

シングルサインオンを実現するには、cookie プロバイダを使用します。cookie プロバイダは、シングルサインオン環境に対応する Web エージェントの拡張機能です。

cookie ドメインが異なる複数のリソースについてドメイン境界を越えたログアウトを可能にするには、個々の cookie ドメインのレールで永続セッションを有効にします。ユーザが、あるドメインでログアウトすると、ポリシー サーバはログアウトイベントを送信してユーザ セッションを終了します。

注: 複数の cookie ドメインにまたがるシングル サインオンでは、シングル サインオン環境全体で同じユーザ ディレクトリを使用する必要はありません。ただし、管理 UI で設定するユーザ ディレクトリ接続は、各 cookie ドメインで同じディレクトリオブジェクト名を共有する必要があります。

ユーザ セッションを検証する方法

ユーザがリソースへのアクセスを要求すると、Web エージェントはセッションがまだアクティブ状態であるかどうかを確認します。Web エージェントはまず、セッションキャッシュのセッション情報をチェックします。Web エージェントが cookie を読み取り、キャッシュに情報が格納されている場合は、セッションを検証することができます。キャッシュに情報がない場合、Web エージェントは、ポリシー サーバに問い合わせるユーザの識別情報を確認し、その他のセッション情報や許可情報を収集します。

セッションの確立時に使用された保護レベルよりも高い保護レベルが設定されたリソースにアクセスする場合には再度認証を受ける必要がありますが、セッション情報は保持されます。

セッション情報を委任する方法

ユーザ セッションは、セッション チケットを使用して、**SiteMinder** インストール環境内のアプリケーション階層の間で委任できます。セッション チケットは、ユーザの識別情報をアプリケーション階層間でやり取りする場合に使用されるメカニズムです。セッション チケットを取得すると、各アプリケーションはポリシー サーバに対して許可呼び出しを行うことができます。

SiteMinder インストール環境でカスタム エージェントを使用している場合、カスタム エージェントには、セッション情報を保持するために、セッション チケット内の情報へのアクセス権が必要になります。

セッションの委任にセッションチケットを使用するだけでなく、**Web** エージェントは、一連のデフォルト HTTP ヘッダを生成します。このヘッダは **Enterprise Java Beans (EJB)** や **Component Object Model (COM)** をベースとする階層など、異なるビジネスアプリケーション階層の境界を越えて渡すことができます。このヘッダには、一意なセッション ID に加え、オプションでユニバーサル ID を含めることができます。セッション ID はアクティブなユーザセッションを識別します。

ユニバーサル ID により、**SiteMinder** 環境内のアプリケーションに対してユーザを識別します。通常、この ID は、ユーザのログイン ID とは異なり、電話番号や顧客アカウント番号など、その他のタイプの一意の ID になります。ユニバーサル ID を使用すれば、古いアプリケーションと新しいアプリケーションの識別が容易になります。アプリケーションにかかわらず、ユーザの ID は自動的に送信されます。また、この ID はアプリケーションに組み込まれるため、常に変化するユーザディレクトリとは別個にアプリケーションごとにユーザを識別することができます。

セッション ID とユニバーサル ID は両方とも、ユーザ セッションの整合性を確保するため、**SiteMinder** 環境内のすべてのアプリケーション間で共有されます。

セッション タイムアウト

各ユーザ セッションには、セッション タイムアウト情報が含まれています。アクティブセッションの長さとしてセッションが有効な状態を保ったまま非アクティブでいられる許容時間は、設定されたタイムアウト値によって決まります。セッションタイムアウトは次のようなタイムアウトオプションを使用してレルム単位で設定します。

名前	目的
最大タイムアウト (すべてのセッション)	Web エージェントがユーザに再認証を要求するまで、ユーザセッションをアクティブにしておく最大時間を指定します。 WebAgent-OnAuthAccept-Session-MaxTimeout レスポンス属性を指定して、この設定よりも優先させることができます。
アイドルタイムアウト (すべてのセッション)	ユーザセッションが非アクティブになってから Web エージェントがそのセッションを終了するまでの時間を示します。セッションが失効すると、ユーザは再認証を受ける必要があります。 注: 永続的セッションの場合、この値は、セッション検証期間で指定される値よりも大きい必要があります。 WebAgent-OnAuthAccept Session-Idle-Timeout レスポンス属性を指定して、この設定よりも優先させることができます。
セッション検証期間 (永続的セッション)	永続的セッションの場合のみ、エージェントがポリシー サーバを呼び出してセッションを検証する最大間隔を指定します。セッション確認呼び出しでは、ユーザがまだアクティブかどうかをポリシー サーバに知らせ、さらに、ユーザのセッションがまだ有効かどうかを確認するという 2 つの機能を実行します。

エージェント キー管理とセッション タイムアウトを整合させる方法

SiteMinder Web エージェントは、SiteMinder 環境内の Web エージェント間で受け渡す cookie の暗号化と復号化にキーを使用します。すべてのキーは、ポリシー サーバと通信するすべての Web エージェントで同じ値に設定する必要があります。

SiteMinder のインストール環境は、定期的に変更されるダイナミック エージェント キーを使用するように設定できます。ダイナミック キーのロールオーバーでは、定期的にダイナミック キーを更新することで、暗号化された **cookie** のセキュリティを確保します。管理 UI の[ロールオーバー間隔の設定]ダイアログ ボックスで、キーのロールオーバーをいつ実行するかを指定します。SiteMinder のインストール環境全体で行われるキー更新は、最大で 3 分かかることがあります。

エージェント キーの更新とセッション タイムアウトを整合させないと、セッション情報が含まれている **cookie** が無効になる場合があります。企業のポリシー設計の担当者とダイナミックキーロールオーバーの設定の担当者が異なる場合があるので、この調整は重要です。

セッション タイムアウトは、エージェント キーのロールオーバー間隔の 2 倍以下にする必要があります。管理者が、セッションが期限切れになる前にエージェント キーのロールオーバーを 2 回実行するように設定した場合、1 回目のキーのロールオーバーの前に Web エージェントによって作成された **cookies** は、無効になります。セッション タイムアウトが、指定されたロールオーバー間隔よりも大きい場合は、ユーザは、セッションが終了する前に識別情報を再要求されることがあります。

たとえば、3 時間ごとにキーのロールオーバーを実行するように設定した場合は、複数のキー ロールオーバーによってセッション **cookie** が無効にならないように、最大セッション タイムアウトを 6 時間に設定する必要があります。

ユーザ セッションを終了する方法

ユーザセッションは次の 3 つのうちいずれかの方法で終了します。

1. ユーザがログアウトする。

ユーザは、ログアウトすることで、そのセッションを終了できます。

2. セッション タイムアウトが期限切れになる。

セッション タイムアウトまたはポリシー ベースのレスポンス属性の設定のために、ユーザのセッションが期限切れになることがあります。この場合、新しいセッションを開始するには、ユーザはクレデンシャルを再入力する必要があります。

3. ユーザが無効になる。

システム マネージャは、ユーザ アカウントを無効にして、そのセッション アカウントをフラッシュし、ユーザの再認証を防ぐことができます。詳細については、「ポリシー サーバ管理ガイド」を参照してください。

Windows ユーザ セキュリティ コンテキスト

Windows ネットワークでは、セキュリティコンテキストによって、ユーザの識別情報と認証情報が定義されます。Microsoft Exchange Server や SQL Server などの Web アプリケーションで Microsoft のアクセス制御リスト (ACL) またはその他のアクセス制御ツールの形式の固有のセキュリティ機能を提供するには、ユーザのセキュリティコンテキストが必要です。

注: Windows セキュリティコンテキストでは、ユーザストアは Active Directory (AD) である必要があります。

SiteMinder Web エージェントは、IIS Web サーバ (Windows 2000 プラットフォーム) 上の Web リソースにアクセスするための Windows ユーザ セキュリティコンテキストを提供できます。ユーザのセキュリティコンテキストを確立することで、サーバはこの識別情報を使用してアクセス制御メカニズムを適用することができます。

SiteMinder は、Windows ユーザ セキュリティコンテキストを提供することで、以下のようなメリットをもたらします。

- 二重のセキュリティ

SiteMinder のすべての機能を使用して Web リソースを保護できるだけでなく、Microsoft のセキュリティツールとも連携してアプリケーションを保護することができます。

- Web エージェントはさまざまな Web ブラウザとの通信が可能な上に、Windows セキュリティコンテキストも提供できます。

SiteMinder は、Internet Explorer だけでなく、Netscape Web ブラウザとも連携できます。

- エージェントは SSL 接続経由でユーザの証明を収集し、以降のリクエストは非 SSL 接続経由で受け付けることができます。SiteMinder は、Web リソースへのアクセスを許可するときに、クレデンシャル収集のこのメカニズムと Windows セキュリティコンテキストを組み合わせで使用します。

- SiteMinder では、フォームベースの認証を使用してシングル サインオンを実装し、ユーザの識別情報をバックオフィス アプリケーションに渡すことができます。

Microsoft は汎用のフォーム証明収集機能を提供していません。

ユーザ セキュリティコンテキストの永続セッションのメンテナンス方法

Web エージェントが特定のリソースにセキュリティコンテキストを提供できるようにするには、これらのリソースが含まれるすべてのレルムで永続セッションを有効にする必要があります。永続セッションは、セッション サーバによって管理されます。

注: 永続セッションの概念については、「[永続セッションと非永続セッション \(P. 105\)](#)」を参照してください。Web エージェントによって保護されるレルムで永続セッションを有効にする方法については、「[レルムの設定 \(P. 487\)](#)」を参照してください。

セッション サーバはポリシー サーバと同じシステムに配置されており、ユーザの暗号化されたクレデンシャルを格納し、ユーザをセッション ID に関連付けます。SiteMinder セッションがクライアントと Web エージェントの間で確立されると、Windows ユーザ アカウントが確立されてセッションにリンクされます。Web エージェントのユーザ セッション キャッシュがいっぱいになると、自動的にエントリがページされます。これにより、エージェントはセッション サーバからユーザの認証情報を取得して、セッションを再確立できます。シングル サインオン環境でセキュリティコンテキストを伝播する必要があるため、セッション サーバにはセキュリティコンテキストも格納されます。

セッションを再検証する方法

セッション サーバにセッションの再確認を求めるように、Web エージェントを明示的に設定することができます。この設定は、ポリシー サーバを通じて行います。ユーザのブラウザに格納されるセッション cookie にはセッション ID があります。セッション cookie はこのセッション ID を使用して、セッション サーバからユーザのクレデンシャルを再取得します。

注: セッション cookie が無効になると、その ID に関連付けられたクレデンシャルも無効になるため、ユーザは再認証を行う必要があります。

エージェントがセッションを再検証する間隔は、*検証期間*と呼ばれる設定可能な値によって決まります。有効期間は、エージェントがキャッシュの情報を使用してセッションをアクティブに保つ時間を示します。この期間が過ぎると、エージェントはセッションサーバに情報の更新を求めます。

検証期間が短すぎると、エージェントがセッション サーバに問い合わせる回数が増えるため、リクエストを処理する際の **SiteMinder** のパフォーマンスが低下します。したがって、有効期間は長めに設定することをお勧めします。アクティブなセッションの数が、エージェントの最大ユーザ セッション キャッシュ値よりも小さい場合、エージェントは、セッション サーバに問い合わせずに、キャッシュ内の情報を使用します。

注: セッション サーバが稼働していない場合、検証期間は無限になり、エージェントはセッション サーバに問い合わせません。

検証期間の設定手順については、「[レルムの設定](#) (P. 487)」を参照してください。

Windows ユーザ セキュリティ コンテキストの要件

Windows セキュリティ コンテキストの機能を使用するためには、IIS Web サーバ環境が次の条件を満たしている必要があります。

- ユーザの認証ドメイン内にあるすべての IIS サーバが信頼されていること。複数のユーザに分散サービスを提供するサーバ間に信頼関係を確立することができます。信頼関係を設定する方法については、**Microsoft** のサーバ関連マニュアルを参照してください。
- ユーザに **Web** サーバへのローカルログオン権限があること。複数のサーバがある場合、ユーザにはすべてのサーバにローカルでログオンする権限が必要になります。

ローカルログオンを可能するには、IIS の管理ツールを使用して機能を設定する必要があります。手順については、**Microsoft** のマニュアルを参照してください。

- **World Wide Web Publishing Service** の起動にシステム アカウントではなく、特定のドメイン アカウントを使用する場合は、サービスを実行しているドメイン アカウントについてユーザのアカウント権限を[オペレーティング システムの一部として機能]に設定する必要があります。この機能は、管理ツールを通じて設定します。手順の詳細については、**Microsoft** のマニュアルを参照してください。

設定の概要

手順	SiteMinder コンポーネント	関連マニュアル
リレーショナルデータベースをセッションストアとして指定する	SiteMinder ポリシー サーバ管理 コンソールの[データ]タブ	SiteMinder ポリシー サーバ管理 ガイド
Windows ユーザが含まれるユーザディレクトリでセキュリティコンテキストを実行できるように設定する	[ユーザディレクトリ]ウィンドウの [ディレクトリのセットアップ]グループ ボックスにある[認証済み ユーザ セキュリティコンテキスト を使用]チェック ボックス	ユーザディレクトリ (P. 165)
サポートされている認証方式 (P. 118)のうちの 1 つを各レلمと関連付ける	[レلم]ウィンドウのリソースグループ ボックス	レلمの設定 (P. 487)
各レلمで永続セッションを有効にして、有効な検証期間を高い値に設定する	[レلم]ウィンドウのセッショングループ ボックス	レلمの設定 (P. 487)
IIS Web サーバが物理的に安全な場所がない場合は、insecureserver を「はい」に設定する	エージェント設定オブジェクトまたは WebAgent.conf	SiteMinder Web エージェント設定ガイド

サポートされている認証方式

以下の認証方式に対応しています。

- 基本認証方式
- SSL を介した基本認証方式
- HTML フォーム認証方式
- X.509 クライアント証明書および基本認証方式
- X.509 クライアント証明書および HTML フォーム認証方式

ユーザ認証情報は提供されないため、Web エージェントは証明書のみを使用することはできません。ユーザ認証情報を収集するには、基本認証またはフォーム認証と証明書認証を組み合わせる必要があります。

イントラネットなどの一部のレルム内にあるリソースへのアクセスには、NTLM 認証方式を使用することもできます。ここで説明する Windows ユーザ セキュリティ コンテキスト設定の中では NTLM 認証方式を選択することはできません。ただし、Web サーバ上の一部のリソースに NTLM 認証方式を使用し、同じ Web サーバ上の別のリソースに Windows ユーザ セキュリティ コンテキストの仕組み (サポートされている認証方式)を使用することができます。この場合、NTLM を使用してアクセスされるリソースと、Windows ユーザ セキュリティ コンテキストメカニズムを通じてアクセスされるリソースは別のレルムに分ける必要があります。

Active Directory と NetBIOS 名

Active Directory ドメインは DNS ネーミング標準に従って名前が設定されますが、Active Directory ドメインの作成時には NetBIOS 名も定義する必要があります。

SiteMinder でシームレスな Microsoft セキュリティ コンテキストをサポートするには、Microsoft の推奨する DNS ドメイン名と同じ名前を NetBIOS 名にも指定する必要があります。

Active Directory ドメインの DNS 名と NetBIOS 名が異なると、SiteMinder が LDAP ユーザ ディレクトリを使用するように設定されている場合に、Web サーバに対してユーザドメインを確立できないため、セキュリティ コンテキストを提供できません。

セキュリティ コンテキストでのシングル サインオン

SiteMinder は、シームレスな Microsoft セキュリティ コンテキストに必要なセキュリティ コンテキストを保持しながら、サポートされるすべての Web サーバでのシングル サインオンを実現します。

ただし、このためには、ユーザの認証が実行される (かつ SiteMinder セッションが確立される) 可能性のあるすべてのレルムを「永続的」に設定する必要があります。ユーザの認証が実行されるレルムが永続的ではないときに、SiteMinder がセキュリティ コンテキストを永続化する必要が生じると、ユーザは再び識別情報の入力求められます。

第 6 章: エージェントおよびエージェントグループ

このセクションには、以下のトピックが含まれています。

- [Web エージェントに対するトラステッド ホスト \(P. 121\)](#)
- [トラステッド ホストホストのホスト設定オブジェクト \(P. 126\)](#)
- [SiteMinder エージェントの概要 \(P. 130\)](#)
- [Web エージェント設定の概要 \(P. 137\)](#)
- [Web エージェントの設定方法 \(P. 140\)](#)
- [エージェント設定オブジェクトの概要 \(P. 148\)](#)
- [Web エージェントの有効化 \(P. 155\)](#)
- [非 Web エージェントのエージェント識別情報の設定 \(P. 155\)](#)
- [RADIUS エージェントの設定 \(P. 157\)](#)
- [エージェント グループ \(P. 159\)](#)
- [カスタム エージェント \(P. 162\)](#)
- [SiteMinder エージェントによるリソースの保護 \(P. 164\)](#)

Web エージェントに対するトラステッド ホスト

トラステッド ホストは、SiteMinder Web エージェントを 1 つ以上インストールできるクライアントコンピュータです。トラステッドホストは、物理システムを指します。

ポリシー サーバへのトラステッド ホストの登録

Web エージェントをインストールするシステムにトラステッド ホストを登録します。ホスト登録プロセスは、エージェントのインストールおよび設定プロセスの一部です。登録が完了したら、登録ツールは `SmHost.conf` ファイルを作成します。このファイルが正常に作成されたら、クライアントコンピュータはトラステッド ホストになります。ポリシー サーバと通信するには、トラステッド ホストを登録する必要があります。

注: 管理 UI を使用してトラステッド ホストを作成することはできません。登録した後でのみトラステッド ホストを表示することができます。また、トラステッド ホストを削除することもできます。

初期設定時に、Web エージェントは、ホスト設定ファイル (`SmHost.conf`) に格納されているトラステッド ホストの設定を使用します。Web エージェントは、ホスト設定ファイルの `PolicyServer` パラメータ下にリストされている最初のポリシー サーバに接続を試みます。最初のポリシー サーバへの接続に失敗した場合、他にリストされているポリシー サーバがあれば、次のポリシー サーバへの接続を試みます。

Web エージェントがそのブートストラップ ポリシー サーバに接続すると、トラステッド ホストはホスト設定オブジェクトを検索します。これは、`Smhost.conf` ファイルの `hostconfigobject` パラメータに指定されています。この値はトラステッドホストの登録時に指定します。ホスト設定オブジェクトが見つかったら、トラステッドホストはその構成設定を取得します。

重要: ホストの登録は、Web エージェントをインストールおよび設定するたびにを行うのではなく、1 回のみ行います。

トラステッド ホスト設定

トラステッド ホスト設定の大半は、ホスト設定オブジェクトに設定します。唯一の例外は `Policy Server` と `RequestTimeout` のパラメータです。これらのパラメータは、ホスト設定ファイル (`SmHost.conf`) に設定します。

ホスト設定ファイルの設定が適用されるのは、トラステッド ホストの初期設定時のみです。トラステッド ホストの初期設定後は、ホスト設定オブジェクトの設定が有効になります。

リクエスト タイムアウト

リクエスト タイムアウト パラメータには秒数を指定します。トラステッド ホストはこの秒数が経過するのを待ってから、ポリシー サーバが使用不可であるものと判断します。このパラメータを設定することで、Web サーバの応答時間を最適化できます。

デフォルト値は 60 秒です。

注: ネットワーク負荷が高いか、ネットワークの接続速度が遅いためにポリシー サーバが使用不可になる場合は、`RequestTimeout` の値を増やしてください。

動作モード

設定された動作モードによって、トラステッド ホストが複数のポリシー サーバと連携する方法が決まります。動作モードには、フェイルオーバーとラウンドロビンの 2 種類があります。

フェイルオーバー

フェイルオーバーは冗長モードです。主ポリシー サーバに障害が起きると、バックアップ ポリシー サーバがポリシー操作を引き継ぎます。フェイルオーバーは、デフォルトの動作モードです。初期化時のトラステッド ホストはフェイルオーバー モードで動作します。

このモードでは、すべてのトラステッド ホストリクエストがリストの最初にあるポリシー サーバに送られます。ポリシー サーバからの応答がない場合、トラステッド ホストはそのポリシー サーバを「使用不可」としてマークし、2 番目にリストされたポリシー サーバにリクエストをリダイレクトします。最初のポリシー サーバが障害から復旧すると、リストの先頭に戻ります。

ラウンドロビン

ラウンド ロビン モードでは、トラステッド ホストが複数のポリシー サーバにリクエストを分散して送信できます。これにより、ポリシー サーバへのアクセスが高速化され、認証と許可も効率化されます。また、1 つのポリシー サーバにリクエストが集中するのを回避できます。

このモードでは、リストの最初にあるポリシー サーバにリクエストが送られます。次のリクエストはリストの 2 番目にあるポリシー サーバに送信されます。このように、トラステッドホストは使用可能なポリシー サーバすべてに順にリクエストを送信していきます。すべてのポリシー サーバにリクエストを送信すると、再びリストの最初にあるポリシー サーバに戻って、同じサイクルが繰り返されます。

ポリシー サーバに障害が起きると、リクエストはリストの次にあるサーバにリダイレクトされます。トラステッド ホストは、障害の起きたサーバを「使用不可」とマークして、以降のすべてのリクエストをほかのサーバにリダイレクトします。サーバが障害から復旧すると、自動的にリストの元の位置に戻ります。

動作モードの実装

設定された動作モードによって、トラステッド ホストが複数のポリシー サーバと連携する方法が決まります。動作モードには、フェイルオーバーとラウンドロビンの 2 種類があります。

動作モードを実装する方法

1. 複数のポリシー サーバを構成します。
2. すべてのポリシー サーバが共通のポリシーストアを使用するように設定します。
3. `EnableFailover` パラメータを設定します。
 - フェイルオーバーを有効にするには、`EnableFailover` パラメータを「はい」に設定します。
 - ロードバランスを有効にするには、`EnableFailover` パラメータを「いいえ」に設定します。

`EnableFailover` パラメータの値は、ホスト設定オブジェクトに指定されたすべてのポリシー サーバに適用されます。

TCP/IP 接続

トラステッドホストとポリシー サーバは **TCP/IP** 接続を経由して通信します。ポリシー サーバの許可ポート、認証ポート、アカウントिंग ポートに使用可能なソケットの数によって、トラステッド ホストとポリシー サーバとの間で確立可能な **TCP/IP** 接続の数が決まります。

また、ポート当たりのソケット数によって、**Web** サーバからポリシー サーバにアクセスする同時スレッドの数が決まります。各ユーザのアクセスリクエストは、別個の **Web** サーバスレッドによって処理されるため、それぞれにソケットが必要になります。**Web** サーバはリクエストに使用するスレッドのプールを保持し、使用可能なスレッドがなくなると新しいスレッドを作成します。トラフィックが増加した場合には、ポート当たりのスレッド数を増やす必要があります。

トラステッドホストとポリシー サーバ間の **TCP/IP** 接続に影響する設定は次のとおりです。

1 ポートあたりの最大ソケット数

トラステッド ホストとポリシー サーバ間の通信に使用される **TCP/IP** 接続の最大数を定義します。デフォルトではこの値は **20** に設定されます。**Web** サイトのトラフィックが少量から中程度であれば、通常はこの設定値で十分です。トラフィック量が多い **Web** サイトを管理する場合や、仮想サーバにエージェント識別情報を定義してある場合には、この値を増やす必要があります。

1 ポートあたりの最小ソケット数

起動時にポリシー サーバに対して開かれる TCP/IP 接続の数を示します。デフォルト値は 2 です。トラフィック量が多い Web サイトを管理する場合は、この値を増やす必要があります。

新規ソケット ステップ

新しい接続が必要になった場合にエージェントが開く TCP/IP 接続の数を示します。デフォルト値は 2 です。ソケットの数が不足する場合は、毎回追加するソケットの数を変更してください。

注: これらの値、および SiteMinder 環境の拡大に応じてこの値を調整する方法の詳細については、「ポリシー サーバ管理ガイド」を参照してください。

トラステッド ホスト オブジェクトの削除

トラステッド ホストを再登録すると、そのトラステッド ホストは削除されます。トラステッドホストを再登録するには、登録ツールの **smregghost** を使用します。このツールは、Web エージェントと共にインストールされます。

注: Web エージェント設定ウィザードを実行すると、トラステッド ホストを再登録できます。ただし、**SmHost.conf** ファイルを削除するか、またはファイル名を変更しないと、ウィザードはトラステッド ホストを登録するようプロンプトするメッセージは表示されません。このため、トラステッドホストの再登録には **smregghost** ツールを使用することをお勧めします。トラステッド ホストの登録および再登録についての詳細は、「Web エージェント インストール ガイド」を参照してください。

トラステッドホストを削除する方法

1. [インフラストラクチャ]-[ホスト]-[トラステッド ホスト]をクリックします。
2. [トラステッド ホストの削除]をクリックします。
[検索]ダイアログが表示されます。
3. 検索条件を指定して[検索]をクリックします。
検索条件に一致するトラステッド ホストのリストが表示されます。

4. リストからトラステッド ホストを選択し、[選択]をクリックします。

注: 一度に複数のトラステッド ホストを選択できます。

ホストの削除を確認するメッセージが表示されます。

5. [はい]をクリックします。

トラステッド ホストが削除されます。

トラステッド ホストホストのホスト設定オブジェクト

ホスト設定オブジェクトは、トラステッド ホストの設定を格納します。トラステッド ホストがポリシー サーバに接続すると、ホスト設定オブジェクトの設定を使用します。

Web エージェント側では、トラステッド ホストが使用するホスト設定オブジェクトは、**SmHost.conf** ファイルの **hostconfigobject** パラメータで識別されます。**SmHost.conf** ファイル内の設定は、各 Web サーバの子プロセスの初期化フェーズで Web エージェントによって使用されます。これは、**SmHost.conf** ファイルが起動時に使用されるだけでなく、容量を追加したり予期せず停止したプロセスを代替するために、Web サーバにより実行時にも使用される場合があることを意味します。初期化が完了すると、ホスト設定オブジェクトの設定が使用されます。

トラステッド ホスト オブジェクトを作成する前に、ホスト設定オブジェクトを作成する必要があります。

ホスト設定オブジェクトのコピー

ホスト設定オブジェクトの作成に推奨される方法は、既存のホスト設定オブジェクトをコピーし、そのプロパティを変更することです。**DefaultHostSettings** オブジェクトをコピーし、そのプロパティを新規オブジェクトのテンプレートとして使用できます。

重要: ホスト設定オブジェクトの名前は一意である必要があります。既存のエージェントオブジェクトと同じ名前を使用しないでください。すでに別の Web エージェントに割り当てられている名前を入力すると、そのトラステッドホストはすでに存在することを知らせるメッセージが表示されます。

ホスト設定オブジェクトをコピーする方法

1. [インフラストラクチャ]-[ホスト]をクリックします。
 2. [ホスト設定]-[ホスト設定の作成]をクリックします。
[ホスト設定の作成]ウィンドウが開きます。
 3. [コピーの作成]ラジオ ボタンを選択して、以下のいずれかを実行します。
 - デフォルトのホスト設定オブジェクトのうちのいずれかを選択します。
 - [検索]をクリックし、表示されるホスト設定オブジェクトのリストから選択します。
 4. [OK]をクリックします。
[ホスト設定の作成: <名前>]ウィンドウが開きます。
- 注:**それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
5. [一般]グループ ボックス上のフィールドに、新しい名前および説明を入力します。
 6. 新規オブジェクトで異なるプロパティを変更し、[サブミット]をクリックします。
ホスト設定の作成タスクを処理できるよう送信します。

ホスト設定オブジェクトへの複数のポリシー サーバの追加

トラステッド ホストは複数のポリシー サーバにアクセスできます。トラステッド ホスト接続、フェイルオーバー、またはラウンド ロビン操作をセットアップするには、ホスト設定オブジェクトにポリシー サーバを追加する必要があります。

ホスト設定オブジェクトにポリシー サーバを追加する方法

1. [インフラストラクチャ]-[ホスト]をクリックします。
2. [ホスト設定]-[ホスト設定の変更]をクリックします。
[ホスト設定の変更]ウィンドウが開きます。
3. 検索条件を指定して[検索]をクリックします。
検索条件に一致するホスト設定オブジェクトのリストが表示されます。

4. ホスト設定オブジェクトを選択し、[選択]をクリックします。

[ホスト設定: <名前> の変更]ウィンドウが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

5. [クラスタ]グループ ボックスで[追加]をクリックします。

[クラスタの追加]ダイアログ ボックスが開きます。

6. [クラスタの追加]グループ ボックスの[ホスト]フィールドおよび[ポート]フィールドに、クラスタに追加するポリシー サーバの IP アドレスおよびポート番号をそれぞれ入力します。

注: クラスタにポリシー サーバを追加するには、[クラスタへ追加]をクリックします。クラスタからポリシー サーバを削除するには、右側のマイナス記号をクリックします。クラスタ内のポリシー サーバの順序を変更するには、上下の矢印をクリックします。

7. [OK]をクリックします。

[クラスタ]グループ ボックスにクラスタが追加されます。

注: クラスタを変更するには、左側の右向き矢印をクリックします。クラスタを削除するには、右側のマイナス記号をクリックします。クラスタを追加するには、[クラスタ]グループ ボックスで[追加]をクリックしてステップ 6 および 7 を繰り返します。

8. [フェイルオーバーのしきい値パーセント]フィールドにフェイルオーバーのしきい値を入力します。

注: クラスタ内のアクティブなサーバの割合が指定された割合を下回ると、そのクラスタは、クラスタのリスト内の次に使用可能なクラスタにフェイルオーバーします。

9. [サブミット]をクリックします。

ホスト設定の変更タスクを処理できるよう送信します。

ホスト設定オブジェクトのポリシー サーバ クラスタの構成

フェイルオーバーを実現するには、複数のポリシー サーバのクラスタを構成します。サーバをクラスタ化することで、1 つのサーバ グループから別のグループに処理をフェイルオーバーすることができます。

ポリシー サーバ クラスタは、ホスト設定オブジェクトの一部として定義されます。SiteMinder Web エージェントが初期化されると、ホスト設定オブジェクトの設定を使用してポリシー サーバとの通信が設定されます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

クラスタを構成する方法

1. [インフラストラクチャ]-[ホスト]をクリックします。
2. [ホスト設定]-[ホスト設定の作成]をクリックします。
[ホスト設定の作成]ウィンドウが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[ホスト設定の作成: <名前>]ウィンドウが開きます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [一般]グループ ボックスのフィールドにホスト設定オブジェクトの名前および説明を入力します。
5. [設定値]グループ ボックスのフィールドで[ホスト設定]を指定します。
6. [クラスタ]グループ ボックスで[追加]をクリックします。
[クラスタの追加]ダイアログ ボックスが開きます。
7. [クラスタの追加]グループ ボックスの[ホスト]フィールドおよび[ポート]フィールドに、クラスタに追加するポリシー サーバの IP アドレスおよびポート番号をそれぞれ入力します。

注: クラスタにポリシー サーバを追加するには、[クラスタへ追加]をクリックします。クラスタからポリシー サーバを削除するには、右側のマイナス記号をクリックします。クラスタ内のポリシー サーバの順序を変更するには、上下の矢印をクリックします。

8. [OK]をクリックします。

[クラスタ]グループ ボックスにクラスタが追加されます。

注: クラスタを変更するには、左側の右向き矢印をクリックします。クラスタを削除するには、右側のマイナス記号をクリックします。クラスタを追加するには、[クラスタ]グループ ボックスで[追加]をクリックしてステップ 7 および 8 を繰り返します。

9. [クラスタ]グループ ボックスの[フェイルオーバーのしきい値パーセント]フィールドに割合を入力します。

注: クラスタ内のアクティブなサーバの割合が指定された割合を下回ると、そのクラスタは、クラスタのリスト内の次に使用可能なクラスタにフェイルオーバーします。

10. [サブミット]をクリックします。

ホスト設定の作成タスクを処理できるよう送信します。

重要: [設定値]グループ ボックスは、[クラスタ]グループ ボックスでクラスタが指定されていない場合のみ使用される単一のポリシー サーバおよび単純なフェイルオーバー操作を指定します。単純なフェイルオーバー操作を使用するためにクラスタをすべて削除する場合は、[クラスタ]グループ ボックスからポリシー サーバ情報をすべて削除してください。

SiteMinder エージェントの概要

SiteMinder 環境のエージェントは、ネットワークアクセス制御または **Web** アクセス制御を実行するフィルタとして機能するネットワーク エンティティです。エージェントは、リソースに対するリクエストを監視します。ユーザが保護されたリソースへのアクセスを要求すると、エージェントは認証方式に基づいて証明をユーザに要求し、そのクレデンシャルをポリシー サーバに送信します。

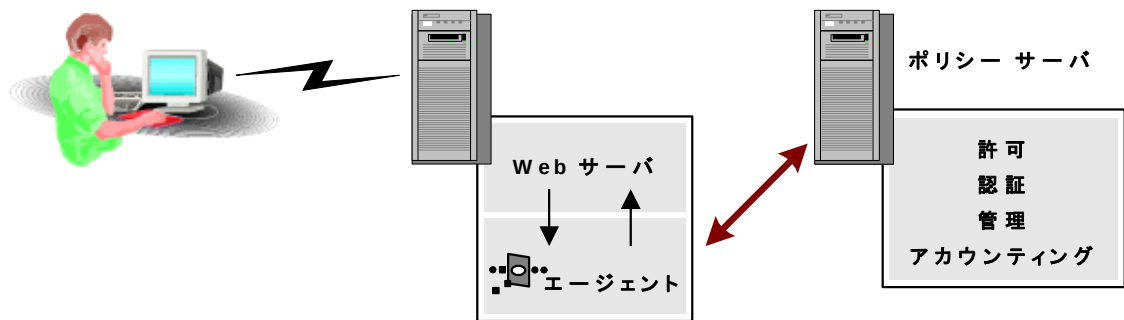
ポリシー サーバはその証明に基づいて、ユーザを認証するかどうか、また要求されたリソースへのアクセスを許可するかどうかを決定します。次にポリシー サーバはエージェントと通信し、要求されたリソースへのアクセスの許可もしくは拒否を行います。

デフォルトで **Web** エージェント、アフィリエイト エージェント、**EJB** エージェント、サーブレット エージェント、**RADIUS** エージェントを使用できます。それ以外のエージェントはすべて、エージェント **API** を使用して作成されたカスタムエージェントとみなされます。作成されたカスタム エージェントは、管理 **UI** で設定できます。

Web エージェント

Web エージェントは、Web サーバと連携する SiteMinder エージェントです。ユーザが Web サーバにページを要求すると、Web エージェントはポリシー サーバと通信して、認証リクエストおよび許可リクエストを処理します。これで、ユーザは Web ブラウザからそのリソースにアクセスできるようになります。さらに、ポリシー サーバは、Web エージェントがユーザの識別情報に基づいてパーソナライズされたコンテンツを提供するために使用する情報を提供することもできます。

次の図は、保護されたリソースへのアクセスを許可するために Web エージェントとポリシー サーバが処理する、最も基本的な 3 つのトランザクションを示しています。コンテンツのカスタマイズを行ったり、SiteMinder の他の機能をサポートしたりするために、これらのトランザクションにより詳細な情報を含めることもできますが、ユーザが、Web サーバを介して、Web エージェントで管理されるリソースへのアクセスを試みる場合は、常に同様のプロセスが発生します。



1. リソースは保護されているか？ されている場合、ユーザの認証情報を要求
2. ユーザは認証されているか？ されている場合、許可のポリシーを確認
3. ユーザは許可されているか？ されている場合、保護されたリソースへのアクセスを許可

前述の図では、ユーザが要求している保護されたリソースの権限がそのユーザにあるとします。Web エージェントはリソースが保護されているかどうかをポリシー サーバに問い合わせ、ポリシー サーバはリソースが保護されていることを Web エージェントに知らせます。次に、Web エージェントはユーザ証明を収集して、それをポリシー サーバに送信します。

ポリシー サーバはユーザを認証して、そのユーザが正しく識別されたことを **Web** エージェントに通知します。最後に、**Web** エージェントはポリシー サーバに確認して、ユーザがリソースへのアクセスを許可されているかどうか判断します。このとき、ポリシー サーバはユーザがリソースへのアクセスを許可されていることを確認し、それを **Web** エージェントに通知します。**Web** エージェントからの許可を受け、**Web** サーバはユーザが要求した保護されたリソースを表示します。

同じリソースを制御する、同じエージェント タイプ (すべての **Web** エージェント、またはすべての **RADIUS** エージェント) のエージェントは、グループ化することができます。

注: 仮想 **Web** サーバのサポートを設定する予定がある場合は、「**Web** エージェント設定ガイド」を参照してください。

SAML アフィリエイトエージェント

SAML アフィリエイト エージェントは、**Federation** セキュリティ サービス ソリューションの一部です。**Federation** セキュリティ サービスにより、企業が複数のドメイン間でセキュリティ情報を共有することが可能になります。

注: **Federation** セキュリティ サービスと **SAML** アフィリエイト エージェントは、別売のパッケージで、ライセンスが別途必要になります。

SAML アフィリエイト エージェントは、ポータルなどのプロデューサ側から、統合ネットワークの提携先 (アフィリエイト) として機能する **SAML** コンシューマへのシームレスなシングルサインオンを実現します。アフィリエイトからは、ポータルに関連するリソースやサービスが提供されます。たとえば、**affiliateA.com** と **affiliateB.com** が契約を交わして、**affiliateA.com** にアクセスした人が **affiliateB.com** で買い物する場合に **10%** の値引きを受けられるとします。この場合、これら 2 つのサイトがアフィリエイト (提携) 関係にあるとすることができます。**SAML** アフィリエイト エージェントを使用すれば、ユーザが複数のサイトにアクセスする際に再認証を受けたり、識別情報を再入力したりする必要がなくなります。

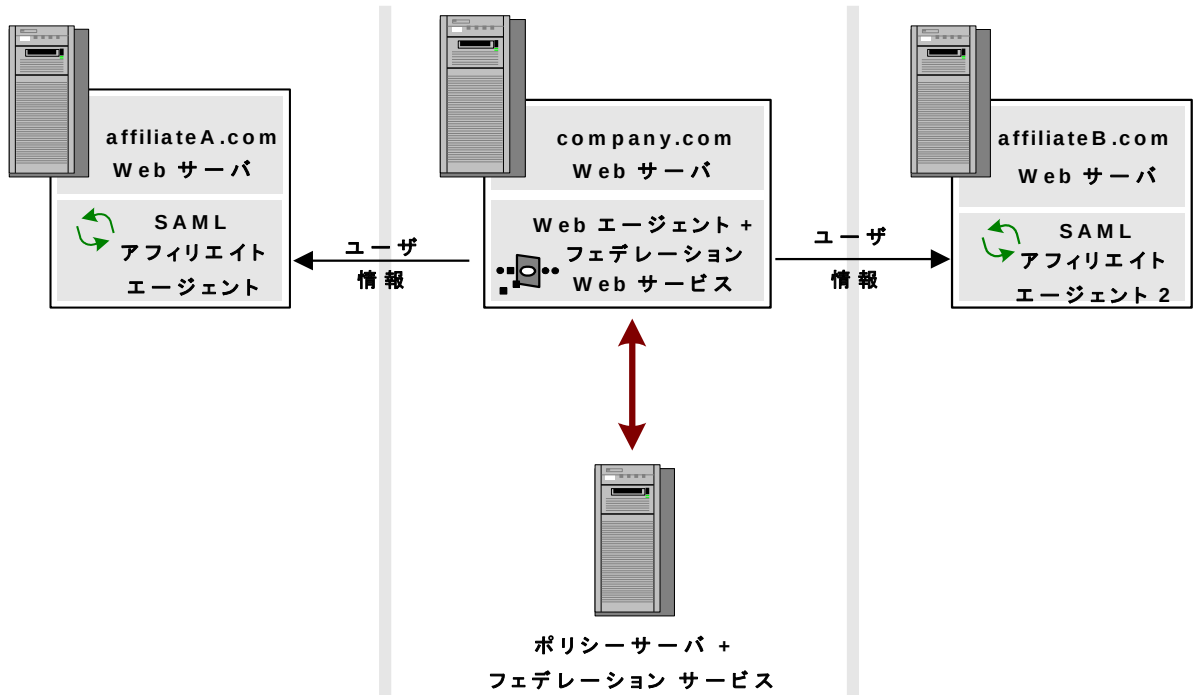
SAML アフィリエイト エージェントと SiteMinder がプロデューサ側で通信すると、SAML アサーションが生成されます。SAML アサーションは、ユーザの許可情報と認証情報が入った XML ドキュメントです。プロデューサはこの文書をコンシューマ側にある SAML アフィリエイト エージェントに送ります。このアサーションから、ユーザがメインポータルで認証を受けていることが確認されます。さらに、アフィリエイトはアサーションに含まれる情報をもとに、Web サーバにユーザ情報を提供し、Web サーバはこの情報を Web アプリケーションに使用できます。

提携サイトに SAML アフィリエイト エージェントをインストールする場合、そのサイトにインストールするのは SiteMinder コンポーネントのみです。SAML アフィリエイト エージェントはパーソナライズを可能にするだけで、リソースを保護することはありません。したがって、提携サイトに SiteMinder をフルインストールする必要はありません。

プロデューサ側で Federation セキュリティ サービスをサポートするには、次のコンポーネントが必要になります。

- ポリシー サーバ
- Web エージェント
- Web エージェントオプションパックには、Federation Web サービスアプリケーションが含まれています。

以下の図に、SAML アフィリエイト エージェントを含むフェデレーション ネットワークを示します。



前述の図では、ポリシー サーバは **company.com** の Web サーバ上の Web エージェントに接続されています。このエージェントは SAML アフィリエイト エージェントを使用して、**company.com** ユーザに関するユーザ情報を **affiliateA.com** と **affiliateB.com** の Web サーバに渡します。

プロデューサのサイトでは、ユーザを認証し、そのユーザに関する情報をアフィリエイト サイトに渡します。このサイトには SiteMinder がフル インストールされています。アフィリエイト サイトには SAML アフィリエイト エージェントと Web サーバのみがインストールされていて、ポリシー サーバはありません。SAML アフィリエイト エージェントは、Web エージェントと同様の方法でリソースを保護しないため、アフィリエイト サイトに SiteMinder をフルインストールする必要はありません。SAML アフィリエイト エージェントは、Web アプリケーションで使用するためのユーザ情報を Web サーバへ提供するだけです。アプリケーションはこの情報を使用して Web コンテンツをパーソナライズし、各ユーザ独自のインタフェースを作成します。

SAML アフィリエイト エージェントと Federation セキュリティ サービスの設定

SAML アフィリエイト エージェントの設定はコンシューマ側で実行します。つまり、プロデューサ側のポリシー サーバでは **SAML** アフィリエイト エージェントを設定しません。ただし、ポリシー サーバ ユーザインタフェースで **Federation** セキュリティ サービス コンポーネントを設定して、アフィリエイトドメインを作成し、そのアフィリエイトドメインのアフィリエイトを設定します。

注: ポリシー サーバのユーザ インタフェースには、エージェントを作成し、エージェントタイプとしてアフィリエイト エージェントを選択できるオプションがあります。これは、**SAML** アフィリエイト エージェント用のオプションではありません。このオプションは、**4.x** アフィリエイト エージェントにのみ適用されます。したがって、このオプションは選択しないでください。詳細については、「*Federation Security Services Guide*」を参照してください。

RADIUS エージェント

RADIUS (リモート認証ダイヤルイン ユーザ サービス) は、NAS (ネットワーク アクセス サーバ) デバイスと RADIUS 認証サーバの間で、セッション認証と設定情報を交換するためのプロトコルです。RADIUS プロトコルは、プロキシサービス、ファイアウォールまたはダイヤルアップ セキュリティ デバイスとして機能する NAS デバイスで頻繁に利用されます。

RADIUS エージェントは、RADIUS プロトコルを使用して通信するアプリケーション全体を保護します。

ポリシー サーバは RADIUS 認証サーバとして使用できます。RADIUS エージェントは、ポリシー サーバと NAS クライアントデバイスの通信を可能にします。

アプリケーション サーバ エージェント

アプリケーション サーバ エージェントは、Java コンポーネントの集合で、WebLogic および WebSphere アプリケーション サーバのリソースを保護するための全機能を備えた SiteMinder エージェントを提供します。アプリケーション サーバ エージェントは、J2EE プラットフォームで SiteMinder と統合します。

アプリケーション サーバ エージェントを使用して、次のコンポーネントを保護できます。

- Web アプリケーション (サーブレット、HTML ページ、JSP、イメージファイルを含む)
- JNDI 検索
- EJB コンポーネント
- JMS 接続ファクトリ、トピック、およびキュー
- JDBC 接続プール

アプリケーション サーバ エージェントは、1 つのエージェントですが、SiteMinder ポリシー サーバから見ると、アプリケーション サーバリソースを保護する複数のエージェント タイプとして認識されます。このエージェント タイプにより、アプリケーション サーバ エージェントでは、サーブレットと EJB コンポーネントの保護において、サーブレットまたは EJB エージェントをそれぞれ使用して保護するか、Web エージェントを使用して保護するという 2 つの方法で柔軟に保護することができます。

注: SiteMinder をアプリケーション サーバ エージェントと連携させる設定についての詳細は、該当するプラットフォームの「*SiteMinder Application Server Agent Guide*」を参照してください。

CA SOA Security Manager SOA エージェント

CA SOA Security Manager SOA エージェントは、Web サーバやアプリケーション サーバと統合され、それらサーバでホストされている Web サービスのリソースへのアクセスを求めるリクエストを認証および許可します。

注: SOA エージェントの詳細については、「*CA SOA Security Manager Policy Configuration Guide*」を参照してください。

Web エージェント設定の概要

Web エージェントを設定するためのオプションは 2 つあります。

中央設定

ポリシー サーバから Web エージェント設定を行います。ポリシー ストアは、一群の Web エージェントによって使用される設定パラメータのセットを保持します。パラメータは 管理 UI を使用して設定します。

エージェント設定はエージェント設定オブジェクトで指定されます。

注： 中央設定は、RADIUS、EJB、Servlet、またはカスタム エージェントには適用されません。これらのエージェントはローカル設定のみ実行できます。

ローカル設定

エージェントがインストールされている各 Web サーバ上のローカル設定ファイルから Web エージェントを設定します。

パラメータは中央とローカルの両方に分けて格納することができます。

注： Web エージェントはローカル エージェント設定ファイルからのみ有効または無効にできます。ポリシー サーバからはこの操作は行えません。これは、中央またはローカルのどちらの設定でも同じです。

Web エージェントを中央で設定した場合の利点

Web エージェントを中央で設定すると、設定はローカルではなく、Web サーバ上にある構成ファイルのポリシー ストアに格納されます。

ローカル設定と比べると、中央設定には以下のような利点があります。

エージェントを中央で設定すると、操作性が向上する

- 複数の Web エージェントの構成設定を簡単かつすばやく更新できます。1 か所で設定を変更したら、変更内容を複数の Web エージェントに伝播できます。
- Web エージェントをインストール際に、管理者が共有秘密キーを指定する必要がなくなります。ポリシー サーバは共有秘密キーを生成します。

エージェントの中央設定ではセキュリティが強化される

- Web エージェント設定へのアクセス制御が容易になります。たとえば、Web サーバの管理者が構成設定を変更するのを阻止できます。
- 構成設定は Web サーバ上ではなく、ファイアウォールの内側に格納できます。
- ハードウェアに関連付けられたキーを使用して共有秘密キーを生成し、クライアント側でこのキーを使ってポリシー サーバへの安全な接続を確立できます。ポリシー サーバへの接続はトラステッドホストによって処理されます。

Web エージェント コンポーネント

SiteMinder ネットワークのエージェント側では、以下のような主なコンポーネントが Web エージェントの動作に関与します。

SiteMinder Web エージェント

Web サーバへの仮想インタフェース。ルールを起動し、ポリシーを実行します。

トラステッド ホスト

1 つ以上の Web エージェントがインストールされているクライアントコンピュータ。ポリシー サーバへの接続を処理します。トラステッドホストは、物理システムを指します。1 台のサーバに複数のトラステッド ホストを設定できますが、それぞれを一意の名前で識別する必要があります。

トラステッド ホストは、ポリシー サーバに登録された信頼されるホストです。ホストにインストールされた Web エージェントがポリシー サーバと通信するためには、そのホストに登録してトラステッド ホストにする必要があります。

トラステッド ホストを識別するデータは次のとおりです。

- ホスト名
- ホストの IP アドレス
- ホストの説明

- 共有秘密キー
- ホストオブジェクト ID (OID)

Web エージェント設定ファイル (WebAgent.conf または LocalConfig.conf)

エージェントがインストールされている Web サーバに格納されます。このファイルは、ローカル設定に使用されます。このファイルには、各 Web エージェントのエージェント設定パラメータが入っています。Web エージェントはすべて WebAgent.conf ファイルを使用します。ただし、IIS 6.0 Web エージェントは、開始してポリシー サーバに接続するのに必要なコア設定についてのみ WebAgent.conf ファイルを使用します。IIS 6.0 Web エージェントは、その設定に LocalConfig.conf ファイルを使用します。IIS 6.0 WebAgent.conf ファイルの中に LocalConfig.conf ファイルへのポインタがあります。

ホスト設定ファイル (SmHost.conf)

エージェントがインストールされている Web サーバに格納されます。このファイルには、トラステッド ホストの初期化パラメータが含まれています。トラステッド ホストはポリシー サーバに接続すると、ポリシー サーバに格納されているの設定を使用します。ホスト設定オブジェクトは、このファイルの hostconfigobject パラメータで指定されます。

Web エージェントに関連するポリシー サーバオブジェクト

ポリシー サーバ側に、Web エージェント設定に関連する 3 つのポリシー オブジェクトがあります。

エージェントオブジェクト

エージェントの名前を指定して、特定の Web サーバにマッピングできるエージェントの識別情報を確立します。

エージェント設定オブジェクト

Web エージェント設定パラメータが含まれます。エージェント設定オブジェクトを使用して Web エージェントのグループを一元管理します。このオブジェクトは主にエージェントの中央設定に使用しますが、ポリシー サーバにローカル設定を使用するように指示するパラメータも含まれています。このオブジェクトは Web エージェントに対してのみ適用されます。

ホスト設定オブジェクト

トラステッド ホスト設定パラメータが含まれます。初期設定パラメータを除き、トラステッド ホスト パラメータは常にホスト設定オブジェクトに保持されます。

設定オブジェクトはポリシーストアに格納されます。設定オブジェクトの作成、変更、表示は、管理 UI で行います。

Web エージェントの設定方法

完全に Web エージェントを設定するために完了する必要があるタスクが多くあります。これらのタスクは、Web エージェントのローカル設定と中央設定に適用されます。

注: Web エージェントをインストールしてトラステッド ホストを登録する前に、Web エージェント用のポリシー サーバをセットアップしておく必要があります。

Web エージェントを設定する方法

1. ポリシー サーバをインストールします。
2. ホスト設定オブジェクトを作成します。
3. ポリシー サーバ管理者にトラステッドホストの登録権限を付与します。管理者にはトラステッドホストの登録権限が必要です。

注: [トラステッドホストの登録]権限しか付与されていない管理者は 管理 UI を使用できません。

4. エージェント オブジェクトを作成してエージェントに名前を付けます。このオブジェクトとエージェント設定オブジェクトは違うものです。混同しないように注意してください。
5. エージェント設定オブジェクトを作成します。
エージェントをローカル設定する場合でも、ローカル設定パラメータの **AllowLocalConfig** を有効にするにはこのオブジェクトを作成する必要があります。
6. クライアント側で、Web エージェントをインストールします。

7. トラストド ホストを登録します。この手順では、ポリシー サーバにすでに作成されているホスト設定オブジェクトに名前を付けます。
8. エージェントに関連するポリシーオブジェクトを構成する場合には、**Web エージェント**を有効にします。この設定は、ローカル エージェント構成ファイルに格納されます。

注: すべてのパラメータについての説明とデフォルト値、パラメータの設定手順については、「**Web エージェント設定ガイド**」を参照してください。**Web エージェント**を中央またはローカルのどちらで設定する場合も、このガイドのパラメータについての説明を参照してください。また、エージェントおよびトラストド ホスト登録処理に関する情報は、「**ポリシー サーバ インストール ガイド**」および「**Web エージェント インストール ガイド**」を参照してください。

Web エージェントの中央設定

Web エージェントを中央で設定するには、「**Web エージェントの設定**」に記載されている手順に従います。これらのタスクは、**Web エージェント**のローカル設定と中央設定に適用されます。

いずれかの設定パラメータをローカルで指定すると、ローカル エージェント環境設定ファイルのパラメータ値によって対応するエージェント設定オブジェクトの値が上書きされ、2 つの設定の入力がマージされます。

エージェント設定オブジェクトとエージェント設定ファイルの入力をマージせずに、ローカル設定を排他的に使用するには、エージェント設定オブジェクトに **AllowLocalConfig** パラメータのみを指定し、このパラメータを「はい」に設定してください。こうすることで、**Web エージェント**にはローカル設定ファイルの設定データのみが適用されます。

中央設定とローカル設定がどのように連携するかの詳細については、「中央設定とローカル設定の組み合わせ」を参照してください。

ホスト設定オブジェクトの作成

新しいホスト設定オブジェクトを作成することも、既存のオブジェクトを複製することもできます。

ホスト設定オブジェクトを作成する方法

1. [インフラストラクチャ]-[ホスト]をクリックします。
2. [ホスト設定]-[ホスト設定の作成]をクリックします。
[ホスト設定の作成]ウィンドウが開きます。
3. 以下のいずれかの操作を行います。
 - (推奨) 既存のホスト設定オブジェクトのコピーを作成して、そのプロパティを変更します。DefaultHostSettings オブジェクトをコピーし、その設定を新規オブジェクトのテンプレートとして使用できます。
DefaultHostSettings オブジェクトはポリシー サーバ インストール プログラムによってインストールされます。
重要: DefaultHostSettings オブジェクトを直接編集するのは避けてください。必ずこのオブジェクトをコピーしてから変更するようにしてください。
 - オブジェクトを新規作成します。
4. [OK]をクリックします。
[<ホスト設定の作成: 名前>]ウィンドウが開きます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
5. [一般]グループ ボックスのフィールドに名前および説明を入力します。
6. [設定値]グループ ボックスで[ホスト設定]を指定します。
7. [サブミット]をクリックします。
ホスト設定の作成タスクを処理できるよう送信します。

ローカルでの Web エージェントの設定

Web エージェントは、エージェント設定オブジェクトとローカル エージェント設定ファイルの両方を読み込みます。エージェント設定オブジェクトの値は、ローカル エージェント設定ファイルの値で上書きされます。Web エージェントはこれらの設定を 1 つの構成ソースにまとめます。これで、エージェント パラメータの一部のみをローカルで変更し、エージェントの残りの設定には中央のエージェント設定オブジェクトを利用することができます。

中央設定とローカル設定がどのように連携するかの詳細については、「中央設定とローカル設定の組み合わせ」を参照してください。

パラメータをローカルで設定する方法

1. ポリシー サーバ管理者からローカル設定の実行許可を取得します。
2. 「Web エージェントの設定方法」の手順を実行します。これらのタスクは、Web エージェントのローカル設定と中央設定に適用されます。
3. エージェント設定オブジェクト内で、AllowLocalConfig パラメータを **yes** に設定します。
4. Web エージェント設定ファイル (WebAgent.conf/LocalConfig.conf) を編集します。

必ず、Web エージェント設定ファイルのコピーに変更を加えて、バックアップコピーを保持してください。

IIS 6.0 を除くすべての Web エージェントで、<web_agent_home>\config ディレクトリに WebAgent.conf.sample ファイルがあります。このファイルを変更し、WebAgent.conf という名前で、Web サーバ上の適切な場所にこのファイルを保存します。

IIS 6.0 の Web エージェントでは、アクティブな設定ファイルとして、<web_agent_home>\bin\IIS ディレクトリの LocalConfig.conf ファイルが使用されます。変更する必要がある場合は、このファイルを変更します。<web_agent_home>\config ディレクトリにある LocalConfig.conf ファイルのコピーはオリジナルなので、変更しないでください。

注: IIS 6.0 の Web エージェントを使用している場合、メインの設定ファイルの名前は LocalConfig.conf です。WebAgent.conf ファイルも使用しますが、使用されるのはエージェントを起動してポリシー サーバに接続するために必要なコア設定のみです。

ローカル設定およびパラメータの詳細については、「Web エージェント設定ガイド」を参照してください。

中央設定とローカル設定の組み合わせ

Web エージェントは有効になった時点で、エージェント設定オブジェクトを検索し、設定情報を取得します。ついで、**AllowLocalConfig** パラメータの値に注目します。このパラメータが[はい]に設定されていると、Web エージェントは対応するエージェントのローカル設定ファイルから変更されたパラメータまたは追加のパラメータを探して、エージェント設定オブジェクトのパラメータをすべて設定ファイルの値で上書きします。

エージェントは中央およびローカルの設定ソースを使用し、それらを統合して、エージェント設定オブジェクトのローカルコピーを 1 つ作成します。それを使用して設定を行います。このローカルコピーは、ポリシー サーバ上にあるエージェント設定オブジェクトに変更を加えることはありません。

中央設定とローカル設定の組み合わせの使用例

シナリオ:

各エージェントを個別に設定することなく、**SiteMinder** ネットワーク全体で複数の **cookie** ドメインのシングル サインオンを設定します。

エージェント設定オブジェクトの **CookieDomain** パラメータを **acmecorp.com** に設定します。ただし、ネットワーク内の 1 つの Web エージェントで、**CookieDomain** パラメータを **test.com** に設定し、その他のすべてのパラメータにはエージェント設定オブジェクトに設定されている値をそのまま使用する必要があります。

解決方法:

このサンプル設定を実装する方法

1. 使用している環境に該当するすべてのパラメータを使用して、エージェント設定オブジェクトを設定します。
2. エージェント設定オブジェクト内で、**AllowLocalConfig** パラメータを **yes** に設定します。
3. 1 つの Web エージェントで、**CookieDomain** パラメータのみを **test.com** に変更します。他のどのパラメータも変更しないでください。

エージェント設定ファイル内にある **CookieDomain** パラメータの値は、エージェント設定オブジェクトの値より優先されますが、他のすべてのパラメータに関しては、エージェント設定オブジェクトが設定値を決定します。

エージェント オブジェクトの作成による Web エージェント ID の確立

Web エージェント ID を作成するには、管理 UI でエージェント オブジェクトを作成する必要があります。オブジェクト名は、**AgentName** のエージェント名または、エージェント設定ファイルまたはエージェント設定オブジェクトの **DefaultAgentName** パラメータと一致する必要があります。ポリシー サーバはエージェント ID を使用して、Web エージェントをホストしている Web サーバの IP アドレスにエージェント名をマッピングし、ポリシーを Web エージェントに正しく関連付けます。Web エージェント オブジェクトおよび ID を作成すると、Web エージェントをレルムに関連付けられます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

Web エージェント オブジェクトおよび ID を作成する方法

1. [インフラストラクチャ]-[エージェント]-[エージェントの作成]をクリックします。
[エージェントの作成]ペインが開きます。
2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[エージェントの作成: <名前>]ペインが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。

注: Web エージェント名には、以下の制限があります。

- エージェント名には、印刷可能な文字を少なくとも 1 文字含む、32 から 127 の範囲内の 7 ビットの ASCII 文字を使用する必要があります。
- エージェント名にアンパサンド(&)およびアスタリスク(*)文字を使用することはできません。
- エージェント名は大文字/小文字を区別しません。たとえば、あるエージェントに **MyAgent** という名前を付け、別のエージェントに **myagent** という名前を付けることはできません。

4. [属性]グループ ボックスで、エージェントタイプに **Web** エージェントを、エージェントスタイルに **SiteMinder** を選択します。
5. [サブミット]をクリックします。
エージェントの作成タスクが、処理のためにサブミットされます。

4.x Web エージェント識別情報を作成するためのエージェントオブジェクトの設定

4.x Web エージェントアイデンティティを作成するために、管理 UI 内のエージェントオブジェクトを作成する必要があります。オブジェクト名は、ローカルの **Web** エージェント設定ファイルのエージェント名と一致していなければなりません。設定パラメータの説明については、「**Web** エージェント設定ガイド」を参照してください。 **Web** エージェントオブジェクトおよび ID を作成すると、**Web** エージェントをレルムに関連付けられます。

重要: 4.x **Web** エージェントのサポート完了日に関する通知が **CA** から發送される予定です。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

4.x の Web エージェント オブジェクトおよび識別情報を作成する方法

1. [インフラストラクチャ]-[エージェント]-[エージェントの作成]をクリックします。
[エージェントの作成]ペインが開きます。
2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[エージェントの作成: <名前>]ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。

制限:

- エージェント名には、印刷可能な文字を少なくとも 1 文字含む、32 から 127 の範囲内の 7 ビットの ASCII 文字を使用する必要があります。
- エージェント名にアンパサンド(&)およびアスタリスク(*)文字を使用することはできません。
- エージェント名は大文字/小文字を区別しません。たとえば、あるエージェントに **MyAgent** という名前を付け、別のエージェントに **myagent** という名前を付けることはできません。

4. 以下の点を確認します。

- SiteMinder ラジオ ボタンが選択されていること。
- Web エージェントが[エージェント タイプ]ドロップダウンリストに表示されていること。

5. [4.x エージェントをサポートします]チェック ボックスをオンにします。

[信頼性の設定]グループ ボックスが表示されます。

6. [IP アドレス]フィールドに、エージェントが置かれているサーバの IP アドレスを入力します。

注: 仮想サーバには、単一サーバと同様、名前と IP アドレスが定義されています。仮想サーバ上の各エージェントには、一意なエージェント名を設定する必要があります。

7. それぞれのフィールドで、共有秘密キーの入力および確認を行います。

制限:

- 指定する秘密キーは、**Web** エージェントを **Web** サーバにインストールしたときに割り当てられた秘密キーと一致する必要があります。
- 秘密キーには、1 文字以上 254 文字以下の文字を使用する必要があります。
- 秘密キーに使用できるのは、英数字のみです。
- 秘密キーにはスペースを使用できません。

注: 同一の **Web** サーバ上にある仮想サーバは、同じ秘密キーを共有する必要があります。4.x エージェントがポリシー サーバに接続を試みる際、エージェントおよびポリシー サーバで相互認証に共有秘密キーが使用されます。

8. [サブミット]をクリックします。

エージェントの作成タスクが、処理のためにサブミットされます。

エージェント設定オブジェクトでの設定パラメータの設定

以下の手順には、エージェント設定オブジェクトの設定パラメータを設定するのに必要な 2 つの一般的な副次的手順を含んでいます。

Web エージェントの設定を定義する方法

1. エージェント設定オブジェクトを作成します。
2. このオブジェクトの設定パラメータを変更します。

注: 一元的またはローカルに **Web** エージェントを設定する場合は、パラメータの説明、デフォルト値、およびパラメータの設定方法の詳細について「**Web** エージェント設定ガイド」を参照してください。

エージェント設定オブジェクトの概要

エージェント設定オブジェクトには、**Web** エージェントの構成を定義するパラメータが格納されます。設定オブジェクトは、ポリシー サーバにおける **Web** エージェント設定ファイルに相当するものです。

Web エージェントを設定すると、エージェント設定オブジェクトの名前を要求されます。設定オブジェクトは、設定している Web エージェントに関連付けられます。

Web エージェントの初期設定および必須の Web エージェントパラメータの詳細については、「Web エージェント インストール ガイド」を参照してください。

エージェント設定オブジェクトのコピー

既存のエージェント設定オブジェクトをコピーして、新しいエージェント設定オブジェクトを作成できます。SiteMinder がサポートする Web エージェント用のサンプル エージェント設定オブジェクトが用意されています。新しいエージェント設定オブジェクトを作成したら、パラメータとパラメータ値のリストを変更することができます。パラメータおよびデフォルト設定の詳細については、「Web エージェント設定ガイド」を参照してください。

エージェント設定オブジェクトをコピーする方法

1. [インフラストラクチャ]-[エージェント設定]-[エージェント設定の作成]をクリックします。
[エージェント設定の作成: 検索]ペインが開きます。
2. コピーの作成ラジオ ボタンを選択し、以下のどちらかを実行します。
 - リスト内のデフォルトのエージェント設定オブジェクトのいずれかを選択します。
 - [検索]をクリックして、表示されたエージェント設定オブジェクトのリストから選択します。
3. [OK]をクリックします。
[エージェント設定の作成: 名前]ペインが開きます。
4. [一般]グループ ボックスの各フィールドに、新しい名前と説明を入力し、[サブミット]をクリックします。
エージェント設定の作成タスクが、処理のためにサブミットされます。

エージェント設定オブジェクトの作成

ポリシー サーバのインストーラは、複数のサポートされている Web サーバ用の複数のエージェント設定オブジェクトを作成します。ここに、デフォルトの設定が含まれています。これらのサンプル オブジェクトの名前は、`IISDefaultSettings`、`iPlanetDefaultSettings` などです。これらのデフォルト オブジェクトを複製し、エージェント設定のテンプレートとして使用することができます。

エージェント設定オブジェクトを作成する方法

- (推奨) 既存のエージェント設定オブジェクトをコピーし、パラメータ値を変更します。

重要: デフォルト オブジェクトを直接変更したり、使用したりしないでください。必ず、オブジェクトのコピーを作成し、そのコピーを変更するようにしてください。

- 新しいエージェント設定オブジェクトを作成します。

注: 新しいエージェント設定オブジェクトを作成するには、既存のオブジェクトのコピーを作成し、そのコピーで、パラメータとパラメータ値のリストを変更することをお勧めします。既存のエージェント設定オブジェクトのコピーを作成せずに、新しいエージェント設定オブジェクトを作成する場合は、パラメータとパラメータ値をすべて手動で入力する必要があります。

エージェント設定オブジェクトを作成する方法

1. [インフラストラクチャ]-[エージェント設定]-[エージェント設定の作成]をクリックします。

[エージェント設定の作成: 検索]ペインが開きます。

2. [オブジェクトの新規作成]が選択されていることを確認し、[OK]をクリックします。

[エージェント設定の作成: 名前]ペインが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。
4. [追加]をクリックします。

[パラメータの作成]グループ ボックスが開きます。

5. [パラメータの作成]グループ ボックスの各フィールドに、パラメータの名前と値を入力します。

6. (省略可)以下のいずれかを実行します。

- パラメータの値を暗号化するには、[暗号化]チェック ボックスをオンにします。
- 1つのパラメータに対して複数の値を作成するには、以下を実行します。
 - a. [複数值]ボタンをクリックします。
 - b. [値]フィールドに、パラメータの値を入力します。
 - c. [追加]をクリックします。
 - d. 必要な値をすべて追加するまで、手順 b および c を繰り返します。
 - e. (省略可)複数の値の順序を変更するには、上向きの矢印と下向きの矢印をクリックします。複数の値は、[パラメータ]グループ ボックスに表示されるときは正方形記号(□)で区切られます。値を削除するには、マイナス符号をクリックします。

注: 暗号化されたパラメータに複数の値を入力することはできません。暗号化された単一の値は、[パラメータ]グループ ボックスに一連の記号として表示されます。

7. [OK]をクリックします。

パラメータが、[パラメータ]グループ ボックスのリストに追加されます。

注: パラメータは、アルファベット順にリストされます。リスト上のパラメータを編集するには、右向きの矢印をクリックします。リストからパラメータを削除するには、マイナス符号をクリックします。エージェント設定オブジェクトにさらにパラメータを追加するには、手順 4 から 7 を繰り返します。

8. [サブミット]をクリックします。

エージェント設定の作成タスクが、処理のためにサブミットされます。

エージェント設定オブジェクトの必須パラメータ

エージェント設定オブジェクトを設定する場合、以下のパラメータを設定する必要があります。

- すべてのエージェントに **DefaultAgentName** が必須
- IIS エージェントは、**DefaultUserName** および **DefaultPassword** を必要とします。
- Domino エージェントは、**DominoDefaultUser** および **DominoSuperUser** を必要とします。

注: これらのパラメータについての詳細は、「**Web エージェント インストール ガイド**」を参照してください。

IIS プロキシ ユーザの指定

IIS Web エージェントを設定するには、エージェント設定オブジェクトで以下のよう

に設定する必要があります。

注: NTLM 認証方式を使用する場合、または Windows ユーザ セキュリティ コンテキスト機能を有効にする場合には、これらの IIS パラメータに値を指定しないでください。

DefaultUserName

IIS プロキシ ユーザのユーザ名を指定します。

DefaultPassword

IIS プロキシ ユーザのパスワードを指定します。

DefaultUserName と **DefaultPassword** によって、**SiteMinder** で保護されている IIS Web サーバ上のリソースにアクセスするための十分な権限を持つ既存の NT ユーザ アカウントを識別します。**SiteMinder** で保護されている IIS Web サーバ上のリソースにアクセスしようとするユーザに、必ずしも必要なサーバアクセス権限があるとは限りません。**Web** エージェントが **SiteMinder** によってアクセスを許可されたユーザのプロキシユーザ アカウントとして機能するためには、NT 管理者によって割り当てられたこの NT ユーザ アカウントを使用する必要があります。

Domino ユーザの指定

Domino サーバ上に **Web** エージェントを設定するには、システムに合わせて以下の設定を編集する必要があります。

DominoDefaultUser

ユーザが **Domino** ディレクトリに含まれず、**SiteMinder** で別のユーザ ディレクトリに対して認証されている場合は、これが、**Domino Web** エージェントがユーザを **Domino** サーバに対して識別する場合に使用する名前になります。この値は暗号化できます。

DominoSuperUser

SiteMinder へのログインに成功したすべてのユーザが **Domino SuperUser** として **Domino** にログインできるようにします。この値は暗号化できます。

エージェント名の指定

すべての **Web** エージェントで、**DefaultAgentName** の値を指定する必要があります。

DefaultAgentName

Web エージェントが **Web** サーバ上でエージェント識別情報が割り当てられていない IP アドレスを検出したときに使用するエージェント識別情報を指定します。

デフォルト: デフォルトのエージェント名は、インストールされている **Web** エージェントの名前です。

エージェント設定パラメータの変更

エージェント設定オブジェクト内のパラメータの名前および値を編集できます。パラメータ名は一意である必要があります。パラメータ値はプレーンでも、暗号化または複数値にすることもできます。非アクティブパラメータをアクティブにするには、パウンド記号(#)を削除します。

注: パラメータおよびデフォルト設定の詳細については、「**Web** エージェント設定ガイド」を参照してください。

エージェント設定オブジェクトを変更する方法

1. [インフラストラクチャ]-[エージェント設定]-[エージェント設定の変更]をクリックします。

[エージェント設定の変更: 検索]ウィンドウが開きます。

2. 検索条件を指定して[検索]をクリックします。

検索条件に一致するエージェント設定オブジェクトのリストが表示されます。

3. エージェント設定オブジェクトを選択し、[選択]をクリックします。

[エージェント設定の変更: 名前]ウィンドウが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [パラメータ]グループ ボックスのパラメータ名の左側にある右向き矢印をクリックします。

[パラメータの編集]ダイアログ ボックスが開きます。

5. パラメータの値を編集します。

注: 値を削除するには、マイナス記号をクリックします。複数の値の順序を変更するには、上向き矢印と下向き矢印をクリックします。1 つのパラメータに対して複数の値を指定するには、[追加]をクリックします。複数の値は、[パラメータ]グループ ボックスに表示されるときは正方形記号(□)で区切られます。

6. (オプション) [パラメータの編集]グループ ボックスで[暗号化]を選択します。

注: [暗号化]を選択すると、1 つのパラメータに対して複数の値を指定するオプションは無効になります。暗号化された単一の値は、[パラメータ]グループ ボックスに一連の記号として表示されます。

7. [OK]をクリックします。

[パラメータ]グループ ボックスにおいてパラメータの値が更新されます。

注: パラメータは、アルファベット順にリストされます。リスト上のパラメータを編集するには、右向きの矢印をクリックします。リストからパラメータを削除するには、マイナス符号をクリックします。エージェント設定オブジェクトの複数のパラメータを編集するには、手順 4 ～ 7 を繰り返します。

8. [サブミット]をクリックします。

エージェント設定の変更タスクを処理できるよう送信します。

Web エージェントの有効化

Web エージェントを中央またはローカルのどちらで設定する場合でも、Web エージェントは **WebAgent.conf** ファイルからのみ、有効または無効に設定できます。

エージェントが有効または無効かは、**EnableWebAgent** パラメータの値によって決まります。エージェントを有効にするには、このパラメータを「はい」に設定します。デフォルトでは、「いいえ」に設定されます。

ポリシー サーバとエージェントが通信するために必要なすべてのオブジェクトのセットアップが完了し、Web エージェントをインストールして設定した後で、Web エージェントを有効にします。

注: **WebAgent.conf** ファイルおよび Web エージェントの有効化についての詳細は、「Web エージェント設定ガイド」を参照してください。

非 Web エージェントのエージェント識別情報の設定

すべての **SiteMinder** エージェントにはエージェント識別情報が必要になります。ポリシー サーバはエージェントの ID を使用して、エージェントをホストしている **Web**、**RADIUS**、またはアプリケーション サーバ インスタンスにエージェント名をマッピングします。ポリシー サーバはエージェント識別情報を使用して、ポリシーを適切なエージェントに関連付けます。

管理 UI でエージェント オブジェクトを作成することにより、エージェント識別情報を設定できます。エージェントオブジェクトでは、エージェントに名前を割り当てて、**Web** エージェント、**EJB** エージェント、**サーブレット**エージェント、**RADIUS** エージェント、カスタムエージェントのいずれかのエージェントタイプを定義できます。

注: アフィリエイト エージェントは、エージェント オブジェクトのエージェントタイプのうちの 1 つです。このエージェントタイプは **4.x** アフィリエイト エージェントにのみ適用されます。このオプションは、**SAML** アフィリエイト エージェントを設定するのに使用しないでください。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

エージェント オブジェクトの設定方法

1. [インフラストラクチャ]-[エージェント]-[エージェントの作成]をクリックします。

[エージェントの作成]ペインが開きます。

2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

[エージェントの作成: <名前>]ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。

制限:

- エージェント名には、印刷可能な文字を少なくとも 1 文字含む、32 ～ 127 の 7 ビット ASCII 文字が含まれている必要があります。
 - エージェント名にアンパサンド(&)およびアスタリスク(*)文字を使用することはできません。
 - エージェント名は大文字/小文字を区別しません。たとえば、あるエージェントに **MyAgent** という名前を付け、別のエージェントに **myagent** という名前を付けることはできません。
4. [エージェント タイプ]グループ ボックスでエージェント設定を指定します。
 5. 該当する手順に従って、残りのフィールドに必要な情報を入力します。
 - Web エージェントの詳細については、「[エージェントオブジェクトの作成による Web エージェント ID の確立](#) (P. 145)」を参照してください。
 - RADIUS エージェントについては、「[RADIUS エージェントの設定](#) (P. 157)」を参照してください。
 - カスタム エージェントについては、「[カスタム エージェント](#) (P. 162)」を参照してください。
 - EJB エージェントおよびサーブレット エージェントについては、WebLogic または WebSphere アプリケーション サーバ向けの「[SiteMinder Application Server Agent guide](#)」を参照してください。

RADIUS エージェントの設定

RADIUS エージェント ID を作成するには、管理 UI でエージェント オブジェクトを作成する必要があります。オブジェクト名は、エージェントをインストールしたときに指定したエージェント名と一致していなければなりません。Web エージェント管理コンソールまたは **WebAgent.conf** ファイルでエージェント名が変更された場合、エージェント名も変更する必要があります。ポリシー サーバは、RADIUS エージェント ID を使用して NAS クライアント デバイスと通信します。RADIUS エージェント オブジェクトおよび ID を作成すると、RADIUS エージェントをレルムに関連付けられます。

RADIUS エージェント ID を作成した後は、RADIUS クライアントまたはアプリケーション サーバに **SiteMinder** エージェントをインストールおよび設定できます。RADIUS エージェントはローカル設定のみに対応しています。より詳細な情報については、「**Web エージェント インストール ガイド**」を参照してください。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「**ポリシー サーバ オブジェクトの複製**」を参照してください。

RADIUS エージェントの設定方法

1. [インフラストラクチャ]-[エージェント]-[エージェントの作成]をクリックします。
[エージェントの作成]ペインが開きます。
2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[エージェントの作成: <名前>]ペインが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。

制限:

- エージェント名には、印刷可能な文字を少なくとも 1 文字含む、32 から 127 の範囲内の 7 ビットの ASCII 文字を使用する必要があります。
- エージェント名にアンパサンド(&)およびアスタリスク(*)文字を使用することはできません。
- エージェント名は大文字/小文字を区別しません。たとえば、あるエージェントに **MyAgent** という名前を付け、別のエージェントに **myagent** という名前を付けることはできません。

4. [RADIUS]ラジオ ボタンをクリックしてから、[エージェントタイプ]ドロップダウンリストから **RADIUS** ベンダーを選択します。

注: エージェントタイプとして **Generic RADIUS** を選択すると、すべてのタイプの **RADIUS** 機器を保護できます。ただし、**Generic RADIUS** エージェントタイプはベンダーに特有のレスポンス属性へのアクセスを提供しません。

[信頼性の設定]および[RADIUS 設定]のグループ ボックスが開きます。

5. [信頼性の設定]グループ ボックスで以下の項目を追加します。
 - フィールドに **RADIUS** クライアント(NAS デバイス)の IP アドレスを入力します。
 - 該当するフィールドに英数字の共有秘密キーを入力して確認します。
6. [RADIUS 設定]グループ ボックスの[レルム ヒント]フィールドに、**RADIUS** レルム ヒントの数を入力します。

注: レルム ヒントの詳細については、「ポリシー サーバ管理ガイド」を参照してください。

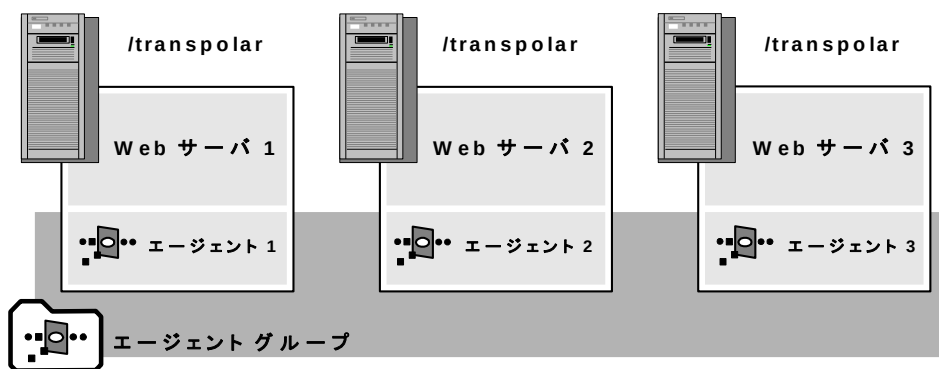
7. [サブミット]をクリックします。

エージェントの作成タスクが、処理のためにサブミットされます。

エージェントグループ

エージェントグループは、共通のリソースを保護するために同じタイプのエージェントを集めてグループ化したものです。エージェントグループの利点は、多くの Web サーバ/Web エージェントにリソースの複製が作成されているため、多くのユーザ ベースに対して同じリソースへのアクセス権を付与できることです。また、すべての Web エージェントに対してポリシーを 1 つだけ定義すればよいので、時間を節約することもできます。たとえば、エージェントグループを使用して、ラウンドロビン処理によって同じリソースへのアクセスを提供している Web サーバのグループ (Web ファーム) を保護することができます。

下図では、1 セットのポリシーで Web ファームを保護し、そのポリシーのセットにバインドされたエージェントグループを作成しています。



グループに含めるエージェントがまだ作成されていなくても、エージェントグループを作成することができます。SiteMinder に新規エージェントを追加すれば、エージェントグループを編集したり、グループに新規エージェントを追加したりできるようになります。

注: エージェントグループを設定しても、エージェントグループに直接、一連のパラメータ値を割り当てることはできません。同じを複数のエージェント設定オブジェクトに割り当てて、オブジェクトを編集することができます。

エージェントグループの設定

エージェントグループを作成すると、共通のリソースを保護するエージェントをより効率よく管理できます。同じグループ内のエージェントはすべて同じタイプである必要があります。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

エージェントグループの設定方法

1. [インフラストラクチャ]-[エージェントグループ]-[エージェントグループの作成]をクリックします。

[エージェントグループの作成]ウィンドウが開きます。

2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

[エージェントグループ: <名前> の作成]ウィンドウが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスのフィールドにエージェントグループの名前および説明を入力します。

4. [エージェントタイプ]グループ ボックスで、エージェントスタイルおよびエージェントタイプを選択します。

注: エージェントグループには同じタイプのエージェントのみ含められます。たとえば、すべてのエージェントを **Web** エージェント、アフィリエイトエージェント、または **RADIUS** エージェントなどにします。

5. [グループ メンバ]グループ ボックスで[追加/削除]をクリックします。

[エージェントの選択]ウィンドウが開きます。

注: 指定されたエージェントタイプのエージェントのみが、[使用可能なメンバー]列に表示されます。たとえば、指定されたエージェントスタイルが **RADIUS** であり、指定されたエージェントタイプが **3-Com** である場合、**3-Com** エージェントのみが表示されます。指定されたエージェントタイプが **Generic RADIUS** である場合、**RADIUS** エージェントがすべて表示されます。

6. [使用可能なメンバー]リストから 1 つまたは複数のエージェントを選択し、右向きの矢印をクリックします。

[使用可能なメンバー]リストからエージェントが削除され、[メンバーの選択]リストに追加されます。

注: 一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。

7. [OK]をクリックします。

選択されたエージェントがエージェントグループに追加されます。

8. [サブミット]をクリックします。

エージェントグループの作成タスクを処理できるよう送信します。

エージェントグループへのエージェントの追加

エージェントグループに既存のエージェントを追加できます。

エージェントグループにエージェントを追加する方法

1. [インフラストラクチャ]-[エージェントグループ]-[エージェントグループの変更]をクリックします。

[エージェントグループの変更]ウィンドウが開きます。

2. 検索条件を指定して[検索]をクリックします。

検索条件に一致するエージェントグループのリストが表示されます。

3. エージェントグループを選択し、[選択]をクリックします。

[エージェントグループ: <名前> の変更]ウィンドウが開きます。

4. [グループ メンバ]グループ ボックスで[追加/削除]をクリックします。

[エージェントの選択]ウィンドウが開きます。

注: 指定されたエージェントタイプのエージェントのみが、[使用可能なメンバー]列に表示されます。たとえば、指定されたエージェントスタイルが **RADIUS** であり、指定されたエージェントタイプが **3-Com** である場合、**3-Com** エージェントのみが表示されます。指定されたエージェントタイプが **Generic RADIUS** である場合、**RADIUS** エージェントがすべて表示されます。

5. [使用可能なメンバー]リストから **1** つまたは複数のエージェントを選択し、右向きの矢印をクリックします。
[使用可能なメンバー]リストからエージェントが削除され、[メンバーの選択]リストに追加されます。
注: 一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。
6. [OK]をクリックします。
選択されたエージェントがエージェントグループに追加されます。
7. [サブミット]をクリックします。
エージェントグループの変更タスクを処理できるよう送信します。

カスタム エージェント

SiteMinder エージェント API の **C** または **Java** バージョンのいずれかを使用して、カスタム エージェントを作成します。SiteMinder エージェント API と関連マニュアルは、別売のソフトウェア開発キットに含まれています。

注: カスタムエージェントについては中央設定はサポートされません。API 使用によるカスタム エージェントの作成についての詳細は、「SiteMinder Programming Guide for C」または「SiteMinder Programming Guide for Java」を参照してください。

独自のエージェントを作成したら、新しいエージェントタイプを設定する必要があります。エージェントタイプはエージェントの動作を定義し、リソースを保護するカスタム エージェントを使用できるようにします。たとえば、カスタム **FTP** エージェントを作成した場合には、**FTP** エージェントタイプを定義する必要があります。

カスタム エージェント タイプの設定

カスタム エージェントタイプを設定して、カスタム エージェントの動作を定義および、エージェントオブジェクトの作成時にカスタム エージェントを識別します。SiteMinder API を使用して、エージェントタイプを作成します。

注: SiteMinder Agent API の **C** バージョン使用によるエージェントタイプの作成についての詳細は、SiteMinder Programming Guide for C を参照してください。

エージェント識別情報用のカスタム エージェント オブジェクトの作成

カスタム エージェントの識別情報を作成するには、管理 UI でエージェント オブジェクトを作成する必要があります。エージェント オブジェクトと識別情報を作成すると、エージェントをレルムに関連付けることができます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

エージェント オブジェクトを作成する方法

1. [インフラストラクチャ]-[エージェント]-[エージェントの作成]をクリックします。
[エージェントの作成]ペインが開きます。
2. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[エージェントの作成: <名前>]ペインが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、エージェントの名前と説明を入力します。

制限:

- エージェント名には、印刷可能な文字を少なくとも 1 文字含む、32 から 127 の範囲内の 7 ビットの ASCII 文字を使用する必要があります。
- エージェント名にアンパサンド(&)およびアスタリスク(*)文字を使用することはできません。
- エージェント名は大文字/小文字を区別しません。たとえば、あるエージェントに **MyAgent** という名前を付け、別のエージェントに **myagent** という名前を付けることはできません。

4. [エージェント タイプ]グループ ボックスで、エージェント スタイルとして [SiteMinder]を選択し、[エージェント タイプ]として[Web エージェント]を選択し、[4.x エージェントをサポートします]チェック ボックスをオンにします。

[IP アドレス]グループ ボックスと[共有秘密キー]グループ ボックスが開きます。

5. [IP アドレス]グループ ボックスのフィールドに、エージェントが置かれているサーバの IP アドレスを入力します。
6. [共有秘密キー]グループ ボックスの各フィールドで、共有秘密キーの入力と確認を行います。

制限:

- 指定する秘密キーは、エージェントを **Web** サーバにインストールしたときに割り当てられた秘密キーと一致する必要があります。
 - 秘密キーには、1 文字以上 254 文字以下の文字を使用する必要があります。
 - 秘密キーに使用できるのは、英数字のみです。
 - 秘密キーにはスペースを使用できません。
7. [サブミット]をクリックします。

エージェントの作成タスクが、処理のためにサブミットされます。

SiteMinder エージェントによるリソースの保護

設定されたエージェントをレルムに関連付けます。レルムとは、保護するリソースの集合体です。レルムはルールで保護されます。ルールはアクセス制御ポリシーに含まれています。

第 7 章: ユーザ ディレクトリ

このセクションには、以下のトピックが含まれています。

[ユーザ ディレクトリ接続の概要](#) (P. 166)

[ディレクトリ属性の概要](#) (P. 183)

[CA Directory ユーザ ディレクトリ接続を設定する方法](#) (P. 184)

[Sun Java System ユーザ ディレクトリ接続を設定する方法](#) (P. 188)

[IBM Directory Server ユーザ ディレクトリ接続を設定する方法](#) (P. 189)

[Domino ユーザ ディレクトリをユーザ ストアとして設定する方法](#) (P. 191)

[Novell eDirectory LDAP ディレクトリ接続を設定する方法](#) (P. 193)

[ADAM ユーザ ディレクトリ接続を設定する方法](#) (P. 198)

[Active Directory ディレクトリ接続を設定する方法](#) (P. 203)

[Active Directory グローバル カタログ ユーザ ディレクトリ接続を設定する方法](#) (P. 210)

[Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法](#) (P. 212)

[ODBC ユーザ ディレクトリ接続を設定する方法](#) (P. 215)

[Windows ディレクトリ接続を設定する方法](#) (P. 220)

[カスタム ユーザ ディレクトリ接続を設定する方法](#) (P. 225)

[SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法](#) (P. 227)

[LDAP ロード バランシングおよびフェイルオーバー](#) (P. 238)

[ODBC データ ソース フェイルオーバーの設定](#) (P. 244)

[SQL クエリ方式](#) (P. 244)

[複数ポリシー ストアの同じユーザ ディレクトリ接続の定義](#) (P. 253)

[ユーザ ディレクトリの内容の表示](#) (P. 254)

[ユーザ ディレクトリの検索](#) (P. 255)

[ユニバーサル ID](#) (P. 255)

[名前付き式](#) (P. 256)

[ユーザ属性マッピング](#) (P. 273)

ユーザ ディレクトリ接続の概要

ユーザ ディレクトリおよびユーザ データベースには、ユーザ データが格納されます。ユーザ データには、組織的な情報、ユーザ属性とグループ属性、およびパスワードなどのクレデンシャルが含まれます。管理 UI を使用して、既存のユーザ ディレクトリおよびユーザ データベースへの接続を設定することができます。

ポリシー サーバがユーザの識別情報のコンテキストを確立する方法を解決するように、ディレクトリ接続を設定します。ポリシー サーバはこれらの接続を使用して、ユーザの識別情報を確認し、ユーザ ストアに含まれているユーザ属性を取得します。

サポートされている任意の数のユーザ ディレクトリに接続するように、ポリシー サーバを設定します。サポートされているユーザ ディレクトリは、以下のとおりです。

- LDAP
- ODBC
- Oracle
- Windows NT
- カスタム

サポートされているディレクトリ タイプのリストについては、SiteMinder のプラットフォーム サポート マトリックスを参照してください。

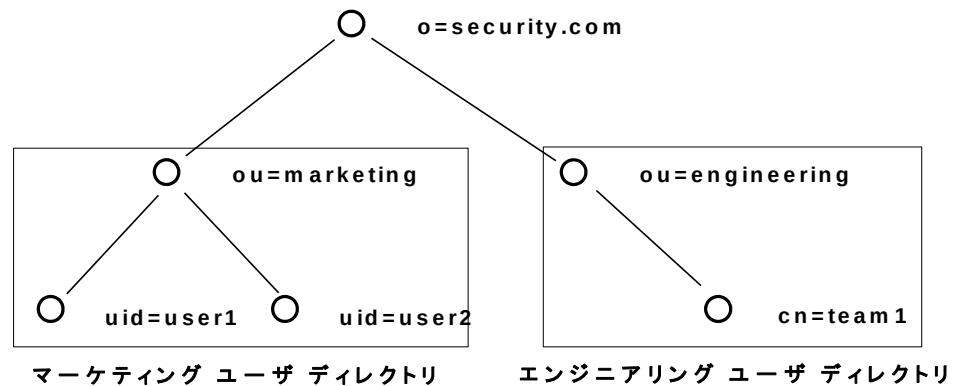
注: SiteMinder プラットフォームのサポート マトリックスに記載された SiteMinder ストアを設定またはアップグレードしようとして、このガイドの手順が見つからない場合は、「*Directory Configuration Guide*」を参照してください。

LDAP の概要

ポリシー サーバは、LDAP (Lightweight Data Access Protocol) を使用するユーザ ディレクトリと通信できます。

LDAP に関する全般情報

LDAP ユーザディレクトリは逆ツリー構造で作成されます。この階層構造により、LDAP 対応のディレクトリは複数のユーザネームスペースを持つことができます。ネームスペースとは、LDAP DIT (ディレクトリ情報ツリー) 内のノードの下にあるエンティティグループのことです。LDAP DIT のどのブランチも、固有のネームスペースとしてユーザディレクトリ接続に定義できます。通常、ユーザ ディレクトリ接続は、組織(o)または組織単位(ou)を表わす DIT ブランチに対して設定されます。ユーザとユーザグループは、ディレクトリ構造内の o= ノードまたは ou= ノードの下に位置します。



LDAP ツリー内のノードは、すべて DN (識別名) によって区別されます。DN は、そのノードとディレクトリツリー内の上位ノードの名前をカンマで区切ったリストで構成されています。この命名方式により、ユーザディレクトリ内の各ポイントが固有の DN を保有することになります。

たとえば前述の図では、マーケティング部門のユーザは次のような DN によって識別されます。

uid=user1,ou=marketing,o=security.com

また、エンジニアリングのユーザグループは、次のような DN によって識別されます。

ou=engineering,o=security.com.

LDAP ディレクトリのユーザの明確化

ユーザの明確化とは、ユーザディレクトリから一意のユーザを検索するプロセスのことです。ユーザ ディレクトリでユーザを検索する方法は 2 種類あります。以下の方法でユーザを検索できます。

- DN
- 検索式

ポリシー サーバは[ユーザ ディレクトリ]ウィンドウの[ユーザの検索]グループボックスで提供された情報および、ログイン名などユーザが提供した値を使用してユーザを見つけます。

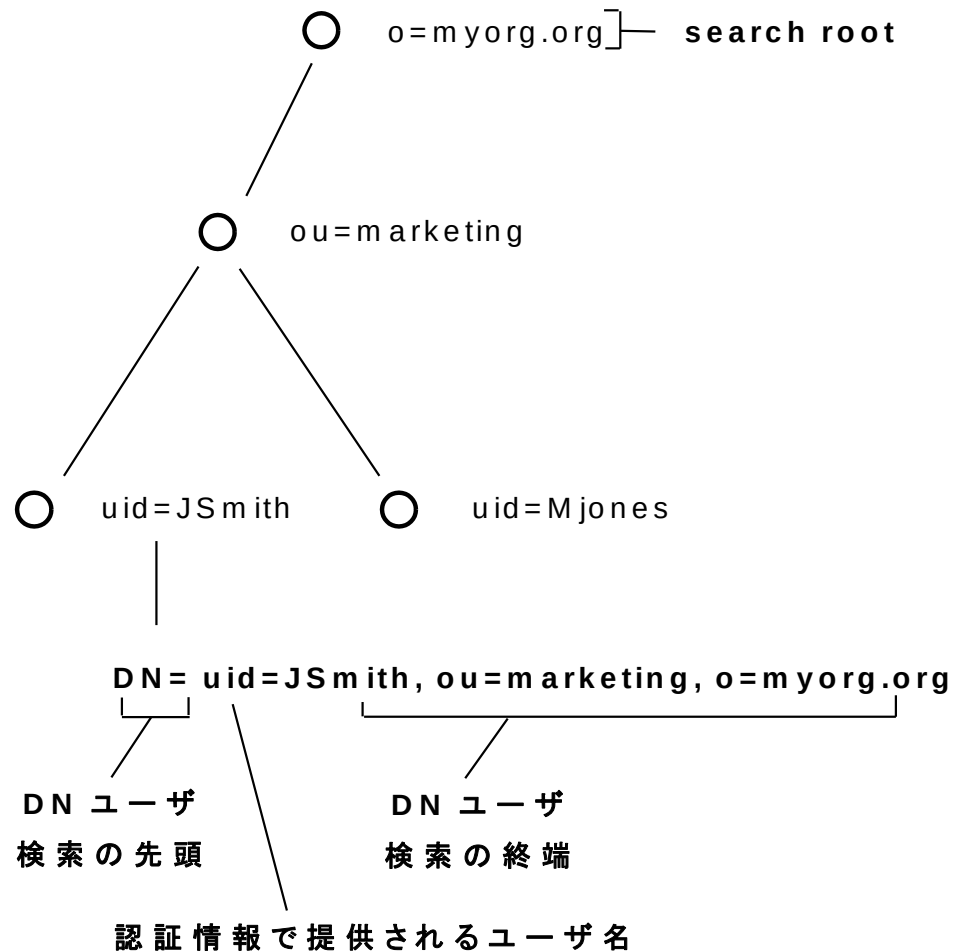
DN によるユーザの検索

[LDAP 設定]領域の[ユーザの検索]グループ ボックスにある[ユーザ ディレクトリ]ウィンドウから、DN によるユーザの検索を構成できます。[ユーザ DN 検索の先頭文字列]で指定された値、ログイン中にユーザが指定したユーザ名、および[ユーザ DN 検索の終端文字列]フィールドで指定された値を連結します。

この結果、DN は以下ようになります。

<ユーザ DN 検索の先頭文字列フィールドの値>、<ユーザ名>、<ユーザ DN 検索の終端文字列フィールドの値>

次に、LDAP ディレクトリ情報ツリー (DIT) の例を示します。



前の図では、LDAP DIT の設計に `uid=JSmith,ou=marketing,o=myorg.org` というフォームの DN が必要となります。

- [ユーザの検索]グループ ボックスの[先頭]プロパティは `uid=` になります。
- [ユーザの検索]グループ ボックスの[終端]プロパティは `ou=marketing, o=myorg.org` になります。

ユーザがログイン時に認証情報で指定する必要があるのは、一意な部分である `JSmith` の値だけです。

検索式によるユーザの検索

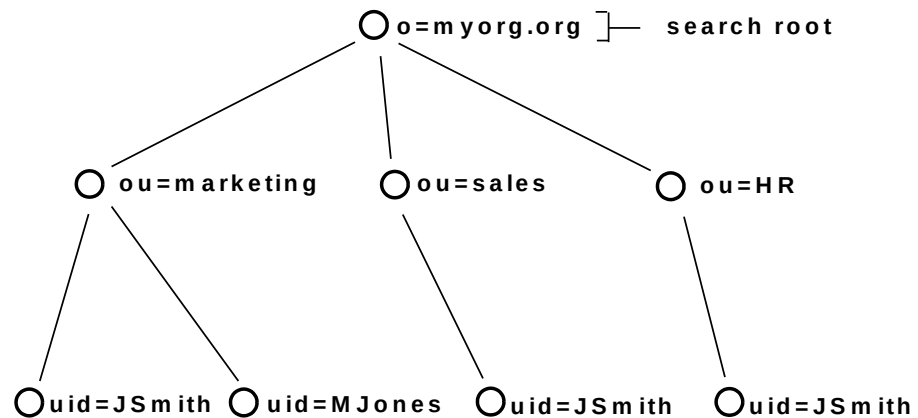
LDAP ディレクトリ サーバの複雑な DIT 内に多数のユーザが含まれている場合があるため、ユーザ ディレクトリ接続をそれに合わせて多数作成するのは実用的ではありません。部署が何百もある企業では、エンド ユーザに詳細な文字列を入力させてログインさせたくない場合があります。このような場合は、代わりに [LDAP ユーザ DN の検索] グループボックスの [先頭文字列] プロパティと [終端文字列] プロパティを用いた LDAP 検索式を定義して、共通ルートを示すユーザ ディレクトリ接続を 1 つ作成することができます。検索式の結果は、ポリシー サーバが認証を試みるユーザ DN のリストになります。

例: ユーザ DN の検索に使用する検索式

多くの部署にまたがってユーザを検索するには、ユーザ検索の [先頭] プロパティに「(&(objectclass=inetOrgPerson)(uid=」、ユーザ検索の [終端] プロパティに「)）」と定義します。ユーザ証明で指定する必要があるのは、一意である uid の値だけです。前の図に示した [ユーザ ディレクトリ] ダイアログ ボックスのセクションでは、これらの値が [LDAP ユーザ DN の検索] グループ ボックスの値に置き換えられます。

注: InetOrgPerson は、LDAP ディレクトリで使用される共通オブジェクト クラスです。

検索式の呼び出しに役立つ、LDAP DIT のタイプに関する以下の図を参照してください。



ユーザ名

JSmith
JSmith
JSmith

識別名

uid=JSmith,ou=marketing,o=myorg.org
uid=JSmith,ou=sales,o=myorg.org
uid=JSmith,ou=HR,o=myorg.org

この例において ou=sales の JSmith がリソースにアクセスする場合、JSmith は自分の名前だけをクレデンシャルに使用して認証を受けることができます (DN 文字列全体は必要ありません)。[LDAP ユーザ DN の検索]グループボックスの [先頭文字列]フィールドと[終端文字列]フィールドの間に uid= 属性と各フィールドに応じた検索式を用いることによって、LDAP クエリ (&(objectclass=inetOrgPerson)(uid=JSmith)) と一致する DN がすべて検索されます。

この後、ポリシー サーバは DN リストを取得し、このリストから DN を選択して保護されたリソースに対するアクセス権を付与します。そのリソースにアクセスできるのが ou=sales の JSmith だけであれば、認証されるのは、DN uid=JSmith,ou=sales,o=myorg.org のユーザ名、パスワードだけになります。

LDAP 検索フィルタ

ポリシー サーバで LDAP ディレクトリ接続を使用して作業する場合、LDAP 検索式用のフィルタを指定する必要がある場合があります。次の表に、一般的な LDAP 検索フィルタについての簡単に説明を示します。

検索フィルタ	形式	説明
完全一致	attribute=value たとえば、jsmith というユーザ ID を持つユーザを検索する場合、検索フィルタは uid=jsmith となります。	このフィルタは、LDAP ディレクトリの属性から特定の値を検索します。
文字列一致	attribute=*value、または attribute=value*、または attribute=val*ue、または attribute=*value* たとえば、uid=*smith を指定すると、smith という文字列で終わるすべての値 (jsmith、msmith など) が検出されます。uid=*smith* を指定すると、jsmith、msmith、bsmithe などの値が検出されます。	LDAP 検索フィルタではワイルドカードを使用できます。これによって、文字列の一部分から属性値を検索することができます。文字列の一部分が一致する値をすべて検索するには、ワイルドカード文字 (*) を使用します。
≧ (以上)	attribute>=value たとえば、ディレクトリに含まれている 21 歳以上のユーザをすべて検索する場合、検索フィルタの一部を age>=21 とします。	このフィルタは、指定された値に等しいか、指定された値より大きい値を検索します。
≦ (以下)	attribute<=value たとえば、ディレクトリに含まれている 21 歳以下のユーザをすべて検索する場合、検索フィルタの一部を age<=21 とします。	このフィルタは、指定された値に等しいか、指定された値より小さい値を検索します。

検索フィルタ	形式	説明
> (より大きい)	<code>(!(attribute<=value))</code> たとえば、ディレクトリに含まれている 21 歳より年上のユーザをすべて検索する場合は、検索フィルタの一部を <code>(!(age<=21))</code> とします。	LDAP では、式に大なり (>) を使用できません。LDAP 属性値を大なりでフィルタリングするには、式に大なりイコールと否定演算子 "!" を組み合わせて使用する必要があります。
< (より小さい)	<code>(!(attribute>=value))</code> たとえば、ディレクトリに含まれている 21 歳未満のユーザをすべて検索する場合は、検索フィルタの一部を <code>(!(age>=21))</code> とします。	LDAP では、式に小なり (<) を使用できません。LDAP 属性値を小なりでフィルタリングするには、式に小なりイコールと否定演算子 "!" を組み合わせて使用する必要があります。
近似	<code>attribute~value</code> たとえば、フィルタ <code>uid~=smith</code> では、 <code>smithe</code> や <code>smitt</code> などの値が返されます。	このフィルタは、フィルタに指定された値の近似値を検索します。
存在確認	<code>attribute=*</code> たとえば、 <code>email=*</code> と指定すると、電子メール アドレスを持つすべてのユーザが検索されます。	このフィルタは、属性が存在するかどうかを調べます。
複合 フィルタ	<code>Filter1 と Filter2 の共通集合 (&(filter1)(filter2))</code> <code>Filter1 と Filter2 の和集合 ((filter1)(filter2))</code> <code>Filter1 を満たすが、Filter2 を満たさない (&(filter1)(!(filter2))</code> 注記 複合フィルタ全体とその中に含まれる各フィルタは、それぞれ括弧で囲む必要があります。 たとえば、文字 <code>s</code> で始まるユーザ ID を持ち、21 歳以上のユーザをすべて検索する場合は、 <code>(&(uid=s*)(!(age<=21)))</code> というフィルタを使用します。	複合検索フィルタを作成します。

オブジェクトクラス検索

LDAP テーブルの各エントリには、オブジェクトクラス属性が 1 つ以上含まれています。存在フィルタをオブジェクトクラス属性と組み合わせて使用すると、LDAP ユーザディレクトリに対する検索フィルタを作成できます。SiteMinder 環境では、オブジェクト クラス属性が以下のような場合に最も役立ちます。

- あるエントリ直下にある全エントリを一覧表示する

あるディレクトリ エントリの 1 レベル下のエントリをすべて取得するには、検索範囲に [1 レベル] を指定し、検索フィルタ (`objectclass=*`) を使用します。どの LDAP ディレクトリ エントリにも必ず 1 つ以上の `objectclass` 属性が含まれているので、この検索フィルタを使用するとルートの下にあるエントリの完全なリストが返されます。

- サブツリー内の全エントリを一覧表示する

ディレクトリエントリの下にあるブランチの全エントリを取得するには、検索範囲に [サブツリー] を指定し、検索フィルタ (`objectclass=*`) を使用します。この検索フィルタを使用すると、サブツリー全体に含まれているエントリの完全なリストが返されます。

ユーザ ID でフィルタ処理される文字

SiteMinder には、LDAP 検索フィルタが LDAP 標準 (RFC) に適合するかどうか分析する LDAP 検索フィルタ チェック機能があります。

デフォルトで、ユーザのログイン ID はすべて、LDAP ユーザストアと照合する前に "("、")"、"\ " の文字が含まれているかどうかチェックされます。このチェックを無効にするには、次の `EnableSearchFilterCheck` 登録値を 0 に設定します。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\Ds\LDAPProvider\EnableSearchFilterCheck
```

重要: このチェックを無効にすると、システムが脆弱化され、攻撃を受ける危険が高くなります。したがって、ユーザ ID にこれらの文字を使用することを禁止することをお勧めします。

LDAP リフェラル

ポリシー サーバでは、LDAP ユーザディレクトリと接続するために LDAP リフェラルをサポートしています。あるディレクトリ内の LDAP リフェラルから、異なる LDAP ディレクトリの場所が提示されます。LDAP リフェラルには、書き込みリフェラルと読み込みリフェラルの 2 種類があります。さらに、ポリシー サーバは、拡張リフェラル処理もサポートしています。

注: マルチ ポリシー ストア環境で LDAP リフェラルを正常に機能させるためには、SiteMinder の LDAP ポリシー ストア スキーマをすべてのレプリカに適用する必要があります。詳細については、「ポリシー サーバ インストール ガイド」の LDAP ポリシー ストアのインストールに関するセクションを参照してください。

書き込みリフェラル

マスタおよびスレーブ LDAP ディレクトリを含むディレクトリの場合、LDAP 書き込みリフェラル機能により、マスタディレクトリを更新してからスレーブディレクトリに複製することができます。SiteMinder 展開では、スレーブ LDAP ディレクトリへの接続を指定できます。LDAP ディレクトリにデータを書き込むことを必要とする SiteMinder の機能のいずれかを使用すると、SiteMinder は、マスタ LDAP ディレクトリを指すリフェラルを自動的に検出します。SiteMinder が LDAP ディレクトリに書き込む情報は、マスタ LDAP ディレクトリに格納され、ネットワークリソースのレプリケーション方式に従って、スレーブ LDAP ディレクトリに複製されます。

読み込みリフェラル

大規模な LDAP ディレクトリでは、情報がいくつかの LDAP ディレクトリに分割して格納されている可能性もあります。たとえば、あるディレクトリにユーザ認証に必要なユーザ情報が格納されており、別のディレクトリにその他の重要なユーザ属性が格納されているとします。このとき、ユーザ属性が収められたディレクトリを示すように認証ディレクトリを設定することができます。このプロセスを読み込みリフェラルと呼びます。読み込みリフェラルが格納された LDAP ディレクトリに対するディレクトリ接続があると、SiteMinder は、その読み込みリフェラルを使用して、関連ディレクトリから情報を取得することができます。

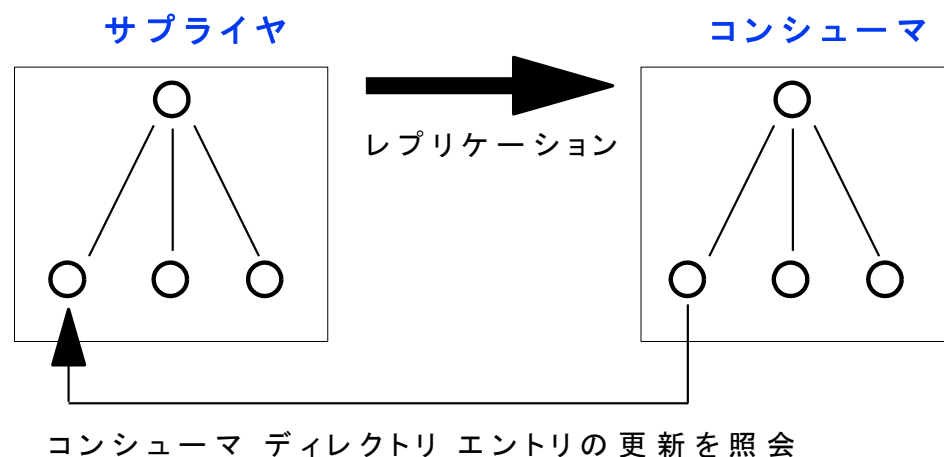
ディレクトリトポロジと LDAP リフェラル

LDAP ディレクトリのトポロジは、複数の物理サーバにわたるディレクトリツリーの区分を表します。ディレクトリツリーの論理的な区分は、パーティションと呼ばれます。LDAP のディレクトリトポロジは、LDAP 環境によって大きく異なります。ただし、パーティション間でリフェラルを使用することで、トポロジに関係なく、ディレクトリが 1 つのサービスとして機能することが可能になります。

ディレクトリトポロジには、3 種類の LDAP リフェラルを使用できます。これらの 3 種類のリフェラルを併用すれば、非常に複雑なディレクトリ構造を作成することができます。使用できるリフェラルの種類は次のとおりです。

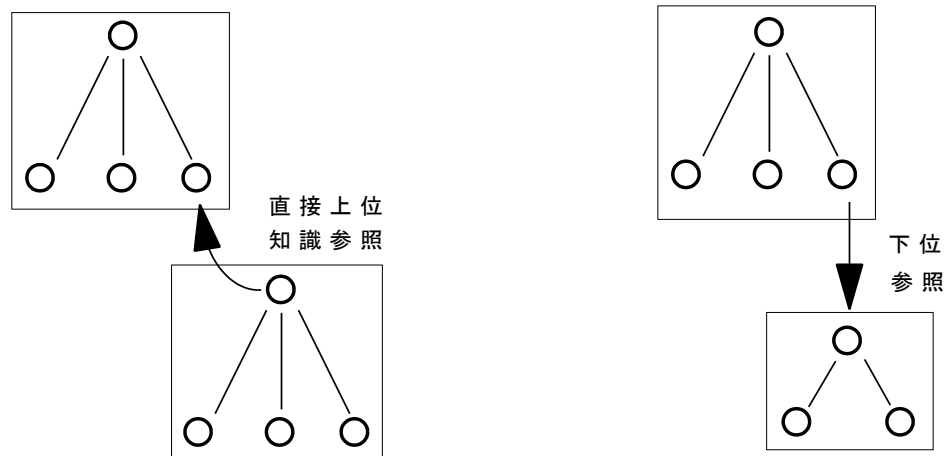
レプリケーション アグリーメント

ディレクトリ トポロジにレプリケーション アグリーメントが含まれている場合は、サプライヤ ディレクトリ内のすべての変更が下位のコンシューマ ディレクトリにレプリケート(複製)されます。コンシューマ ディレクトリとサプライヤ ディレクトリを使用して、リクエストの負荷分散や、ディレクトリ間でのフェイルオーバー関係の確立が可能になります。更新リクエストを受け取ると、コンシューマ ディレクトリはそのリクエストをサプライヤ ディレクトリに照会し、サプライヤ ディレクトリで更新が行われます。これはごく一般的な LDAP リフェラルです。



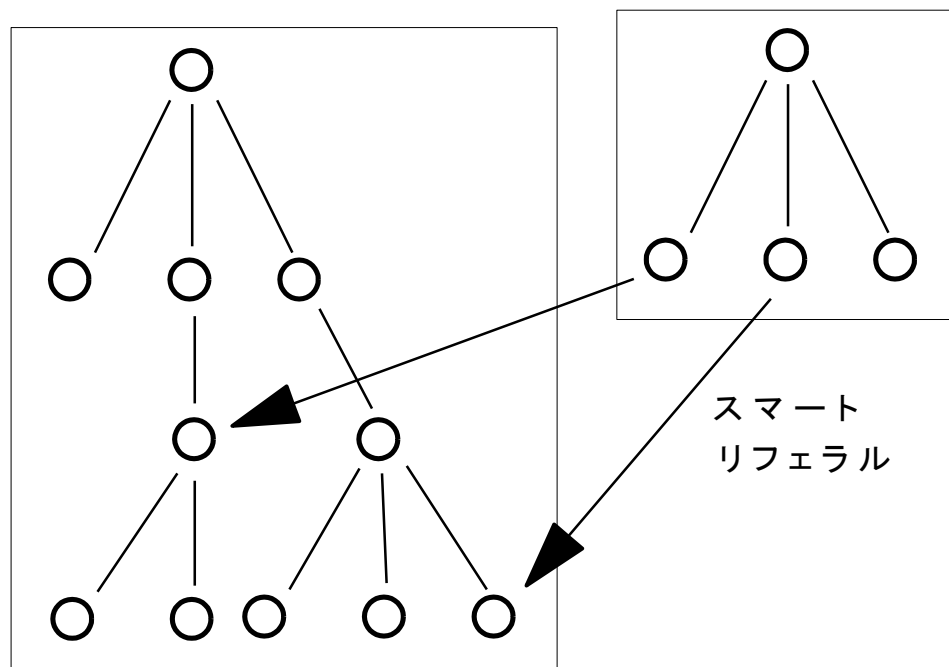
知識参照

1つのディレクトリパーティションから別のディレクトリパーティションを指すポインタのことを、知識参照と呼びます。DITのルートの直上にあるノードを指す知識参照は、直接上位知識参照と見なされます。DITの下位にある他のパーティションを指す知識参照は、下位参照と見なされます。



スマートリフェラル

元のパーティションの直上または直下でないディレクトリの部分を指すポインタのことを、スマートリフェラルと呼びます。スマートリフェラルには、ディレクトリトポロジのいずれかにあるノードを参照できるだけの十分な情報が格納されています。



拡張 LDAP リフェラル処理

ポリシー サーバの LDAP リフェラル処理機能が強化され、パフォーマンスと冗長性が改善されました。ポリシー サーバの以前のバージョンでは、LDAP SDK レイヤを通じて LDAP リフェラルの自動処理がサポートされていました。LDAP リフェラルが発生すると、これまでは LDAP SDK 層が、参照先サーバへのリクエストの実行を、ポリシー サーバと通信せずに処理していました。

SiteMinder は、非自動(機能強化型)LDAP リフェラル処理をサポートしています。非自動リフェラル処理では、LDAP リフェラルは、LDAP SDK 層ではなくポリシー サーバに返されます。リフェラルには、リフェラルの処理に必要なすべての情報が含まれています。ポリシー サーバは、リフェラルで指定されている LDAP ディレクトリが使用できるかどうかを調べて、該当する LDAP ディレクトリが機能していない場合は、リクエストを中断させることができます。この機能により、オフラインのシステムへの LDAP リフェラルによってリクエスト待ち時間が恒常的に増加することによるパフォーマンスの低下が解消されます。このような待ち時間の増加は、SiteMinder でリクエストの飽和状態を発生させることがあります。

たとえば、SiteMinder の展開には、サプライヤ/消費者レプリケーション方式を使用して展開され、フェイルオーバーを実行する 2 つの LDAP ディレクトリが含まれます。リクエストはすべて消費者ディレクトリに対して発行されます。消費者ディレクトリがリクエストを処理できないと、ポリシー サーバはフェイルオーバー設定を使ってサプライヤディレクトリに問い合わせを行います。

パスワードサービスが有効になっている場合、サプライヤディレクトリにもパスワードの変更が書き込まれるように、消費者ディレクトリで LDAP リフェラルが発生します。LDAP リフェラルの自動処理のみがサポートされていたポリシー サーバの以前のバージョンでは、サプライヤディレクトリが使用不可な場合には、新しいパスワード情報をサプライヤディレクトリに書き込む必要がある消費者ディレクトリからのリフェラルが機能していないことをポリシー サーバが認識できず、サプライヤからの応答を待っていました。このような処理の遅れが続くと、システムに処理待ちのリクエストが蓄積されていきます。

拡張 LDAP リフェラル処理機能を指定してポリシー サーバを設定すると、ポリシー サーバはサプライヤ LDAP ディレクトリが使用不可な状態にあることを認識し、サプライヤサーバがパスワード変更を記録できるようになるまで、パスワードの変更が必要なリクエストを自動的に終了します。

機能強化された LDAP リフェラル処理の設定については、「ポリシー サーバ管理ガイド」を参照してください。

ポリシー サーバが LDAP ユーザ ストアにバインドする方法

ポリシー サーバは、LDAP ユーザ ストアに接続するときに 3 つの接続を開きます。

- 最初の接続では、ユーザ ストアが稼働していることを確認します。デフォルトでは、ポリシー サーバはこの接続で 30 秒ごとにユーザ ストアに Ping を発行します。
- 2 つ目の接続は、検索と更新に使用されます。たとえば、ポリシー サーバは、バインドに失敗した場合に、ユーザの検索と属性の設定にこの接続を使用します。
- 3 つ目の接続は、クレデンシャルのテストに使用されます。ポリシー サーバは、ユーザのクレデンシャルを使用して、ユーザ ストアにバインドを試みます。バインドの結果によって、ユーザのクレデンシャルが承諾されるか、拒否されるかが識別されます。

ODBC データベースの概要

SiteMinder は、認証や許可のユーザ ディレクトリとして、ODBC 対応データベースの独自スキーマを使用することができます。このオプションは、ユーザ名、パスワード、グループ メンバーシップなどのユーザ情報が ODBC データベースに格納されているサイトで役立ちます。このようなサイトでは、この機能によってポリシー サーバが独自のデータベーススキーマをユーザ ディレクトリとみなすことができます。

ポリシー サーバは、次のタイプの ODBC 対応データベースとの接続をサポートしています。

- Microsoft SQL Server – Windows ポリシー サーバのみ
- Oracle RDBMS

注: ユーザのユーザ ディレクトリ データが Oracle データベースに格納されている場合、ODBC ではなく OCI を使用してユーザ ディレクトリに接続することをお勧めします。

対応するデータベースのバージョンに関する最新情報は、CA のサポートサイトで確認してください。

ユーザ ディレクトリとしてデータベースを使用するようにポリシー サーバを設定するには、次の手順に従います。

- SiteMinder SQL クエリ方式ウィンドウのフィールドに SQL クエリ情報を入力します。このウィンドウには、SiteMinder ユーザ ディレクトリウィンドウからアクセスできます。ポリシー サーバに必要な情報を ODBC 対応データベースのフィールドにマッピングするためのフィールドが表示されます。

注: クエリ方式が異なる場合は同じデータソースを使用しないでください。クエリ方式ごとに一意のデータソースを作成する必要があります。

- ユーザ情報が収められているデータベースを指すように、ODBC データソースを定義します。

注: ODBC データソースは「System DSN」として設定します。このデータソースは Microsoft SQL Server データベースまたは Oracle データベースを指している必要があります。ODBC データソースの設定については、使用している Windows オペレーティングシステムのマニュアルを参照してください。

Windows ディレクトリの概要

次の Windows NT 環境でユーザ ディレクトリ接続を設定できます。

- Windows NT ドメイン
- ドメイン内の各 Windows NT コンピュータ
- スタンドアロン Windows NT コンピュータ

ポリシー サーバが Windows NT ドメインに接続するには、次の要件を満たしている必要があります。

- ポリシー サーバが含まれるドメインとユーザが含まれるドメインの間に単方向の信頼関係が確立されている必要があります。
- SiteMinder で認証されるドメイン内のすべてのユーザに、ポリシー サーバが稼動するマシンに対する「ネットワーク経由でコンピュータへアクセス」ユーザ権限が必要です。
- SiteMinder がユーザ ディレクトリ オブジェクトへのアクセスに使用するアカウントは、WinNT ドメイン ユーザが存在するドメイン コントローラ マシンの IPC\$ 共有にアクセスする必要があります。

注: WinNT ドメインのデフォルト設定を使用すると、デフォルトでこれらの要件が満たされることがあります。Windows NT ドメイン管理者は、ドメインが上記の要件を満たしているかどうかを確認する必要があります。

ポリシー サーバは Windows NT に対して認証を行い、各個人の識別情報やグループ メンバーシップに基づいてユーザを許可します。

Windows NT ネームスペースに対する認証を行うとき、ポリシー サーバは認証用にユーザ証明を Windows NT に渡します。この証明が、そのユーザの Windows NT ユーザ名とパスワードです。SiteMinder 環境に複数の WinNT ネームスペースが定義されているときに、SiteMinder に提供するユーザ名にドメイン名が含まれていると(つまり、*domain\username*)、ユーザ認証にかかる時間は短縮されます。この場合、SiteMinder は、指定したドメイン名に一致しないすべての WinNT ネームスペースをスキップします。

Windows NT ユーザ名とパスワードをクレデンシャルとして使用

注: WinNT ドメインに対してユーザを認証するには、ポリシー サーバが WinNT 上で稼働している必要があります。

Active Directory の概要

SiteMinder は、Microsoft Active Directory プラットフォーム上のユーザ ディレクトリをサポートしています。管理 UI を使用した Active Directory (AD) ネームスペースと LDAP ネームスペースの設定はほとんど同じですが、機能面で異なる点があります。

Active Directory ユーザストアに AD ネームスペースを使用すると、次のようなメリットがあります。

- Windows ネイティブな証明データベースを使用する SSL 接続

注: ポリシー サーバと Active Directory ユーザ ストアをホスティングするシステムを信頼できるものとして確立する必要があります。Windows システムと Active Directory を SSL 接続用に設定する方法についての詳細は、Windows のマニュアルを参照してください。

- 安全な LDAP バインド操作が可能な Windows ネイティブ SASL のサポート

ただし、次のようなデメリットもあります。

- 拡張 LDAP リフェラル処理はサポートされない。
- LDAP のページング操作とソート操作はサポートされない。

注: Active Directory ユーザ ストアに LDAP ネームスペースを使用する方法についての詳細は、「[Active Directory 接続の設定](#) (P. 209)」を参照してください。

カスタム ディレクトリの概要

管理 UI では、SiteMinder ディレクトリ API (Software Development Kit で別途使用可能)を使用して共有ライブラリを作成することによって、カスタムのユーザ ディレクトリ接続を作成することができます (Software Development Kit がインストールされている場合、詳細については「*API Reference Guide for C*」を参照してください)。カスタム接続を通じて、ポリシー サーバとレガシー ディレクトリの対話が可能になります。SiteMinder API を使用してディレクトリ接続を作成した後、[ユーザ ディレクトリ] ペインでカスタム ネームスペースを設定することができます。

ディレクトリ属性の概要

SiteMinder 機能の一部では、7 つの SiteMinder 属性への読み取りまたは読み書きアクセス権が必要とされます。これらの属性の値は、ポリシー サーバに接続されたユーザ ディレクトリに格納されています。ポリシー サーバからユーザ ディレクトリへの接続を設定する場合、そのユーザ ディレクトリ内で、7 つの SiteMinder 属性に対応するユーザ属性の名前を指定する必要があります。これは、[ユーザ属性]グループ ボックスのフィールドで行います。

たとえば、「ユニバーサル ID」の名前は、あるユーザ ディレクトリでは「学生 ID」であり、他のディレクトリでは「アカウント番号」である場合があります。ディレクトリ接続が設定されると、SiteMinder はユニバーサル ID を見つけるたびに、選択したユーザ ディレクトリ内の正しいユーザ属性にアクセスできます。

SiteMinder のこの機能は、ユーザ属性マッピングによって拡張できます。ユーザ属性マッピングにより独自の共通名を定義でき、それぞれの名前を異なる方式の複数のユーザ ディレクトリ内のユーザ属性名にマッピングできます。

SiteMinder 属性はそれぞれデータ型および 1 つ以上のディレクトリタイプに関連付けられています。以下の表で各 SiteMinder 属性を説明します。(R)は、属性が読み取りアクセスを必要とすることを示します。(RW)は、属性が読み取り/書き込みアクセスを必要とすることを示します。

属性名	データのタイプ	ディレクトリタイプ	説明
ユニバーサル ID (R)	文字列	LDAP データベース Windows NT	ユーザの ID を管理するために保護されているアプリケーションに、SiteMinder により渡されるユニバーサル ID またはユーザ識別子を指定します。この機能は SiteMinder とレガシーアプリケーションの間のブリッジであり、属性を使用してユーザを識別することが多くあります。ユニバーサル ID は、ディレクトリマッピングの設定にも使用されます。
無効フラグ (RW)	文字列	LDAP データベース	ユーザのアカウントステータスを指定します。より詳細な情報については、「ポリシー サーバ管理ガイド」を参照してください。

属性名	データのタイプ	ディレクトリタイプ	説明
パスワード属性 (RW)	バイナリ	LDAP データベース	ユーザのパスワードを指定します。
パスワード データ (RW)	バイナリ	LDAP データベース	パスワード ポリシー情報を追跡するのに使用します。
匿名 ID (RW)	文字列	LDAP データベース	匿名の認証方式を使用して認証されたユーザの DN を格納します。
電子メール (R)	文字列	LDAP データベース	この属性は現在 SiteMinder の機能で使用されていません。
チャレンジ/レスポンス (RW)	文字列	LDAP	パスワード サービスおよび DMS の「忘れたパスワード」機能で使用する質問と答えの組み合わせを指定します。チャレンジの文字列は、ユーザに渡されるパスワードのヒントです。

注: ユーザ ディレクトリへの接続の設定時には、ポリシー サーバがディレクトリにアクセスするときに使用する管理者認証情報を指定できます。これらのクレデンシャルには、テーブル内の対応するユーザ属性と同じ読み取り/書き込みアクセス権が必要です。

CA Directory ユーザ ディレクトリ接続を設定する方法

ユーザストアとして、CA Directory ユーザ ディレクトリを使用できます。以下に、ポリシー サーバへのユーザストア接続を作成するための手順を示します。

1. ユーザストアシステムに Ping を発行します。
2. CA Directory ユーザ ディレクトリ接続を設定
3. (オプション) CA Directory ユーザストアのキャッシュを有効にする
4. (オプション) CA Directory キャッシュ設定を確認する

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

CA Directory ユーザ ディレクトリ接続の設定

ポリシー サーバが CA Directory ユーザ ストアと通信するように、ユーザ ディレクトリ接続を設定することができます。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は FIPS 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。
6. (省略可) [管理者クレデンシャル]グループ ボックスで[クレデンシャルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可) [ユーザ属性]グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。

8. (省略可) [属性マッピング リスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
9. [サブミット] をクリックします。
ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

ユーザ ストア DSA パラメータの有効化

SiteMinder が Sun Java System LDAP SDK を使用することで、クライアントは、ディレクトリ サーバへの 1 つの管理接続を開き、その接続下でユーザ バインドを実行することができます。CA Directory をユーザ ストアとして使用している場合、ポリシー サーバは、認証要求ごとにバインド要求を実行して、CA Directory に接続します。これらの要求を処理するように CA Directory を設定します。そうしないと、CA Directory は接続不足になり、認証が失敗します。

ユーザ ストア DSA パラメータを有効にする方法

1. ユーザ ストア DSA 用の.dxc ファイルを開きます。
2. ファイルの下部で、以下を定義します。

```
#SiteMinder
set mimic-netscape-for-siteminder = true;
set concurrent-bind-user = DN;
set hold-ldap-connections = true;
```
3. .dxc ファイルを保存して閉じます。
これで、ユーザ ストア DSA パラメータが有効になります。
注: DN は x500 形式です。
例: <o acme><cn smadmin>

CA Directory ユーザ ストアのキャッシュの有効化

CA Directory DXcache 機能を有効にすることにより、SiteMinder による大規模なユーザ ストアの認証および認可のパフォーマンスを向上させることができます。5 MB のユーザ ストアは、大規模であると見なされます。

キャッシュを有効にする方法

1. dsa ユーザとして、ユーザ ストア DSA の DXI ファイル (たとえば、`<dxserver_install>\config\servers\eTrustDsa.dxi`) を編集し、ファイルの末尾に以下の行を追加します。

```
# 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;
```

注: max-cache-size エントリは MB 単位の合計キャッシュ サイズです。CA Directory サーバで使用可能な合計メモリとユーザ ストアの全体的なサイズに基づいて、この値を調整します。さらに、cache-index フィールドを、SiteMinder がユーザ ストアでユーザ検索を実行する場合に使用するフィールドに設定します。たとえば、共通名 (cn=*) に基づいてユーザの認証および許可を行う場合は、cache-index に commonName が設定されていることを確認します。

2. dsa ユーザとして、ユーザ DSA を停止してから再起動し、DXcache の設定変更を有効にします。

```
dxserver stop eTrustDsa
dxserver start eTrustDsa
```

CA ディレクトリ キャッシュ設定の確認

ユーザ ストアの CA DXcache 機能を設定したら、DXmanager ユーザ インターフェイスを使用して、キャッシュが有効であることを確認できます。

キャッシュを確認する方法

1. Web ブラウザを使用して、CA DXmanager Web インターフェイスに接続します。

以下に例を示します。

```
http://<CA_host>:8080/dxmanager/ManagerServlet?hostgroup=All
```

2. DSA 設定ページに移動し、ポリシー ストア DSA に対する DXcache ステータスフィールドが[有効]に設定されていることを確認します。

Sun Java System ユーザ ディレクトリ接続を設定する方法

Sun Java System ユーザ ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. Sun Java ユーザ ディレクトリ接続を設定します。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Sun Java System ユーザ ディレクトリ接続の設定

ポリシー サーバが Sun Java System ユーザ ストアと通信するように、ユーザ ディレクトリ接続を設定することができます。

Sun Microsystems では、cn=Directory Manager 以外の管理者アカウントを使用することを推奨しています。cn=Directory Manager アカウントを使用すると、このアカウントに適用されるセキュリティポリシーによってパフォーマンスが低下することがあります。代わりに、ディレクトリを管理することのできる十分な権限を持つ新しいユーザを作成して、[ユーザ名]フィールドにそのユーザを指定します。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。

[LDAP 設定]が開きます。

4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリストアが使用する証明書は FIPS 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。
6. (省略可)[管理者クレデンシヤル]グループ ボックスで[クレデンシヤルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。
[属性マッピングの作成]ペインが開きます。
9. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

IBM Directory Server ユーザ ディレクトリ接続を設定する方法

IBM Directory Server をユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. IBM Directory Server ユーザ ディレクトリ接続を設定します。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシヤルを提示することが必要になる場合もあります。

IBM Directory Server ユーザ ディレクトリ接続の設定

ポリシー サーバが IBM Directory Server ユーザ ストアと通信するように、ユーザ ディレクトリ接続を設定することができます。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注:ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は FIPS 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。
6. (省略可)[管理者クレデンシャル]グループ ボックスで[クレデンシャルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。

[属性マッピングの作成]ペインが開きます。

9. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Domino ユーザ ディレクトリをユーザ ストアとして設定する方法

Domino ユーザ ディレクトリをユーザ ストアとして設定する方法は、3 段階のプロセスです。

1. Domino のユーザ ディレクトリがポリシー サーバの要件を満たすことを確認
2. ユーザ ストア システムに Ping を発行します。
3. ポリシー サーバから Domino ユーザ ストアへの接続を設定します。

Domino のユーザ ディレクトリがポリシー サーバの要件を満たすことを確認

Domino のユーザ ディレクトリは、LDAP ディレクトリです。Domino ユーザ ディレクトリをユーザ ストアに設定する前に、以下のポリシー サーバ要件を満たすことを確認します。

- Domino ユーザ ディレクトリはバージョン 5.02 以上である必要があります。
- Domino ユーザ グループは、ポリシー サーバから Domino ユーザ ストアへの接続を設定する場合に指定するルート DN を共有する必要があります。

例: Lotus Notes においてアドレス帳 (names.nsf) にグループ *marketing/myorg.org* を追加する場合、[ユーザ ディレクトリ] ウィンドウの [ルート] フィールドに「*o=myorg.org*」と入力する必要があります。

- 新規ユーザはそれぞれ、Domino ユーザ ディレクトリ内にユーザ名エントリおよびインターネット パスワード エントリの両方を保有する必要があります。

注: Domino ユーザ ディレクトリにユーザを追加する際に、そのユーザを登録することをお勧めします。この追加のステップにより、Domino ユーザ ディレクトリに重複するユーザ名エントリが含まれないようにします。ディレクトリに複数のエントリがある場合、ポリシー サーバは最初のエントリを使用します。他のユーザ名でログインしようとすると、失敗します。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Domino ディレクトリ接続の設定

ポリシー サーバが **Domino** ユーザ ストアと通信するように、ユーザ ディレクトリ 接続を設定することができます。

注: 以下の手順では、新しいオブジェクトを作成しているとしします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが **FIPS** モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な **SSL** 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は **FIPS** 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。

注: [ルート]フィールドに指定する値は、**Lotus Notes** で割り当てた組織名と一致する必要があります。**Lotus Notes** で国を指定した場合は、[ルート]フィールドにも国を指定する必要があります。

例:「myorg」という組織があります。この組織の所在地は米国です。[検索ルート]は、o=myorg,c=us と指定されています。

注: [ユーザ DN 検索の先頭文字列]フィールドおよび[ユーザ DN 検索の終端文字列]フィールドに指定する検索文字列は、**Lotus Notes** の省略形式ではなく、正式な **LDAP** 形式に従う必要があります。検索文字列の詳細については、「**LDAP Search Filters**」を参照してください。

6. (省略可) [設定] をクリックして、ロード バランシングとフェイルオーバーを設定します。

注: ロード バランシングとフェイルオーバーの詳細については、「LDAP ロード バランシングおよびフェイルオーバー」を参照してください。

7. (省略可) [管理者クレデンシャル] グループ ボックスで [クレデンシャルが必要] を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
8. (省略可) [ユーザ属性] グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
9. (省略可) [属性マッピング リスト] グループ ボックスで [作成] をクリックします。

[属性マッピングの作成] ペインが開きます。

10. [サブミット] をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Novell eDirectory LDAP ディレクトリ接続を設定する方法

Novell eDirectory LDAP ユーザ ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. NetWare の設定
2. Novell eDirectory における匿名 LDAP アクセスの設定

または

特定の SiteMinder 管理者のアクセス権を作成します。

- SiteMinder 管理者の特別なアクセス権
- SiteMinder 管理用の Novell eDirectory ユーザ アカウントを作成します。

3. ユーザ ストア システムに Ping を発行します。
4. Novell eDirectory LDAP ディレクトリ接続を設定します。

NetWare の設定

このセクションで説明する設定の目的は、ポリシー サーバが **Novell eDirectory** にログインしてディレクトリの内容を確認し、ディレクトリ属性を取得することです。**SiteMinder** の一部の拡張機能では、ポリシー サーバにディレクトリへの書き込み権限を付与するように **Novell eDirectory** を設定することも必要になる場合があります。

Novell eDirectory のインストールの一環として **LDAP** をインストールした場合は、**Novell eDirectory** に **LDAP Server** というサーバと、**LDAP Group** という **LDAP** グループがあります。**LDAP Server** は、**LDAP Group** のメンバーである必要があります。

Novell eDirectory に LDAP サーバと LDAP グループを作成する方法

1. **Novell eDirectory** に **LDAP** サーバを作成します（この例では、**LDAP Server** という名前です）。
2. **Novell eDirectory** に **LDAP** グループを作成します（この例では、**LDAP Group** という名前です）。
3. **LDAP Server** に **LDAP Group** を割り当てます。
 - a. **NetWare Admin** ツールで **LDAP Server** をクリックします。

注： **Novell eDirectory** の変更には、**NW Admin** ツールではなく **Netware ConsoleOne** ツールを使用している場合は、**ConsoleOne** で使用可能なツールを使用して同じタスクを完了する必要があります。両方のツールのインタフェースは似ています。詳細については、**Novell** のマニュアルを参照してください。
 - b. ポップアップ メニューから、**[Details]** を選択します。
 - c. **[LDAP Group]** フィールドに、「**LDAP Group**」と入力します。
 - d. **[OK]** をクリックします。

Novell eDirectory における匿名 LDAP アクセスの設定

ポリシー サーバが **Novell eDirectory** と通信するには、そのディレクトリへのアクセスに必要な管理権限を持つアカウントを作成する必要があります。

これには、**LDAP** サーバ上に匿名ユーザを生成して、このユーザをプロキシユーザにする方法が最も簡単です。**LDAP** サーバ上で **SiteMinder** に必要な機能をすべて実行できるように、ユーザには十分な権限を割り当てます。

次に、匿名ユーザに管理権限を割り当てる方法を説明しますが、これよりもユーザに割り当てる権限を制限することも可能です。これによって、LDAP ディレクトリに対するすべての匿名アクセスに、SiteMinder に付与したのと同じ権限を持たせることができます。

匿名の LDAP アクセスを設定する方法

1. LDAP_Anonymous というユーザを作成します。

次の手順は、使用している Novell 製品のバージョンによって異なる場合があります。

- a. NetWare Admin ツールのメニューバーで、[Object]、[Create]、[User] の順に選択します。
- b. LDAP_Anonymous という名前を追加します。
- c. パスワードは割り当てません。
- d. 右側のフレームで[Security Equal To]を選択して、管理者ユーザ (Admin.transpolar など)を追加します。
- e. [OK]をクリックします。

2. 以下のとおり、プロキシアカウントを設定します。

次の手順は、使用している Novell 製品のバージョンによって異なる場合があります。

- a. NetWare Admin ツールで[LDAP Group]を選択します。
- b. ポップアップメニューから、[Details]を選択します。
- c. [Continue]をクリックします。
- d. [Proxy Username]フィールドに「LDAP_Anonymous」と入力します。
- e. 右側のフレームで[Access Control]を選択して、[Add]をクリックします。
- f. [LDAP ACL Name]フィールドに「LDAP_Anonymous」と入力します。
- g. [LDAP Distinguished Name]チェックボックスをオンにして、「cn=LDAP_Anonymous」と入力します。
- h. [All Attributes and Object Rights]チェックボックスをオンにします。
- i. [OK]をクリックします。
- j. 右側のフレームで[Access Control]を選択して、[Add]をクリックします。
- k. [LDAP ACL Name]というラベルのボックスに、「Everyone」と入力します。
- l. [Everything]チェックボックスをオンにします。

- m. [All Attributes and Object Rights] チェックボックスをオンにします。
- n. [OK] をクリックします。
- o. [OK] をクリックします。

ポリシー サーバで Novell eDirectory を使用するための設定を継続する方法については、「[Novell eDirectory LDAP ディレクトリ接続の設定 \(P. 197\)](#)」を参照してください。

SiteMinder 管理者による特別なアクセス

以下に、ポリシー サーバのみへの特別なアクセスを許可する手順について説明します。環境によっては、こちらの設定の方が適切な場合があります。

1. SiteMinder 管理者を表す Novell eDirectory ユーザ (SiteMinder_admin など) を作成します。
2. このユーザに、SiteMinder 管理者によって生成されたパスワードを付与します。このパスワードを、管理 UI に入力します。

SiteMinder 管理用の Novell eDirectory ユーザ アカウントの作成

NW Admin ツールを使用して、SiteMinder 管理者にユーザ アカウントを付与することができます。

SiteMinder 管理用の Novell eDirectory ユーザ アカウントを作成する方法

1. NetWare Admin ツールで [LDAP Group] を右クリックします。
2. ポップアップ メニューから、[Details] を選択します。
3. 右側のパネルで [Access Control] をクリックします。
4. ACL を追加します。
5. ACL の名前を入力します。
6. [Access By List] 画面で [Add] をクリックします。

7. [Access By List]画面で、[LDAP Distinguished Name]をクリックします。
8. 「cn=SiteMinder_admin」と入力します。

デフォルトでは、アクセスレベルは読み取りに設定されます。SiteMinder の基本機能を使用するには、このアクセスレベルで十分です。アクティブな API や SiteMinder の一部の高度な機能(パスワード サービス、ユーザ無効化、登録サービスなど)を使用するユーザには、書き込み権限が必要になることもあります。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Novell eDirectory の LDAP ディレクトリ接続の設定

ポリシー サーバが Novell eDirectory ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。
[ユーザ ディレクトリの作成]ペインが表示されます。
注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は FIPS 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。

注: ユーザ ディレクトリに複数の組織が含まれる場合、[ルート]フィールドは空白のままにすることができます。これにより、ポリシー サーバは複数の組織のユーザを検索できます。

6. (省略可)[管理者クレデンシヤル]グループ ボックスで[クレデンシヤルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。
[属性マッピングの作成]ペインが開きます。
9. [サブミット]をクリックします。
ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

ADAM ユーザ ディレクトリ接続を設定する方法

ADAM ユーザ ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ADAM インスタンスおよびユーザ ストア接続の作成
2. ADAM ユーザ ストアへ接続するための SiteMinder ユーザを作成します。
3. SiteMinder ユーザに管理者の権限を割り当てます。
4. ADAM ユーザ ストアにユーザをロードします。
5. ADAM ユーザ ストア ディレクトリ接続を設定します。

ADAM インスタンスおよびユーザ ストア接続の作成

ADAM ADSI Edit の[接続の設定]ダイアログ ボックスを使用して、ADAM インスタンスおよびユーザ ストア接続を作成します。

ADAM インスタンスおよびユーザ ストア接続を作成する方法

1. 空いているポート番号に ADAM インスタンスをインストールします。この例では、ポート 390 を使用しています。
2. [スタート]-[すべてのプログラム]-[ADAM]-[ADAM ADSI Edit]を選択して ADAM ADSI ユーティリティを開きます。
3. ADAM ADSI Edit を右クリックし、[接続]を選択します。[接続の設定]ダイアログ ボックスが開きます。
4. [接続の設定]ダイアログ ボックスで以下を実行します。
 - a. [接続名]フィールドに名前を入力します。この例では、「My Connection」を使用しています。
 - b. [サーバ名]フィールドで localhost を選択します。
 - c. [ポート]フィールドに空いているポート番号を入力します。この例では、ポート 390 を使用しています。
 - d. [識別名 (DN) または名前付けコンテキスト]を選択し、フィールドに「O=userstore」を入力します。
 - e. [OK]をクリックします。

ADAM ADSI Edit において、設定した接続 (MyConnection [localhost:390]) のルート要素が表示されます。

ADAM ユーザ ストアに接続するための SiteMinder ユーザの作成

前のトピックで示した[接続の設定]ダイアログ ボックスで SiteMinder ユーザを作成できます。

ADAM ユーザ ストアに接続するための SiteMinder ユーザを作成する方法

1. 「o=userstore」の下で「CN=Roles」を右クリックし、[新規作成]-[オブジェクト]を選択します。
2. [オブジェクトの作成]ダイアログ ボックスで以下を実行します。
 - a. [user]を選択して[次へ]をクリックします。
 - b. [値]フィールドで、SiteMinder が ADAM ユーザ ストアに接続するのに使用するユーザ名を入力し、[次へ]をクリックします。この例では、admin を使用しています。
 - c. [完了]をクリックします。
3. ADAM ADSI Edit の右側の列で[CN=admin]を右クリックし、[パスワードのリセット]を選択します。
4. [パスワードのリセット]ダイアログ ボックスで新しいパスワードを入力し確認し、[OK]をクリックします。

これで、ADAM に SiteMinder ユーザが作成されました。以下のステップではこのユーザに管理者権限を割り当てます。

SiteMinder ユーザへの管理者権限の割り当て

ADAM ADSI Edit を使用して、SiteMinder ユーザに管理者権限を割り当てます。

SiteMinder ユーザに管理者権限を割り当てる方法

1. ADAM ADSI Edit の右側の列で「CN=Administrators」を右クリックし、[プロパティ]を選択します。
2. [CN=Administrators のプロパティ]ダイアログ ボックスで Member 属性までスクロールし、[編集]をクリックします。
3. [ADAM アカウントの追加]をクリックします。

4. [ADAM アカウントの追加]ダイアログ ボックスでは、「ADAM ユーザ ストアに接続するための SiteMinder ユーザの作成」で作成した SiteMinder ユーザを ADAM ユーザ ストアに追加します。
5. [OK]をクリックします。

この例では、CN=admin、CN=Role、O=userstore を使用しています。

ADAM ユーザ ストアへのユーザの読み込み

ADAM Tools コマンド プロンプトを使用して ADAM ユーザ ストアにユーザを読み込みます。

ADAM ユーザ ストアにユーザを読み込む方法

1. [スタート]-[すべてのプログラム]-[ADAM]-[ADAM Tools Command Prompt]を選択して、ADAM Tools コマンド プロンプトを起動します。
2. 以下のコマンドを実行して、ストアのユーザを含む .ldif ファイルを読み込みます。

```
ldifde -i -f user_store.ldif -s IP_address -t port_number
```

ユーザ ストア

ユーザ ストアの名前を指定します。

IP アドレス

ADAM をホストしているマシンの IP アドレスを指定します。

ポート番号

ADAM がリスンするポート番号を指定します。

ADAM ユーザ ストア ディレクトリ接続の設定

ポリシー サーバが **Active Directory** グローバル カタログ ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注:ポリシー サーバが **FIPS** モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な **SSL** 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は **FIPS** 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。
6. (省略可)[管理者クレデンシャル]グループ ボックスで[クレデンシャルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、**SiteMinder** 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。

[属性マッピングの作成]ペインが開きます。

9. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Active Directory ディレクトリ接続を設定する方法

以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. Active Directory ユーザ ディレクトリがポリシー サーバ要件を満たしていることを確認します。
2. Active Directory または LDAP ネームスペースを指定します。
3. ユーザ ストア システムに Ping を発行します。
4. ポリシー サーバから Active Directory ユーザ ストアへの接続を設定します。

Active Directory に関する考慮事項

Active Directory への接続を設定する前に考慮する点は、以下のとおりです。

Windows での展開

SiteMinder は、ユーザの完全修飾 Windows ID とパスワードを IIS に渡すことで、Windows のユーザ コンテキストを確立します。SiteMinder は、完全修飾 Windows ID をユーザの DN エントリから取得します。DN の最初の cn 値と dc 値が連結されて Windows ID が生成されます。たとえば、ユーザ DN が次のような場合

```
cn=<username>,cn=<usergroup>,dc=<server>,dc=<domain>,  
dc=<extension>
```

生成される Windows ID は、<server>\<username> です。IIS では、<username> が Windows ユーザ ID と同一で、<server> がログオンドメイン名である必要があります。

マルチバイト文字のサポート

AD ネームスペースはマルチバイト文字セットをサポートしていません。Active Directory でマルチバイト文字セットを使用するには、LDAP ネームスペースを使用してディレクトリ接続を設定します。

注: 使用するコード ページにかかわらず、SiteMinder は、Unicode での定義と同じように文字を処理します。コード ページは特殊文字を 1 バイトとして参照できますが、SiteMinder は、Unicode でその文字がマルチバイト文字として定義されていれば、マルチバイト文字として処理します。

AD ネームスペースに対する認証

ポリシー サーバは SASL を使用して、Active Directory にバインドします。ユーザの共通名 (CN) がユーザの Windows ログオン名と異なる場合は、EnableSaslBind レジストリの設定がポリシー サーバのマシン上に存在しても、ユーザは認証可能です。

EnableSaslBind の設定は DWORD レジストリ キーです。このレジストリ キーは、0 または 1 に設定できます。

HKLM\Software\Netegrity\SiteMinder\CurrentVersion\Ds\LDAPProvider\EnableSaslBind

この設定は、ユーザを認証する際に SASL プロトコルを無効または有効にします。たとえば、EnableSaslBind が不在状態で、この設定を 1 にすると、バインドは SASL で実行されます。EnableSaslBind がある状態で、この設定を 0 にすると、バインドは単純認証メカニズムで実行されます。

管理者クレデンシャル

Active Directory (AD) ネームスペースにユーザ ディレクトリを設定する場合は、[管理者クレデンシャル] グループ ボックスの [ユーザ名] フィールドに、管理者の完全修飾ドメイン名 (FQDN) を指定します。この要件を満たさないと、ユーザ認証が失敗することがあります。

LDAP 検索ルート設定

アカウントのロック ステータスを読み取るには、ポリシー サーバが AD ネームスペースの AD ドメインを識別する必要があります。このためには、ユーザ ディレクトリの LDAP 検索ルートをドメインの DN として設定します。LDAP 検索ルートを他の DN に設定すると、ポリシー サーバは AD ドメインを識別できず、したがって、ドメインに関連付けられている Windows ロックアウト ポリシーを読み取ることができません。この状況になると、AD コンソールでロックされているユーザが、管理 UI の [User Management] ダイアログ ボックスで、有効として表示される可能性があります。

たとえば、AD コンソールで、DN ou=People,dc=clearcase,dc=com に 5 人のユーザを作成し、そのうちの 2 人をロックするとします。SiteMinder のユーザ管理ダイアログ ボックスでは、LDAP 検索ルートが AD ドメインの DN (つまり、dc=clearcase,dc=com) として設定されている場合にのみ、ロックされたユーザが無効として表示されます。LDAP 検索ルートが ou=People,dc=clearcase,dc=com として設定されている場合、ロックされたユーザは、誤って有効として表示されます。

元から無効な未認証ユーザに対するパスワード サービスリダイレクトの無効化

デフォルトでは、SiteMinder は、ディレクトリ サーバで元から無効であるためにユーザが許可されていない場合、そのユーザにクレデンシャルの再入力を求めます。この動作は、Active Directory に格納されているユーザに対しては発生しません。言い換えれば、SiteMinder は、リソースを保護する認証方式でパスワード サービスが無効であっても、元から無効なユーザをパスワード サービスにリダイレクトします。Active Directory のこの動作を防ぐには、IgnoreDefaultRedirectOnADnativeDisabled を作成して有効にします。

IgnoreDefaultRedirectOnADnativeDisabled

場所:

HKEY_LOCAL_MACHINE/SOFTWARE/Netegrity/Siteminder/CurrentVersion/Ds/LDAPProvider

値: 0(無効)または 1(有効)

デフォルト: 0 レジストリ キーが無効の場合は、デフォルトの動作が有効になります。

注: パスワード ポリシーが有効で、パスワード サービスへのリダイレクトが指定されている場合、SiteMinder は、レジストリ キーの設定にかかわらず、元から無効なユーザをパスワード サービスにリダイレクトします。

Active Directory ネームスペースがページングをサポートしない

Active Directory ネームスペースはページングをサポートしません。そのため、ユーザの数が 1000 を超えると、検索が失敗します。Active Directory ネームスペース内のユーザ数が多い場合の検索をサポートするには、以下のレジストリ キーを 1 に設定して、有効にします。

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds\LDAPProvider\EnablePagingADNameSpace

値: 0(無効)または 1(有効)

デフォルト値: 0

Active Directory ユーザ ディレクトリ接続用の LDAP ネームスペース

LDAP ネームスペースを使用して Active Directory ユーザ ディレクトリにアクセスする場合は、以下のレジストリ キーを無効にします。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Ds\
LDAPProvider\EnableADEnhancedReferrals
```

値: 0(無効)または 1(有効)

デフォルト値: 1

この手順に従えば、LDAP 接続エラーは発生しません。

Active Directory 接続のための LDAP ネームスペース

SiteMinder は、Microsoft Active Directory プラットフォーム上のユーザ ディレクトリをサポートしています。管理 UI を使用した Active Directory (AD) ネームスペースと LDAP ネームスペースの設定はほとんど同じですが、機能面で異なる点がいくつかあります。

Active Directory ユーザ ストアに LDAP ネームスペースを使用すると、以下のよう
なメリットがあります。

- 拡張 LDAP リフェラル処理がサポートされる
- LDAP のページング機能とソート機能がサポートされる

ただし、次のようなデメリットもあります。

- Windows SASL はサポートされない

LDAP ネームスペースではネイティブ Windows SASL はサポートされません。Windows SASL を使用すると、安全な LDAP バインド操作を容易に実行して、Kerberos や NTLM などの Windows ネイティブな認証方式をサポートできます。

- オブジェクトクラス属性のインデックス付け

Microsoft Active Directory がオブジェクト クラスを識別するために使う方法は標準的なものではありません。したがって、Active Directory ディレクトリ内のオブジェクトクラス属性は、デフォルトでインデックス付けされません。このため、数多くのユーザやグループが格納されている Active Directory の LDAP 実装環境で検索を実行すると、管理 UI のタイムアウトが発生することがあります。

Active Directory ユーザ ディレクトリを使用して SiteMinder を効率よく稼働させるには、Active Directory の objectClass 属性をインデックス付けする必要があります。詳細については、Active Directory のマニュアルを参照してください。

- Active Directory とパスワードサービス

Microsoft Active Directory では、格納されたユーザパスワードを変更するために SSL 接続が必要です。Active Directory ユーザディレクトリでパスワードサービスが機能するためには、パスワードポリシーが適用されるすべての Active Directory LDAP ユーザディレクトリに SSL 接続を設定する必要があります。

また、パスワードサービスと Active Directory ユーザディレクトリの連携を可能にするには、unicodePWD というパスワード属性を定義する必要があります。

注：Microsoft Active Directory の設定の詳細については、Active Directory のマニュアルを参照してください。

- Windows ユーザ セキュリティ コンテキストの使用

SiteMinder Web エージェントは、IIS Web サーバ上の Web リソースにアクセスするために Windows ユーザ セキュリティ コンテキストで実行することができます。SiteMinder で Windows ユーザ セキュリティ コンテキストを使用できるようにするには、事前にセッション ストアを設定し、レム単位で永続的セッションを有効にしておく必要があります（「Windows ユーザ セキュリティ コンテキスト」に関する説明を参照してください）。また、[ユーザ ディレクトリ] ダイアログ ボックスのクレデンシャルと接続タブで、この機能を有効にすることも必要です。

Active Directory 接続のための AD ネームスペース

SiteMinder は、Microsoft Active Directory プラットフォーム上のユーザ ディレクトリをサポートしています。管理 UI を使用した Active Directory (AD) ネームスペースと LDAP ネームスペースの設定はほとんど同じですが、機能面で異なる点があります。

Active Directory ユーザストアに AD ネームスペースを使用すると、次のようなメリットがあります。

- Windows ネイティブな証明データベースを使用する SSL 接続
注: ポリシー サーバと Active Directory ユーザ ストアをホスティングするシステムを信頼できるものとして確立する必要があります。Windows システムと Active Directory を SSL 接続用に設定する方法についての詳細は、Windows のマニュアルを参照してください。
- 安全な LDAP バインド操作が可能な Windows ネイティブ SASL のサポート

ただし、次のようなデメリットもあります。

- 拡張 LDAP リフェラル処理はサポートされない
- LDAP のページング操作とソート操作はサポートされない

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Active Directory 接続の設定

ポリシー サーバが **Active Directory** ユーザ ストアと通信できるようにするユーザ デirectory接続を設定できます。

ユーザ デirectory接続を設定する方法

1. [インフラストラクチャ]-[directory]をクリックします。
2. [ユーザ directory]、[ユーザ directoryの作成]をクリックします。

[ユーザ directoryの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [directoryのセットアップ]グループ ボックスの[ネームスペース]リストから、**AD** または **LDAP** のいずれかを選択します。

[LDAP 設定]が開きます。

注: Microsoft Active Directory は LDAP 準拠のユーザ directoryであるため、AD ネームスペースまたは LDAP ネームスペースを使用して **Active Directory** 接続を設定できます。

4. [一般]グループ ボックスと[directoryのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: 以下の点について考慮してください。

- 認証されたユーザのセキュリティ コンテキストの詳細については、**Windows** ユーザのセキュリティ コンテキストの取得方法を参照してください。
- **SSL** 接続でポリシー サーバから **Active Directory** ネームスペースに接続する予定である場合は、[directoryのセットアップ]グループ ボックスの[サーバ]フィールドで **FQDN** および **port number** を指定する必要があります。**FQDN** が指定されていない場合、ユーザ directoryにアクセスできないという内容のエラーがログ記録されます。証明書がサーバ名と一致しないことを報告する **Windows** イベントもログ記録されます。

注: ポリシー サーバが **FIPS** モードで動作し、ポリシー サーバと通信する際にdirectory接続に安全な **SSL** 接続を使用する場合、ポリシー サーバとdirectory ストアが使用する証明書は **FIPS** 準拠である必要があります。

5. (オプション) [ディレクトリ] セットアップ グループ ボックスで [設定] をクリックしてロード バランシング およびフェイルオーバーを設定します。

注: ロード バランシング とフェイルオーバーの詳細については、「LDAP ロード バランシング およびフェイルオーバー」を参照してください。

6. [管理者クレデンシャル] グループ ボックスで [クレデンシャルが必要] を選択して、グループ ボックス上のフィールドで管理者のアカウントのユーザ名およびパスワードを入力します。

注: Active Directory (AD) ネームスペースでユーザ ディレクトリを設定する場合、[ユーザ名] フィールドで管理者の完全修飾ドメイン名 (FQDN) を指定する必要があります。しない場合、ユーザ認証に失敗する場合があります。

7. [LDAP 設定] グループ ボックスの [LDAP 検索] フィールドと [LDAP ユーザ DN の検索] の各フィールドに、設定値を入力します。
8. (省略可) [ユーザ属性] グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
9. (省略可) [属性マッピングリスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
10. [サブミット] をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Active Directory グローバル カタログ ユーザ ディレクトリ接続を設定する方法

Active Directory グローバル カタログをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. Active Directory グローバル カタログ接続を設定します。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Active Directory グローバル カタログ ディレクトリ接続の設定

ポリシー サーバが **Active Directory** グローバル カタログ ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ポリシー サーバのユーザ ストアは、**Active Directory** のグローバル カタログ サポート機能に対応しています。ただし、パスワード サービスなどの **Active Directory** への書き込みを必要とする **SiteMinder** 機能には対応しません。これは、グローバル カタログが **Active Directory** への書き込みに対応していないためです。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。
[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。
注: ポリシー サーバが **FIPS** モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な **SSL** 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は **FIPS** 準拠である必要があります。
5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。

6. (省略可) [設定] をクリックして、ロード バランシングとフェイルオーバーを設定します。

注: ロード バランシングとフェイルオーバーの詳細については、「LDAP ロード バランシングおよびフェイルオーバー」を参照してください。

7. (省略可) [管理者クレデンシヤル] グループ ボックスで [クレデンシヤルが必要] を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
8. (省略可) [ユーザ属性] グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
9. (省略可) [属性マッピング リスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
10. [サブミット] をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法

Oracle Internet Directory (OID) ユーザ ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. Oracle Internet Directory で組織単位を作成します。
3. Oracle Internet Directory 接続を設定します。

Oracle Internet Directory ユーザ ディレクトリの LDAP リフェラル制限

Oracle Internet Directory Server 10g (9.0.4) がユーザ ストアとして設定され、拡張リフェラルが有効である場合、LDAP リフェラルは機能しません。これは、OID による制限です。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注：一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

OID ディレクトリ用の組織単位を作成します。

OID ディレクトリにユーザを追加するための組織単位を作成できます。

OID ディレクトリ用の組織単位を作成する方法

1. ADD を使用して、ドメイン下に組織単位を作成します。
例： OracleSchemaVersion
2. 組織単位を選択し、識別名を入力します。
例： ou=people,cn=OracleSchemaVersion
3. [Entry Management]を右クリックし、[Create]を選択します。
4. [Distinguished Name]ダイアログ ボックスで[Add]をクリックして、[inetOrgPerson]を選択します。
5. [Mandatory Properties]タブに、以下を入力します。
 - cn=user1
 - sn=user1
 - uid=user1
 - userpassword=user1
6. DN を cn=user1,ou=people,cn=OracleSchemaVersion として指定します。

Oracle インターネット ディレクトリ接続の設定

ポリシー サーバが **OID** ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[LDAP]を選択します。
[LDAP 設定]が開きます。
4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注:ポリシー サーバが **FIPS** モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な **SSL** 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は **FIPS** 準拠である必要があります。

5. [LDAP 設定]グループ ボックスの[LDAP 検索]フィールドと[LDAP ユーザ DN の検索]の各フィールドに、設定値を入力します。
6. (省略可)[管理者クレデンシャル]グループ ボックスで[クレデンシャルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。
7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、**SiteMinder** 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。

[属性マッピングの作成]ペインが開きます。

9. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

ODBC ユーザ ディレクトリ接続を設定する方法

ODBC ユーザ ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. ODBC ディレクトリ接続を設定します。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

ODBC ディレクトリ接続の設定

ポリシー サーバが ODBC ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

監査ログ用に SQL データベースを使用していて、キャッシュが有効である場合に高い負荷がかかると、ポリシー サーバが記録するメッセージをキューに入れるため SiteMinder のパフォーマンスが低下します。この問題を解決するには、多数のユーザがアクセスするリソースに関連付けられたレルムの非同期監査をオンにします。

ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]-[ユーザ ディレクトリの作成]をクリックします。

[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[ODBC]を選択します。
[ODBC 設定]が開きます。

4. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリストアが使用する証明書は FIPS 準拠である必要があります。

5. SQL クエリ方式を選択します。
6. (オプション)[管理者クレデンシヤル]グループ ボックスで[クレデンシヤルが必要]を選択し、グループ ボックスのフィールドに、ユーザ ディレクトリ上にアカウントのある管理者のユーザ名およびパスワードを入力します。

注: ユーザ名は、ユーザ ディレクトリ データを含むテーブルを所有するユーザと一致する必要があります。たとえば、SmSampleUsers 方式を使用する場合、SmUser、SmUserGroup、および SmGroup のテーブルを所有するユーザの名前を入力します。管理者のアカウントには、ユーザ ディレクトリの読み取りまたは読み取り/書き込みの権限が必要です

7. (省略可)[ユーザ属性]グループ ボックスの各フィールドに、SiteMinder 用に予約されているユーザ ディレクトリ プロファイル属性を指定します。
8. (省略可)[属性マッピングリスト]グループ ボックスで[作成]をクリックします。
[属性マッピングの作成]ペインが開きます。
9. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

SQL Server ユーザ ストアの大文字/小文字の区別なしに関する問題および末尾に余分な空白のあるパスワードに関する問題

SQL Server ユーザ ストアは、大文字と小文字を区別しません。また、ユーザのパスワードの末尾の余分な空白は無視されます。このため、ユーザは、大文字/小文字の区別がないパスワードや末尾に余分な空白のあるパスワードを入力して、保護されているリソースへのアクセス権を入手できるので、問題が発生します。たとえば、パスワード ポリシーで、ユーザのパスワードが大文字/小文字の区別がある「ABCD」でなければならないことが設定されていても、ユーザが「ABcd」と入力すると、SQL Server はアクセスを許可します。また、パスワード ポリシーで、ユーザのパスワードが「A B C」でなければならないように設定されていても、ユーザが「A B C」と入力すると、SQL Server は末尾の余分な空白を無視して、アクセスを許可します。

これらの 2 つの問題の解決方法を、以下に示します。

1 つ目の問題: SQL Server ユーザ ストアには大文字/小文字の区別がない

SQL Server データベースは適切な照合を実行せず、パスワードで大文字/小文字の区別を認識しません。

解決方法 1

SQL Server ユーザ ストアで大文字/小文字が区別されるようにするには、データベースをインストールするときにテーブル作成で適切な照合を選択します。

照合の詳細については、以下を参照してください。

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/instsql/_collation_30a6.asp

解決方法 2

デフォルトの照合が指定された SQL Server がすでにインストール済みで、データベースがパスワードの大文字/小文字を認識しない場合は、SiteMinder ユーザ ストアのテーブルを作成するときに適切な照合を作成します。

照合を指定するには、以下のスクリプトのいずれかを変更して、インポートします。

siteminder_install\db\SQL\smsampleusers_sqlserver.sql

siteminder_install\db\SQL\smsampleusers_sqlserver_upgrade.sql

注: これらの 2 つのスクリプト ファイルは、デフォルトの米国英語のローカライゼーションを使用します。

smsampleusers_sqlserver.sql スクリプト ファイル

smsampleusers_sqlserver.sql スクリプト ファイル内で、太字で強調された以下の行を変更します。

```
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
```

スクリプトを変更した後、SQL Server データベースにそのスクリプトをインポートする必要があります。

重要: 既存のデータはすべてバックアップしてください。このスクリプトを実行すると、既存のデータが削除され、新しいテーブルが作成されます。

smsampleusers_sqlserver_upgrade.sql スクリプト ファイル

太字で強調された以下の行は、smsampleusers_sqlserver_upgrade.sql スクリプトファイルに加える必要がある変更です。

```
/* 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
```

スクリプトを変更した後、SQL Server データベースにそのスクリプトをインポートする必要があります。

重要： 既存のデータはすべてバックアップしてください。このスクリプトを実行すると、既存のデータが削除され、新しいテーブルが作成されます。

2 つ目の問題: SQL Server はパスワードの末尾の空白を無視する

SQL Server は、比較する文字列に空白を埋め込んで、文字列の長さを等しくします。

解決方法

SQL Server がデータベースに格納されているパスワードの末尾の空白を認識できるようにするには、管理 UI を使用して、ODBC クエリ方式オブジェクトの[ユーザの認証]クエリを変更します。SQL Server が埋め込みまたは切り取りを行わずに文字列を比較できるようにするには、= 演算子の代わりに、LIKE 述部を組み込みます。LIKE 述部式の右側にある値の末尾に空白があっても、SQL Server は、比較を実行する前に、長さが同じになるように 2 つの値に空白を埋め込みません。認証クエリの例を、以下に示します。

```
select Name from SmUser where Name = '%s' and Password LIKE '%s'
```

重要: パスワード照合クエリに LIKE 述部式を使用すると、セキュリティホールが開くことがあります。ユーザがパスワードに「%」を使用すると、SQL Server は、その文字を LIKE 述部のワイルドカード文字として扱うため、このユーザは複数のパスワードを使用して認証できるようになります。

以下の点に注意してください。

- SDK を使用して ODBC クエリ方式オブジェクトを作成する場合は、Sm_PolicyApi_ODBCQueryScheme_t オブジェクトの pszQueryAuthenticateUser 属性を使用して、認証クエリを指定できます。
- Perl スクリプト インターフェースを使用して ODBC クエリ方式オブジェクトを作成する場合は CreateODBCQueryScheme() API を呼び出す際に認証クエリを指定できます。

Windows ディレクトリ接続を設定する方法

ユーザストアとして Windows ディレクトリを使用できます。次の Windows NT 環境でユーザディレクトリ接続を設定できます。

- Windows NT ドメイン
- ドメイン内の各 Windows NT コンピュータ
- スタンドアロン Windows NT コンピュータ

以下は、ポリシー サーバでユーザ ディレクトリ接続を作成するための手順を示しています。

1. WinNTドメイン接続要件を満たしていることを確認します
2. ユーザ ストア システムに Ping を発行します。
3. Windows ディレクトリ接続を設定します

WinNT ドメイン接続の要件

ポリシー サーバが Windows NTドメインに接続するには、以下の要件を満たす必要があります。

- ポリシー サーバを含むドメインとユーザを含むドメインの間に一方向の信頼関係が確立されていること。
- SiteMinder で認証されるドメイン内のすべてのユーザに、ポリシー サーバが稼動するマシンに対する「ネットワーク経由でコンピュータへアクセス」ユーザ権限が必要です。
- SiteMinder がユーザ ディレクトリ オブジェクトへのアクセスに使用するアカウントは、WinNTドメイン ユーザが存在するドメイン コントローラ マシンの IPC\$ 共有にアクセスする必要があります。

注: WinNTドメインのデフォルト設定を使用すると、デフォルトでこれらの要件が満たされることがあります。Windows NTドメイン管理者は、ドメインが上記の要件を満たしているかどうかを確認する必要があります。

注: Windows 展開では、SiteMinder はユーザの完全修飾 Windows ID とパスワードを IIS に渡して、Windows ユーザ コンテキストを確立します。SiteMinder は、完全修飾 Windows ID をユーザの DN エントリから取得します。DN の最初の cn 値と dc 値が連結されて Windows ID が生成されます。たとえば、ユーザ DN が次のような場合

```
cn=<username>,cn=<usergroup>,dc=<server>,dc=<domain>,  
dc=<extension>
```

生成される Windows ID は、<server>\<username> です。IIS では、<username> と Windows ユーザ ID が一致していて、かつ <server> がログオンドメイン名と同じである必要があります。

ポリシー サーバは Windows NT に対して認証を行い、各個人の識別情報やグループ メンバーシップに基づいてユーザを許可します。

Windows NT ネームスペースに対する認証を行うとき、ポリシー サーバは認証用にユーザ証明を Windows NT に渡します。この証明が、そのユーザの Windows NT ユーザ名とパスワードです。SiteMinder 環境に複数の WinNT ネームスペースが定義されているときに、SiteMinder に提供するユーザ名にドメイン名が含まれていると(つまり、`domain\username`)、ユーザ認証にかかる時間は短縮されます。この場合、SiteMinder は、指定したドメイン名に一致しないすべての WinNT ネームスペースをスキップします。

Windows NT ユーザ名とパスワードをクレデンシャルとして使用

注: WinNTドメインに対してユーザを認証するには、ポリシー サーバが WinNT 上で稼働している必要があります。

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

Windows ディレクトリ接続の設定

ポリシー サーバが WinNT ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。
[ユーザ ディレクトリの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [ネームスペース]リストから[WinNT]を選択します。
[WinNT 設定]が開きます。

4. WinNTドメイン、WinNTドメイン内のコンピュータ、またはスタンドアロン WinNT コンピュータの[一般]および[ディレクトリのセットアップ]グループ ボックス内の必須接続情報の残りを入力します。

- WinNTドメイン

Windows NTドメインに含まれるグローバル ユーザおよびグループを表すネームスペースを定義するには、[NTドメイン名]フィールドにドメイン名を指定します。SiteMinder はドメイン認証および、ドメインのユーザアカウントのドラステッドドメイン認証に対応します。

注: ポリシー サーバがインストールされるシステムは、ドメイン認証が動作するべき適切なドメイン内に、コンピュータ アカウントを持っている必要があります。ユーザ認証を行う必要があるすべてのドメインに、システムがコンピュータアカウントを所有していない場合は、ドメイン間に適切な信頼関係が確立する必要があります。

コントロール パネルから[サービス]ダイアログ ボックスを開いて、ポリシー サーバが実行されているアカウントを、指定されたドメインにアクセスするのに必要な権限を持っているアカウント変更することにより、ポリシー サーバを実行する NT アカウントを変更できます。このアカウントには、[オペレーティング システムの一部として機能]のシステム権限が必要です。

- ドメイン内の各 Windows NT コンピュータ

ドメインに属しているコンピュータのローカルユーザおよびグループを表すネームスペースを定義するには、[NTドメイン名]フィールドにドメイン名とコンピュータ名を続けて指定します。右上がりスラッシュ(/)によってドメイン名とコンピュータ名を区切る必要があります。コンピュータの名前を指定しない場合、検索中のパフォーマンスが低下する場合があります。

例: SampleDomain/Comp1

- スタンドアロン Windows NT コンピュータ

スタンドアロン コンピュータに含まれるローカルユーザおよびグループを表すネームスペースを定義するには、[NTドメイン名]フィールドにコンピュータ名を指定します。このネームスペースにアクセスするため、スタンドアロン コンピュータにポリシー サーバをインストールしておく必要があります。ポリシー サーバが初めてこのネームスペースにアクセスするときには、初期遅延が発生する場合があります。

5. (オプション)[管理者クレデンシヤル]グループ ボックスで[クレデンシヤルが必要]を選択し、グループ ボックスのフィールドに、ユーザ ディレクトリ上にアカウントのある管理者のユーザ名およびパスワードを入力します。

6. (省略可) [ユーザ属性] グループ ボックスの各フィールドに、**SiteMinder** 用に予約されているユーザ ディレクトリ プロファイル 属性を指定します。
7. (省略可) [属性マッピング リスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
8. [サブミット] をクリックします。
ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

LanMan ユーザディレクトリ

Windows NT 環境でディレクトリ リクエストを配信する代わりに、**LanMan** ユーザ ディレクトリ 接続を使用する方法があります。**LanMan** ユーザディレクトリ接続オプションを使用すると、**Windows NT** レジストリで各ユーザディレクトリの検索に使用される **BDC** (バックアップドメイン コントローラ) のフェイルオーバー リストを指定できます。**BDC** のリストは、ほとんどの **Windows NT** 環境においてディレクトリ リクエストを処理する **PDC** (プライマリドメイン コントローラ) よりも優先されます。**LanMan** ディレクトリ接続を使用すると、ポリシー サーバは、**Windows NT** ディレクトリ リクエストを **PDC** に渡す代わりに、レジストリ リスト内の最初のアクティブ **BDC** に送信します。

Windows NT ユーザディレクトリのフェイルオーバー

PDC (プライマリドメイン コントローラ) と **BDC** (バックアップドメイン コントローラ) のリストを定義することで、**Windows NT LanManager** は **WinNT** ユーザ ディレクトリのフェイルオーバーを自動的に実行します。**SiteMinder** に、**WinNT** ディレクトリフェイルオーバー用の追加機能はありません。フェイルオーバーを設定するには、**Windows NT LanManager** を使用してください。

注: **LanManager** の詳細については、**Windows NT** のマニュアルを参照してください。

レスポンスの WinNT 属性

レスポンスによって **Web** アプリケーションに情報を返すため、**SiteMinder** は、以下のデフォルトのユーザ属性へのアクセス権を所有しています。

- Description
- FullName
- AccountExpirationDate
- UserFlags

- LoginWorkstations
- BadPasswordAttempts
- MaxLogins
- MaxStorage
- PasswordExpired
- LastLogin
- LastLogoff
- HomeDirectory
- Profile
- LoginScriptMinPasswordLength
- MaxPasswordAge
- MinPasswordAge
- PasswordHistoryLength

これらのユーザ属性には、[レスポンス]ペイン上の[ユーザ属性]機能からアクセスできます。レスポンスは、これらのどの属性の値でも **SiteMinder** エージェントに返すことができます。**Web** 実装では、**SiteMinder** はこれらの属性を HTTP ヘッダ変数の中の名前/値のペアとして返します。

カスタム ユーザ ディレクトリ接続を設定する方法

カスタム ディレクトリをユーザ ストアとして使用することができます。以下に、ポリシー サーバへのユーザ ストア接続を作成するための手順を示します。

1. ユーザ ストア システムに Ping を発行します。
2. カスタム ディレクトリ接続の設定

ユーザ ストア システムに Ping を発行します。

ユーザ ストア システムに Ping を発行することによって、ポリシー サーバとユーザ ディレクトリまたはユーザ データベースとの間にネットワーク接続が確立されていることを確認します。

注: 一部のユーザ ストア システムでは、ポリシー サーバがクレデンシャルを提示することが必要になる場合もあります。

カスタム ディレクトリ接続を設定します

ポリシー サーバがカスタム ユーザ ストアと通信できるようにするユーザ ディレクトリ接続を設定できます。

ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。
[ユーザ ディレクトリの作成]ペインが表示されます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[ユーザ ディレクトリの作成: <名前>]ペインが表示されます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [ネームスペース]リストから[カスタム]を選択します。
[カスタム設定]が開きます。
5. [一般]グループ ボックスと[ディレクトリのセットアップ]グループ ボックスに、その他の必要な接続情報を入力します。

注: ポリシー サーバが FIPS モードで動作し、ポリシー サーバと通信する際にディレクトリ接続に安全な SSL 接続を使用する場合、ポリシー サーバとディレクトリ ストアが使用する証明書は FIPS 準拠である必要があります。

6. (省略可) [管理者クレデンシヤル]グループ ボックスで[クレデンシヤルが必要]を選択し、そのグループ ボックスの各フィールドに、ユーザ ディレクトリ上の管理者のアカウントのユーザ名とパスワードを入力します。

注: ポリシー サーバは共有ライブラリを使用して、カスタム ディレクトリが使用できるユーザ属性を決定します。ユーザ属性を入力する前に、ユーザ ディレクトリ接続を作成する必要があります。

7. (省略可) [属性マッピングリスト]グループ ボックスで[作成]をクリックします。

[属性マッピングの作成]ペインが開きます。

8. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法

LDAP ユーザ ディレクトリの SSL 接続を設定するには、証明書データベース ファイルを使用するように SiteMinder を設定する必要があります。

SSL 接続を設定するには、以下の手順に従います。

1. SSL 接続を設定する前に以下を実行します。
2. NSS ユーティリティをインストールします。
3. 証明書データベース ファイルを作成します。
4. 証明書データベースに証明機関(CA)を追加します。
5. 証明書データベースにサーバ証明書を追加します。
6. 証明書データベース内の証明書を一覧表示します。
7. ユーザ ディレクトリの SSL 接続を設定します。
8. ポリシー サーバが証明書データベースを参照するようにします。
9. SSL 接続を検証します。

SSL 接続を設定する前に

SSL による LDAP ユーザ ディレクトリ接続を設定する前に、以下の点を確認してください。

- ディレクトリ サーバが SSL 対応であること。

注: SSL を介して通信するようにディレクトリ サーバを設定する方法の詳細については、ベンダー別の資料を参照してください。

- SiteMinder は、Netscape LDAP SDK を使用して、LDAP ディレクトリと通信します。その結果、SiteMinder では、データベース ファイルを Netscape バージョン ファイル形式 (cert7.db) にする必要があります。

重要: cert7.db データベース ファイルへの証明書のインストールに Microsoft Internet Explorer を使用しないでください。

- SSL 証明書の管理には、Netscape と互換性のある、サードパーティの証明書ユーティリティが必要です。Mozilla® Network Security Services (NSS) ユーティリティ、バージョン 3.2.2 の使用をお勧めします。

注: cert7.db 形式をサポートするには、バージョン 3.2.2 が必要です。それ以降のバージョンを使用しないでください。

- (Active Directory) 以下の点を考慮してください。
 - SiteMinder ユーザ ディレクトリ接続が AD ネームスペースを使用して設定されている場合は、以下のプロセスは適用されません。SSL 接続を確立するときに、AD ネームスペースは、ネイティブの Windows 証明書レジストリを使用します。SSL を介して通信するように AD ネームスペースを設定する場合は、以下の点を確認してください。
 - SiteMinder ユーザ ディレクトリ接続が、安全な接続になるように設定されていること。詳細については、「ユーザ (236P.) ディレクトリの SSL 接続の設定」を参照してください。
 - Active Directory インスタンスをホストしているマシンで、ルート CA 証明書およびサーバ証明書が、サービスの証明書ストアに追加されていること。

注: SSL を介して通信するように Active Directory を設定する方法の詳細については、Microsoft の資料を参照してください。

- SiteMinder ユーザ ディレクトリ接続が LDAP ネームスペースを使用して設定されている場合は、以下のプロセスを実行して、SSL 接続を設定します。

NSS ユーティリティのインストール

証明書データベース ファイルを管理する NSS ユーティリティをインストールします。

注: Netscape Portable Runtime (NSPR) またはポリシー サーバがインストールされているシステムに、このユーティリティをインストールします。システムに、どちらかのコンポーネントと一緒にこのユーティリティをインストールすると、必要な DLL または共有オブジェクトが使用可能になります。

NSS ユーティリティをインストールする方法

1. [Mozilla](#) NSS 3.2.2 FTP サイトにアクセスします。
2. ご使用のオペレーティング システムに対応する zip または tar をダウンロードします。

注: Windows Server 2003 用の zip は用意されていません。Windows NT 用の zip をダウンロードしてください。

3. 証明書データベース ファイルを管理しているシステム上の一時ロケーションに、NSS ユーティリティを解凍します。

証明書データベース ファイルの作成

ポリシー サーバでは、証明書データベース ファイルを Netscape バージョン ファイル形式(cert7.db)にする必要があります。NSS ユーティリティを使用して、証明書データベース ファイルを作成することができます。

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、[NSS プロジェクト ページ](#)にある Mozilla マニュアルを参照してください。

証明書データベース ファイルを作成する方法

1. コマンド プロンプトから、NSS ユーティリティを解凍したロケーション内の bin ディレクトリに移動します。

例: C:\nss\bin

注: Windows には固有の certutil ユーティリティがあります。NSS ユーティリティの bin ディレクトリから作業するようにしてください。間違えて Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを入力します。

```
certutil -N -d certificate_database_directory
```

-N

cert7.db、key3.db、および secmod.db の証明書データベース ファイルを作成します。

-d certificate_database_directory

NSS ユーティリティが証明書データベース ファイルを作成するディレクトリを指定します。

注: ファイル パスにスペースがある場合は、そのパスを引用符で囲ってください。

このユーティリティは、データベース キーを暗号化するためにパスワードの入力を求めます。

3. パスワードを入力および確認します。

NSS は、必要な証明書データベース ファイルを作成します。

- cert7.db
- key3.db
- secmod.db

例: 証明書データベース ファイルの作成

```
certutil -N -d C:\certdatabase
```

証明書データベースへのルート証明機関の追加

ルート証明機関(CA)を追加して、SSL 通信でできるようにします。

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、[NSS プロジェクト ページ](#)にある **Mozilla** マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

証明書データベースにルート CA 証明書を追加する方法

1. コマンド プロンプトから、NSS ユーティリティを解凍したロケーション内の bin ディレクトリに移動します。

例: C:\nss\bin

注: Windows には固有の certutil ユーティリティがあります。NSS ユーティリティの bin ディレクトリから作業するようにしてください。間違えて Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行して、データベースファイルにルート CA を追加します。

```
certutil -A -n alias -t trust_arguments -i root_CA_path -d  
certificate_database_directory
```

-A

証明書データベースに証明書を追加します。

-n alias

証明書の別名を指定します。

注: 別名にスペースがある場合は、その別名を引用符で囲んでください。

-t trust_arguments

証明書データベースに証明書を追加するときに証明書に適用する信頼性属性を指定します。各証明書には、3 つの使用可能な信頼性カテゴリがあります。これらのカテゴリを表記する順序は、「SSL、電子メール、オブジェクト署名」です。ルート CA が信頼されて SSL 証明書を発行できるように、適切な信頼性引数を指定します。それぞれのカテゴリ位置に、以下の属性引数を 0 個以上使用することができます。

p

有効なピア。

P

信頼されたピア。この引数は p を意味します。

c

有効な CA。

T

クライアント証明書を発行する信頼された CA。この引数は c を意味します。

C

サーバ証明書を発行する信頼された CA (SSL のみ)。この引数は c を意味します。

重要: これは SSL 信頼性カテゴリに必須の引数です。

u

証明書は認証または署名に使用できます。

-i root_CA_path

ルート CA ファイルのパスを指定します。以下の点について考慮してください。

- パスには証明書名も含める必要があります。
- 証明書の有効な拡張子には、.cert、.cer、および .pem があります。

注: ファイルパスにスペースがある場合は、そのパスを引用符で囲ってください。

-d certificate_database_directory

証明書データベースが入っているディレクトリのパスを指定します。

注: ファイルパスにスペースがある場合は、そのパスを引用符で囲ってください。

NSS によって、証明書データベースにルート CA が追加されます。

例: 証明書データベースへのルート CA の追加

```
certutil -A -n "My Root CA" -t "C,," -i C:\certificates\cacert.cer -d C:\certdatabase
```

証明書データベースへのサーバ証明書の追加

証明書データベースにサーバ証明書を追加して、SSL 通信でできるようにします。

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、[NSS プロジェクト ページ](#)にある Mozilla マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

証明書データベースにサーバ証明書を追加する方法

1. コマンド プロンプトから、NSS ユーティリティを解凍したロケーション内の bin ディレクトリに移動します。

例: C:\nss\bin

注: Windows には固有の certutil ユーティリティがあります。NSS ユーティリティの bin ディレクトリから作業するようにしてください。間違えて Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行して、データベース ファイルにルート証明書を追加します。

```
certutil -A -n alias -t trust_arguments -i server_certificate_path -d  
certificate_database_directory
```

-A

証明書データベースに証明書を追加します。

-n *alias*

証明書の別名を指定します。

注: 別名にスペースがある場合は、その別名を引用符で囲んでください。

-t *trust_arguments*

証明書データベースに証明書を追加するときに証明書に適用する信頼性属性を指定します。各証明書には、3 つの使用可能な信頼性カテゴリがあります。これらのカテゴリを表記する順序は、「SSL、電子メール、オブジェクト署名」です。証明書が信頼されるように、適切な信頼性引数を指定します。各カテゴリ位置では、以下の属性引数を 0 個以上使用することができます。

p

有効なピア。

P

信頼されたピア。この引数は p を意味します。

重要: これは SSL 信頼性カテゴリに必須の引数です。

-i server_certificate_path

サーバ証明書のパスを指定します。以下の点について考慮してください。

- パスには証明書名も含める必要があります。
- 証明書の有効な拡張子には、.cert、.cer、および .pem があります。

注: ファイルパスにスペースがある場合は、そのパスを引用符で囲ってください。

-d certificate_database_directory

証明書データベースが入っているディレクトリのパスを指定します。

注: ファイルパスにスペースがある場合は、そのパスを引用符で囲ってください。

NSS によって、証明書データベースにサーバ証明書が追加されます。

例: 証明書データベースへのサーバ証明書の追加

```
certutil -A -n "My Server Certificate" -t "P,," -i C:\certificates\servercert.cer -d C:\certdatabase
```

証明書データベース内の証明書の一覧表示

証明書が証明書データベースに追加されたことを確認するために、証明書を一覧表示します。

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、[NSS プロジェクト ページ](#)にある **Mozilla** マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

証明書データベース内の証明書を一覧表示する方法

1. コマンド プロンプトから、NSS ユーティリティを解凍したロケーション内の bin ディレクトリに移動します。

例: C:\nss\bin

注: Windows には固有の certutil ユーティリティがあります。NSS ユーティリティの bin ディレクトリから作業するようにしてください。間違えて Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行します。

```
certutil -L -d certificate_database_directory
```

-L

証明書データベース内のすべての証明書を一覧表示します。

-d certificate_database_directory

証明書データベースが入っているディレクトリのパスを指定します。

注: ファイルパスにスペースがある場合は、そのパスを引用符で囲ってください。

NSS によって、証明書データベースに証明書を追加するときに指定した、ルート CA の別名、サーバ証明書の別名、および信頼性属性が表示されます。

例: 証明書データベース内の証明書の一覧表示

```
certutil -L -d C:\certdatabase
```

ユーザ ディレクトリの SSL 接続の設定

ポリシー サーバとユーザ ストアが通信するときに SSL 接続が使用されるように、ユーザ ストア接続を設定します。

注: FSS 管理 UI にポリシー サーバオブジェクトを作成または変更するときは、ASCII 文字を使用します。ASCII 文字以外でのオブジェクトの作成または変更はサポートされていません。

ユーザ ストアの SSL 接続を設定する方法

1. 管理 UI にログインします。
2. [インフラストラクチャ]-[ディレクトリ]をクリックします。

3. [ユーザ ディレクトリ]-[ユーザ ディレクトリの変更]をクリックします。
[ユーザ ディレクトリの変更]ペインに、既存のユーザ ディレクトリ接続の一覧が表示されます。
4. 目的のユーザ ディレクトリ接続を選択し、[選択]をクリックします。
ユーザ ディレクトリの設定が表示されます。
5. [安全な接続]チェック ボックスをオンにして、[サブミット]をクリックします。
ユーザ ディレクトリ接続が、SSL を介して通信するように設定されます。

ポリシー サーバから証明書データベースへの参照の設定

ポリシー サーバが証明書データベースを参照するように設定し、SSL を介してユーザ ディレクトリと通信するようにポリシー サーバを設定します。

注: FSS 管理 UI にポリシー サーバ オブジェクトを作成または変更するときは、ASCII 文字を使用します。ASCII 文字以外でのオブジェクトの作成または変更はサポートされていません。

ポリシー サーバから証明書データベースへの参照を設定する方法

1. ポリシーサーバー管理コンソールを起動します。
重要: Windows Server 2008 上でこのグラフィカル ユーザ インターフェースにアクセスする場合は、管理者としてシステムにログインしている場合でも、管理者権限でショートカットを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。
2. [データ]タブをクリックします。
3. Netscape 証明書データベース ファイルのフィールドに、Netscape 証明書データベース ファイルのパスを入力します。
例: C:\certdatabase\cert7.db
注: key3.db ファイルも、cert7.db ファイルと同じディレクトリに含まれている必要があります。
4. ポリシー サーバを再起動します。

ポリシー サーバが、SSL を介してユーザ ディレクトリと通信するように設定されます。

SSL 接続の検証

SSL 接続を検証して、ユーザ ディレクトリとポリシー サーバが SSL を介して通信していることを確認します。

注: FSS 管理 UI にポリシー サーバ オブジェクトを作成または変更するときは、ASCII 文字を使用します。ASCII 文字以外でのオブジェクトの作成または変更はサポートされていません。

SSL 接続を検証する方法

1. 管理 UI にログインします。
2. [インフラストラクチャ]-[ディレクトリ]をクリックします。
3. [ユーザ ディレクトリ]-[ユーザ ディレクトリの表示]をクリックします。
[ユーザ ディレクトリの表示]ペインに、既存のユーザ ディレクトリ接続の一覧が表示されます。
4. 目的の接続を選択し、[選択]をクリックします。
ユーザ ディレクトリの設定が表示されます。
5. [内容の表示]をクリックします。
SSL が正しく設定されている場合は、[ディレクトリのコンテンツ]ウィンドウに、ユーザ ディレクトリの内容が表示されます。

LDAP ロード バランシングおよびフェイルオーバー

ポリシー サーバでは、複数の LDAP サーバに LDAP クエリを送信して、フェイルオーバーやロードバランシングを実行できます。フェイルオーバーが設定されていると、ポリシー サーバは、サーバが応答に失敗するまで 1 台の LDAP サーバを使用してリクエストに応答し続けます。デフォルト サーバが応答しない場合、フェイルオーバー用に指定された次のサーバにリクエストがルーティングされます。この処理を複数のサーバで繰り返すことができます。デフォルト サーバが再びリクエストを実行できるようになると、ポリシー サーバは元のサーバにリクエストをルーティングします。

ロードバランシングが設定されていると、ポリシー サーバは指定された複数の LDAP サーバにリクエストを分散して送信します。これによって、リクエストが各 LDAP サーバに対して均等に配信されます。フェイルオーバーとロードバランシングを組み合わせると、より素早く効率的に LDAP ユーザディレクトリ情報にアクセスできるようになり、サーバエラーが発生した場合の冗長性を高めることもできます。

ポート番号に関する考慮事項

ポートは、個々の LDAP サーバやフェイルオーバー グループに割り当てることができます。また、LDAP サーバにデフォルトのポート番号を使用するようにポリシー サーバを設定することもできます。

ポート番号を指定する際のガイドラインは次のとおりです。

状況	結果
フェイルオーバー グループ内の最後のサーバ以外のすべてのサーバにポート番号があります。	ポリシー サーバは、具体的なポートが割り当てられていないグループ内のサーバにはデフォルトのポートを使用すると見なします。SSL のデフォルトは 636 です。非 SSL のデフォルトは 389 です。 たとえば、サーバのフェイルオーバー グループに次のようなポート番号が含まれているとします。 123.123.12.12:350 123.123.34.34 フェイルオーバー グループ内の最初のサーバのポート番号が 350 です。そのサーバとの通信は、ポート 350 で行われます。 最初のサーバで障害が発生すると、ポリシー サーバは、デフォルトポートの 389 を使用して 2 番目のサーバと通信します。これは、フェイルオーバー グループ内の 2 番目のサーバにポートが指定されていないためです。

フェイルオーバーの設定

プライマリ LDAP ディレクトリ接続が使用できなくなった場合に備えて、フェイルオーバーを設定して冗長性を確保します。

注: フェイルオーバー用にサーバを追加する場合、そのフェイルオーバー ディレクトリは、プライマリ ディレクトリと同じ通信タイプ (SSL または SSL 以外) を使用する必要があります。これは、両方のディレクトリが同じポート番号を共有するためです。

フェイルオーバーを設定する方法

1. [ユーザ ディレクトリ] ペインの[ディレクトリのセットアップ]グループ ボックスにある[設定]をクリックします。

[ディレクトリのフェイルオーバーとロード バランシングのセットアップ] ペインが開きます。[フェイルオーバー グループ] にプライマリ ユーザ ディレクトリが表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ] をクリックします。

2. [フェイルオーバーの追加] をクリックします。

[ホスト] フィールドと[ポート] フィールドが表示されます。

3. ポリシー サーバがフェイルオーバーするサーバのホスト名とポートを入力します。

注: ポート番号を指定しない場合、ポリシー サーバはデフォルトのポートを使用します。SSL のデフォルトのポートは 636 です。SSL 以外のデフォルトのポートは 389 です。

4. さらにフェイルオーバー サーバを定義するには、手順 2 と 3 を繰り返します。

注: グループの最後のサーバにのみポートを指定し、その他のサーバにはポートを指定しない場合、ポリシー サーバは指定されているポートをそのグループ内のすべてのサーバに使用します。

5. [OK] をクリックします。

[ユーザ ディレクトリ] ペインが表示されます。[サーバ] フィールドには、フェイルオーバー用として指定されたサーバが一覧されます。フェイルオーバーに指定された各サーバは、スペースで区切られます。

ロード バランシングの設定

ポリシー サーバが複数の LDAP サーバ間でリクエストを均等に分散できるように、ロード バランシングを設定します。

ロード バランシングを設定する方法

1. [ディレクトリのセットアップ]グループ ボックスで[設定]をクリックします。
[ディレクトリのフェイルオーバーとロード バランシングのセットアップ]ペインが開きます。[フェイルオーバー グループ]にプライマリ ユーザ ディレクトリが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
2. [ロード バランシングの追加]をクリックします。
新しいフェイルオーバー グループが開きます。
3. ポリシー サーバがロード バランシングを適用するサーバのホスト名とポートを入力します。
4. 手順 2 および 3 を繰り返して、追加のロード バランシング サーバを定義します。
5. [OK]をクリックします。
[ユーザ ディレクトリ]ペインが表示されます。[サーバ]フィールドに、ロード バランシング対象として指定されたサーバがリストされます。ロード バランシングに指定された各サーバは、カンマ(,)で区切られます。

ロード バランシングとフェイルオーバーの設定

ロード バランシングおよびフェイルオーバーを設定して、複数のサーバにリクエストを分散して送信し、プライマリ ディレクトリ接続が使用できなくなった場合に冗長性を提供します。

ロード バランシングとフェイルオーバーを設定する方法

1. [ディレクトリのセットアップ]グループ ボックスで[設定]をクリックします。

[ディレクトリのフェイルオーバーとロード バランシングのセットアップ]ペインが開きます。[フェイルオーバー グループ]にプライマリ ユーザ ディレクトリが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. ポリシー サーバがフェイルオーバーするサーバのホスト名とポートを入力します。

注:ポート番号を指定しない場合、ポリシー サーバはデフォルトのポートを使用します。SSL のデフォルトのポートは 636 です。SSL 以外のデフォルトのポートは 389 です。

3. さらにフェイルオーバー サーバを定義するには、手順 2 と 3 を繰り返します。

注:グループの最後のサーバにのみポートを指定し、その他のサーバにはポートを指定しない場合、ポリシー サーバは指定されているポートをそのグループ内のすべてのサーバに使用します。

4. [ロード バランシングの追加]をクリックします。

新しいフェイルオーバー グループが開きます。

5. ポリシー サーバがロード バランシングを適用するサーバのホスト名とポートを入力します。

注:ロード バランシングに同じサーバを複数回追加して、特定システムで処理されるリクエストを強制的に増やすことができます。たとえば、グループ内に 2 つのサーバ、Server1 と Server2 があるとします。Server1 は高性能サーバで、Server2 は性能が低いシステムです。Server1 をロード バランシングリストに 2 回追加すると、Server2 でリクエストが 1 つ処理される間に Server1 で 2 つ処理させることができます。

6. ステップ 5 および 6 を繰り返して、追加のロード バランシング サーバを定義します。

7. [OK]をクリックします。

[ユーザ ディレクトリ]ペインが表示されます。[サーバ]フィールドが、フェイルオーバーおよびロード バランシング用に指定されたサーバを示します。フェイルオーバーに指定された各サーバは、スペースで区切られます。ロード バランシングに指定された各サーバは、カンマ(,)で区切られます。

ユース ケース - ロード バランシングとフェイルオーバー

この例では、SiteMinder 環境に、A と B の 2 つのユーザ ディレクトリがあります。これらの 2 つのディレクトリは、以下の要件を満たしている必要があります。

- ユーザ ディレクトリ A は、(1) ユーザ ディレクトリ B にフェイルオーバーし、(2) ユーザ ディレクトリ B とロード バランシングする必要があります。
- ユーザ ディレクトリ B は、(1) ユーザ ディレクトリ A にフェイルオーバーし、(2) ユーザ ディレクトリ A とロード バランシングする必要があります。

ここで、スペースはフェイルオーバーを表し、カンマはロード バランシングを表します。要件は、以下のように記述されます。

A B, B A

解決方法:

この設定には、2 つのフェイルオーバー グループが必要です。

1. 最初のフェイルオーバー グループに、ユーザ ディレクトリ B を追加します。

現在の設定は A B です。

2. ロード バランシング グループを追加します。

注: ロード バランシング グループは、新しいフェイルオーバー グループとして開きます。

3. ロード バランシング グループの最初のサーバとして、ユーザ ディレクトリ B をリストします。

現在の設定は、A B, B です。

4. ロード バランシング グループの 2 つ目のサーバとして、ユーザ ディレクトリ A をリストします。

この結果、2 つのフェイルオーバー グループ「A B」と「B A」が設定されます。これらのグループは相互にロード バランシングを行います。両方のディレクトリが使用可能な場合は、各フェイルオーバー グループ内の最初のディレクトリの間、つまり A と B の間でロード バランシングが発生します。ユーザ ディレクトリ A が使用不能になると、ユーザ ディレクトリ B に対するフェイルオーバーが発生します。この結果、ユーザ ディレクトリ B は、ユーザ ディレクトリ A が使用可能になるまで、すべてのリクエストを処理します。

ODBC データ ソース フェイルオーバーの設定

プライマリの ODBC データ ソースも、後続の ODBC データ ソースも使用できなくなった場合に冗長性を提供するように、フェイルオーバーを設定します。

フェイルオーバーを設定する方法

1. [ディレクトリのセットアップ]グループ ボックスで[設定]をクリックします。
[ODBC フェイルオーバーのセットアップ]ペインが開きます。データ ソースグループ内のプライマリ データ ソースが開きます。
注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
2. [追加]をクリックします。
データ ソース フィールドが開きます。
3. ポリシー サーバのフェイルオーバー先とするデータ ソースを入力します。
4. 手順 2 および 3 を繰り返して、別のデータ ソースを追加します。
5. [OK]をクリックします。
[ユーザ ディレクトリ]ペインが表示されます。[サーバ]フィールドに、フェイルオーバー用に指定されたデータ ソースがリストされます。フェイルオーバー用に指定された各データ ソースは、カンマ(,)で区切られます。

SQL クエリ方式

ポリシー サーバは SQL クエリ方式を使用して、リレーショナル データベースでユーザ データを検索するクエリを作成します。SiteMinder の[SQL クエリ方式]ダイアログ ボックスを使用して、SQL クエリ方式を作成および編集します。

注: 列名の接頭辞である「SM_」は、SiteMinder から要求される追加の特殊名のために予約されています。したがって、ユーザ ディレクトリの列名には、「SM_」という接頭辞を使用しないでください。この接頭辞が列名に使用されていると、ユーザ検索時にポリシー サーバエラーが発生します。

SQL クエリ方式の設定

ユーザストアとして使用しているリレーショナル データベースでユーザ データを検索する SQL クエリ方式を設定できます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

SQL 認証方式の設定方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [SQL クエリ方式]-[SQL クエリ方式の作成]をクリックします。
[SQL クエリ方式の作成]ウィンドウが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[SQL クエリ方式: <名前>]ウィンドウが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [一般]グループ ボックスのフィールドに名前および説明を入力します。
5. 使用しているデータベース スキーマに合わせて、クエリ フィールドの内容を更新します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

リレーショナル データベースを使用するには、各クエリを設定する必要があります。そのためには、以下のデータベース テーブル名および列名を、使用しているリレーショナル データベースにある名前と置き換えます。

■ Name

リレーショナル データベースに配置されているユーザ ストアを使用するようポリシー サーバを設定する場合、SiteMinder が正しくユーザを識別できるよう、ユーザの **Name** パラメータは一意である必要があります。このため、複数のユーザが同じユーザ名を持つことはできません。

■ SmUser—table

■ SmGroup—table

■ Password

■ SmUserGroup—table

- ID
 - UserID
 - FirstName
 - LastName
 - TelephoneNumber
 - EmailAddress
 - Mileage
 - PIN
 - GroupID
 - Disabled
6. クエリ タイプを選択し、[サブミット]をクリックします。
クエリが保存されました。クエリ方式をユーザ ディレクトリ接続に関連付けることができます。
 7. ポリシー サーバの管理コンソールを使用して、ポリシー サーバを再起動します。

ODBC ユーザ ディレクトリ接続への SQL クエリ方式の追加

[ユーザ ディレクトリ]ダイアログ ボックスを使用して、SQL クエリ方式を選択することができます。

SQL クエリ方式を選択する方法

1. 既存のユーザ ディレクトリ接続オブジェクトに対する[ユーザ ディレクトリ]ペインを開きます。
2. [SQL クエリ方式]リストから、SQL クエリ方式を選択して、[サブミット]をクリックします。

SQL クエリ方式がディレクトリ接続に保存されます。

3. ポリシー サーバの管理コンソールを使用して、ポリシー サーバを再起動します。

注: Microsoft SQL Server を使用しているときに、クエリによってアポストロフィ文字を含む名前 (O'Neil など) が返された場合は、クエリ文字列の '%s' インスタンスをすべて "%s" に置き換える必要があります。この問題を回避するには、アポストロフィを含まないユーザ ID に基づくクエリを作成するか、'%s' を含むクエリ文字列を変更します。

ストアド プロシージャを介して認証するように SQL クエリ方式を設定する方法

ODBC ユーザ ディレクトリによる認証にストアド プロシージャが必要な場合は、以下のように、ストアド プロシージャを呼び出すように SQL クエリ方式を設定します。

SQLServer

構文: Call *Procedure_Name* %s , %s

例: Call EncryptPW %s , %s

SQLServer のストアド プロシージャは、以下の要件を満たしている必要があります。

- 最初のパラメータがユーザ名で、2 番目のパラメータがパスワードであること。
- すべてのパラメータは、キーワード OUT を使用して定義すること。
- ストアドプロシージャは整数値を返すこと。

以下の例に、SQLServer ユーザ ディレクトリのストアド プロシージャを作成する方法を示します。

```
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

構文: Call *Procedure_Name* %s, %s

例: Call EncryptPW %s, %s

MySQL のストアド プロシージャは、以下の要件を満たしている必要があります。

- 最初のパラメータがユーザ名で、2 番目のパラメータがパスワードであること。
- ストアド プロシージャは値を返さないこと。

以下の例に、MySQL ユーザ ディレクトリのストアド プロシージャを作成する方法を示します。

```
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 の関数

Oracle ユーザ ディレクトリでは、以下のテンプレートを使用して、以下の関数を作成することができます。

- EncryptPW
- ChangePW

EncryptPW 関数

EncryptPW 関数は、以下のように、整数値を返す必要があります。

- value = 0
成功を指定します。
- value = 1
失敗を指定します。

以下のテンプレートを使用して、EncryptPW 関数を作成することができます。

```
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 関数

ChangePW 関数は、以下のように、整数値を返す必要があります。

- value = 1
成功を指定します。
- value = 0
失敗を指定します。

以下のテンプレートを使用して、ChangePW 関数を作成することができます。

```
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;
```

フェイルオーバーおよび接続プーリング時の非同期呼び出しの対応

同期呼び出しは、信頼性が高く、リクエストの処理が完了してから返されます。
非同期呼び出しはただちに返されます。呼び出し元は、非同期呼び出しを放棄して、ネットワーク障害に伴う遅延を回避できます。

SiteMinder 以下のデータベースへの非同期呼び出しに対応しています。

- SQLServer
- Windows NT および Solaris の上の Oracle 8

非同期呼び出し対応の設定

レジストリ サブキー *Netegrity\SiteMinder\CurrentVersion\Database* の下には、以下のようなレジストリ オプションが格納されています。

AsynchronousCalls

データベースコールが非同期で行われるかどうかを指定します。

値: 0 (no)、1 (yes)

デフォルト: 0

AsynchronousSleepTime

呼び出しの間隔を指定します。この時間を過ぎても SQL 呼び出しが処理されない場合、ステータスがチェックされます。

値: 0 ~ n ミリ秒

デフォルト: 15 ミリ秒

LoginTimeout

データベースにログインするまでの許容時間。この時間を過ぎると、接続はタイムアウトします。

値: 最小で 1 秒

デフォルト: 15 秒

QueryTimeout

クエリの実行が完了するまでの時間。この時間を過ぎるとクエリは取り消されます。

値: 最小で 1 秒

デフォルト: 15 秒

注: Windows NT で動作する SQL Server の場合、非同期呼び出しを使用しても、放棄された接続ごとに生じるメモリリークはごくわずかです。ネットワークが不安定な場合には、上記の設定を調整してタイムアウトを延長し、フェイルオーバーの数を減らしてください。

Solaris で動作する Oracle 8 に非同期呼び出しを設定

Solaris 2.6 および 2.7 対応 Oracle の Merant ODBC ドライバは、非同期呼び出しをサポートした場合にコアダンプを発生させることがあります。これは、Oracle のバグによるもので、次のように修正できます。

Oracle 8.0.5

<install_directory>/db ディレクトリの system_odbc.ini ファイルに含まれる各データソースに、次のエントリを追加します。

ArraySize=1

注: この変更を行うと、複数行フェッチがオフになり、大きなポリシー ストアを読み込むときのパフォーマンスが低下します。

Oracle 8.1.5

1. *siteminder/bin* または *siteminder/odbc/lib*、あるいはその両方で *libclntsh.so* を削除するか、名前を変更します。
2. Oracle クライアントの *\$ORACLE_HOME/lib* に *libclntsh.so* がインストールされていることを確認します。インストールと再ビルドの方法については、Oracle のマニュアルを参照してください。
3. *LD_LIBRARY_PATH* が Oracle クライアントライブラリディレクトリの *\$ORACLE_HOME/lib* を参照していることを確認します。

次のようなログ メッセージが表示された場合

```
[MERANT][ODBC Oracle 8 driver][Oracle 8]ORA-03106: 致命的な 2 タスク  
通信プロトコルエラーが発生しました
```

install_directory>/db ディレクトリの *system_odbc.ini* ファイルに含まれる関連するデータソースに、次のエントリを追加します。

```
ArraySize=<value_greater_than 60000>
```

このように変更すると、複数行フェッチバッファが拡張されるため、エラーを回避できます。この変数のデフォルト値は 60000 バイトです。最大許容値は 4 G バイトです。

ODBC 接続プーリング

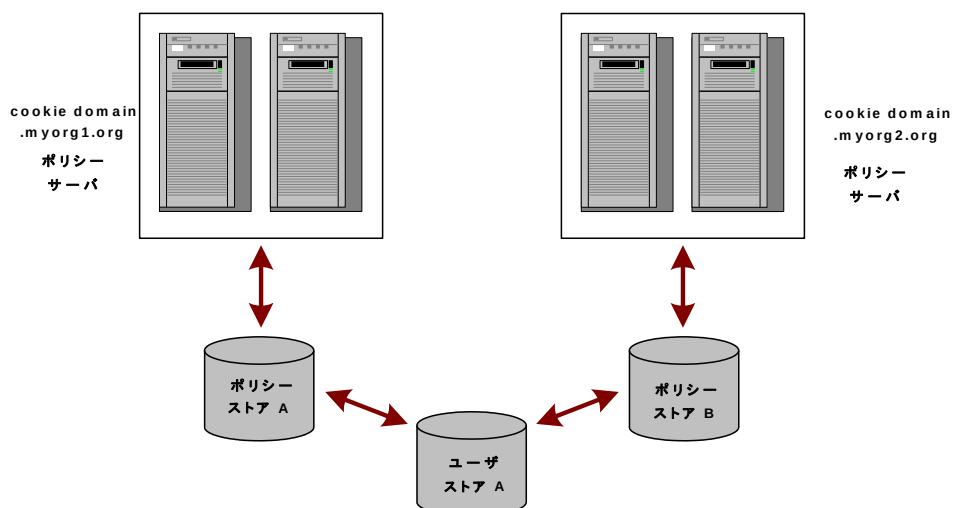
以下は ODBC 接続プーリングに関する説明です。

- 接続は ODBC データソースによってプールされます。1 つのデータソースが 1 つ以上のユーザディレクトリ(およびポリシーストアまたは別のポリシー サーバ上のストア)によって使用されると、使用可能な接続の数がユーザディレクトリおよびストアに指定された個々の数値の合計に達することがあります。
- 接続の共有については、SQL Server は Oracle よりも制限を受けます。クライアントに結果がフェッチされる間、2 つの呼び出し元は 1 つの開いている結果セットを共有できません。ポリシー サーバは SQL Server にサーバ側のカーソルを使用しますが、同時アクティビティの数は制限されます。マルチプロセッサマシンでは特に注意が必要です。接続数を増やせば、同時実行性は大幅に向上します。
- Oracle では、複数の呼び出し元が 1 つの接続を共有できますが、呼び出しは内部的に直接化できます。やはり、接続数を増やせば同時実行性は向上します。

- 接続が共有可能な場合、アクティブリクエストの数はプール内の接続の間で分散され、均衡化されます。
- いったん開いた接続はポリシー サーバを停止するまで閉じることができません。

複数ポリシー ストアの同じユーザ ディレクトリ接続の定義

すべてのポリシー サーバはポリシー ストアに接続されます。複数のポリシー サーバが 1 つのポリシー ストアを指すよう設定される場合があります。管理 UI のインスタンスを開くと、追加および変更したオブジェクトはポリシー サーバに関連付けられたポリシー ストア内に格納されます。下図で示すとおり、ユーザの SiteMinder 環境には、ポリシー サーバのデータを管理するための複数の独立したポリシー ストアが含まれている場合があります。



myorg1 のポリシー サーバはポリシー ストア A に接続されます。myorg2 のポリシー サーバはポリシー ストア B に接続されます。ただし、どちらの組織もユーザ ストア A からのデータを必要とします。

複数のポリシーストアから単一のユーザ ディレクトリへの接続を定義する方法

1. SiteMinder の展開に含まれるポリシー ストアの 1 つに関連付けられた 管理 UI を開きます。
2. ユーザ ディレクトリ接続を設定します。

ユーザ ディレクトリ接続を定義する際、[名前]フィールドに入力する値をメモしておきます。

3. SiteMinder の展開に含まれる他のポリシー ストアに関連付けられた 管理 UI を開きます。

4. 同じユーザ ディレクトリ接続を設定します。

ユーザ ディレクトリ接続を定義する際、ステップ 2 で使用したものと同じ名前を使用します。

たとえば、最初のポリシー ストアでユーザ ディレクトリ接続を定義する際に [名前] フィールドでユーザ ストア A の値を使用した場合、シングル サインオンを管理するには、[ユーザ ディレクトリ] ダイアログ ボックスの [名前] フィールドでユーザ ストア A の値を使用して、2 番目のポリシー ストアを設定する必要があります。

5. 同じユーザ ストアにアクセスする SiteMinder 展開内のすべての独立したポリシー ストアに対し、このプロセスを繰り返します。

個々の独立したポリシー ストアでユーザ ストアへの接続を定義する場合に同じユーザ ディレクトリ名を使用すると、SiteMinder は別のポリシー ストアのポリシーによって保護されたリソースにアクセスするユーザのためのシングル サインオンを維持できます。

ユーザ ディレクトリの内容の表示

管理 UI では、ユーザ ディレクトリの内容を表示できます。

ユーザ ディレクトリ内容の表示方法

1. 既存のユーザ ディレクトリ接続の [ユーザ ディレクトリ] ダイアログ ボックスを開きます。

注: ディレクトリ接続を保存するまで、ディレクトリの内容は表示できません。

2. [内容の表示] をクリックします。

[ディレクトリのコンテンツ] ウィンドウが開いてディレクトリの内容を表示します。

注: デフォルトではグループを表示します。個人を表示するには、検索機能を使用します。

ユーザ ディレクトリの検索

管理 UI にはユーザ ディレクトリの検索機能があります。この機能を使用すると、検索式またはディレクトリ属性に基づいてユーザやユーザのグループを表示できます。ユーザ ディレクトリ検索はユーザ ディレクトリのタイプごとに異なります。

注: [ポリシーのユーザ/グループ]ウィンドウから、ユーザ ディレクトリの検索機能にアクセスすることもできます。このウィンドウは、ポリシーのユーザおよびグループを追加または削除する場合に使用します。[ポリシーのユーザ/グループ]ウィンドウには、次で説明されているユーザ検索機能へのアクセスに使用する[検索]ボタンと同じボタンがあります。

ユーザ ディレクトリを検索する方法

1. 検索するディレクトリの[ユーザ ディレクトリ]ウィンドウを開きます。

2. [内容の表示]をクリックします。

[ディレクトリのコンテンツ]ウィンドウが開きます。このウィンドウの内容は、表示しているディレクトリタイプによって異なります。

3. 検索条件を入力し、[実行]をクリックします。

検索条件と一致する結果が開きます。

ユニバーサル ID

UID(ユニバーサル ID)は、SiteMinder が制御しているアプリケーション用の顧客固有ユーザ識別子です。多くの場合、UID はユーザ ログイン名とは異なります。

UIDを使用することにより、SiteMinder では新しいアプリケーションとレガシー アプリケーションの間のギャップを埋めたり、基になるユーザリポジトリの変更を回避したりします。つまり、アプリケーションの数や種類に関わらず、自動的にこの ID をアプリケーションに配信できるようにすることを目的としています。たとえば、ある会社で、従業員 ID 番号に基づいてユーザ情報を調べるレガシー アプリケーションを使用しているとします。ポリシー サーバではログイン名を使用してディレクトリ内のユーザを識別するため、UID は、ポリシー サーバがユーザを識別するための手段として使用されます。一方、他のアプリケーションで使用するため、ユーザ ディレクトリからは従来どおり従業員 ID 番号が取得されます。

管理 UI でユーザ ディレクトリ接続を設定する場合は、[ユーザ ディレクトリ]ウィンドウの[ユーザ属性]グループ ボックスで UID を指定できます。

SiteMinder での UID の使用方法

UID を使用してユーザ ディレクトリ接続を設定した場合は、ユーザが SiteMinder にログインすると、ポリシー サーバはユーザのディレクトリ プロファイル内の指定された属性から UID を取り出します。

この値は、セッション チケット (SESSIONSPEC) に保存されて、要求元の SiteMinder エージェントに返されます。Web エージェントはこの値を Web アプリケーションで利用できるようにヘッダ変数 (HTTP_SM_UNIVERSALID) に組み込みます。この値はセッション チケットの確認や許可要求のために、エージェント API を使用して設計されたオブジェクトやアプリケーションに渡されます。いずれの場合も処理が正常終了した結果として UID が返されます。

名前付き式

ユーザ ディレクトリは、組織的な情報、ユーザおよびグループの属性、個別のクレデンシャルといったユーザ属性を格納します。SiteMinder で使用するユーザ属性値には、ユーザ ディレクトリから直接読み取ることができるもののほか、必要になるたびに計算する必要があるものもあります。これらの計算式は名前を付けて、または名前を付けずに保存できます。

名前のない式は、レスポンスやルールなどのポリシー ストア オブジェクトに格納されます。名前付き式は、名前で参照して再利用できるポリシー ストア オブジェクトです。SiteMinder は名前付きと名前なしの両方のすべての式を評価し、ユーザ属性の計算値を判断します。

名前付き式を作成するには、管理者に適切な権限が必要です。

注: アクティブな式と名前付き式は同じではありません。どちらのタイプの式もランタイムに評価されるのは同じですが、以下の点で異なります。

- アクティブな式はブール式ですが、名前付き式は文字列、数値、ブール値を返すことができます。
- アクティブな式はそのまま参照されるため、使用するたびに再入力する必要がありますが、名前付き式は名前で参照されるため、どこからでも参照でき、再利用できます。

名前付き式のメリット

名前付き式のメリットは、以下のとおりです。

- 複数のユーザ ディレクトリへの適用

名前付き式は、名前によって参照でき、再使用することのできるオブジェクトとして、ポリシー ストアに格納されます。**SiteMinder** は、名前付き式を評価して、計算されたユーザ属性の値を判別します。

- 再使用性の推進

システム管理者は、名前付き式をそれぞれ 1 回作成します。ドメイン管理者は、ユーザ情報を取得するために、基礎となる式ではなく、式名を参照します。管理者は、ユーザ情報が必要になるたびに、式全体を再入力する必要がありません。

- データ入力エラーの削減

システム管理者は、名前付き式を 1 つの場所に作成して管理します。式を変更する必要がある場合、管理者は変更を 1 回のみ行います。

- 保守タスクの簡易化

ビジネス ロジックで式の変更が必要になった場合、システム管理者は 1 回のみ変更します。ドメイン管理者は、基礎となる変更に関係なく、式名を参照し続けることができます。

- セキュリティの強化

適切な権限を持つ管理者のみが名前付き式を作成できます。名前付き式は、権限が付与されている組み込み関数および任意の名前付き式 (秘密として指定される式を含む) を呼び出すことができます。

たとえば、名前付き式は、現在のユーザをグループに追加する秘密の式を呼び出せますが、名前なし式はこの処理を実行できません。この制約により、ドメイン管理者は、管理グループへの現在のユーザの追加など、セキュリティを無視する作業を行うことができなくなります。

名前付き式の定義

名前付き式は、名前で参照して再利用できるポリシー ストア オブジェクトです。
SiteMinder は、名前付き式を評価して、計算されたユーザ属性の値を判別します。

名前付き式には、以下の 2 つのタイプがあります。

- [仮想ユーザ属性](#) (P. 258)
- [ユーザ クラス](#) (P. 261)

仮想ユーザ属性

仮想ユーザ属性では、ユーザ情報を計算するための再利用可能な式を定義できます。ユーザ属性がユーザ ディレクトリによって一意に参照されない場合は、このタイプの式を使用します。この場合は、ユーザ属性を、属性、およびビジネスロジックによって確立される他の基準を使用して計算する必要があります。

仮想ユーザ属性名の式では、以下のいずれかのデータ型の値が返されます。

- 文字列
- 数値
- ブール値

仮想ユーザ属性には、プレフィクスとして「ポンド」記号(#)が付加されます。「ポンド」記号により、ユーザ属性名とマッピングで名前が競合することがなくなります。また、ユーザ属性値が計算されることをひとめで確認できるリマインダにもなります。

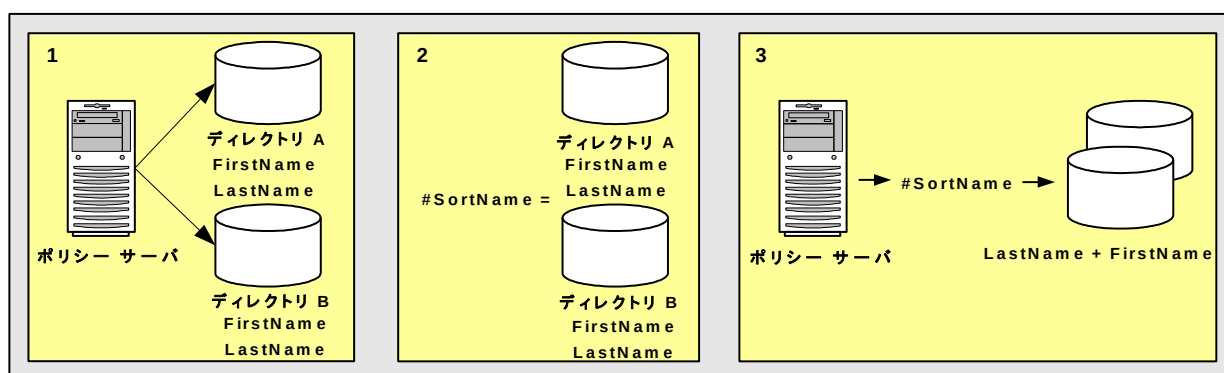
1 つの式として、仮想ユーザ属性には以下を含めることができます。

- ユーザ属性 (ディレクトリ固有のユーザ属性またはマップされたユーザ属性)
- ほかに名前付き式への参照
- SiteMinder 組み込み関数および式構文

仮想ユーザ属性ユース ケース

このユース ケースは、基本スキーマが異なる 2 つの LDAP ユーザ ディレクトリで、ユーザの姓と名を識別する方法を示す基本的なシナリオです。

以下の図は、ユーザ ディレクトリが異なるユーザの仮想ユーザ属性 **#SortName** (LastName、FirstName) をユーザ属性マッピングを使って計算する方法を示します。ユーザ属性マッピングでは、異なるユーザ ディレクトリの異なるユーザ属性名に共通名をマップできます。



- 2 つのユーザ ディレクトリでは、ユーザの姓と名の識別の方法が異なります。ユーザ属性マッピングを作成することで、この情報の共通表示を作成できます。
 - **FirstName** は、ディレクトリ A とディレクトリ B のユーザの名を識別する基本ディレクトリスキーマにマップされます。
 - **LastName** は、ディレクトリ A とディレクトリ B のユーザの姓を識別する基本ディレクトリスキーマにマップされます。
- #SortName** は、以下の式で両方のディレクトリのユーザのソート名を計算できる仮想ユーザ属性です。
(LastName + "," + FirstName)
- (FirstName + "," + LastName) として定義した **#SortName** という名前の仮想ユーザ属性を作成すれば、式 (LastName + "," + FirstName) を何度も入力する必要がなくなります。この式が必要になったときは、**#SortName** を入力するだけで済みます。

仮想ユーザ属性の定義

1 つ以上のユーザ ディレクトリで一意に参照されないユーザ情報を計算するには、仮想ユーザ属性を定義します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

仮想ユーザ属性を定義する方法

1. [ポリシー]-[式]をクリックします。
2. [名前付き式]-[名前付き式の作成]をクリックします。
[名前付き式の作成] ペインが開きます。
3. タイプが[式]である新しいオブジェクトが選択されていることを確認し、[OK]をクリックします。
[名前付き式の作成: <名前>] ペインが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. 仮想ユーザ属性を選択し、[一般]グループ ボックスの各フィールドに名前と説明を入力します。
5. [名前付き式の追加]グループ ボックスの[式]フィールドに、式を入力します。
6. (省略可) [名前付き式の追加]グループ ボックスで、[無効]チェック ボックスをオンにします。

名前付き式は無効として指定され、式エディタに表示されなくなります。名前付き、名前なしに関係なく、別の式から呼び出せなくなります。

7. (省略可) [名前付き式の追加]グループ ボックスで、[秘密]チェック ボックスをオンにします。

名前付き式は秘密として指定されます。他の名前付き式からのみ呼び出すことができ、名前なし式から呼び出せなくなります。

8. (省略可) [名前付き式の追加] グループ ボックスで、[編集] をクリックします。

[式エディタ] ペインが開きます。

9. [サブミット] をクリックします。

名前付き式の作成タスクが、処理のためにサブミットされます。

ユーザ クラス

ユーザ クラスでは、ユーザ情報を計算する再使用可能な式を定義できます。ユーザ属性がユーザ ディレクトリによって一意に参照されない場合は、このタイプの式を使用します。この場合は、ユーザ属性を、属性、およびビジネス ロジックによって確立される他の基準を使用して計算する必要があります。

ユーザ クラスは、ユーザが指定したクラスのメンバの場合は真を、そうでない場合は偽を返す式を指定します。

ユーザ クラスには、接頭辞としてアット マーク(@) が付きます。アット マークにより、ユーザ クラスの名前とのユーザ属性名やマッピングの競合が避けられます。また、アット マークにより、ユーザ属性値を計算することが視覚的にわかります。

ユーザ クラスは式で以下を含むことができます。

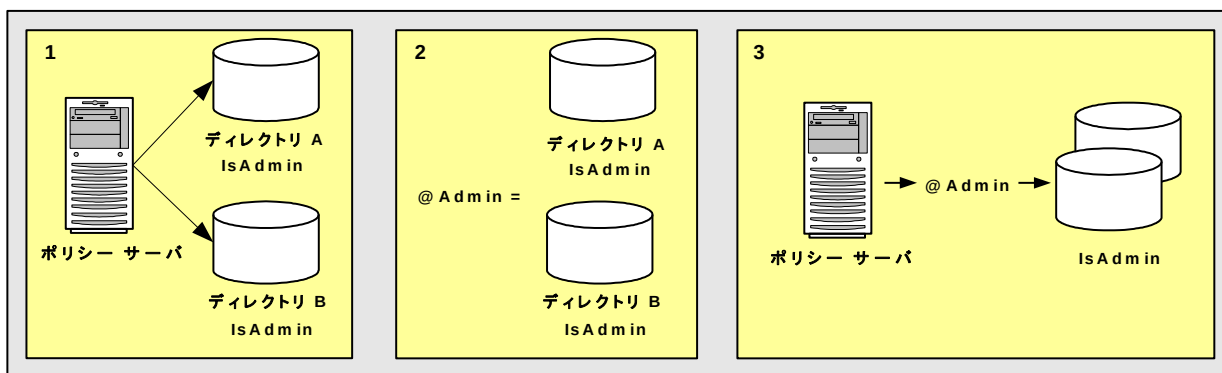
- ユーザ属性 (ディレクトリ固有のユーザ属性またはマップされたユーザ属性)
- ほかの名前付き式への参照
- SiteMinder 組み込み関数および式構文

ユーザ クラスはロールではありません。ロールはエンタープライズ ポリシー管理の機能です。ロールはユーザ クラスを使用できますが、その他にも関連付けられている情報があります。ロールの詳細については、「エンタープライズ ポリシー管理」を参照してください。

ユーザ クラス ユース ケース

このユース ケースは、基本スキーマが異なる 2 つの LDAP ユーザ ディレクトリで、管理者グループのメンバシップを識別する方法を示す基本的なシナリオです。

以下の図は、ユーザ ディレクトリが異なるユーザのユーザ クラス **@Admin** をユーザ属性マッピングを使って計算する方法を示します。ユーザ属性マッピングでは、異なるユーザ ディレクトリの異なるユーザ属性名に共通名をマップできます。



- 2 つのユーザ ディレクトリでは、管理者グループのメンバシップの識別の方法が異なります。ユーザ属性マッピングを作成することで、この情報の共通表示を作成できます。
 - **IsAdmin** は、ディレクトリ A の管理者グループのメンバシップを識別する基本ディレクトリスキーマにマップされます。
 - **IsAdmin** は、ディレクトリ B の管理者グループのメンバシップを識別する基本ディレクトリスキーマにマップされます。
- @Admin** は、両方のディレクトリ内のユーザが管理者かどうか判断するために、SiteMinder が評価するユーザ クラスのタイプを示す名前付き式です。
(IsAdmin)
- (IsAdmin) として定義した **@Admin** という名前のユーザ クラスを作成すれば、式 (IsAdmin) を何度も入力する必要がなくなります。この式が必要になったときは、**@Admin** を入力するだけで済みます。

ユーザ クラスの定義

1 つ以上のユーザ ディレクトリで一意に参照されないユーザ情報を計算するには、ユーザ クラス属性を定義します。計算の結果は **TRUE** または **FALSE** のみになります。結果がユーザに適用されるか、適用されないかのいずれかです。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ユーザ クラスを作成する方法

1. [ポリシー]-[式]をクリックします。
2. [名前付き式]-[名前付き式の作成]をクリックします。
[名前付き式の作成]ペインが開きます。
3. タイプが[式]である新しいオブジェクトが選択されていることを確認し、[OK]をクリックします。

[名前付き式の作成: <名前>]ペインが開きます。

注: フィールド、コントロール、およびそれぞれの要件については、[ヘルプ]をクリックしてください。

4. [ユーザ クラス]を選択し、[一般]グループ ボックスの各フィールドに、名前と説明を入力します。
5. [名前付き式の追加]グループ ボックスの[式]フィールドに、式を入力します。

注: 式はブール式である必要があります。

6. (省略可) [名前付き式の追加]グループ ボックスで、[無効]チェック ボックスをオンにします。

名前付き式は無効として指定され、[式エディタ]ペインに表示されなくなります。名前付き、名前なしに関係なく、別の式からは呼び出せません。

7. (省略可) [名前付き式の追加]グループ ボックスで、[秘密]チェック ボックスをオンにします。

名前付き式は秘密として指定されます。他の名前付き式から呼び出すことはできますが、名前なし式からは呼び出せません。

8. (省略可) [名前付き式の追加]グループ ボックスで、[編集]をクリックします。
[式エディタ]ペインが開きます。
9. [サブミット]をクリックします。
名前付き式の作成タスクが、処理のためにサブミットされます。

式エディタの使用方法

式エディタを使用して、以下を実行できます。

- SiteMinder の関数とオペレータ、ユーザ定義の名前付き式を検索する
- ブール式の作成

注: 式を直接入力する場合は、[キャンセル]をクリックし、[名前付き式の作成:<名前>]に戻ります。ここでは、[名前付き式の追加]グループ ボックスの[式]フィールドに式を入力できます。

式エディタでのブール式の作成プロセスは、2 つの部分で構成されています。プロセスのこれら 2 つの部分を実行する順番は自由です。

1. 条件の作成
2. 式の編集

プロセスの最初の部分では、条件を作成して[Infix 形式]グループ ボックスに追加できます。条件は、単一の SiteMinder 関数または演算子で構成される単純なブール式です。エディタで関数に設定できるパラメータは 3 つまでです。フォーマットは以下のとおりです。

```
FUNCTION_NAME(parameter_1[, parameter_2][, parameter_3])
```

演算子には、2 つのオペランドが必要です。フォーマットは以下のとおりです。

```
left_operand operator right_operand
```

条件はブール式であるため、結果はブール値になります。条件に結果が文字列になる関数や演算子が含まれている場合は、ブール値に変換されます。特に、「TRUE」、「true」、「YES」、「yes」は「TRUE」に変換されます。他の文字列値はすべて、FALSE に変換されます。

同様に、条件に結果が数値になる関数や演算子が含まれている場合は、ブール値に変換されます。ゼロ以外の数値はすべて TRUE に、またゼロは FALSE に変換されます。

以下のように、各条件は[Infix 形式]グループ ボックス上のフィールドに改行して表示され、1 つまたは 2 つのブール演算子によって上の行の条件に連結されます。

```
condition_1  
AND | OR | XOR [NOT] condition_2
```

プロセスのもう 1 つの部分では、条件の変更と削除、条件をグループ化するかこの変更、[Infix 形式]グループ ボックスのフィールドの条件を連結するブール演算子の変更などにより、式を編集できます。たとえば、以下のように条件をグループ化する方法を変更できます。

```
(condition_1  
AND condition_2)  
OR NOT condition_3
```

can become

```
condition_1  
AND (condition_2  
OR NOT condition_3)
```

関数を含む条件の作成

組み込みの SiteMinder 関数を含む条件を作成し、式エディタでその条件を式に追加することができます。

組み込みの SiteMinder 関数を含む条件を作成する方法

1. 関数のドロップダウンリストから関数名を選択するか、[式エディタ]ペインの[条件]グループ ボックスの[関数]フィールドに関数名を入力します。
2. [名前付き式]をクリックするか、[条件]グループ ボックスの[最初のパラメータ]フィールドに入力して、最初のパラメータを指定します。

注: [名前付き式]をクリックすると、[変数の検索]グループ ボックスが開きます。

3. (省略可) 名前付き式をクリックするか、[条件]グループ ボックスの[2 番目のパラメータ]フィールドに入力して、2 つ目のパラメータを指定します。

注: [名前付き式]をクリックすると、[変数の検索]グループ ボックスが開きます。

4. (省略可)ドロップダウンリストから[真]または[偽]を選択するか、[条件]グループ ボックスの[3 番目のパラメータ]フィールドに入力して、最後のパラメータを指定します。
5. [追加]をクリックします。
指定した関数が、[infix 形式]グループ ボックスおよび[結果として得られる形式]グループ ボックスに追加されます。

演算を含む条件の作成

組み込みの SiteMinder 演算を含む条件を作成し、式エディタでその条件を式に追加することができます。

組み込みの SiteMinder 演算を含む条件を作成する方法

1. [式エディタ]ペインの[条件]グループ ボックスのドロップダウンリストから、演算子のタイプと演算子を選択します。
2. [名前付き式]をクリックするか、[条件]グループ ボックスの[左パラメータ]フィールドに入力して、左オペランドを指定します。
注: [名前付き式]をクリックすると、[変数の検索]グループ ボックスが開きます。
3. [名前付き式]をクリックするか、[条件]グループ ボックスの[右パラメータ]フィールドに入力して、右オペランドを指定します。
注: [名前付き式]をクリックすると、[変数の検索]グループ ボックスが開きます。
4. [追加]をクリックします。
指定した演算が、[infix 形式]グループ ボックスおよび[結果として得られる形式]グループ ボックスに追加されます。

式を編集する方法

式エディタで作成した条件はそれぞれ、[infix 形式]グループ ボックスのフィールド内の別々の行に表示されます。式を作成する場合は、[infix 形式]グループ ボックスのボタンを使用することで、条件をグループ化すること、式を接続するブール演算子を変更することができます。

式の編集は、3 段階のプロセスです。1 つ目の手順には、4 つのオプションがあります。これらのオプションは、任意の順序で繰り返すことができます。

1. オプションを選択します。
 - [式で条件を変更する](#) (P. 267)
 - [式から条件を削除する](#) (P. 267)
 - [式の中で条件をグループ化する](#) (P. 268)
 - [式でブール演算子を変更する](#) (P. 269)
2. (省略可) 手順 1 を繰り返します。
3. [OK]をクリックして、式エディタを閉じます。

式で条件を変更する

式に含まれる条件は、式エディタの[infix 形式]グループ ボックスにある[変更]ボタンをクリックして変更できます。

式に含まれる条件を変更する方法

1. 条件をクリックして選択します。
2. [変更]をクリックします。

[編集]グループ ボックスに条件が表示されます。

式から条件を削除する

式に含まれる 1 つまたは複数の条件は、式エディタの[infix 形式]グループ ボックスにある[削除]ボタンをクリックして削除できます。

式から条件を削除する方法

1. 条件をクリックして選択します。

注: 複数の隣接した条件を選択するには、Shift キーを押したままクリックします。

2. [削除]をクリックします。

選択した条件が式から削除されます。

注: 複数の条件を選択して[削除]をクリックすれば、一度に削除できます。

式の中で条件をグループ化する

式に含まれる条件のグループは、式エディタの[infix 形式]グループ ボックスでかっこを追加または削除するボタンをクリックして変更できます。

式に含まれる条件のグループを変更する方法

1. 2 つ以上の隣接した条件をクリックして選択します。

注: 複数の隣接した条件を選択するには、**Shift** キーを押したままクリックします。

2. 以下の 2 つのボタンのいずれかをクリックします。

()

選択した条件の外側に丸かっこを追加します。

例:

condition_1

AND condition_2

変更後

(condition_1

AND condition_2)

Remove()

選択した条件の外側から丸かっこを削除します。

例:

```
(condition_1  
OR condition_2  
OR condition_3)
```

変更後

```
condition_1  
OR condition_2  
OR condition_3
```

編集後の式が、式エディタの[結果として得られる形式]および[**infix** 形式]グループ ボックスに表示されます。

式でブール演算子を変更する

式に含まれるブール演算子は、式エディタの[**infix** 形式]グループ ボックスにある以下のボタンのいずれかをクリックして変更できます。

- And/Or
- Not
- XOR
- 条件付き?:はい:いいえ

式に含まれるブール演算子を変更する方法

1. 1 つ、または複数の条件をクリックして選択します。

注: 複数の隣接した条件を選択するには、**Shift** キーを押したままクリックします。

2. 以下のいずれかのボタンをクリックします。

And/Or

ブール演算子 AND と OR を切り替えます。

例:

AND condition_1

変更後

OR condition_1

注: [And/Or] ボタンは XOR を AND に切り替えます。

Not

ブール演算子 NOT の追加と削除を切り替えます。

例:

AND condition_1

変更後

AND NOT condition_1

XOR

ブール演算子 AND と OR を XOR に切り替えます。

例:

AND condition_1

変更後

XOR condition_1

注: 排他的論理和 (XOR) 演算子は 2 つのブール オペランドを取り、両方ではなく一方のオペランドが TRUE の場合、TRUE を返します。

条件付き?はい:いいえ

条件付きの決定演算子を追加します。

例:

condition_1

変更後

condition_1 ? "YES" : "NO"

編集後の式が、式エディタの[結果として得られる形式]および[infix 形式]グループ ボックスに表示されます。

名前付き式の適用

このユースケースでは、衣料品小売会社で、顧客がクレジット限度に達するかそれを超えたら Web 上でクレジットによる購入を行えないようにするルールを定義するシナリオを表します。この会社のポリシーでは、顧客のクレジット限度を 1,000 ドルとし、従業員のクレジット限度を 2,000 ドルとしています。

このユースケースでは、SiteMinder 環境に 2 つのユーザ ディレクトリが含まれています。

- ディレクトリ A には、従業員が格納されます。従業員は顧客になることもあります。そのため、ディレクトリ A では、グループ `cn=Customers,ou=Groups,o=acme.com` のメンバである従業員を顧客として識別します。
- ディレクトリ B は顧客のみ格納します。すべてのユーザが顧客であるため、ディレクトリ B には顧客を識別するユーザ属性がありません。

属性マッピング、仮想ユーザ属性、およびユーザ クラスを使用して、どのように会社のクレジット ポリシーを満たすかを、以下に詳しく説明します。

1. ユーザ属性マッピングと、各ユーザ ディレクトリの顧客を識別するユニバーサルなスキーマまたは共通名を作成します。
 - a. ディレクトリ A (従業員) にはグループ名属性マッピングを作成します。
 - このマッピングの名前を **IsCustomer** とします。
 - **IsCustomer** を `cn=Customers,ou=Groups,o=acme.com` として定義します。
 - b. ディレクトリ B (顧客) には定数属性マッピングを作成します。
 - このマッピングの名前を **IsCustomer** とします。
 - **IsCustomer** を真に定義します。

注: **IsCustomer** が、ディレクトリ A とディレクトリ B 内の同じユーザ情報にマッピングする共通名です。この情報にアクセスするために、式で **IsCustomer** を使用することができます。
2. 定数属性マッピングと、各ユーザ ディレクトリの会社のクレジット限度を識別するユニバーサルなスキーマまたは共通名を作成します。
 - a. ディレクトリ A (従業員) には定数属性マッピングを作成します。
 - このマッピングの名前を **CreditLimit** とします。
 - **CreditLimit** を **2000** として定義します。

b. ディレクトリ B (顧客) には定数属性マッピングを作成します。

- このマッピングの名前を **CreditLimit** とします。
- **CreditLimit** を **1000** として定義します。

注: **CreditLimit** は、ディレクトリ A とディレクトリ B 内の同じユーザ情報にマッピングする共通名です。この情報にアクセスするために、式で **CreditLimit** を使用することができます。

3. **#CreditBalance** は仮想ユーザ属性とします。この属性によって、アカウント
イング データベースからユーザのクレジット残高を取得します。
4. 顧客のクレジット残高がクレジット限度よりも低い場合に **TRUE** 値を返すユー
ザ クラスを作成します。

- ユーザ クラスの名前を **@IsUnderCreditLimit** とします。
- **@IsUnderCreditLimit** を以下のように定義します。

(IsCustomer AND (#CreditBalance < CreditLimit))

注: この式は、SiteMinder 式の構文ルールに従います。

5. クレジット残高がクレジット限度よりも少なければ顧客が Web 上での購入を
行うことを可能にする EPM ロールを作成します。
- ロールの名前を **PurchaseWithCredit** とします。
 - ロールを **@IsUnderCreditLimit** として定義します。

注: EPM ロールの詳細については、「エンタープライズ ポリシー管理」を参照してください。

ユーザ属性マッピング

ポリシー サーバからユーザ ディレクトリに接続を設定する場合、SiteMinder の 7 つの組み込み属性をディレクトリ固有のユーザ属性名にマップできます。

- ユニバーサル ID
- 無効フラグ
- パスワード属性
- パスワード データ
- 匿名 ID
- E-MAIL(電子メール)
- チャレンジ/レスポンス

ユーザ属性マッピングは、SiteMinder で独自の共通名を定義し、その共通名を基本スキーマが異なる複数のユーザ ディレクトリのユーザ属性名にマップできるようにすることで、この機能を拡張しています。ポリシー サーバとユーザ ディレクトリの接続を設定した後、異なるユーザ ディレクトリでも共通名を使用して同じユーザ情報を参照できるようになります。

ユーザ属性マッピングの概要

ユーザ属性マッピングは 5 タイプ、名前付き式は 2 タイプあります。属性マッピング タイプは以下のとおりです。

- エイリアス
- グループ名
- マスク
- 定数
- 式

名前付き式タイプは以下のとおりです。

- 仮想ユーザ属性
- ユーザ クラス

ユーザ属性マッピングは名前付き式と似ていますが、以下に示す重要な相違点があります。

■ アクセス

- ユーザ属性マッピングには、変更できないユーザ属性値にマップされている読み取り専用 (R) タイプのものと、読み取りや変更が可能なユーザ属性値にマップされている読み取り/書き込み (R/W) 可能タイプのものがあります。

読み取り専用 (R) : ターゲットを読み取ることはできるが変更できないマッピングを示します。

読み取り/書き込み (RW) : ターゲットの読み取りや変更が可能なマッピングを示します。

- すべての名前付き式は読み取り専用 (R) です。

■ データ型

- ユーザ属性マッピングは、データ型が指定されているユーザ属性にマップされています。
- 名前付き式は、評価時に指定されたデータ型の結果になります。

■ 可視性

- ユーザ属性マッピングはグローバルではないため、適用するユーザ ディレクトリごとに定義する必要があります。
- 名前付き式はグローバルであるため、どのユーザ ディレクトリのどのユーザにも適用できます。

■ プレフィクス

- ユーザ属性マッピングはユーザ属性名と同じ構文ルールに従います。
- 名前付き式はユーザ属性名と同じ構文ルールに従い、以下のプレフィクスがあります。
 - 仮想ユーザ属性は番号記号 (#) で始まる必要があります。
 - ユーザ クラスはアットマーク (@) で始まる必要があります。

これら相違点の概要については、以下の表を参照してください。

ユーザ属性マッピング タイプ	データ型	可視性	プレフィクス
エイリアス (RW)	文字列、数値、ブール	ディレクトリ固有	いいえ

ユーザ属性マッピング タイプ	データ型	可視性	プレフィクス
グループ名 (RW)	ブール値	ディレクトリ固有	いいえ
マスク (RW)	ブール値	ディレクトリ固有	いいえ
定数 (R)	文字列、数値、ブール	ディレクトリ固有	いいえ
式 (R)	文字列、数値、ブール	ディレクトリ固有	いいえ

名前付き式タイプ	データ型	可視性	プレフィクス
仮想ユーザ属性 (R)	文字列、数値、ブール	global	#
ユーザ クラス (R)	ブール値	global	@

属性マッピングのしくみ

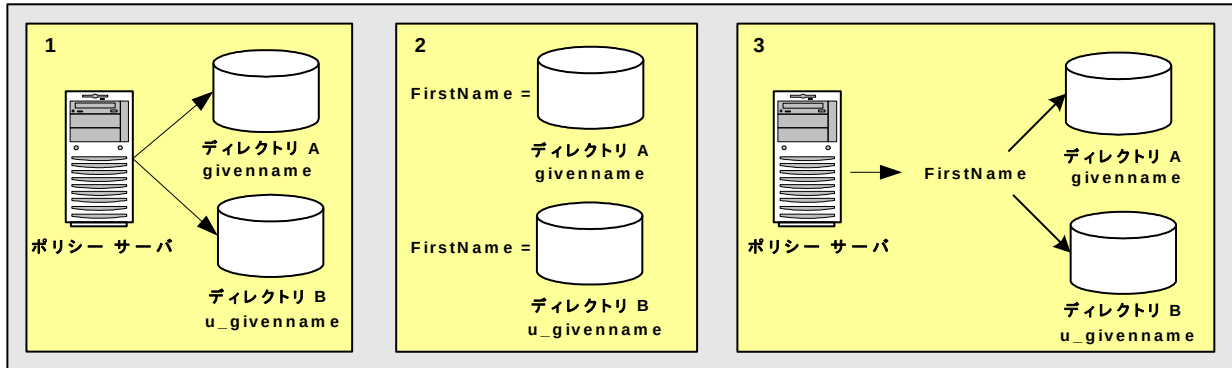
ユーザ ディレクトリには、組織的な情報、ユーザ属性とグループ属性、個々のクレデンシャルなど。ユーザ情報が格納されます。**SiteMinder** 環境内の複数のユーザ ディレクトリでは、格納するユーザ情報は同じなのに、そのユーザ情報を識別するために使用する基礎スキーマとユーザ属性名が異なることがよくあります。この結果、**SiteMinder** から見ると、同じユーザ情報の異なるビューが生じることになります。

ユーザ属性マッピングの目的は、ユニバーサルなスキーマを定義することによって、同じユーザ情報の共通のビューを作成することです。**SiteMinder** では、このユニバーサルなスキーマを使用して、複数のユーザ ディレクトリ間でユーザ情報を解決します。

ユーザ属性マッピングを定義するには、**共通名**を、ユーザ属性を識別する基礎となるディレクトリ スキーマにマッピングします。同じ共通名を、環境内の各ユーザ ディレクトリの基礎スキーマにマッピングすることで、ユーザ属性のユニバーサルなスキーマが生成されます。これにより、同じユーザ情報の共通のビューが作成されます。

このようなビューを作成することで、**SiteMinder** は、ディレクトリ タイプに関係なく、ユーザ属性を参照でき、ポリシーの数や、複数のユーザ ディレクトリを考慮するように設定する必要があるその他のオブジェクトの数を大幅に削減することができます。ユーザ属性マッピングはそれぞれ、それが定義されているユーザ ディレクトリに固有です。

ユーザ属性マッピングの基本的な概念を、以下に示します。



- 2つのユーザディレクトリは、別々の方法でユーザのファーストネーム(名)を識別します。
 - ディレクトリ A は、ユーザの名を **givenname** で識別します。
 - ディレクトリ B は、ユーザの名を **u_givenname** で識別します。この結果、同じユーザ情報の2つの異なる表現とビューが生成されます。
- FirstName** が、基礎となるディレクトリスキーマにマッピングする共通名です。
 - **FirstName** は、ディレクトリ A では **givenname** にマッピングされます。
 - **FirstName** は、ディレクトリ B では **u_givenname** にマッピングされます。
- FirstName** により、同じユーザ情報の共通のビューが生成されます。ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザの名を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **FirstName** を参照できます。**SiteMinder** は、**FirstName** が、ディレクトリ A では **givenname** であり、ディレクトリ B では **u_givenname** であると判断します。

属性マッピングの定義

ユーザ属性マッピングでは、1 つの共通名を、複数のユーザ ディレクトリの基礎となるディレクトリ スキーマにマッピングすることができます。同一の共通名を、環境内の各ユーザ ディレクトリの基礎スキーマにマッピングすることで、ユーザ情報のユニバーサルなスキーマを生成できます。ユーザ属性マッピングはそれぞれ、それが定義されているユーザ ディレクトリに固有です。

以下の 1 つ以上の属性マッピング タイプを使用して、複数のユーザ ディレクトリ間で同じユーザ情報を識別するマッピングを定義することができます。

- [エイリアス](#) (P. 277)
- [グループ名](#) (P. 280)
- [マスク](#) (P. 283)
- [定数](#) (P. 288)
- [式](#) (P. 291)

エイリアス

「エイリアス」属性マッピングでは、共通名を、基礎となるディレクトリ スキーマが使用する名前にマッピングしてユーザ属性を識別することができます。「エイリアス」属性マッピングにより、共通名は、読み取りまたは変更可能なユーザ属性値にマッピングされます。このタイプのアクセスのことを、読み取り/書き込み (RW) と呼びます。

「エイリアス」属性マッピングでは、以下のいずれかのデータ型のユーザ属性にマッピングできます。

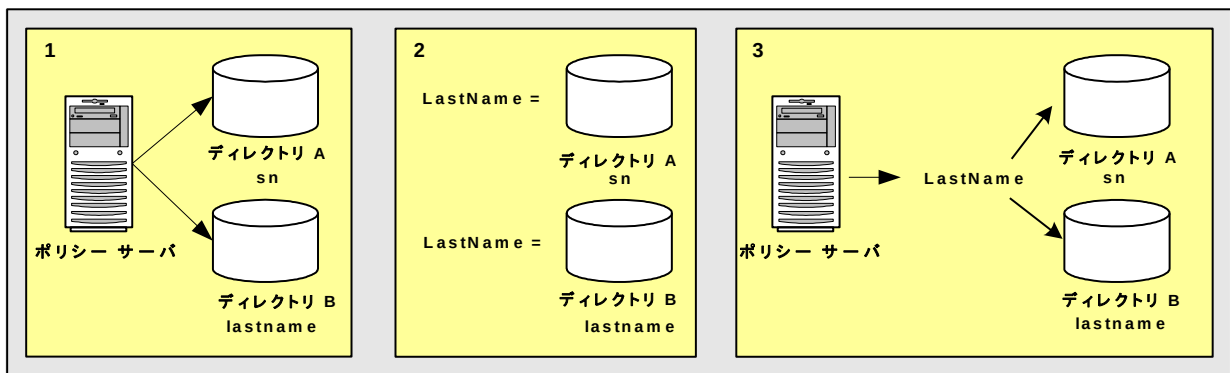
- 文字列
- 数値
- ブール値

エイリアス属性のユース ケース

このユース ケースでは、2 つの LDAP ユーザ ディレクトリが、別々の基礎スキーマを使用して、ユーザのラスト ネーム(姓)を識別する場合の基本的なシナリオを表します。

注: 高度なユース ケースは、「Apply Attribute Mapping」のセクションで説明されています。高度なユース ケースでは、異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法を詳しく説明します。

以下の図に、2 つの「エイリアス」属性マッピングによってユニバーサルなスキーマを定義し、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



- 2 つのユーザ ディレクトリは、別々の方法でユーザの姓を識別します。
 - ディレクトリ A は、ユーザの姓を `sn` で識別します。
 - ディレクトリ B は、ユーザの姓を `lastname` で識別します。この結果、同じユーザ情報の 2 つの異なる表現とビューが生成されます。
- `LastName` が、基礎となるディレクトリスキーマにマッピングする共通名、つまり「エイリアス」です。
 - `LastName` は、ディレクトリ A では `sn` にマッピングされます。
 - `LastName` は、ディレクトリ B では `lastname` にマッピングされます。
- `LastName` により、同じユーザ情報の共通のビューが生成されます。ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザの姓を必要とするポリシー、式、またはその他のオブジェクトを定義するときに `LastName` を参照できます。SiteMinder は、`LastName` が、ディレクトリ A では `sn` であり、ディレクトリ B では `lastname` であると判断します。

エイリアス属性マッピングの作成

共通名つまりエイリアスを、ユーザ ディレクトリの基礎スキーマで指定されたユーザ属性名にマッピングすることができます。ユーザ属性の有効なデータ型は、文字列、数値、またはブール値です。エイリアス属性マッピングを使用して、ユーザ ディレクトリ内のユーザ属性値を読み取りまたは変更することができます。ユーザ属性マッピングはディレクトリ固有です。

エイリアス属性マッピングを作成する方法

1. [ユーザ ディレクトリ: <名前>] ペインに移動します。
2. [属性マッピング リスト] グループ ボックスで[作成]をクリックします。
[属性マッピングの作成] ペインが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[属性マッピングの作成: <名前>] ペインが開きます。
4. [一般] グループ ボックスの各フィールドに、属性マッピングの共通名と説明を入力します。
注: 共通名はユーザ属性名と同じルールに従う必要があります。
5. [プロパティ] グループ ボックスで[エイリアス]を選択します。
6. [プロパティ] グループ ボックスの[定義]ボックスに、定義を入力します。

例:

名前: **FirstName**

定義: **givenname**

説明: 共通名 **FirstName** はユーザ属性名 **givenname** にマッピングされます。

7. (省略可) [無効]を選択して、この属性マッピングを無効にします。
8. [OK]をクリックします。

属性マッピングの作成タスクが、処理のためにサブミットされます。また、新しい属性マッピングが、[属性マッピング リスト] グループ ボックスに追加されます。

グループ名

グループ名属性では、共通名を、基礎となるディレクトリスキーマにマッピングして、ユーザが特定のグループに属しているかどうかを識別することができます。グループ名属性マッピングにより、共通名は、読み取りまたは変更可能なユーザ属性値にマッピングされます。このタイプのアクセスのことを、読み取り/書き込み(RW)と呼びます。

グループ名属性マッピングでは、結果としてブール値が返されます。ユーザが指定されたグループのメンバであると、マッピングの結果は **TRUE** 値になります。それ以外の場合、結果は **FALSE** になります。

グループ名属性マッピングとユーザクラスは、名前付き式の 1 つのタイプです。類似点は、以下のとおりです。

- 両方とも、グループメンバシップを決定するために使用されます。
- 両方とも、結果としてブール値が返されます。

グループ名属性マッピングとユーザクラスとの相違点は、以下のとおりです。

- グループ名属性マッピングは、特定のユーザディレクトリに対して定義されます。ユーザクラスはグローバルで、任意のユーザディレクトリ内の任意のユーザに適用できます。
- グループ名属性マッピングは、ユーザディレクトリ内のグループのユーザのメンバシップを変更できます。ユーザクラスは、ブール式であり、変更できません。
- ユーザクラスはアットマーク(@)で始まる必要があります。グループ名マッピングは異なります。

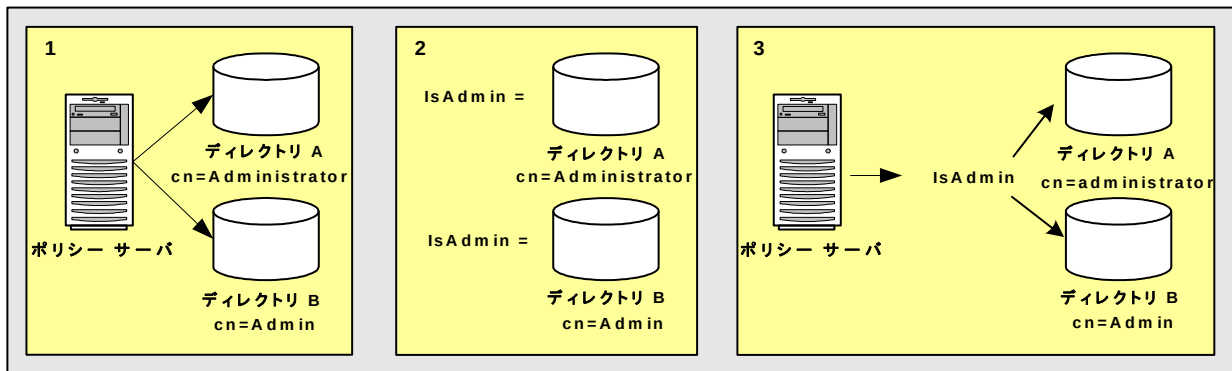
注: ユーザクラスの詳細については、「名前付き式」を参照してください。

グループ名のユースケース

このユースケースでは、2 つの LDAP ユーザディレクトリが、別々の基礎スキーマを使用して、管理者グループに属するユーザを識別する場合の基本的なシナリオを表します。

注: 高度なユースケースは、「Apply Attribute Mapping」のセクションで説明されています。高度なユースケースでは、異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法を詳しく説明します。

以下の図に、2つのグループ名属性マッピングによってユニバーサルなスキーマを定義し、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



- 2つのユーザディレクトリは、別々の方法で管理者グループのメンバシップを識別します。
 - ディレクトリ A は、管理者グループのメンバシップを `cn=Administrators,ou=groups,o=acme.com` として識別します。
 - ディレクトリ B は、管理者グループのメンバシップを `cn=Admin,ou=groups,o=acme.com` として識別します。

この結果、同じユーザ情報の2つの異なる表現とビューが生成されます。
- IsAdmin が、基礎となるディレクトリスキーマにマッピングする共通名です。
 - IsAdmin は、ディレクトリ A では `cn=Administrators,ou=groups,o=acme.com` にマッピングされます。
 - IsAdmin は、ディレクトリ B では `cn=Admin,ou=group,o=acme.com` にマッピングされます。
- IsAdmin により、管理者グループの共通のビューが生成されます。ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、管理者グループに適用するポリシー、式、またはその他のオブジェクトを定義するときに IsAdmin を参照できます。SiteMinder は、IsAdmin を、ディレクトリ A では `cn=Administrators,ou=groups,o=acme.com` であり、ディレクトリ B では `cn=Admin,ou=group,o=acme.com` であると判断します。

グループ名属性マッピングの作成

共通名を基礎となるディレクトリスキーマにマッピングして、ユーザが特定のグループに属しているかどうかを識別するグループ名属性を定義します。グループ名属性マッピングの結果はブール値です。グループ名属性マッピングを使用して、ユーザディレクトリ内のユーザグループ名を読み取りまたは変更することができます。ユーザ属性マッピングはディレクトリ固有です。

注: グループメンバシップを返すグローバルオブジェクトの作成については、本書の「名前付き式」の章のユーザクラスを参照してください。

グループタイプのユーザ属性マッピングを作成する方法

1. [ユーザディレクトリ: <名前>] ペインに移動します。
2. [属性マッピングリスト] グループボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK] をクリックします。
[属性マッピングの作成: <名前>] ペインが開きます。
4. [一般] グループボックスの各フィールドに、属性マッピングの共通名と説明を入力します。

注: 共通名はユーザ属性名と同じルールに従う必要があります。

5. [プロパティ] グループボックスで [グループ] を選択します。
6. [プロパティ] グループボックスの [定義] ボックスに、定義を入力します。

例:

名前: IsAdmin

定義: cn=administrators,ou=groups,o=acme.com

説明: 共通名 IsAdmin はグループ名

cn=administrators,ou=groups,o=acme.com にマッピングされます。ユーザが cn=administrators,ou=groups,o=acme.com のメンバである場合、IsAdmin は TRUE になります。

7. (省略可) [無効] を選択して、この属性マッピングを無効にします。
8. [OK] をクリックします。

属性マッピングの作成タスクが、処理のためにサブミットされます。また、新しい属性マッピングが、[属性マッピングリスト] グループボックスに追加されます。

マスク

マスク属性マッピングでは、共通名を、基礎となるディレクトリスキーマが使用する名前にマッピングして、ビットパターンが格納されているユーザ属性を識別することができます。マスク属性マッピングにより、共通名は、読み取りまたは変更可能なユーザ属性値にマッピングされます。このタイプのアクセスのことを、読み取り/書き込み (RW) と呼びます。

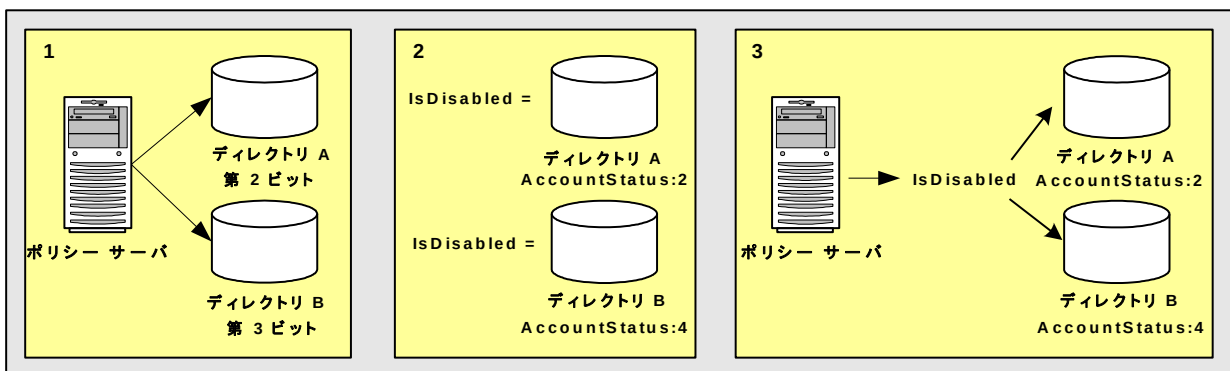
マスク属性マッピングでは、結果としてブール値が返されます。ユーザディレクトリ内のビットパターンが指定されたマスクと一致すると、マッピングの結果は TRUE 値になります。それ以外の場合、結果は FALSE になります。

マスクのユース ケース

このユース ケースでは、2 つの Active Directory ユーザディレクトリが、異なる基礎スキーマを使用して、無効なユーザアカウントを識別する場合の基本的なシナリオを表します。

注: 高度なユース ケースは、「Apply Attribute Mapping」のセクションで説明されています。高度なユース ケースでは、異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法を詳しく説明します。

以下の図に、2 つのマスク属性マッピングによってユニバーサルなスキーマを定義し、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



- 2つのユーザディレクトリには、**AccountStatus** というユーザ属性が含まれています。**AccountStatus** で、ユーザ情報はビットパターンで格納されます。ここで、各ビットはフラグです。
 - ディレクトリ A では、2 番目のビットが無効なアカウントにフラグを立てます。2 番目のビットが 1 に等しい場合、アカウントは無効です。
 - ディレクトリ B では、3 番目のビットが無効なアカウントにフラグを立てます。3 番目のビットが 1 に等しい場合、アカウントは無効です。この結果、同じユーザ情報の 2 つの異なる表現とビューが生成されます。
- IsDisabled** が、基礎となるディレクトリ スキーマにマッピングする共通名です。両方のディレクトリとも、**IsDisabled** は **AccountStatus** にマッピングされます。
 - ディレクトリ A では、**SiteMinder** は、**AccountStatus** の 2 番目のビットが設定されて、アカウントが無効であるかどうかを、ビットマスク 2 (10 進数) を使用して判断します。
 - ディレクトリ B では、**SiteMinder** は、**AccountStatus** の 3 番目のビットが設定されて、アカウントが無効であるかどうかを、ビットマスク 4 (10 進数) を使用して判断します。
- IsDisabled** により、無効なユーザ アカウントの共通のビューが生成されます。ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザのアカウント状況を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **IsDisabled** を参照できます。**SiteMinder** は、**IsDisabled** が、ディレクトリ A では **AccountStatus:2** であり、ディレクトリ B では **AccountStatus:4** であると判断します。

マスク属性マッピングの作成

共通名を、ユーザ属性名がユーザディレクトリの基礎スキーマで指定されているビットパターンにマッピングすることができます。マスク属性マッピングの結果はブール値です。マスクマッピングを使用して、ユーザディレクトリ内のユーザ属性値を読み取りまたは変更することができます。ユーザ属性マッピングはディレクトリ固有です。

マスクタイプのユーザ属性マッピングを作成する方法

- [ユーザディレクトリ: <名前>] ペインに移動します。
- [属性マッピングリスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。

3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

[属性マッピングの作成: <名前>] ペインが開きます。

4. [一般]グループ ボックスの各フィールドに、属性マッピングの共通名と説明を入力します。

注: 共通名はユーザ属性名と同じルールに従う必要があります。

5. [プロパティ]グループ ボックスで[マスク]を選択します。
6. [プロパティ]グループ ボックスの[定義]ボックスに、定義を入力します。

例:

名前: `IsDisabled`

定義: `AccountStatus:4`

説明:

共通名 `IsDisabled` はユーザ属性名 `AccountStatus` にマッピングされます。`AccountStatus` は、ビット パターン内に 1 つ以上の状態を格納します。たとえば、`AccountStatus` は 3 番目のビットにアカウント状態を格納します。アカウントが無効になると、3 番目のビットは 1 に設定されます。反対に、アカウントが有効になると、3 番目のビットは 0 に設定されます。

`SiteMinder` は、アカウントが無効かどうか判断するために、`AccountStatus` と指定された状態つまりマスク(この例では 4)のビット単位の AND 演算を実行します。アカウントが無効な場合、`IsDisabled` は `TRUE` です。

7. (省略可) [無効]を選択して、この属性マッピングを無効にします。
8. [OK]をクリックします。

属性マッピングの作成タスクが、処理のためにサブミットされます。また、新しい属性マッピングが、[属性マッピングリスト]グループ ボックスに追加されます。

マスク属性マッピングのビット マスク

マスク属性マッピングのビット マスクの目的は、ユーザ属性の他のビットの値をマスクすることにより、1 つまたは複数のビットの値をテストすることです。

マスク属性マッピングは、以下のように定義されます。

```
user_attribute_name:bit_mask
```

たとえば、AccountStatus という名前のユーザ属性があり、この AccountStatus のビット パターンに以下の 3 つのフラグが格納されているとします。

ビット パターン	フラグ
00?	アカウントは無効ですか?
0?0	パスワードは期限切れですか?
00	ゴールド メンバですか?

ビットが 1 に等しいとき、フラグは以下のように TRUE になります。

ビット パターン	アカウント ステータス
000 (0)	TRUE のフラグなし
001 (1)	アカウント無効
010 (2)	パスワード期限切れ
100 (4)	ゴールド メンバー
011 (3)	パスワード期限切れ、アカウント無効
101 (5)	ゴールド メンバ、アカウント無効
110 (6)	ゴールド メンバ、パスワード期限切れ
111 (7)	ゴールド メンバ、パスワード期限切れ、アカウント無効

注: 同等の 10 進数値をカッコ内に示します。

ユーザがゴールドメンバかどうかのみをテストする場合を考えます。このビットをテストするには、ゴールドメンバに対応するビットパターンをビットマスクとして選択します。たとえば、2進数では100、10進数では4を指定します。結果として作成されるマスク属性マッピングは以下のように定義されます。

AccountStatus:4

SiteMinder は AccountStatus とビットマスクの各ビットについて AND 演算を行い、結果がビットマスクと等しいかどうかテストします。これらが等しい場合、以下の表に示すように、テストしたビットの値は1、フラグは TRUE になります。

アカウントステータス	ビットマスク	ビット単位の AND の結果	ゴールドメンバですか?
000 (0)	100 (4)	000 (0)	FALSE
001 (1)	100 (4)	000 (0)	FALSE
010 (2)	100 (4)	000 (0)	FALSE
011 (3)	100 (4)	000 (0)	FALSE
100 (4)	100 (4)	100 (4)	TRUE
101 (5)	100 (4)	100 (4)	TRUE
110 (6)	100 (4)	100 (4)	TRUE
111 (7)	100 (4)	100 (4)	TRUE

注: 同等の10進数値をカッコ内に示します。

また、ビット マスクを使用して、ビット セットの値や一度に複数のビットをテストできます。アカウントが無効かどうか、パスワードの期限が切れているかどうかを知りたいとします。これらのビットをテストするには、**011** (2 進数) または **3** (10 進数) のビット マスクを指定します。結果として作成されるマスク属性マッピングは以下のように定義されます。

AccountStatus:3

SiteMinder は AccountStatus とビット マスクの各ビットについて AND 演算を行い、結果がビット マスクと等しいかどうかテストします。これらが等しい場合、以下の表に示すように、テストしたビットの値は 1、両方のフラグは TRUE になります。

アカウント ステータス	ビット マスク	ビット単位の AND の結果	両方のフラグが設定されていますか?
000 (0)	011 (3)	000 (0)	FALSE
001 (1)	011 (3)	001 (1)	FALSE
010 (2)	011 (3)	010 (2)	FALSE
011 (3)	011 (3)	011 (3)	TRUE
100 (4)	011 (3)	000 (0)	FALSE
101 (5)	011 (3)	001 (1)	FALSE
110 (6)	011 (3)	010 (2)	FALSE
111 (7)	011 (3)	011 (3)	TRUE

注: 同等の 10 進数値をカッコ内に示します。

定数

定数属性マッピングでは、共通名を、ディレクトリ内のどのユーザでも同一の値、つまり定数にマッピングすることができます。定数属性マッピングにより、共通名は、読み取り専用 (R) である定数にマッピングされるため、システム管理者以外は、このマッピングを変更できません。

定数属性マッピングでは、以下のいずれかのデータ型の定数にマッピングできます。

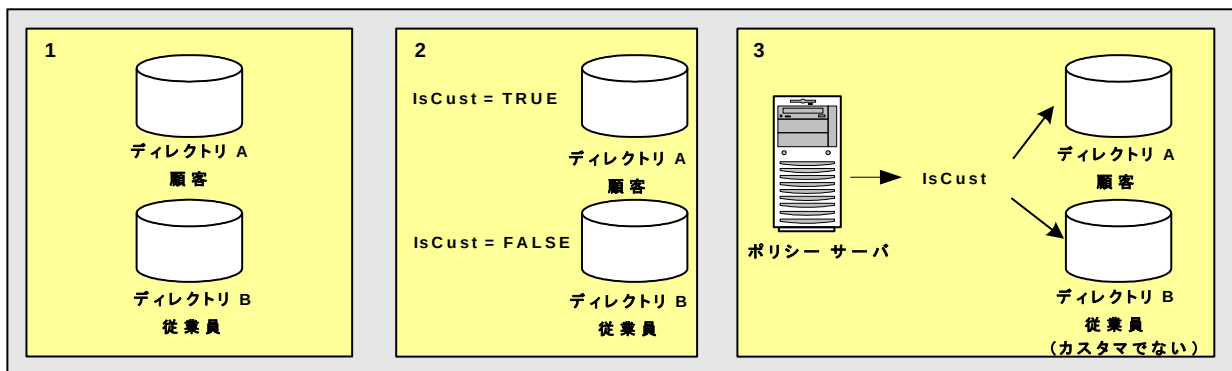
- 文字列
- 数値
- ブール値

定数のユース ケース

このユース ケースでは、一方のユーザ ディレクトリに顧客のみを格納し、もう一方のディレクトリに従業員のみを格納する場合の基本的なシナリオを表します。

注: 高度なユース ケースは、「Apply Attribute Mapping」のセクションで説明されています。高度なユース ケースでは、異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法を詳しく説明します。

以下の図に、2 つの定数属性マッピングによって異なるユーザ ディレクトリの異なる値をどのように表すかを詳しく示します。



1. ディレクトリ A には、顧客のみが格納されます。ディレクトリ B には、従業員のみが格納されます。
2. IsCust が、異なるディレクトリ内の異なる値にマッピングする共通名です。
 - IsCust は、ディレクトリ A では TRUE にマッピングされます。
 - IsCust は、ディレクトリ B では FALSE にマッピングされます。
3. ユーザが格納されている特定のディレクトリに関与することなく、ユーザが顧客かどうかを判断する必要があるポリシー、式、またはその他のオブジェクトを定義するときに IsCust を参照することができます。SiteMinder は、ディレクトリ A 内のユーザはすべて顧客であり、ディレクトリ B 内のユーザはすべて顧客ではない、と判断します。

定数属性マッピングの作成

共通名を定数値にマッピングして、ディレクトリ内のすべてのユーザに関する情報を伝達することができます。定数の有効なデータ型は、文字列、数値、またはブールです。定数属性マッピングを使用して、ユーザ ディレクトリ内のすべてのユーザに適用されるユーザ属性値を読み取ることができます。ユーザ属性マッピングはディレクトリ固有です。

定数属性マッピングを作成する方法

1. [ユーザ ディレクトリ: <名前>] ペインに移動します。
2. [属性マッピングリスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK] をクリックします。
[属性マッピングの作成: <名前>] ペインが開きます。
4. [一般] グループ ボックスの各フィールドに、属性マッピングの共通名と説明を入力します。
注: 共通名はユーザ属性名と同じルールに従う必要があります。
5. [プロパティ] グループ ボックスで [定数] を選択します。
6. [プロパティ] グループ ボックスの [定義] ボックスに、定義を入力します。

例:

名前: IsCustomer

定義: TRUE

説明: 共通名 IsCustomer は定数値 TRUE にマッピングされます。このユーザ ディレクトリには顧客のみが格納されているので、ユーザは常に顧客であり、IsCustomer は常に TRUE にマッピングされます。

7. (省略可) [無効] を選択して、この属性マッピングを無効にします。
8. [OK] をクリックします。

属性マッピングの作成タスクが、処理のためにサブミットされます。また、新しい属性マッピングが、[属性マッピングリスト] グループ ボックスに追加されます。

式

式属性マッピングでは、式に共通名をマップできます。式は、ユーザ ディレクトリの基本スキーマで指定した 1 つ以上のユーザ属性名を含むことができ、SiteMinder 式の構文ルールに従う必要があります。

式属性マッピングは、読み取れるが変更できない式に共通名をマップします。このタイプのアクセスは読み取り専用 (R) と呼ばれます。評価された式は、文字列、数値、またはブール値になります。

式属性マッピングと仮想ユーザ属性 (名前付き式の 1 タイプ) は、以下の点で類似しています。

- いずれも SiteMinder 式です。
- いずれの式も読み取り専用 (R) です。
- いずれの式の結果も以下のデータ型のうちの 1 つになります。
 - 文字列
 - 数値
 - ブール値

式属性マッピングは、仮想ユーザ属性と以下の点で異なります。

- 式属性マッピングは特定のユーザ ディレクトリに対して定義されます。仮想ユーザ属性はグローバルで、任意のユーザ ディレクトリの任意のユーザに適用できます。
- 仮想ユーザ属性はパウンド記号 (#) から始まる必要があります。式マッピングはそうではありません。

式のユース ケース

このユース ケースでは、式属性マッピングを使用して、1 つのディレクトリ内の複数のユーザ属性への参照を簡略化する方法を示します。

例:

このユース ケースでは、保護されているリソースがユーザのソート名 (last name, first name) を必要とする場合の基本的なシナリオを表します。ユーザ ディレクトリは、このユーザ属性を一意に参照するのではなく、ユーザのラスト ネーム (姓) を surname として、ユーザのファースト ネーム (名) を givenname として格納しています。

解決方法:

ユーザ ディレクトリの基礎スキーマによって指定されたユーザ属性名を使用してソート名を作成する式に、共通名をマッピングします。

- このマッピングの名前を **SortName** とします。
- 以下のように、**SortName** を定義します。

```
{surname + "," + givenname}
```

注: この式は、SiteMinder 式の構文ルールに従う必要があります。

結果:

ディレクトリ固有のスキーマに関係なく、ユーザのソート名を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **SortName** を参照できます。

SiteMinder は、式を使用してソート名を計算します。

注: 高度なユースケースは、「**Apply Attribute Mapping**」のセクションで説明されています。高度なユースケースでは、異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法を詳しく説明します。

式属性マッピングの作成

共通名を式にマッピングして、ユーザ ディレクトリの基礎スキーマで指定された 1 つ以上のユーザ属性名を参照することができます。式属性マッピングのデータ型は、文字列、数値、またはブールです。式属性マッピングを使用して、式の結果を読み取ることはできますが、値をユーザ ディレクトリに書き込むことはできません。

注: ユーザ属性マッピングはディレクトリ固有です。式として定義されるグローバルオブジェクトを作成するには、名前付き式を作成します。

式属性マッピングを作成する方法

1. [ユーザ ディレクトリ: <名前>] ペインに移動します。
2. [属性マッピングリスト] グループ ボックスで [作成] をクリックします。
[属性マッピングの作成] ペインが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK] をクリックします。
[属性マッピングの作成: <名前>] ペインが開きます。

4. [一般]グループ ボックスの各フィールドに、属性マッピングの共通名と説明を入力します。

注: 共通名はユーザ属性名と同じルールに従う必要があります。

5. [プロパティ]グループ ボックスで[式]を選択します。

注: [式]を選択すると[編集]ボタンがアクティブになります。[編集]ボタンをクリックすると式エディタが開きます。式は、SiteMinder 式の構文ルールに従う必要があります。

6. [プロパティ]グループ ボックスの[定義]ボックスに、定義を入力します。

例:

名前: SortName

定義: (surname + ',' + givenname)

説明: 共通名 SortName は、ユーザ属性名 surname および givenname を参照する式にマッピングされます。

7. (省略可)[無効]を選択して、この属性マッピングを無効にします。

8. [OK]をクリックします。

属性マッピングの作成タスクが、処理のためにサブミットされます。また、新しい属性マッピングが、[属性マッピングリスト]グループ ボックスに追加されます。

ユーザ属性マッピングの適用

SiteMinder 環境内の複数のユーザ ディレクトリでは、格納するユーザ属性は同じなのに、そのユーザ属性を識別するために使用する基礎スキーマが異なることがよくあります。この例では、衣料品小売会社が、異なるタイプの 2 つのユーザ ディレクトリを使用しています。ディレクトリ A は、従業員専用の内部 LDAP ユーザ ディレクトリです。ディレクトリ B は、顧客専用の ODBC ユーザ ディレクトリです。ユーザ属性マッピングはそれぞれ、それが定義されているユーザ ディレクトリに固有です。

以下の表に、ディレクトリ A とディレクトリ B が、異なる基礎スキーマを使用して、同じユーザ情報を識別する場合について詳しく示します。これに伴うユースケースでは、異なる属性マッピングを使用して、同じユーザ情報の共通のビューを作成するユニバーサルなスキーマを定義する方法を説明します。SiteMinder から見ると、ディレクトリは操作的には同一です。

属性の説明	ディレクトリ A (LDAP)	ディレクトリ B (ODBC)
ユーザのファースト ネーム (名)	givenname	u_first_name
ユーザのラスト ネーム (姓)	surname	u_last_name
ユーザのソート名 (last name, first name)	ユーザ ディレクトリは、ユーザ属性を一意に格納しません。	sort_name
ユーザが顧客であるか	group:cn=customer,ou=groups,o=acme.com	ユーザは常に顧客である
ユーザ アカウントが無効か	ユーザ ディレクトリには AccountStatus 属性があります。これは、フラグのセットです。2 番目のビットは無効なアカウントを示します。	u_disabled

ファースト ネーム (名) のユース ケース

このユース ケースでは、2 つの別名属性マッピングを使用して、ディレクトリ A とディレクトリ B でファースト ネーム (名) ユーザ属性を表す方法を示します。

例:

ユーザ ディレクトリ A は、ユーザの名を givenname で識別します。ユーザ ディレクトリ B は、ユーザの名を u_first_name で識別します。この結果、同じユーザ情報の 2 つの異なる表現とビューが生成されます。

解決方法:

1. ディレクトリ A 用の別名属性マッピングを作成します。
 - このマッピングの名前を **FirstName** とします。
 - **FirstName** を **givenname** として定義します。
2. ディレクトリ B 用の別名属性マッピングを作成します。
 - このマッピングの名前を **FirstName** とします。
 - **FirstName** を **u_first_name** として定義します。

結果:

FirstName により、同じユーザ情報の共通のビューが生成されます。ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザの名を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **FirstName** を参照できます。**SiteMinder** は、ディレクトリ A 内のユーザを参照するときはユーザの名を **givenname** によって識別し、ディレクトリ B 内のユーザを参照するときは **u_first_name** によって識別する、と判断します。

ラストネーム(姓)のユースケース

このユースケースでは、2 つの別名属性マッピングを使用して、ディレクトリ A とディレクトリ B でラストネーム(姓)ユーザ属性を表す方法を示します。

例:

ユーザディレクトリ A は、ユーザの姓を **surname** で識別します。ディレクトリ B は、ユーザの姓を **u_last_name** で識別します。この結果、同じユーザ情報の 2 つの異なる表現とビューが生成されます。

解決方法:

1. ディレクトリ A 用の別名属性マッピングを作成します。
 - このマッピングの名前を **LastName** とします。
 - **LastName** を **surname** として定義します。
2. ディレクトリ B 用の別名属性マッピングを作成します。
 - このマッピングの名前を **LastName** とします。
 - **LastName** を **u_last_name** として定義します。

結果:

ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザの姓を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **LastName** を参照できます。**SiteMinder** は、ディレクトリ A 内のユーザを参照するときはユーザの姓を **surname** によって識別し、ディレクトリ B 内のユーザを参照するときは **u_last_name** によって識別する、と判断します。

ソート名のユース ケース

このユース ケースでは、**計算式属性マッピング**と**別名属性マッピング**を使用して、ディレクトリ A とディレクトリ B 内のユーザのソート名を表す方法を示します。

例:

1. ディレクトリ A は、ユーザのソート名を一意に識別しません。ディレクトリ A では、ユーザの名が **givenname** として、ユーザの姓が **surname** として格納されます。
2. ディレクトリ B は、ユーザのソート名を **sort_name** で識別します。

この結果、同じユーザ情報の 2 つの異なる表現とビューが生成されます。

解決方法:

1. ディレクトリ A 用の**計算式属性マッピング**を作成します。

- このマッピングの名前を **SortName** とします。
- 以下のように、**SortName** を定義します。

```
(surname + "," + givenname)
```

注: この式は、**SiteMinder** 式の構文ルールに従う必要があります。

2. ディレクトリ B 用の**別名属性マッピング**を作成します。

- このマッピングの名前を **SortName** とします。
- **SortName** を **sort_name** として定義します。

結果:

ディレクトリは操作的には同一なので、ディレクトリ固有のスキーマに関係なく、ユーザのソート名を必要とするポリシー、式、またはその他のオブジェクトを定義するときに **SortName** を参照できます。**SiteMinder** は、ディレクトリ A 内のユーザを参照するときはソート名を計算する必要があり、ディレクトリ B 内のユーザを参照するときはソート名は **sort_name** であると判断します。

顧客ユース ケース

このユース ケースでは、グループメンバシップ属性マッピングと定数属性マッピングを使用して、ディレクトリ A とディレクトリ B の顧客を識別できる方法について示します。

例：

1. ディレクトリ A は従業員を格納します。また、会社の従業員は顧客にもなれるため、ディレクトリ A では、以下に属す従業員を顧客として識別します。

`cn=Customers, ou=Groups, o=acme.com`

2. ディレクトリ B は顧客のみ格納します。ディレクトリ B には、顧客を識別するユーザ属性はありません。これは、ディレクトリ B に格納されている以上、そのユーザは必ず顧客であるためです。

解決方法：

1. ディレクトリ A のグループメンバシップ属性マッピングを以下のように作成します。

- このマッピングの名前を **IsCustomer** とします。
- 以下のように、**IsCustomer** を定義します。

`cn=Customers, ou=Groups, o=acme.com`

2. ディレクトリ B の定数属性マッピングを以下のように作成します。

- このマッピングの名前を **IsCustomer** とします。
- **IsCustomer** を真に定義します。

結果：

ディレクトリは操作上は同一であるため、ユーザが顧客かどうか判断する必要があるポリシー、式、その他のオブジェクトを定義する際は、ディレクトリ固有のスキーマを考慮することなく、**IsCustomer** を参照できます。SiteMinder は、ディレクトリ A の参照時は `cn=Customers, ou=Groups, o=acme.com` に属しているユーザを顧客と判断します。また、ディレクトリ B の参照時にはすべてのユーザを顧客と判断します。

アカウント ステータス ユース ケース

このユース ケースでは、マスク属性マッピングと計算式属性マッピングを使用して、ディレクトリ A とディレクトリ B で無効になっているユーザ アカウントを識別できる方法について示します。

例:

1. ディレクトリ A は一連のフラグである、**AccountStatus** という名前のユーザ属性で無効なアカウントを識別します。2 番目のビットは無効なアカウントを示します。
2. ディレクトリ B は無効なアカウントを **u_disabled** という名前のユーザ属性で識別します。**u_disabled** が「y」と等しいとき、アカウントは無効になります。**u_disabled** が「n」と等しいとき、アカウントはアクティブです。

解決方法:

1. ディレクトリ A のマスク属性マッピングを以下のように作成します。
 - このマッピングの名前を **IsDisabled** とします。
 - マッピングを **AccountStatus:2** と定義します。これでは、
 - **AccountStatus** にビット パターンを格納します。
 - ビット マスクは 2 (10 進数) です。
2. ディレクトリ B の計算式属性マッピングを以下のように作成します。
 - このマッピングの名前を **IsDisabled** とします。
 - マッピングを以下のように定義します。

```
(u_disabled = "y")
```

ここで **u_disabled** はブール式です。

結果:

ディレクトリは操作上は同一であるため、ユーザのアカウント ステータスを判断する必要があるポリシー、式、その他のオブジェクトを定義する際は、ディレクトリ固有のスキーマを考慮することなく、**IsDisabled** を参照できます。**SiteMinder** は、ユーザが無効かどうかを、ディレクトリ A の参照時はビット パターンをチェックして判断し、ディレクトリ B の参照時は計算を実行して判断します。

第 8 章：ディレクトリマッピング

このセクションには、以下のトピックが含まれています。

[ディレクトリ マッピングの概要](#) (P. 299)

[ディレクトリ マッピングの要件](#) (P. 301)

[サポートされているディレクトリ マッピング](#) (P. 301)

[認証と認可のディレクトリ マッピングを設定する方法](#) (P. 301)

[AuthValidate ディレクトリ マッピングを設定する方法](#) (P. 303)

[ディレクトリ マッピングの例](#) (P. 305)

[ディレクトリ マッピングとレスポンス](#) (P. 308)

ディレクトリ マッピングの概要

SiteMinder では、ユーザの認証と許可を同じユーザ ディレクトリに対して実行することを前提としています。ほとんどの場合はこのデフォルトの動作で問題ありませんが、SiteMinder では、ユーザの認証と許可に別々のディレクトリを使用することもできます。この機能をディレクトリ マッピングといいます。この機能が特に役立つのは、認証情報が中央ディレクトリに格納されている一方で、許可情報が特定のネットワークアプリケーションと関連付けられた異なるユーザディレクトリに格納されている場合です。

注：ディレクトリ マッピングでは、インパーソネーションはサポートされていません。インパーソネーションの対象ユーザは、ドメインに関連付けられている認証ディレクトリ内で一意である必要があります。そうでない場合、インパーソネーションは失敗します。

認証ディレクトリから許可ディレクトリにマッピングするには、次の 3 つの手順に従います。

1. ユーザ ディレクトリ接続をセットアップします。

マッピングで認証ディレクトリまたは許可ディレクトリとして指定するディレクトリ接続は、[ユーザ ディレクトリ] ペインで設定する必要があります。

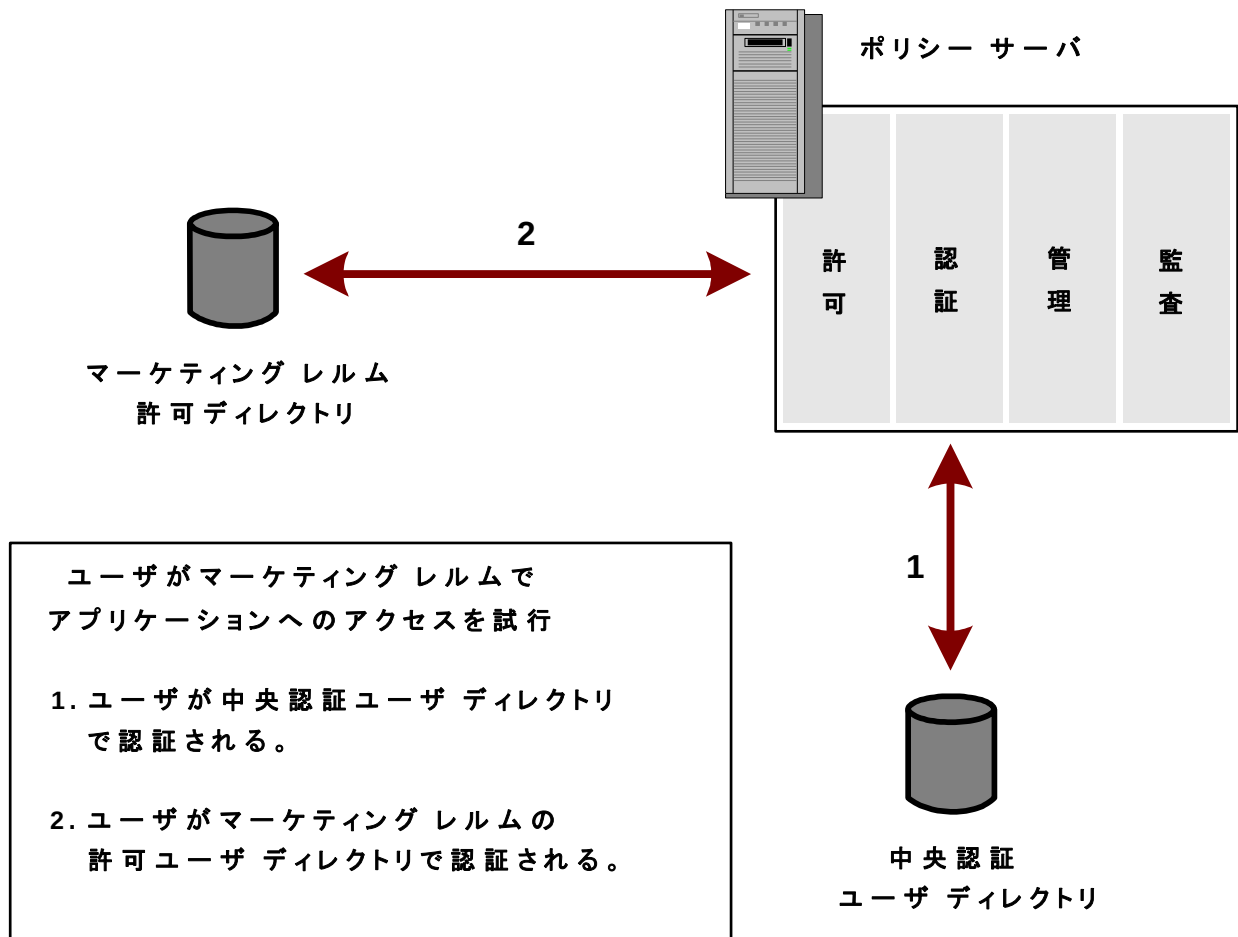
2. ディレクトリ マッピングの設定

ポリシー サーバは、ディレクトリ マッピングを使用して、認証済みのユーザを別の許可ディレクトリで探します。

3. レルムにディレクトリ マッピングを割り当てます。

ディレクトリ マッピングを特定のレルムに関連付けることで、ネットワーク内の特定のリソースに対してユーザを許可するときに使用するディレクトリを定義することができます。

たとえば、次の図に示すように、企業内の全ユーザが単一の中央ユーザ ディレクトリに対して認証される一方で、マーケティング部はユーザディレクトリを別に所有し、ここに部署内のスタッフの許可データが収められているとします。ポリシー サーバを使用して、マーケティング許可ユーザ ディレクトリに対するディレクトリ マッピングを設定し、マッピングで指定された許可ディレクトリを使用するマーケティング アプリケーション用のレルムを作成することができます。これによって、ユーザがマーケティング アプリケーションにアクセスするたびに、ポリシー サーバは中央のユーザ ディレクトリに対してユーザを認証し、マーケティング ユーザ ディレクトリに対してユーザを許可することになります。



ディレクトリ マッピングの要件

ディレクトリ マッピングでは、ポリシー サーバへのユーザ ディレクトリ接続が、許可ディレクトリや検証ディレクトリだけでなく、認証ディレクトリにも必要になります。

注: ユーザ ディレクトリ接続の作成の詳細については、「ユーザ ディレクトリ接続の概要」を参照してください。

サポートされているディレクトリ マッピング

以下の表では、サポートされているディレクトリ マッピングのタイプと、認証ディレクトリを許可ディレクトリまたは検証ディレクトリにマッピングする方法について説明します。

許可ディレクトリ/検証ディレクトリ			
認証 ディレクトリ	LDAP	リレーショナル データベース	Windows NT
LDAP	同一の DN ユニバーサル ID	ユニバーサル ID	該当なし
AD	同一の DN ユニバーサル ID	ユニバーサル ID	該当なし
リレーショナル データベース	ユニバーサル ID	同一の DN ユニバーサル ID	該当なし
Windows NT	ユニバーサル ID	ユニバーサル ID	同一の DN

認証と認可のディレクトリ マッピングを設定する方法

認証と認可のディレクトリ マッピングの設定は 2 段階のプロセスです。

1. ディレクトリ マッピングの設定
2. 許可ディレクトリのレルムへの割り当て

ディレクトリ マッピングの設定

ディレクトリ マッピングを設定して、あるディレクトリに対してユーザを認証する一方で、別のディレクトリに対してユーザを許可できます。

ディレクトリ マッピングの設定方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [認証/許可マッピング]-[ディレクトリ マッピングの作成]をクリックします。

[ディレクトリ マッピングの作成]ウィンドウが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. それぞれのリストから、認証と許可のディレクトリを選択します。
4. [同一の DN]ラジオ ボタンまたは[ユニバーサル ID]ラジオ ボタンを選択します。
5. [サブミット]をクリックします。

ディレクトリ マッピングの作成タスクを処理できるよう送信します。

許可ディレクトリのレルムへの割り当て

ディレクトリ マッピングをレルムに割り当て、ポリシー サーバが 1 つのディレクトリではユーザを認証し、別のディレクトリでユーザを許可できるようにします。ポリシー サーバはレルム内で指定されている許可ディレクトリを使用してユーザの許可を行います。

ディレクトリ マッピングをレルムに割り当てる方法

1. ディレクトリ マッピングを割り当てるレルムを開きます。
2. [ディレクトリ マッピング]リストの認証されたユーザを許可するためにレルムが使用するユーザ ディレクトリを選択します。

デフォルトの値は、ディレクトリ マッピングがないことを表します。つまり、ユーザがそのレルム内のリソースにアクセスを試みると、認証ディレクトリが許可ディレクトリとして使用されます。このリストには、既存のディレクトリ マッピングで許可ディレクトリとして設定されたユーザ ディレクトリだけが表示されます。

重要: 1 つのレルムごとに 1 つの許可ディレクトリのみをマップできます。

3. [サブミット]をクリックします。

ポリシー サーバがディレクトリ マッピングを保存します。レルムにアクセスするユーザは正常に認証され、レルムで指定されたディレクトリに対して許可されます。

詳細情報:

[レルムの設定](#) (P. 487)

AuthValidate ディレクトリ マッピングを設定する方法

AuthValidate ディレクトリ マッピングは、認証と許可ディレクトリ マッピングの拡張です。これら 2 タイプのディレクトリ マッピングにより、ユーザはあるユーザ ディレクトリを認証したうえで別のユーザ ディレクトリを許可できます。どちらの場合でも、ディレクトリ マッピング タイプは、さらに同一の DN またはユニバーサル ID として指定できます。

AuthValidate ディレクトリ マッピングは、以下の 3 つの点で認証と許可ディレクトリ マッピングを拡張します。

- AuthValidate ディレクトリ マッピングにより、あるポリシー サーバに接続されている認証ユーザ ディレクトリを、別のポリシー サーバに接続されている検証ユーザ ディレクトリにマップできます。ユーザ ディレクトリは OID とディレクトリ名を基に探します。
- AuthValidate ディレクトリ マッピングにより、さまざまなタイプのユーザ ディレクトリが可能になります。
- AuthValidate ディレクトリ マッピングでは、DN によるユーザの検索に失敗した場合にユニバーサル ID によるユーザの検索が実行されます。

認証/検証ディレクトリ マッピングの設定

ユーザの認証に使用するディレクトリと、ユーザの検証に使用するディレクトリを別々にするように、認証/検証ディレクトリ マッピングを設定することができます。

認証/検証ディレクトリ マッピングを設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [認証/検証マッピング]-[認証/検証ディレクトリ マッピングの作成]をクリックします。

[認証/検証ディレクトリ マッピングの作成]ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [認証ディレクトリ]フィールドに、ユーザの認証に使用するディレクトリの名前を入力します。
4. [検証ディレクトリ]リストから、ユーザの検証に使用するディレクトリを選択します。
5. [同一の DN]ラジオ ボタンまたは[ユニバーサル ID]ラジオ ボタンを選択します。
6. [サブミット]をクリックします。

認証/検証ディレクトリ マッピングの作成タスクが、処理のためにサブミットされます。

既存のレルムへの認証/検証ディレクトリ マッピングの割り当て

既存のレルムに認証/検証ディレクトリ マッピングを割り当てることができます。これにより、ポリシー サーバはあるユーザ ディレクトリに対してユーザを認証したうえで、別のユーザ ディレクトリに対してそのユーザを検証できます。ポリシー サーバが認証済みユーザを検証する際、そのユーザにはレルムにあるリソースへのアクセス権が付与されます。

既存のレルムに認証/検証ディレクトリ マッピングを割り当てる方法

1. [ポリシー]、[ドメイン]、[レルム]、[レルムの変更]をクリックします。
2. 変更するレルムを選択します。

3. [ディレクトリ マッピング]ドロップダウンリストから、検証ディレクトリとして使用するユーザ ディレクトリを選択します。

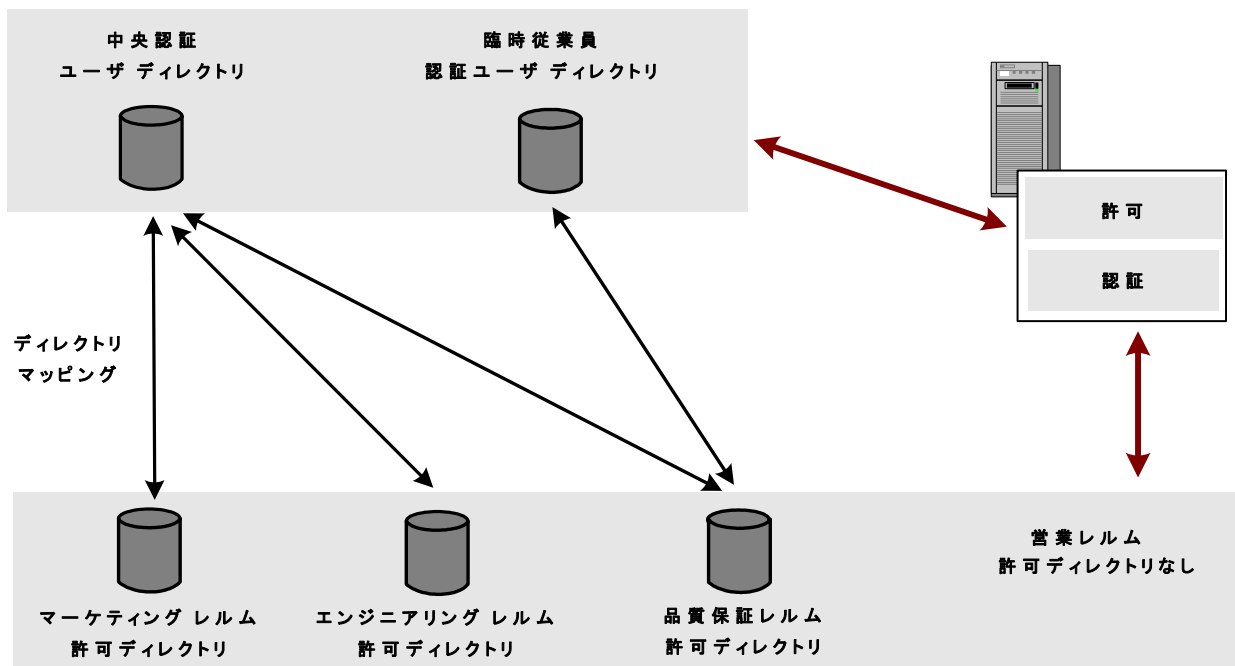
注: [ディレクトリ マッピング]ドロップダウンリストには、既存のディレクトリ マッピングで認証ディレクトリ、または検証ディレクトリとして設定されているユーザ ディレクトリのみが含まれます。選択したディレクトリ マッピングがデフォルトの場合、ユーザがレルムにあるリソースにアクセスしようとすると、認証ディレクトリが検証ディレクトリとして使用されます。

4. [サブミット]をクリックします。

レルムの変更タスクが処理のためにサブミットされます。

ディレクトリ マッピングの例

ネットワークリソースへのアクセス権があるユーザを適切に認証して許可するためには、複数のディレクトリ マッピングが必要になることもあります。以下の図に、複数のディレクトリ マッピングが必要になる可能性がある単純なサンプル ケースを示します。



この例では、それぞれ許可ユーザ ディレクトリが異なる 3 つのレルムと、専用の許可ユーザ ディレクトリがない 1 つのレルムがあります。ユーザ認証情報は、2 つの異なる認証ディレクトリに格納されています。ポリシー サーバは、ユーザ情報が格納されている場所に基づいて、認証ディレクトリのいずれかと、要求されたリソースがマーケティング、エンジニアリング、または品質保証のレルムにある場合は許可ディレクトリのいずれかを使用します。

従業員がエンジニアリング レルムのリソースにアクセスする

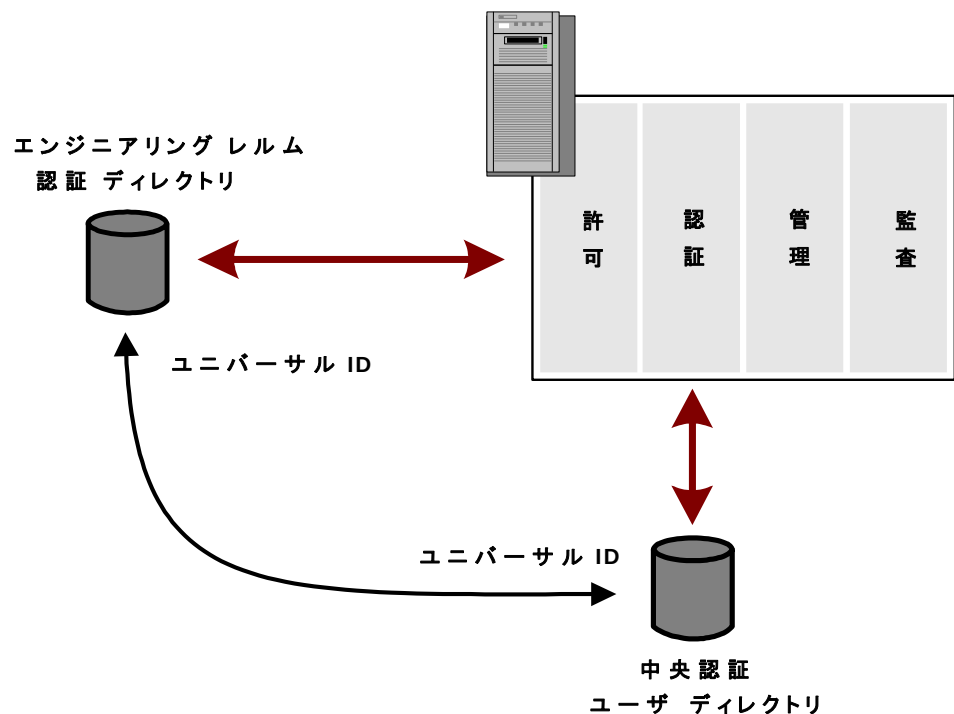
前述の図に表した SiteMinder 保護ネットワークで、正規従業員の認証データは中央認証ユーザ ディレクトリに格納されています。従業員は、エンジニアリング レルム内のリソースへのアクセスを試みます。このとき、その社員が正しく認証されると、ポリシー サーバはエンジニアリング レルムには専用の許可ディレクトリが使用されることを認識します。また、ポリシー サーバは、中央認証ユーザ ディレクトリとエンジニアリング レルムの許可ユーザ ディレクトリ間のディレクトリ マッピングを検索し、ユーザ ID を許可ディレクトリにマップします。この処理が終了すると、ポリシー サーバはその社員が要求したレルムへのアクセス権限を持っているかどうかを確認します。

臨時従業員が品質保証レルムのリソースにアクセスする場合

前述の図に表した SiteMinder 保護ネットワークで、臨時従業員の認証データは臨時従業員認証ユーザ ディレクトリに格納されています。従業員は、営業レルム内のリソースへのアクセスを試みます。営業レルムには専用の許可ディレクトリは関連付けられていません。SiteMinder は、臨時従業員認証ディレクトリでこの従業員を認証し、ディレクトリ マッピングが設定されているかどうかを確認します。営業レルムには専用の許可ディレクトリがないため、SiteMinder は、その従業員の認証に使用したディレクトリの情報を使用して従業員の許可を試みます。

ユニバーサル ID によるディレクトリ マッピング

前述の図に表した SiteMinder 保護ネットワークで、従業員の認証データは中央認証ユーザ ディレクトリに格納されています。従業員は、エンジニアリングレルム内のリソースへのアクセスを試みます。従業員の認証が成功すると、ポリシーサーバはディレクトリ マッピングを使用して、従業員のユニバーサル ID に基づいて、エンジニアリング許可ディレクトリでその従業員を検索します(以下の図を参照してください)。



上記の図で、ディレクトリ マッピングはユニバーサル ID を使用するものとします。ポリシーサーバは、認証ディレクトリ内でユニバーサル ID として定義されている属性を使用して、許可ディレクトリ内で一致するユニバーサル ID を検索します。許可ディレクトリでユニバーサル ID が見つかったと、SiteMinder は、ポリシーの処理を終了し、保護されているリソースにユーザがアクセスできるかどうかを決定します。

ディレクトリ マッピングの大文字/小文字の区別

Oracle データベースなど大文字/小文字を区別するディレクトリは、「ROBIN」と「robin」を異なる 2 つのユーザ名として扱います。LDAP ディレクトリなどのその他のディレクトリは大文字/小文字を区別しないため、「Robin」、「ROBIN」、「robin」、「RobIn」はすべて同じユーザ名として扱われます。これは、ユーザが認証では大文字/小文字を区別しないディレクトリを使用し、許可では大文字/小文字を区別するディレクトリを使用する場合に問題になります。

認証ディレクトリが大文字/小文字を区別するために認証が失敗する場合、ユーザは、ディレクトリの必須形式でユーザ名を再入力すれば回復できます。ユーザ名を「Name」という形式にする必要があるディレクトリの場合は、「Robin」と正しく入力し直します。ただし、許可ディレクトリが大文字/小文字を区別するために許可が失敗する場合、ポリシー サーバで回復する方法はありません。

許可ディレクトリが大文字/小文字を区別する場合は、認証したユーザ名の形式を変更して許可ディレクトリの必須形式に合わせることができます。認証したユーザ名は「RoBiN」であるが、許可ディレクトリの必須形式が「Name」の場合は、最初に「RoBiN」を「Robin」に変更すれば、ユーザを許可できます。

ディレクトリ マッピングとレスポンス

SiteMinder では、ユーザ ディレクトリ属性内の情報を収集するレスポンスを設定できます。ディレクトリ マッピングを使用する場合は、マッピングがレスポンスに与える影響を考慮する必要があります。たとえば、前の図に示したディレクトリ マッピングでは、OnAuth イベントの場合は中央認証ユーザ ディレクトリから属性値を取得し、OnAccess イベントの場合はエンジニアリング レルム認証ディレクトリから属性値を取得することとなります。イベントとレスポンスの関連付けについては、[レスポンスおよびレスポンス グループ](#) (P. 521)を参照してください。

第 9 章：認証方式

このセクションには、以下のトピックが含まれています。

- [認証方式の概要](#) (P. 309)
- [基本認証方式](#) (P. 320)
- [SSL を介した基本認証方式](#) (P. 322)
- [HTML フォーム認証方式](#) (P. 324)
- [Windows 認証方式](#) (P. 342)
- [Information Card 認証方式](#) (P. 346)
- [MS Passport 認証方式](#) (P. 361)
- [RADIUS CHAP/PAP 認証方式](#) (P. 365)
- [RADIUS サーバ認証方式](#) (P. 367)
- [SafeWord サーバ認証方式](#) (P. 368)
- [SafeWord サーバ認証方式と HTML フォーム認証方式](#) (P. 370)
- [SecurID 認証方式](#) (P. 372)
- [X.509 クライアント証明書認証方式](#) (P. 381)
- [X.509 クライアント証明書および基本認証方式](#) (P. 385)
- [X.509 クライアント証明書または基本認証方式](#) (P. 387)
- [X.509 クライアント証明書および HTML フォーム認証方式](#) (P. 390)
- [X.509 クライアント証明書または HTML フォーム認証方式](#) (P. 393)
- [匿名認証方式](#) (P. 397)
- [カスタム認証方式](#) (P. 399)
- [Federation セキュリティ サービス認証方式](#) (P. 400)
- [SOA Security Manager 認証方式](#) (P. 400)
- [偽装認証方式](#) (P. 401)

認証方式の概要

通常、ユーザがネットワークリソースにアクセスしようとするとき、ネットワークの所有者はそのユーザの身元を確認しようとします。特に、企業に所属する従業員については、使用できるリソースを特定するために身元を確認する必要があります。顧客に対しては、アクセスするときに内容をパーソナライズするために身元を確認する必要があります。たとえ匿名のユーザであっても個々に利用記録を追跡すべきです。これによりそのユーザがネットワークに再度アクセスしたときに、そのユーザの履歴を基に質の高い利用環境を提供できます。ユーザを識別するために、SiteMinder では認証方式を使用します。

認証方式とは、ユーザの認証情報を収集してユーザを識別する方法です。**SiteMinder** では、さまざまな種類の認証方式に対応しています。その中には、ユーザ名とパスワードを使った基本認証や **HTML** フォームベースの認証からデジタル証明書やトークン認証まで含まれています。重要度の低いネットワークリソースでは、簡単な認証方式を利用します。これに対し、重要なネットワークリソースの安全性を高めるためには、複雑な認証方式を利用します。

認証方式は 管理 UI を使用して設定する必要があります。認証時に、**SiteMinder Web** エージェントはポリシー サーバと通信して、リソースを要求しているユーザから取得する必要がある認証情報を調べます。

この章では、管理 UI での認証方式の処理について概説した後、サポートされる認証方式ごとにセクションを分けて、テンプレートを使用した設定方法を説明します。これらのテンプレートを使うと、各認証方式に必要な情報の大半がポリシー サーバに提供されます。管理者は、サーバ IP アドレスなどの環境固有の情報や、認証方式の初期化に必要な共有秘密キーを使って認証方式の設定を完了させる必要があります。

サポートされる認証方式とパスワード ポリシー

認証方式のタイプによってパスワード ポリシーをサポートするものとしがないものがあります。特定のタイプの認証方式がパスワード ポリシーをサポートするかどうかは、管理 UI の認証方式プロパティダイアログ ボックスで確認できます。特定の認証方式タイプについて確認するには、[各方式共通セットアップ]グループ ボックスのドロップダウンリストから該当するタイプを選択し、[この認証方式に対してパスワード ポリシーを有効にする]チェック ボックスの設定を参照します。認証方式でパスワード ポリシーをサポートしない場合、このチェック ボックスのラベルはグレーアウトされ、チェックボックスは使用できません。

以下の表では、サポートされる認証方式タイプと、各タイプでパスワード ポリシーがサポートされるかどうかを 管理 UI にアクセスせずに確認できます。

認証方式タイプ	パスワード ポリシーのサポート
匿名	いいえ
基本	はい
SSL を介した基本	はい
カスタム	はい

HTML フォーム	はい
インパーソネーション(偽装)	いいえ
MS Passport	いいえ
RADIUS CHAP/PAP	はい
RADIUS サーバ	はい
SafeWord	いいえ
SafeWord と HTML フォーム	いいえ
SecurID	いいえ
SecurID と HTML フォーム	いいえ
X.509 クライアント証明書	いいえ
X.509 クライアント証明書および基本	はい
X.509 クライアント証明書または基本	はい
X.509 クライアント証明書および HTML フォーム	はい
X.509 クライアント証明書または HTML フォーム	はい
Windows 認証	はい

認証時のポリシー サーバ検索を 1 つのユーザ ストアに限定

1 人のユーザを、ポリシー ドメインと関連付けられた複数のユーザ ディレクトリまたはデータベース内に格納することができます。このユーザは、各ユーザ ストア内に同じパスワードを持ちます。認証中に、ポリシー サーバで、ユーザ ストアの 1 つでユーザが無効になっていることが検出された場合、デフォルトではポリシー ドメインに関連付けられたすべてのストアでユーザの検索を続行します。関連するすべてのユーザ ストアでユーザが無効になっている場合のみ、そのユーザを認証できません。ユーザが関連ユーザ ストアのいずれかで有効である場合は認証されます。

ポリシー サーバのこのデフォルトの動作は設定可能です。1 つのユーザ ストアでユーザが無効になっていることが検出された場合にポリシー サーバで検索を停止するよう設定するには、次のレジストリ キーを追加して値を 1 に設定します: ReturnOnDisabledUser。

認証時のポリシー サーバ検索を 1 つのユーザ ストアに限定する方法

1. レジストリ キー ReturnOnDisabledUser を手動で追加します。

Windows

レジストリ キー ReturnOnDisabledUser を以下の場所に追加します。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion  
¥PolicyServer
```

Solaris

以下の行を sm.registry ファイルに追加します。

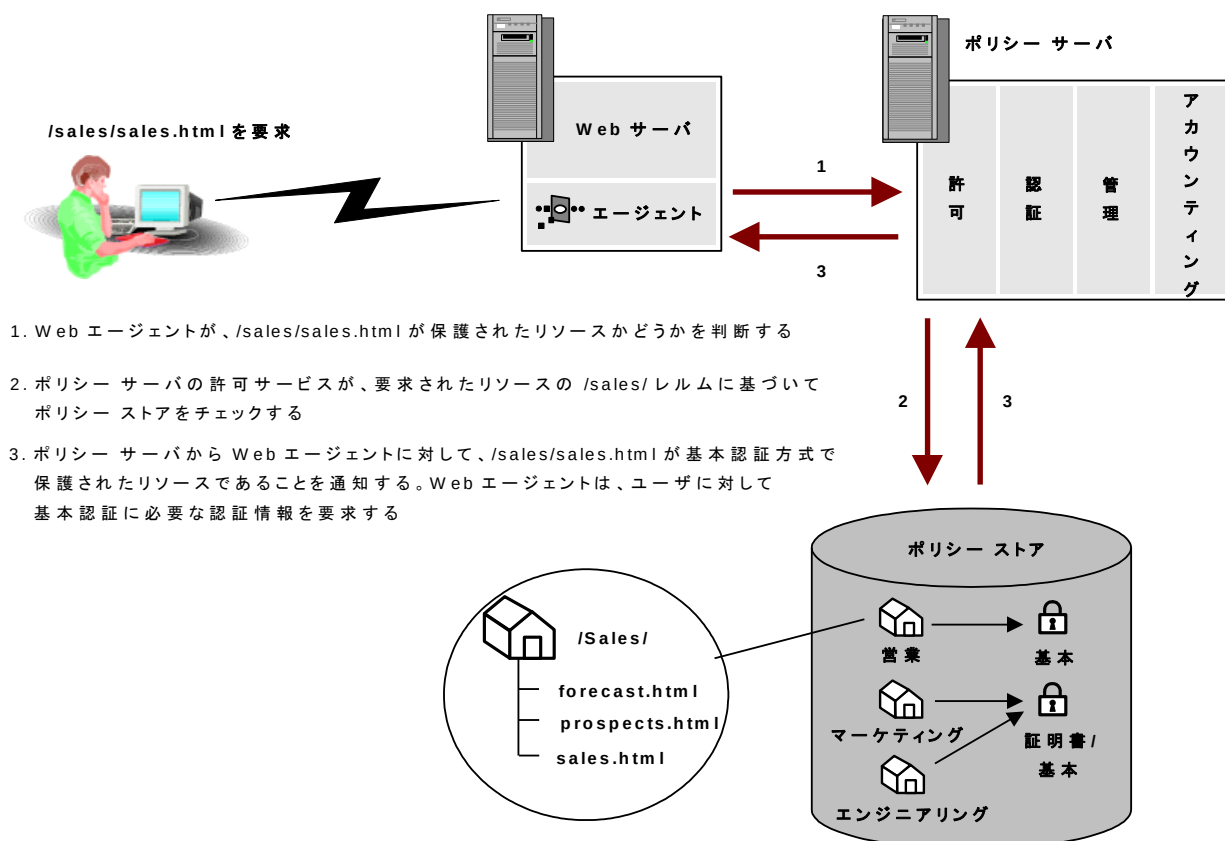
```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion  
¥PolicyServer  
ReturnOnDisabledUser=0x1; REG_DWORD
```

2. ReturnOnDisabledUser に値 1 を割り当てます。

認証方式処理

ユーザが保護されたネットワークリソースへのアクセスを要求すると、ポリシー サーバはそのリソースのレلمに関連付けられている認証方式を使用してユーザの身元を確認します。認証方式は、ユーザが提供する必要がある認証情報を指定するほかに、ポリシー サーバがユーザを識別する方法も指定します。

管理 UI を使用して認証方式を設定し、その認証方式をレلمに割り当てること
ができます。次の図に、ユーザが保護されたリソースにアクセスしようとしたとき、
認証方式がどのように呼び出されるかを示します。



この図の例では、ユーザが `/Sales/` レلمの保護されたリソース `sales.html` を要求しています。このレلمには、基本認証が必要です。ポリシー サーバは、Web エージェントにリソースが保護されていることを通知します。そして、Web エージェントを介してユーザの基本証明を要求し、ユーザにユーザ名とパスワードの入力を求めます。

認証方式タイプ

SiteMinder でサポートされる認証方式は、いくつかのカテゴリに分類されます。これらのカテゴリは、利用可能な認証方法の一般的な特徴を表しています。各認証方式の詳細については、この章の各セクションで説明します。

基本認証方式

基本認証は、ユーザ名とパスワードに基づいてユーザを識別します。ユーザ識別情報は、ユーザ ディレクトリに保存されています。基本認証方式の場合、ポリシー サーバはユーザ名を使用してディレクトリからユーザを検索し、入力されたパスワードがユーザ ディレクトリに保存されているパスワードと一致するかどうかを確認します（つまり、ユーザをディレクトリに関連付けます）。ユーザが入力したユーザ名とパスワードがユーザ ディレクトリのデータと一致すると、SiteMinder はユーザを認証します。

管理 UI には、以下の基本方式に対応した認証方式テンプレートが用意されています。

- 基本 (HTTP 基本)
- SSL を介した基本

HTML フォームベース認証方式

SiteMinder は、カスタマイズした HTML フォームの使用による認証情報の収集をサポートしています。フォームベース認証方式では、ユーザは社会保障番号、所属組織、口座番号などの詳細情報の入力を要求されます。ポリシー サーバは、ユーザディレクトリの属性に対して詳細情報が正しいことを確認してから、ユーザを認証します。

Windows 認証方式

統合 Windows 認証 (IWA) は、純粋な Windows 環境でユーザを検証するための Microsoft 独自開発のメカニズムです。IWA では、最初の対話形式のデスクトップログインプロセス時、およびその後のセキュリティレイヤへの情報の転送時に Windows がユーザ クレデンシャルを取得することで、シングル サインオンを実現しています。SiteMinder は Windows 認証方式を使用して、Microsoft 統合 Windows 認証インフラストラクチャが取得したユーザ クレデンシャルを処理することによってリソースを保護します。

SiteMinder の以前のバージョンは NTLM 認証方式によって Windows 認証に対応していました。ただし、このサポートは、NT ドメインを備えた環境や、混合モードのレガシー NT ドメインをサポートするように Active Directory サービスが設定されている環境に制限されていました。

Windows 認証方式では、SiteMinder がネイティブ モードで実行されている Active Directory および、NTLM 認証に対応するよう設定された Active Directory の展開におけるアクセス コントロールを提供できるようにします。Windows 認証方式は、SiteMinder の以前の NTLM 認証方式に代わるものです。既存の NTLM 認証方式は引き続きサポートされ、新しい Windows 認証方式を使用して設定することができます。

NTLM 認証方式は、IIS Web サーバ上の Web エージェントで保護されているリソースに使用できます。この場合、ユーザは Web ブラウザ Internet Explorer を介してリソースにアクセスします。この方式では、ユーザのクレデンシャルを取得して確認できるように IIS Web サーバを正しく設定する必要があります。ポリシー サーバは、IIS サーバで保証されたユーザの識別情報に基づいて許可を決定します。

X.509 クライアント証明書認証方式

SiteMinder では、X.509 V3 クライアント証明書を使用できます。デジタル証明書は、ユーザの身元を示す暗号化された証明としての役割を果たします。証明書がクライアントにインストールされると、リソースにアクセスするユーザの識別に、その証明書を使用できます。証明書認証は SSL 通信を使用します。証明書認証と基本認証を組み合わせて使用すると、より高い安全性を提供することができます。

管理 UI は、以下の証明書ベース認証方式に対応した認証方式テンプレートを提供します。

- X.509 クライアント証明書
- X.509 クライアント証明書および基本
- X.509 クライアント証明書または基本
- X.509 クライアント証明書および HTML フォーム
- X.509 クライアント証明書または HTML フォーム、あるいはその両方

注: 証明書のみの認証方式の場合には、Web エージェントは「HTTP エラー 403: アクセスが拒否されたか、または禁止されています。」を返します。これは、Web エージェントがユーザに新しい証明書を要求できないためです。

プロキシ認証方式

プロキシ認証方式とは、サードパーティの認証製品で必要となるプロキシサーバとして、ポリシーサーバを使用する方式です。プロキシ認証方式では、ポリシーサーバがこの方式特有のライブラリを使用してサードパーティサーバの認証機能を実行します。

SiteMinder は、以下のプロキシ認証方式をサポートしています。

- RSA 社の ACE/Server (SecurID トークンと組み合わせて使用)
- RSA 社の ACE/Server (SecurID トークンと組み合わせて使用) とパスワードリセット可能な HTML フォームとの組み合わせ
- Secure Computing 社の SafeWord サーバ
- RADIUS サーバ

ダイジェスト認証方式

ダイジェスト認証方式では、ディレクトリに保存されている暗号化されたユーザ属性文字列を読み込み、その文字列とユーザから受信した暗号化された文字列を比較します。これらの暗号化された文字列が一致すると、ポリシーサーバはユーザを認証します。ダイジェスト方式は、クライアントワークステーション上で暗号化された文字列と、サーバ上で暗号化された文字列を、暗号化された送信手段を介さずに比較します。

SiteMinder は、以下のダイジェスト認証方式をサポートしています。

- RADIUS CHAP
- RADIUS PAP

匿名認証方式

匿名認証方式は、未登録ユーザに特定の Web コンテンツへのアクセスを許可します。ユーザが匿名認証を行うリソースにアクセスすると、SiteMinder はそのユーザに GUID (グローバル固有識別子) を割り当てます。SiteMinder は、この GUID をユーザのブラウザ上の永続的な cookie に保存します。これにより、ユーザは認証要求をされることなく特定のリソースにアクセスできます。

カスタム認証方式

使用したい認証方式が SiteMinder にない場合は、CA の API を使用して、カスタムの認証方式を作成できます。

注: Software Development Kit をインストール済みの場合は、カスタム認証方式の作成方法の詳細について、「API Reference Guide for C」または「API Reference Guide for Java」を参照してください。

SSL を介した認証

SSL (Secure Sockets Layer) 接続を介して認証を実行するように設定できます。SSL は、クライアントとサーバの間に暗号化された接続を確立する方法で、接続の確立と身元の証明にはデジタル証明書が使用されます。

次の認証方式は、SSL 接続を使用するように設定されています。

- 基本*
- HTML フォーム*
- X.509 クライアント証明書
- X.509 クライアント証明書および基本
- X.509 クライアント証明書または基本
- X.509 クライアント証明書または HTML フォーム
- X.509 クライアント証明書および HTML フォーム

注: アスタリスクは、SSL 接続がオプションであることを示します。

保護レベル

認証方式には保護レベルが必要です。このレベルの範囲は 0 から 1000 です。値が大きいほど、方式の保護レベルが高いことを示します。保護レベルにより、同じポリシードメイン内の同等かより低い保護レベルの認証方式に対するシングル サインオンは認められるのに対し、方式の保護レベルがより高いリソースにアクセスする場合は追加認証が要求されます。

注: 匿名認証方式の保護レベルは、常に 0 です。カスタム認証方式には、0 から 1000 までの保護レベルを設定できます。その他のすべての認証方式には、1 から 1000 の保護レベルを設定できます。

たとえば、ネットワーク上にすべてのユーザが使用できる一連のリソースがある場合、低い保護レベル(5 など)が設定された基本認証方式(ユーザ名とパスワード)を割り当てることができます。企業の上層部のみが使用できる収益情報に対しては、高い保護レベル(15 など)が設定されている X.509 クライアント証明書認証方式を割り当てることができます。

ユーザがある認証方式で認証された場合、その認証方式の保護レベルと同等もしくは低い保護レベルのリソースには再度認証を要求されることなくアクセスできます。ただし、アクセス権を取得するには、リソースに対してユーザを許可する必要があります。

前述の例では、ユーザ名とパスワードで認証を得たユーザ(保護レベル 5)が、X.509 クライアント証明書認証方式(保護レベル 15)が割り当てられた収益情報にアクセスする場合、デジタル証明書を使用して再度認証を得なければなりません。しかし、保護レベル 5 の認証方式が割り当てられている別の部署のリソースにアクセスする場合は、再度認証を要求されることはありません。

認証方式とクレデンシャル要件

次の表に、サポートされる認証方式とその証明の要件を示します。

クレデンシャル要件					
認証方式	ディレクトリ ユーザ名	ディレクトリ パスワード	トークンから のコード	X.509 証明書	ユーザプロ ファイル属性
匿名					
基本	はい	はい			
SSL を介した基本	はい	はい			
カスタム	任意	任意	任意	任意	任意
HTML フォーム(必要に 応じて SSL を介す)	カスタム証明 書	カスタム証明 書			任意
インパーソネーション(偽 装)	はい				任意
MS Passport	はい	はい			はい
NTLM または Windows	必須*	必須*			
RADIUS CHAP/PAP	はい	はい			
RADIUS サーバ	はい	はい			
SafeWord サーバ	はい	はい			
SafeWord およびフォー ム	はい	はい			任意

クレデンシャル要件					
認証方式	ディレクトリ ユーザ名	ディレクトリ パスワード	トークンから のコード	X.509 証明書	ユーザプロ ファイル属性
SecurID	はい		はい		
SecurID およびフォーム	はい		はい		任意
TeleID	はい		はい		
X.509 クライアント証明書				はい	
X.509 クライアント証明書 および基本 (SSL を使用)	はい	はい		はい	
X.509 クライアント証明書 または基本 (必要に応じて SSL を介す)	はい (基本に 対して)	はい (基本に 対して)		はい (証明書 に対して)	
X.509 クライアント証明書 および HTML フォーム	カスタム証明 書	カスタム証明 書		はい	任意
X.509 クライアント証明書 または HTML フォーム	HTML フォーム のカスタム証 明	HTML フォー ムのカスタム 証明		はい (証明書 に対して)	HTML フォー ムについて は任意

* NTLM または Windows では、ユーザがリソースへのアクセスを試みたときに、SiteMinder はユーザにユーザ名とパスワードの入力を求めません。この方式では、ユーザのクレデンシャルを取得して確認できるように IIS Web サーバを正しく設定する必要があります。ポリシー サーバは、IIS サーバで保証されたユーザの識別情報に基づいて許可を決定します。

ポリシー サーバ ユーザ インターフェースでの認証方式オブジェクトのセットアップ

管理 UI で新しい認証方式をセットアップするには、以下の順序で各コンポーネントを設定する必要があります。

1. Web サーバ (証明書認証方式、SSL 認証方式、HTML フォームベース認証方式のみ)
2. ポリシー サーバ (X.509 証明書認証方式の証明書マッピングを含む)

Web サーバ

SiteMinder Web エージェントで SSL ベースの認証方式をサポートするには、SSL をサポートするように Web サーバを設定する必要があります。

注: Web サーバの設定の詳細については、「ポリシー サーバ インストール ガイド」で、Web サーバの設定に関する説明を参照してください。

ポリシー サーバ

認証方式をサポートするように Web サーバを設定したら、ポリシー サーバもその認証方式をサポートするように設定します。

単一の認証方式の設定の複数のインスタンス

管理 UI では、ほとんどの認証方式の複数のインスタンスを設定できます。たとえば、ログイン処理、パスワードを忘れた場合の処理、ログアウト処理などに対応するように、複数の HTML フォームベース認証方式を作成できます。1 つの認証方式で複数のインスタンスを作成する場合は、各自のセキュリティ要件に合った保護レベルを設定してください。

基本認証方式

ポリシー サーバのインストール処理では、基本認証方式が自動的に設定されます。これは、ユーザ ディレクトリ サービスにユーザ名とパスワードを渡して認証処理を行い、ユーザを識別する方式です。LDAP ユーザ ディレクトリでは、この処理を LDAP の「バインド」操作を介して行います。Windows NT のディレクトリでは、LogonUser() 関数呼び出しを介して行います。ブラウザから SiteMinder Web エージェントで保護された Web サーバにユーザの認証情報を渡す際には、HTTP 基本認証プロトコルが使用されます。基本認証方式は、ASCII 文字にのみ対応しています。

ユーザが基本認証方式で保護されたリソースにアクセスしようとする、**SiteMinder** エージェントはユーザ名とパスワードの入力を要求します。ユーザ名とパスワードを入力すると、暗号化された接続を介してエージェントからポリシー サーバに証明が渡されます。ポリシー サーバは、アクセス対象のリソースが含まれるポリシードメインに関連付けられたディレクトリのユーザ名と渡されたユーザ名を照合します。ユーザ名が一致したら、ユーザディレクトリ内のパスワードとユーザが入力したパスワードを比較します。パスワードが一致するとそのユーザは認証され、ポリシー サーバから **Web** エージェントに処理の続行を示すメッセージが送信されます。認証されなかった場合、ユーザは再度ユーザ名とパスワードの入力を要求されます。

注: この認証方式は、デフォルトでは認証情報の暗号化送信は行われません。**Web** サーバ上の保護されたすべての **URL** が **SSL** を要求するようにセットアップされている場合を除き、ユーザ名とパスワードは標準の **HTTP** 基本プロトコルを介してブラウザから **Web** エージェントで保護された **Web** サーバに渡されます。ただし、**Web** エージェントとポリシー サーバ間の通信は常に暗号化接続を経由して行われます。単純なユーザ名とパスワードに基づきながらも、そのデータが暗号化される認証方式の詳細については、「[SSL を介した基本認証方式 \(P. 322\)](#)」を参照してください。

管理 **UI** で作成したレルムは、デフォルトで基本認証方式を使用します。認証方式は、新規レルムの作成時や既存レルムの修正時に変更できます。

基本認証方式のセキュリティレベルを向上させるために、パスワード ポリシーを作成できます。また、この **SiteMinder** 機能を使用してパスワード ルールを管理できます。

基本認証方式の前提条件

基本認証方式を設定する前に、以下の必要条件を満たしていることを確認します。

- クライアントのユーザ名とパスワード情報がユーザ ディレクトリにあること。
- ポリシー サーバとユーザ ディレクトリの間ディレクトリ接続が確立されていること。

基本認証方式の設定

基本認証方式を使用して、ユーザ ディレクトリ内にあるユーザ名およびパスワードに対してユーザ **ID** を確認します。

注: 以下の手順では、新しいオブジェクトを作成しているとしします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[基本テンプレート]を選択します。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

SSL を介した基本認証方式

SSL を介した基本認証方式は、基本認証方式と同様に、ユーザ名とパスワードをユーザ ディレクトリに渡してユーザを識別します。保護された URL が SSL (Secure Sockets Layer) を要求するように設定されていなくても、認証情報は常に暗号化された SSL 接続を介して渡されます。SiteMinder Web エージェントでは、認証情報を送信する前にユーザのブラウザをリダイレクトして SSL 接続を確立することにより、この処理を実行します。証明を配信したあとは、Web エージェントがブラウザをリダイレクトして元の URL に戻します。

SSL を介した基本認証方式でのセキュリティレベルを向上させるために、パスワード ポリシーを作成できます。また、この SiteMinder 機能を使用してパスワード ルールを管理できます。

SSL を介した基本認証方式の前提条件

SSL を介した基本認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- クライアントユーザ名とパスワード情報がユーザ ディレクトリに保存されていること。
- ポリシー サーバとユーザ ディレクトリの間にディレクトリ接続が確立されていること。
- X.509 サーバ証明書が **SSL Web** サーバ上にインストールされていること。
- ネットワークが、**HTTPS** プロトコルを使用したクライアントブラウザへの **SSL** 接続に対応していること。
- **SSL** 接続にリソースがリダイレクトされる **Web** サーバに **SiteMinder Web** エージェントがインストールされていること。この **Web** エージェントにより、サーバはこの認証方式で使用する必要がある **.scc MIME** タイプを処理できるようになります。

SSL を介した基本認証方式の設定

SSL を介した基本認証方式を使用して、ユーザ ディレクトリ内にあるユーザ名およびパスワードに対してユーザ ID を確認します。クレデンシャルの送信は暗号化された **SSL** レイヤで行われます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[SSL を介した基本テンプレート]を選択します。

認証方式固有の設定が開きます。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスにサーバ名とターゲット情報を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

HTML フォーム認証方式

HTML フォーム認証方式は、カスタム HTML フォームで収集される認証情報に基づいて認証を行います。この方式は、収集する認証情報を変更するなどの柔軟性があるため、次のことが可能になります。

- 企業のブランド イメージ沿った表示が可能です。企業のロゴを入れることもできます。
- ユーザ名とパスワードの収集用にカスタムラベルを使用できます(たとえば、名前とパスワードではなく、アカウント番号と PIN を使用したい場合など)。
- 名前とパスワード以外の認証情報に基づいた認証処理ができます(ユーザディレクトリ属性経由)。この場合、ポリシー サーバ マシン上の認証方式ライブラリが、ユーザのデータを DN にマップします。ユーザを DN にマップすると、ポリシー サーバは属性リストとユーザディレクトリの対応する値を照合できます。この処理をバックエンドマッピングといいます。
- ユーザ名とパスワードに加え、さらに別のユーザ属性を含めた認証情報に基づいた認証処理が可能です。これは追加の属性検証であるものとみなされます。これは単に属性の検証が追加されたものと考えられますので、カスタム認証方式ライブラリは必要ありません。

たとえば、パスワードを忘れたユーザの名前とシークレットフレーズ(本人のみ知っている情報)を収集するために、カスタム フォームを使用できます。

- ログイン、ログアウト、パスワードを忘れた場合など、さまざまな状況に対応する、複数の HTML フォームを作成できます。

注: HTML フォーム認証方式では、マルチバイト文字に対応しています。

ポリシー サーバがインストールされた環境では、複数のフォームベースの認証方式を設定できます。各方式は、以下のコンポーネントで構成されています。

フォーム認証情報コレクタ(FCC)

FCC プロセス ファイルは、HTML やいくつかのカスタム表記法を含む簡単なマークアップ言語で構成されます。

HTML フォーム方式ごとに固有の .fcc ファイルが必要です。このファイルには、カスタムフォームの定義と、FCC が HTML フォーム認証を処理するときに使用する詳細情報が保存されています。FCC は、.fcc ファイルから生成されたカスタムフォームにユーザが入力した認証情報を抽出します。

HTML フォーム認証方式では、.fcc ファイルのデフォルトの拡張子は .fcc です。別の拡張子を使用するには、次の手順に従います。

- Apache または iPlanet Web サーバの場合には、目的の拡張子を使用するように Web サーバを設定します。
- Domino または IIS Web サーバの場合には、Web エージェント構成ファイルまたはオブジェクトの FCCExtensions パラメータに目的の拡張子を指定します。Web エージェント設定パラメータの詳細については、「Web エージェント設定ガイド」を参照してください。

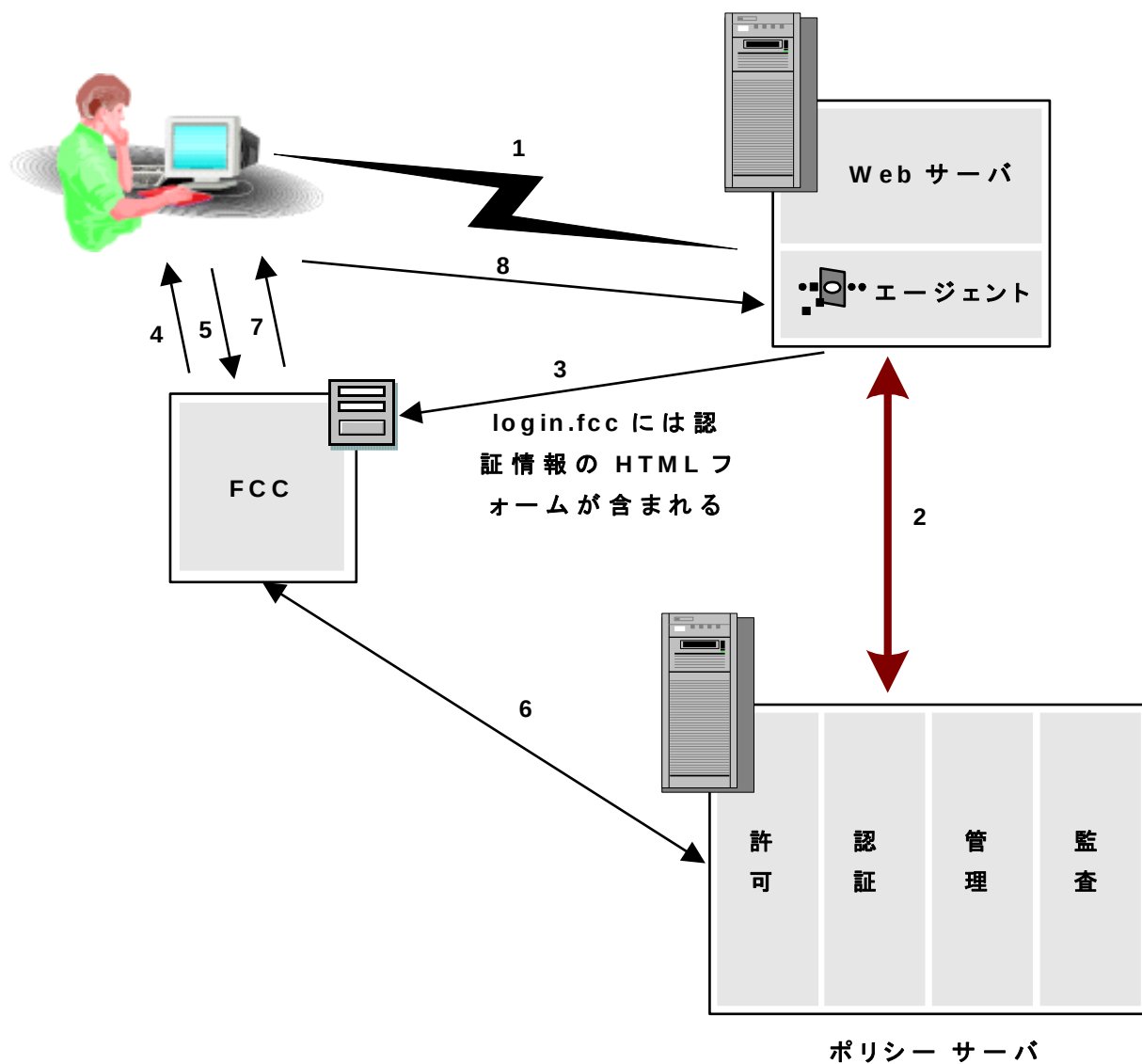
.unauth ファイル

認証の失敗回数が認証方式に指定された最大回数を超えたユーザに対して、SiteMinder はこのファイルの内容を表示します。.unauth ファイルは .fcc ファイルごとに必要です。たとえば、login.fcc ファイルが Web サーバ上にある場合、login.unauth ファイルも同じ場所に保存する必要があります。

smerrorpage 変数が .fcc ファイルに定義されている場合は、.unauth ファイルは必要ありません。

認証方式ライブラリ

これは、ポリシー サーバ マシン上で稼働し、認証処理を実行する共有ライブラリです。



この図は、HTML フォーム認証の処理について示しています。

1. ユーザは、HTML フォーム認証方式で保護されたレルム内のリソースを要求します。
2. Web エージェントはポリシー サーバに問い合わせ、ユーザのリクエストを認証情報コレクタにリダイレクトする必要性を判断します。

3. Web エージェントは認証情報コレクタファイルの URL にリクエストをリダイレクトします。
4. 認証情報コレクタは、.fcc ファイルに記述されたフォームをユーザのブラウザに表示します。
5. ユーザは、カスタム フォームに必要な事項を入力してから、それを通知(サブミット)します。認証情報コレクタが認証情報を処理します。
6. 認証情報コレクタ(FCC)は、ポリシー サーバにユーザをログインします。ポリシー サーバは、認証情報コレクタにユーザ セッション データを返します。
7. ユーザが認証されると、認証情報コレクタはセッション cookie を作成し、そのセッション cookie をブラウザに渡して、ユーザがアクセスを要求していたリソースにユーザをリダイレクトします。
8. ユーザはセッション cookie を使用して認証を受けます。次に、Web エージェントはユーザ許可を処理します。

HTML フォーム認証方式の前提条件

HTML フォーム認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- カスタマイズした .fcc ファイルが、HTML フォーム認証を実行する cookie ドメイン内の Web エージェントのサーバ上にあること。サンプルの .fcc ファイルは、Web エージェントをインストールした Samples/Forms サブディレクトリの下にあります。
 - カスタマイズした.unauth ファイルが、Web エージェントのサーバ上にあること。
- 注: このファイルは、.fcc ファイルが smerrorpage ディレクティブを使用する場合は必要ありません。
- ポリシー サーバとユーザ ディレクトリの間でディレクトリ接続が確立されていること。

- デフォルトの HTML フォーム ライブラリがインストールされていること。このライブラリが HTML フォーム認証を処理します。

- Windows では SmAuthHTML.dll。

- Solaris では smauthhtml.so。

これらのファイルは Web エージェントの設定時に自動的にインストールされます。

- (Sun Java Systems) Sun Java Systems の Web サーバを使用している場合は、magnus.conf ファイル内の StackSize パラメータの値を 131072 より大きい値に増加してください。値を変更しないと、Web サーバはそのコアをダンプし、SiteMinder がフォームを使用して認証リクエストを行うたびに再起動します。

カスタム認証方式ライブラリの作成とインストール

FCC によって収集されたユーザ名とパスワードのデータはポリシー サーバに渡され、そこから認証方式ライブラリに渡されます。

バックエンド マッピングが必要な場合を除き、SmAuthHTML 認証方式ライブラリを使用できます。このライブラリはポリシー サーバにバンドルされており、ポリシー サーバマシンにはすでにインストールされています。

注: バックエンド マッピングにはカスタム認証方式ライブラリが必要です。Software Development Kit をインストール済みの場合は、「API Reference Guide for C」を参照してください。

カスタム認証方式を作成し、ユーザ名とパスワード以外のデータも収集する場合には、FCC がそのデータをパックして、[ユーザ名]フィールドと[パスワード]フィールドに格納する必要があります(いずれのフィールドも 511 文字未満でなければなりません)。カスタム認証方式ライブラリは、そのデータをアンパックして、ユーザ名とパスワードにマップする必要があります。

FCC はポリシー サーバと同じマシンにインストールできます。

HTML フォーム認証テンプレート

フォーム認証に使用されるテンプレートは、HTML の拡張版を使用するテキストファイルです。これらのファイルは.fcc ファイルと呼ばれます。

HTML フォーム認証方式で使用する .fcc ファイルのデフォルトの拡張子は、.fcc です。ただし、他の拡張子を使用することもできます。

- **Apache Web** サーバの場合は、希望する拡張子を使用するように **Web** サーバを設定します。

注：より詳細な情報については、「*Web エージェント インストール ガイド*」を参照してください。

- **Domino** または **IIS Web** サーバの場合は、**Web** エージェント構成ファイルまたはオブジェクトの **FCCExtensions** パラメータに希望する拡張子を指定します。

注：Web エージェント設定パラメータの詳細については、「*Web エージェント ガイド*」を参照してください。

SiteMinder Web エージェントには、英語、フランス語、日本語のサンプル .fcc ファイルが用意されています。これらのファイルは、次のディレクトリにあります。

言語	ディレクトリ
英語版	Windows: Web Agent¥Samples¥Forms UNIX: webagent/samples/forms
フランス語	Windows: Web Agent¥Samples¥Formsfr UNIX: webagent/samples/formsfr
日本語	Windows: Web Agent¥Samples¥Formsja UNIX: webagent/samples/formsja

デフォルトでは、SiteMinder は英語の .fcc ファイルを使用します。

サンプル .fcc ファイルを使用するには、ターゲット ディレクトリを正しく指定するフォーム認証方式を作成します。

安全な HTML フォーム認証テンプレート

HTML フォーム認証テンプレートの安全なバージョンを使用することもできます。安全な HTML フォーム認証テンプレートは、以下の点が標準のバージョンと異なります。

- 安全なバージョンでは、返されるメッセージにユーザ名を表示しません。
- 安全なバージョンでは、フォーム テンプレートの右上にログアウト用ハイパーリンクが含まれます。このリンクは、ユーザをログアウトしてカスタムのログオフ ページにリダイレクトします。
- オートコンプリートは安全なバージョン内のすべてのテキスト フィールドでオフになっています。

安全なテンプレートファイルは、以下のディレクトリにあります。

- Windows : `webagent\secureforms`
- UNIX : `webagent/secureforms`

HTML フォーム認証テンプレートの安全なバージョンを使用するには、`secureforms` ディレクトリから以下の場所にファイルをコピーし、その場所にある標準バージョンを上書きします。

- Windows : `webagent\samples\forms`
- UNIX : `webagent/samples/forms`

注: 安全な HTML フォーム認証テンプレートのローカライズされたバージョンは用意されていません。英語版のみが提供されます。

SiteMinder .FCC ファイル

SiteMinder Web エージェントに組み込まれた FCC (SiteMinder フォーム認証情報コレクタ) は、`.fcc` ファイルと呼ばれるテンプレートファイルを読み込みます。`.fcc` ファイルは、標準の HTML タグといくつかの SiteMinder 独自の表記法で書かれています。この表記法は、SiteMinder が属性を確認し、このセクションの後半で説明するカスタム機能を利用するために必要です。

重要: Windows システムで作成または編集した `.fcc` ファイルを UNIX システムに移動すると、テキストの行末に `^M` が自動的に追加されます。この文字は Windows テキストファイルを識別するものですが、認証時に `.fcc` ファイルが失敗します。Windows テキストエディタから UNIX システムにファイルを移動する際には、追加された文字を削除してください。この問題を回避するために、UNIX 環境で使用する `.fcc` ファイルは UNIX システムで作成してください。

HTML フォーム認証方式では、.fcc ファイルのデフォルトの拡張子は .fcc です。別の拡張子を使用するには、次の手順に従います。

- Apache または iPlanet Web サーバの場合には、目的の拡張子を使用するように Web サーバを設定します。
- Domino または IIS Web サーバの場合には、Web エージェント構成ファイルまたはオブジェクトの FCCExtensions パラメータに目的の拡張子を指定します。Web エージェント設定パラメータの詳細については、「Web エージェントガイド」を参照してください。

HTML フォーム方式で保護されたリソースへのアクセスをユーザが要求すると、Web エージェントはユーザを .fcc ファイルにリダイレクトします。.fcc ファイルは、Web サーバ上で FCC を呼び出し、ユーザからの認証情報をカスタマイズされたフォームを介して収集します。FCC はブラウザのページを生成し、.fcc ファイルの内容に基づいてユーザ名とパスワードを作成します。このファイルは Web サーバのネームスペースに格納され、他の HTML ファイルと同様にアクセスできます。.fcc ファイルは 2 つの部分で構成されます。両方ともオプションです。

FCC ファイルの最初の部分には制御文があり、この制御文を使用して .fcc ファイルで POST 操作を実行します。この制御文は、クライアントに渡されることはありません。制御文の位置はファイルの先頭に、形式は @name=value にしてください。

name は変数の名前です。value は変数の値です。value には、%name1% の形式で文字列を指定します。この文字列は、name1 に関連付けられた変数の値に置き換わります。

.fcc ファイルの 2 つ目の部分は、.fcc ファイル上で GET 操作を実行すると返される HTML コードで構成されます。この部分には、name に関連付けられた値に置き換わる引用符 (") と共に、"\$\$name\$\$" 形式のテキストが含まれます。name では大文字と小文字は区別されません。

以下の図に示す非表示項目は、認証情報コレクタのステータスを保持するために使用されます。

名前	動的に設定するために使用する必要がある値 *	保持されるデータ
target	"\$\$target\$\$"	ユーザがアクセスを要求したリソース
smauthreason	"\$\$smauthreason\$\$"	ログインが失敗した理由

名前	動的に設定するために使用する必要がある値 :*	保持されるデータ
postpreservationdata	"\$\$postpreservationdata\$\$"	ユーザが POST リクエストを通じて送信したデータ
smagentname	\$\$smagentname\$\$	ユーザのログインに使用されたエージェントの名前

*引用符(")は必ず入力する必要があります。

.fcc ファイルは、少なくとも次の情報を収集する必要があります。

- ユーザ名
- パスワード
- ターゲット

重要: 認証情報コレクタを使用する認証方式(以下の図を参照)で保護されたリソースにユーザが POST リクエストを送信する場合には、**postpreservationdata**を使用します。この入力を使用しないと、対象リソースへの POST 要求は失われます。

方式

SSL を介した基本認証方式

HTML フォーム認証方式

X.509 クライアント証明書認証方式

X.509 クライアント証明書および基本認証方式

X.509 クライアント証明書または基本認証方式

X.509 クライアント証明書および HTML フォーム認証方式

X.509 クライアント証明書または HTML フォーム認証方式

次に、有効かつ簡単な .fcc ファイルの例を示します。

フォーム データに
基づいてユーザ名を
作成

ユーザの
パスワードを収集

ユーザの
組織を収集

```
@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>
      <tr><td><input type=submit value=LOGIN></td></tr>
    </table></form></body>
  </html>
```

ここに例示した **usermap.fcc** サンプルファイルは、**SiteMinder Web** エージェントのインストール時にデフォルトでインストールされます。

この .fcc ファイルでは、HTML フォームの **User Name** フィールドおよび **Organization** ドロップダウンリストにユーザが入力した情報に基づいて、ユーザの **DN**（識別名）が作成されます。この **DN** は、ユーザ名認証の認証情報です。ユーザのパスワードは、HTML フォームの **Password** フィールドから収集されます。非表示になっているレلمとターゲットへの入力値も収集され、認証が完了すると、ユーザは目的のリソースにアクセスできます。

FCC ファイルにおける名前/値ペアの生成方法

LoginFCC は、`$$name$$` を展開し、特殊な名前/値ペアによって使用されるネームスペースを生成します。名前が複数回入力された場合は、最後の値が使用されます。名前/値ペアは、以下の順序でネームスペースに追加されます。

1. クエリ文字列の名前/値 (Get の場合のみ)
2. ターゲットの値をコピーすると作成される `smtarget` の名前 (Get の場合のみ)
3. Post データの名前/値
4. `cookie` の名前/値
5. `@` 命令文の名前/値 (Post の場合のみ)
6. 「`smheaders`」の名前/値ペアで名づけられる SiteMinder レスポンス (Post の場合のみ)。

特殊な名前/値ペア

SiteMinder の FCC では、数多くの特殊な名前/値の組み合わせ (`@directives`) を解釈して、非標準的な処理を呼び出すことができます。次に、特殊な `@directives` とその意味を示します。

`username`

ログイン ユーザ名として使用する名前。

`password`

ログインするときに使用するパスワード。

`target`

ログイン後にアクセスするリソース。

`smheaders`

ネームスペースに含まれる SiteMinder レスポンス名のコロン区切りのリスト。コロン区切りのリストには、SiteMinder トランザクションに組み込む各ヘッダのエントリを含める必要があります。たとえば、SiteMinder トランザクションの一部として `header1` と `header2` の値を渡す場合は、FCC に以下のようなリストを含めます。

`@smheaders=header1:header2`

smerrorpage

カスタムフォームへの Post 操作でエラーが発生すると、ユーザのブラウザはこのページにリダイレクトされます。この特殊な値が .fcc ファイルに指定されていない場合、SiteMinder は、.fcc ファイルに関連付けられた .unauth ファイルをエラー ページとして使用します。

smretries

ログインの最大許容試行回数を指定します。この命令文を 0 に設定すると、試行回数は無制限になります。1 以上の値を設定すると、それが許容最大回数になります。

注: ユーザが .fcc フォームへの POST 操作を使用してログインする場合、ログインを試行できる回数が smretries ディレクティブの値を超えるように感じることがありますが、smretries で指定されているのは、ユーザが有効なクレデンシャルを入力できる回数です。この条件が満たされた場合にのみ、ユーザはアクセスを許可されます。

smpasswordfcc

データのポスト元が、パスワード サービスの FCC のファイルであるか、別の FCC のファイルであるかを指定します。

デフォルト: 1

重要: デフォルト値は変更しないままにしておくことをお勧めします。デフォルト値が変更されると、SafeWord 認証方式が正常に動作しないことがあります。

smusrmsg

ユーザに認証情報を要求した理由/ユーザがログインに失敗した理由を示すテキスト。

smauthreason

ログインの失敗に関連付けられた理由コード。

smsavecreds

ユーザのブラウザ上の永続的な cookie にユーザのクレデンシャルを保存するには、「Yes」に設定します。

smsave

永続的な cookie として保存する、名前のコロン区切りのリスト。

save

smsave の別名。

smtransient

一時的な cookie として保存される、名前のコロン区切りのリスト。

smagentname

ユーザがクレデンシャルを入力して認証用フォームを送信するときにポリシー サーバに提供するエージェント名を指定します。エージェントパラメータ FCCCompatMode=NO の場合は、このディレクティブを使用して値を指定する必要があります。

smlogout

SiteMinder からユーザをログアウトさせます。logoffuri と同じです。.fcc テンプレートに @smlogout=true と指定すると、FCC は SiteMinder からユーザをログアウトさせて、ターゲットにリダイレクトします。このため、通常、@smlogout ディレクティブは @target ディレクティブ (@target=<yoururlhere>) と組み合わせて使用します。

urlencode(name)

name に指定した変数を URL エンコード値に置き換えます。

注: 追加の属性またはパスワードに、特殊文字 (" . & = + ? ; / : @ = , \$ %) が含まれる場合、.fcc ファイル内の追加の属性値ごとに URL エンコーディングが必要になります。SiteMinder では US-ASCII エンコーディングが使用されています。

urldecode(name)

name に指定した変数を URL デコード値に置き換えます。

注: 名前/値ペアのプレフィクスである「sm」は、SiteMinder に必要な追加の特殊名用に予約されています。ログイン ページの名前を作成するときは、プレフィクス「sm」を使用しないでください。

ローカライゼーション用の名前/値ペア

.fcc ファイルには、2 種類のローカライゼーション パラメータが含まれています。

SMLOCALE

ユーザ情報を収集またはステータスメッセージを表示する HTML フォームで使用される言語を決定するのに使用されます。

smlocale とペアになっている値は、ローカライズされたプロパティファイルの名前の一部と対応しています。ローカライズされたプロパティ ファイルには、指定した言語の文字列にマップされた ID が含まれています。

SMLOCALE 値は次のようなフォーマットになります。

COUNTRY-LANGUAGE

たとえば、米語用の **SMLOCALE** 値は次のように表されます。

SMLOCALE=US-EN

SMENC

使用する言語エンコーディングをブラウザに指示する情報が含まれています。この変数のデフォルト値を変更すると、次の **META** タグのエンコーディングセットが上書きされます。

```
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1">
```

追加属性の収集

ユーザ名とパスワードだけでなく、ユーザの電子メールアドレスや役職などの詳細属性を収集することができます。

追加属性の収集方法

1. HTML フォーム認証方式と対応する .fcc ファイルを設定して、収集する属性を指定します。
2. HTML フォーム認証方式を設定するには、[方式のセットアップ]ダイアログボックスの[詳細属性リスト]フィールドに新しい属性を指定します。

注: 追加の属性またはパスワードに、特殊文字 (" . & = + ? ; / : @ = , \$ %) が含まれる場合、.fcc ファイル内の追加の属性値ごとに URL エンコーディングが必要になります。SiteMinder では US-ASCII エンコーディングが使用されています。

3. 詳細属性を収集するように .fcc ファイルを変更します。

.fcc ファイルを変更するには、.fcc ファイルの始めに次の行を追加します。

```
@password=PASSWORD=%PASSWORD%&newattr1=%newattr1%&newattr2=%newattr2%
```

詳細属性に特殊文字が含まれる場合は、次のようになります。

```
@password=PASSWORD=%urlencode(PASSWORD)%&newattr1=%urlencode(newattr1)%&newattr2=%urlencode(newattr2)%
```

`newattr1=%newattr1%` と `newattr2=%newattr2%` は詳細属性です。等号の前にある値は属性名で、パーセント記号(%)の間にある記号は属性値です。

FCC は、属性値から新しい属性の名前を解析します。

注: アンパサンド(&) 文字を使用して、@password 制御文に詳細属性を追加してください。

FCC に詳細属性を追加する場合は、以下のことに注意してください。

- 属性名は、`%attribute_name%` というフォーマットで識別されます。この属性名は、FCC に追加する input name エントリと一致している必要があります。たとえば、address という新しい属性を追加するには、次のように指定します。

```
@password=PASSWORD=%PASSWORD%&mail=%address%
```

または

```
@password=PASSWORD=%urlencode(PASSWORD)%&mail=%urlencode(address)%
```

この場合には、次のような行を .fcc ファイルに追加します。

```
<input name="address" type="text">
```

- 詳細属性の名前は、ユーザ ディレクトリにある属性名と一致する必要があります。たとえば、LDAP ディレクトリからユーザの電子メール アドレスを収集する場合は、次のように指定します。

```
@password=PASSWORD=%PASSWORD%&mail=%address%
```

または

```
@password=PASSWORD=%urlencode(PASSWORD)%&mail=%urlencode(address)%
```

mail は、電子メールアドレスが格納されている LDAP 属性の名前です。

- `@password` ディレクティブに指定する値の後にスペースを追加しないでください。追加のスペースは意味のある文字として解釈され、ユーザが認証されるのを妨げる場合があります。
- HTML フォーム認証方式の属性名は、`.fcc` ファイルの詳細属性の名前と一致していなければなりません。

たとえば、上記の例の属性 `mail` を認証方式に追加するには、[詳細属性] フィールドに次のように入力します。

`AL=PASSWORD,mail`

注：詳細属性の名前では、大文字と小文字が区別されます。

詳細情報：

[HTML フォーム認証方式 \(P. 324\)](#)

ログインに失敗した理由をユーザに伝える

フォームベース認証のデフォルト動作では、認証されていない、または許可されていないユーザは元のログインフォームにリダイレクトされます。ユーザがさらにログインを試行できるように、`smretries` ディレクティブ (`@smretries`) を設定できますが、デフォルト動作ではログインに失敗した理由をユーザに通知するメッセージが表示されません。

SiteMinder Web エージェントには、`DynamicRetry.fcc` ファイルと `DynamicRetry.unauth` ファイルが付属しています。このサンプルの `.fcc` ファイルのペアで、リダイレクトの動作を変更します。ログインの試行に 1 回失敗すると、ユーザを非許可ページ (`DynamicRetry.unauth`) に送るようにログイン ページ (`DynamicRetry.fcc`) を設定します。非許可ページは、ログイン ページとは異なる HTML ページです。このため、非許可ページには、ログインを失敗した理由を伝えるメッセージを含むことができます。デフォルトで、非認可のページには、無効なクレデンシャルを入力してリソースにアクセスしようとしていることをユーザに伝えるメッセージが設定されています。

注：このメッセージは、`DynamicRetry.unauth` を開き、`h3` タグに囲まれているテキストを更新すれば変更できます。

ログインに失敗した理由をユーザに伝えるには、認証方式を設定するときに **DynamicRetry.fcc** ファイルへのターゲットパスを指定します。**DynamicRetry.fcc** ファイルへのデフォルトパスは、**agent_home\samples\forms\DynamicRetry.fcc** です。

agent_home

SiteMinder Web エージェントのインストールパスを指定します。

.fcc ファイルの **DynamicRetry** ペアを使用するときは、以下の制限があるため注意してください。

- ユーザログインの試行回数は、**SMTRYNO cookie** を基準にしてカウントできません。
- ログインの試行回数は制限できません。

注: この方法とパスワードサービスを組み合わせれば、指定した回数を失敗したユーザをパスワードポリシーによって無効にできます。パスワードポリシーの詳細については、「**Password Policies**」を参照してください。

HTML フォーム認証方式の設定

カスタムの HTML フォームでユーザを認証するには、HTML フォーム認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「**ポリシー サーバ オブジェクトの複製**」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[HTML フォーム テンプレート]を選択します。

認証方式固有のフィールドとコントロールが開きます。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに、サーバ名、ターゲット、および属性リストの情報を入力します。

注: [ターゲット]フィールドに指定する .fcc ファイルが、前提条件に記載されているガイドラインに準拠していることを確認してください。

7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

ブラウザ以外のクライアントのサポートの有効化

ブラウザ以外の HTTP クライアントを使用してユーザを認証するように、基本クレデンシヤル(ユーザ名とパスワード)を収集する HTML フォーム方式を設定することができます。これらのクライアントは、Perl スクリプト、C++、Java プログラムなどを使用して開発され、HTTP プロトコルを使用して通信します。

カスタム クライアントは、最初のリクエストと一緒に、HTTP 許可ヘッダを介して基本クレデンシヤルを送信する必要があります。そうしないと、SiteMinder はユーザを認証しません。クレデンシヤルが HTTP 許可ヘッダを介して送信されないと、SiteMinder は、ブラウザ以外のクライアントをサポートせず、HTML フォーム方式にリダイレクトします。

ブラウザ以外のクライアントのサポートを有効にする方法

1. HTML フォーム認証方式を開きます。
2. [ブラウザ以外のクライアントのサポート]チェック ボックスをオンにします。
3. [サブミット]をクリックします。

ブラウザ以外のクライアントのサポートが有効になります。

Windows 認証方式

統合 Windows 認証 (IWA) は、純粋な Windows 環境でユーザを検証するための Microsoft 独自開発のメカニズムです。IWA では、最初の対話形式のデスクトップ ログイン プロセス時、およびその後のセキュリティレイヤへの情報の転送時に Windows がユーザ クレデンシヤルを取得することで、シングル サインオンを実現しています。SiteMinder は Windows 認証方式を使用して、Microsoft 統合 Windows 認証インフラストラクチャが取得したユーザ クレデンシヤルを処理することによってリソースを保護します。

SiteMinder の以前のバージョンは NTLM 認証方式によって Windows 認証に対応していました。ただし、このサポートは、NT ドメインを備えた環境や、混合モードのレガシー NT ドメインをサポートするように Active Directory サービスが設定されている環境に制限されていました。

Windows 認証方式では、SiteMinder がネイティブ モードで実行されている Active Directory および、NTLM 認証に対応するように設定された Active Directory の展開におけるアクセス コントロールを提供できるようにします。Windows 認証方式は、SiteMinder の以前の NTLM 認証方式に代わるものです。既存の NTLM 認証方式は引き続きサポートされ、新しい Windows 認証方式を使用して設定することができます。

注: 状況によっては、Windows 認証方式を使用するのではなく、Windows ユーザ セキュリティ コンテキスト機能と他の認証方式を併用するほうがよい場合もあります。

Windows 認証方式は、IIS Web サーバ上の Web エージェントで保護されていて、ユーザが Internet Explorer Web ブラウザを介してリソースにアクセスする場合に使用します。この方式では、ユーザのクレデンシヤルを取得して確認できるように IIS Web サーバを正しく設定する必要があります。ポリシー サーバは、IIS サーバで保証されたユーザの識別情報に基づいて許可を決定します。

Kerberos のサポート

Kerberos は、Windows 2000 でドメイン認証に使用される認証方式で、ユーザとコンピュータのプリンシパルは Active Directory に格納されます。Kerberos は、プラットフォームに関係なく、認証とシングルサインオンを実現できるアーキテクチャを提供します。ただし、Kerberos をサポートするのは、Windows 2000、Windows XP、および .NET で動作する Microsoft Web サーバとブラウザのみです。

SiteMinder は、Windows 認証方式を使用して、間接的に Kerberos 認証をサポートしています。SiteMinder 4.61 で Kerberos 認証を使用できるのは、Web サーバが Microsoft IIS で、ブラウザが Microsoft IE の場合のみです。SiteMinder リリース 5.x では、IIS サーバへの NTLM 認証のリダイレクトがサポートされるため、任意の SiteMinder Web エージェントを使用できます。

ユーザが、NT 認証を使用してデスクトップにログインし、IE を使用して、任意の Web サーバ(IIS 以外の Web サーバを含む)に展開された e-Business アプリケーションにアクセスする場合に、SiteMinder を使用するように設定されている IIS Web サーバがあれば、そのユーザは再認証を要求されることなく SiteMinder にログインできます。この強力な機能を使用すれば、企業はアプリケーションに適切なプラットフォームを自由に選択できる柔軟性が得られる上に、ユーザはデスクトップのパスワードを覚えておくだけで済みます。

Windows 認証方式の前提条件

SSL を介した基本認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- 混合モードのレガシー WinNT ディレクトリまたは Active Directory:
 - 管理 UI で作成するユーザ ディレクトリ接続で、WinNT ネームスペースが指定されていること。
 - 要求されたリソースは、任意のタイプの Web サーバに置くことができます。しかし、それらのリソースを保護する認証サーバと Web エージェントは、Microsoft IIS Web サーバ上にある必要があります。
- ネイティブ モードで稼働している Active Directory:
 - ユーザ データが Active Directory にあること。
 - ユーザ ディレクトリ接続で、LDAP または AD のどちらかのネームスペースが指定されていること。
 - 要求されたリソースは、任意のタイプの Web サーバに置くことができます。しかし、それらのリソースを保護する認証サーバと Web エージェントは、Microsoft IIS Web サーバ上にある必要があります。
 - クライアント アカウントとサーバ アカウントが委任に対応していること。
- ユーザは Internet Explorer (バージョン 4.0 以降)を使用してログインすること。

- Windows 2003 で IIS を使用する場合は、[ワイルドカード アプリケーション マップ]の[ファイルの存在を確認する]オプションが設定されていないこと。
- Windows 認証方式では、creds.ntc ファイルを含む IIS Web サーバ上のすべての仮想ディレクトリが保護されないままであることが必要になります。
- Internet Explorer ブラウザのオプションが、ユーザの現在のユーザ名とパスワードを使用して自動的にログオンできるように設定されていること。

Internet Explorer 5.x および 6.x の各ブラウザで自動ログオンを設定する方法

1. Internet Explorer のメニュー バーで、[ツール]-[インターネット オプション]を選択します。
2. [インターネットオプション]ダイアログ ボックスが開きます。
3. [セキュリティ]タブをクリックして表示します。
4. 使用しているインターネット ゾーンを選択して[レベルのカスタマイズ]をクリックします。
5. [セキュリティの設定]ダイアログ ボックスが開きます。
6. [ユーザ認証]で、[ログオン]が表示されるまで下にスクロールします。
7. [現在のユーザ名とパスワードで自動的にログオンする]ラジオボタンをオンにします。
8. [OK]をクリックします。

Internet Explorer 4.x のブラウザで自動ログオンを設定する方法

1. Internet Explorer のメニュー バーで、[ツール]-[インターネット オプション]を選択します。
2. [インターネットオプション]ダイアログ ボックスが開きます。
3. [セキュリティ]タブをクリックして表示します。
4. ドロップダウンリストから使用しているインターネット ゾーンを選択します。
5. [インターネットゾーン]グループボックスで[カスタム]ラジオボタンをオンにし、[設定]をクリックします。
6. [セキュリティの設定]ダイアログ ボックスが開きます。
7. [ユーザ認証]で、[ログオン]が表示されるまで下にスクロールします。
8. [現在のユーザ名とパスワードで自動的にログオンする]ラジオボタンをオンにします。
9. [OK]をクリックします。

Windows 認証方式に関する考慮事項

ポリシー サーバではなく、IIS Web サーバが、Internet Explorer ブラウザから受け取ったクレデンシャルに基づいて認証を実行します。したがって、**OnAuthAttempt** 認証イベントを使用して、ユーザ ストアに存在しないユーザをリダイレクトすることはできません。

Windows 認証方式の設定

Windows 認証方式を使用して、純粋な Windows 環境にあるユーザを許可します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。
認証方式の設定が開きます。
注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [認証タイプのスタイル]リストから[Windows 認証テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスにサーバ名、ターゲット、およびユーザ DN 情報を入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

Information Card 認証方式

SiteMinder の Information Card 認証方式 (ICAS) 機能を使用すると、複数の Information Card 認証方式を作成できます。各方式はカスタム認証方式として設定されます。

情報カード (Information Card) の概要

情報カードは、私たちが財布に入れて持ち運ぶ物理的なカードに似ています。情報カードのそれぞれが識別情報のセットです。たとえば、運転免許である情報カードには、写真、生年月日、氏名、運転免許番号といった重要な個人情報が含まれています。

情報カードにより、ユーザはそれぞれの識別情報を管理できます。ユーザは情報カードと関連する識別情報を表示できます。情報交換する際には、その情報について使用可能なカードから選択します。また、ユーザは選択したカードに関連付けられている識別情報のリリースを許可できます。

情報カードはアイデンティティ セレクタで表示されます。

アイデンティティ セレクタの概要

アイデンティティ セレクタは、ユーザがそれぞれの識別情報のほか、**Relying Party** およびアイデンティティ プロバイダとのオンライン関係を管理できるアプリケーションです。**Relying Party (RP)** は、ユーザの認証に識別情報を必要とする **Web** サイト アプリケーション、またはサービスです。アイデンティティ プロバイダ (**IdP**) は、識別情報を認証し、ユーザが **Relying Party** と共有できるセキュリティ トークンを作成する第三者です。

アイデンティティ セレクタでは、**Web** ベースのリソースにアクセスする際、ユーザが多くのユーザ名とパスワードを管理する必要がありません。同様に、企業はユーザの識別情報のデータベースを保守する必要がなくなるため、不正確な情報、古い情報、誤用による脆弱性などのリスクや問題が減り、使いやすさが向上します。

また、アイデンティティ セレクタにより、ユーザは各 **Relying Party** にリリースする識別情報を正確に制御できます。さらに、アイデンティティ セレクタのユーザ インターフェースは一貫しており、ユーザの使いやすさが増しています。

Windows CardSpace

Microsoft が提供しているアイデンティティセクタの Windows CardSpace は、任意の Relying Party またはアイデンティティプロバイダと対話できる一貫したユーザ インターフェースをユーザに提供します。SiteMinder は、Information Card 認証方式 (ICAS) と呼ばれるカスタム認証方式を使って Windows CardSpace をサポートしています。

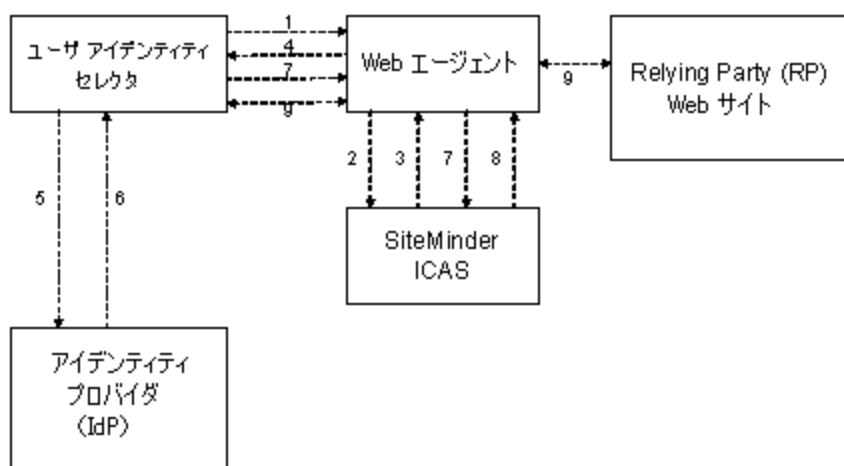
SiteMinder Information Card 認証方式 (ICAS)

SiteMinder Information Card 認証方式 (ICAS) は Windows CardSpace をサポートする SiteMinder 認証方式です。ICAS の各インスタンスを 管理 UI でカスタム認証方式として設定し、他の SiteMinder カスタム認証方式と同様に実装します。

ICAS の概要

SiteMinder ICAS によるユーザ認証プロセスには、以下のコンポーネントと手順が含まれます。

- ユーザ
- アイデンティティ セレクタ
- Web エージェント
- Relying Party (RP)
- アイデンティティ プロバイダ (IdP)



1. ユーザが SiteMinder で保護された Web サイト、または Relying Party (RP) を訪問しようとします。
2. Web エージェントはユーザのリクエストをインターセプトし、ICAS を呼び出します。
3. ICAS は RP のポリシー要件を Web エージェントに送信します。
4. Web エージェントは、ユーザのブラウザにコンピュータでアイデンティティ セレクタを起動するように指示し、RP のポリシー要件を送信します。

- アイデンティティセレクトはポリシー要件を読み取り、要件を満たしている情報カードをユーザにわかるようにハイライトします。ユーザはハイライトされているカードを1つ選択します。アイデンティティセレクトはユーザのクレデンシャルを収集し、認証を受けるためにアイデンティティプロバイダ(IdP)に送信します。アイデンティティセレクトは、さらにRPのポリシー要件をIdPに送信し、トークンをリクエストします。

注: ユーザは、RPで必要としないオプションのクレームを含んでいるカードを選択できます。

- IdPはユーザを認証し、ポリシー要件を処理します。必要なクレームを含むトークンを生成してアイデンティティセレクトに送り返します。
- アイデンティティセレクトはクレームを表示し、ユーザはRPへのクレームのリリースを承認します。
- ICASはトークンを復号化し、トークンの信頼性と整合性を検証したうえで、ユーザデータベースでユーザのクレームをユーザの身元に関連付けます。その後、SiteMinderは標準のポリシーベースの許可を実行し、許可された場合はユーザにアクセスを付与します。
- ユーザはWebサイトにアクセスします。

ICAS の用語

ICASの説明で使われる用語の説明は、以下のとおりです。

アイデンティティメタシステム

ユーザ、Relying Party、アイデンティティプロバイダで識別情報を共有できる方法を指定するアーキテクチャです。

ユーザ

識別情報が共有されている人です。ユーザは対象と呼ばれることもあります。

Relying Party (RP)

識別情報をリクエストして使用するWebサイトです。

アイデンティティプロバイダ(IdP)

識別情報を認証し、セキュリティトークンを作成してRelying Partyとその情報を共有する第三者です。クレジットカード会社、銀行、政府系機関、雇用者および保険会社は、すべてアイデンティティプロバイダの例です。

セキュリティトークン サービス(STS)

セキュリティトークンを作成するためにアイデンティティプロバイダによって使われる技術です。セキュリティトークン サービスでは以下を実行します。

- ユーザの認証
- セキュリティトークンの作成
 - 作成されるセキュリティトークンには識別情報のさまざまなサブセットが含まれ、これらは **Relying Party** の要件およびユーザの制限によって異なります。
 - 作成されるセキュリティトークンにはさまざまなタイプがあります。
注: SiteMinder は、SAML 1.0 および 1.1 をサポートしています。
 - これらセキュリティトークンはセキュリティ目的で暗号化されており、信頼性と整合性について署名済みです。

セキュリティトークン

暗号化されたクレームのセットで、暗号を使って署名されています。

クレーム

真であることのアサーションです。各トークンにはそれぞれ、ユーザの身元に関する 1 つ以上のクレームが含まれています。クレームには、名、姓、電子メール アドレス、生年月日などがあります。クレームは、ユーザまたは第三者のアイデンティティプロバイダが作成できます。

情報カード

一連の識別情報です。情報カードは、私たちが財布に入れて持ち運ぶ物理的なカードに似ています。たとえば、運転免許に相当する情報カードには、写真、生年月日、氏名、運転免許番号、状態、身長、性別といった重要な個人情報が含まれています。

個人カード

ユーザが自分の身元を保証するクレームを含む個人カードです。ただし、このクレームは第三者による確認を受けていません。個人カードには、カードの作成時に生成される秘密個人識別子 (PPID) が含まれます。電子メールアドレスなど、重要性の低い識別情報には個人カードが適切です。

注: 個人カードは、自己発行カードとも呼ばれます。

管理カード

ユーザが自分の身元を保証するクレームを含む情報カードです。このクレームは第三者による確認を受けています。管理カードには、カードの作成時に生成される秘密個人識別子 (PPID) とアイデンティティプロバイダの STS へのポインタが含まれます。クレジットカード番号など、重要性の高い識別情報には管理カードが適切です。

アイデンティティ セレクタ

ユーザに、**Relying Party** およびアイデンティティプロバイダとの関係を管理できるようにするほか、それぞれの識別情報を共有して使用する方法を制御できるようにするアプリケーションです。アイデンティティ セレクタでは以下が可能です。

- 通信プロトコルやセキュリティ技術が異なるパーティ間で情報を共有できるようにする。
- 情報カードを使用して、さまざまなアイデンティティプロバイダや **Relying Party** に一貫したユーザ インターフェースを提供する。
- 識別情報の各交換で利用できる情報カードをハイライトする。
- ユーザが識別情報を表示し、その共有に同意できるようにする。

Windows CardSpace

Windows オペレーティング システム用の Microsoft のアイデンティティ セレクタです。

Information Card 認証方式 (ICAS)

Microsoft のアイデンティティ セレクタである **Windows CardSpace** をサポートしており、**SiteMinder** にカスタム認証方式として実装されています。

秘密個人識別子 (PPID)

情報カードの作成時にアイデンティティ セレクタによって生成される識別子です。

ICAS ファイル

SiteMinder は、ICAS の各インスタンスを設定するために 2 つのファイル、つまり fcc ファイルとプロパティファイルを使用します。

filename.fcc

SiteMinder で必要とし、ICAS のインスタンスごとにカスタマイズできる認証設定を指定します。

InfoCard.fcc

Web エージェント キットに付属しているサンプルの fcc ファイル

filename.properties

ICAS のインスタンスの動作を指定します。

InfoCard.properties

サンプルのプロパティファイル

注: 管理 UI で ICAS のインスタンスを設定する際、管理者はプロパティファイルへのパスを指定します。

ICAS に関する要件

ICAS を実装するには、まず以下の条件を満たしている必要があります。

Web ブラウザの設定

使用する Web ブラウザは以下のいずれかでなければなりません。

- Internet Explorer 7.0
- CardSpace プラグインを備えた Firefox 2.x

Web サーバ設定

Web サーバには、SSL 通信を設定する必要があります。この設定は fcc ファイルを保護します。

注: 詳細については、「Web エージェント設定ガイド」を参照してください。

Web エージェント設定

Web エージェント キットに付属している InfoCard.fcc ファイルは、ICAS の各インスタンスに合わせてカスタマイズする必要があります。

注: 詳細については、「Web エージェント設定ガイド」を参照してください。

Java Runtime Environment (JRE) の設定

Java Runtime Environment は、強力な暗号化アルゴリズムで暗号化されているセキュリティトークンを復号化するように設定する必要があります。

ポリシー サーバ設定

ポリシー サーバ設定には、以下のタスクが含まれます。

- ICAS プロパティファイルの設定
- SiteMinder キー データベースを設定する方法
- ICAS で使用するユーザ ディレクトリの設定
- ICAS のインスタンスの作成
- クレーム値を取得するアクティブ レスポンスの設定

ICAS 用の Java Runtime Environment (JRE) の設定

Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction ポリシー ファイルをインストールして、Java Runtime Environment (JRE)を設定します。これらのファイルは強力な暗号化アルゴリズムで暗号化されたセキュリティトークンを復号化するために必要です。

重要: 新しいポリシー ファイルをインストールする場合は、その前に JRE に付属していたデフォルトのポリシー ファイルをバックアップしてください。

ICAS 用の Java Runtime Environment (JRE)を設定する方法

1. Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction ポリシー ファイルを http://java.sun.com/javase/downloads/index_jdk5.jsp からダウンロードし、\$NETE_JRE_ROOT\lib\security ディレクトリにインストールします。
2. \$NETE_JRE_ROOT\lib\security\java.security ファイルに以下の行を追加します。

```
security.provider.7=com.rsa.jsafe.provider.JsafeJCE
```

ICAS で使用するポリシー サーバを設定する方法

ICAS で使用するポリシー サーバの設定に含まれる手順は以下のとおりです。

- ICAS プロパティファイルの設定
- 後からアクティブ レスポンスで使用するクレームの保存
- ICAS で使用する SiteMinder キー データベースを設定する方法
- ICAS で使用するユーザ ディレクトリの設定
- ICAS のインスタンスの作成
- クレーム値を取得するアクティブ レスポンスの設定

ICAS プロパティファイルの設定

ICAS プロパティファイルは、ICAS インスタンスの動作方法を指定します。管理 UI で ICAS のインスタンスを設定する際、管理者は関連するプロパティファイルへのパスを指定します。新しいプロパティファイルを設定するには、サンプルのプロパティファイルをテキスト エディタで開き、内容を編集します。必ず名前を変更して新しいプロパティファイルとして保存します。

注: 複数の ICAS インスタンスで同じプロパティファイルを共有できます。

例: InfoCard.properties という名前のプロパティファイルには、以下のプロパティとサンプル値が含まれています。

fcc

fcc ファイルの場所を指定します。

例: fcc=https://web_server_home/siteminderagent/forms/InfoCard.fcc

注: アイデンティティ セレクタを有効にするには、「https」を指定します。

vppid_claim

ユーザを判別するために使用するクレームを指定します。

例:

vppid_claim=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname

vppid_claim=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname

vppid_claim=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

alias

Relying Party の SSL 証明書を取得するために使用する SiteMinder キー ストアのキーを指定します。

例: alias=rpssl

tokenPrim

tokenPrim インターフェースのプロバイダを指定します。

例: tokenPrim=com.ca.sm.authscheme.infocard.higgins.TokenAdapter

後からアクティブ レスポンスで使用するクレームの保存

アクティブ レスポンスで後から使用できるようにクレームを保存できます。後から使用できるようにクレームを保存するには、プロパティファイルに以下のプロパティを追加します。

postprocessingchain

ユーザ認証中に実行するコマンドのチェーンを定義します。この段階にはクレーム変換コマンドとストレージコマンドが含まれます。

例:

postprocessingchain=com.ca.sm.authscheme.infocard.command.StoreClaimsToContext

ICAS で使用する SiteMinder キー データベースを設定する方法

Relying Party は、SSL を使用して fcc ファイルを保護する必要があります。
Relying Party は、Web サイトに関連付けられている SSL 証明書を pfx ファイルにエクスポートする必要があります。SiteMinder 管理者は、smkeytool を使用して、pfx ファイルから smkeydatabase に SSL 証明書をインポートできます。

証明書は smkeydatabase にインポートする際、別名に関連付けられます。この別名は、fcc ファイルに保存されます。証明書の秘密キーを使用してセキュリティトークンを復号化し、デジタル署名を検証します。

SiteMinder キー データベースの設定は 2 段階のプロセスです。

1. IIS Web サーバからローカル マシン上の pfx ファイルに SSL 証明書をエクスポートするには、Web サーバ証明書ウィザードを使用できます。詳細については、Microsoft のマニュアルを参照してください。
2. smkeytool を使用して、pfx ファイルから smkeydatabase に SSL 証明書をインポートするには、以下の例にあるオプションを指定して、smkeytool.bat を実行します。

```
smkeytool.bat -addPrivKey -alias example -keycertfile  
c:\Temp\www-example-com.pfx  
-password CAdemo123
```

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

addPrivKey

smkeytool で実行するアクションを指定します。

alias

smkeydatabase に SSL 証明書の名前を指定します。

注: これはプロパティ ファイルで指定される別名です。

keycertfile

ローカル マシンの pfx ファイルの場所を指定します。

password

SSL 証明書を pfx ファイルにエクスポートする際に指定したパスワードを指定します。

注: SSL 証明書を pfx ファイルにエクスポートするときに指定するパスワードは、後で pfx ファイルから SSL 証明書をインポートするときに SiteMinder で使用されます。

注: smkeydatabase がいない場合は、ポリシー サーバ設定ウィザードを使用して作成できます。詳細については、「ポリシー サーバ インストール ガイド」を参照してください。

注: smkeydatabase と smkeytool の詳細については、「*Federation Security Services Guide*」を参照してください。

ICAS で使用するユーザ ディレクトリの設定

ユーザの認証は、ICAS に提示されるクレームの 1 つとユーザ データベース内のユーザ属性が一致するかどうかによります。トークンの逆アセンブリ時、指定されたクレーム値はユーザ ディレクトリで検索値として使われます。このため、指定されたクレームに対応するユーザ属性が LDAP 検索文字列または SQL クエリ方式で指定されるように、ユーザ ディレクトリを設定する必要があります。以下の例は、電子メール アドレスに LDAP 参照文字列と SQL クエリ方式を設定する方法を示します。

LDAP の例

LDAP ユーザ DN 検索グループ ボックス

先頭

(mail=

終端

)

SQL の例

SQL クエリ グループボックス

ユーザ/グループ情報の取得

```
SELECT EmailAddress, 'User' FROM SmUser WHERE EmailAddress = '%s'  
UNION SELECT Name, 'Group' FROM SmGroup WHERE Name = '%s'
```

ユーザの認証

```
SELECT EmailAddress FROM SmUser WHERE EmailAddress = '%s' AND  
Password = '%s'
```

ICAS のインスタンスの作成

管理 UI でカスタム認証方式を指定し、ICAS のインスタンスを作成します。

制限: ポリシー ストアごとに、ICAS のインスタンスを最大 10 までサポートできます。

ICAS のインスタンスを作成する方法

1. [インフラストラクチャ]、[認証]、[認証方式]、[認証方式の作成]をクリックします。

[認証方式の作成]ペインが表示されます。

2. 認証方式タイプのオブジェクトを作成するように選択し、[OK]をクリックします。

[認証方式の作成: <名前>]ペインが表示されます。

3. [一般]グループ ボックスのフィールドに、認証方式の名前と説明を入力します。

4. [各方式共通セットアップ]グループ ボックスの[認証方式タイプ]ドロップダウンリストから、[カスタム テンプレート]を選択します。

[方式のセットアップ]グループ ボックスは、[詳細]グループ ボックスになります。

5. [各方式共通セットアップ]グループ ボックスの[保護レベル]フィールドに値を入力します。

6. [各方式共通セットアップ]グループ ボックスの[パスワード ポリシー]チェックボックスをオフにします。

注: ICAS はパスワード ポリシーをサポートしません。

7. [方式のセットアップ]グループ ボックスのフィールドに以下の値を入力します。

ライブラリ

smjavaapi

注: カスタム認証方式は Java Authentication API を使用します。

秘密キーと秘密キーの確認入力

これらのフィールドは空白のままにします。

注: カスタム認証方式は共有秘密キーを使用しません。

パラメータ

[パラメータ]フィールドに以下の 2 つのパラメータをスペースで区切って入力します。

`com.ca.sm.icas.SmAuthInfoCard`

これは、`SmAuthScheme` インターフェースを実装するクラスの完全修飾名です。

`policy_server_home¥config¥icas¥InfoCard.properties`

これはプロパティファイルの場所です。

例:

`com.ca.sm.icas.SmAuthInfoCard`
`policy_server_home¥config¥icas¥InfoCard.properties`

8. [サブミット]をクリックします。

認証方式の作成タスクが処理のためにサブミットされます。

クレーム値を取得するアクティブ レスポンスの設定

カスタム クラスの `com.ca.sm.icas.GetClaimValue` を使用して、認証の完了後にクレーム値を取得するアクティブ レスポンスを設定します。

注: クレーム値の格納と取得には、セッション サーバが必要です。セッション サーバの詳細については、「[ポリシー サーバ管理ガイド](#)」を参照してください。

クレーム値を取得するアクティブ レスポンスを設定する方法

1. [ポリシー]、[ドメイン]、[レスポンス]、[レスポンスの作成]をクリックします。
[レスポンスの作成:ドメインの選択]ペインが表示されます。
2. ドメインを選択し、[次へ]をクリックします。
[レスポンスの作成:レスポンスの定義]ペインが表示されます。
3. [一般]グループ ボックスのフィールドにレスポンスの名前および説明を入力します。
4. [属性]グループ ボックスの[エージェントタイプ]で[Web エージェント]を指定します。
5. [属性リスト]グループ ボックスの[レスポンス属性の作成]をクリックします。
[レスポンス属性の作成:<名前>]ペインが表示されます。

6. [属性タイプ]グループ ボックスの属性ドロップダウンリストから **WebAgent-HTTP-Header-Variable** または **WebAgent-HTTP-Cookie-Variable** を選択します。
7. [属性のセットアップ]グループ ボックスの[属性の種類]で[アクティブ レスポンス]を選択します。
8. [属性のセットアップ]グループ ボックスの[属性フィールド]に以下の値を入力します。

cookie 名または変数名

クレームの名前を指定します。

例: emailaddress

ライブラリ名

ライブラリ名を指定します。

値: smjavaapi

関数名

関数名を指定します。

値: JavaActiveExpression

パラメータ

カスタム ICAS コマンドのほか、情報カード モデルに従って標準のクレームタイプを定義する、claims.xsd ファイルの場所を指定します。

例: com.ca.sm.authscheme.infocard.GetClaimValue

http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

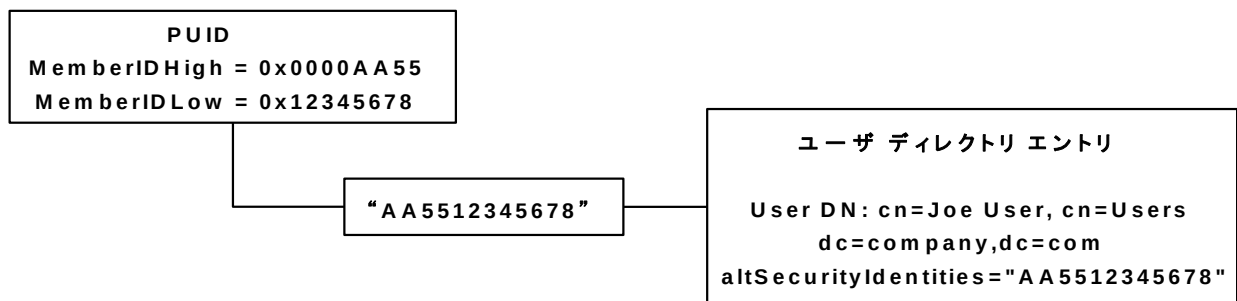
9. [OK]をクリックします。
レスポンスの作成タスクが処理のためにサブミットされます。

MS Passport 認証方式

Microsoft® .NET Passport は、参加サイトに共通のインターネット認証を提供するオンライン サービスです。.NET Passport アカウントを使用すれば、ユーザは一度ユーザ名とパスワードを入力するだけで、以降は認証情報を求められることなく、加入サイト間を自由に移動できます。.NET Passport 対応サイトとなるためには、.NET Passport シングル サインイン (SSI) を通じて .NET Passport 認証サービスのサポートを実装する必要があります。これには、共同ブランディングをサポートする共通のインターフェイスを使用して Passport 経由でログインするためのリンクの実装が含まれます。いったん Passport にユーザがログインすると、ブラウザがリダイレクトされて Passport 識別情報が取得され、Passport 識別情報を含むサイト cookie が作成されます。これにより、ユーザはそのサイトに自動的に認証されます。ただし、特定のユーザのアカウントについて参加サイトまたはリソースへのアクセスを許可または拒否する判断は行われません。

Passport 識別情報にはアクセス許可を判断するためのデータは一切含まれていません。識別情報を取得し、それをもとに許可を取得するための情報を提供するフィールドは、Passport Unique ID (PUID) だけです。SiteMinder はこの PUID を使用して、サイトをパーソナライズするのに使用するローカル ID をマッピングし、SiteMinder ポリシーで保護されたリソースへのアクセスを許可します。

PUID は文字列に変換され、ユーザ ディレクトリ属性を通じて SiteMinder 識別情報にマッピングされます。次の図では、"Joe User" のユーザ DN がディレクトリ属性 "altSecurityIdentities" を通じて Passport PUID にマッピングされます。



SiteMinder での Passport 認証は、Passport 識別情報を SiteMinder ユーザにマッピングすることを基本としているため、Passport を使用して認証やパーソナライズを行うには登録が必要になります。

登録は、認証方式に設定された登録 URL を通じて制御されます。設定パラメータは ?registrationurl=? です。この後に登録ページまたはキーワードの "FORM="、そして SiteMinder フォーム URL が続きます。このモデルでは、Passport ユーザは 2 つの方法で登録を行うことができます。

1 つ目の方法は、サイト提供の Web アプリケーションを使用し、認証方式に設定された属性を通じて Passport 識別情報を SiteMinder ユーザ アカウントに関連付ける方法です。これには、登録データを受け入れ、ユーザ ディレクトリ属性を設定するためのサービスを提供する Web ページを各サイトで用意する必要があります。SiteMinder を使用してユーザ ディレクトリへのインターフェースを提供し、トンネル サービスを利用してファイアウォールの内側に安全なトンネルを確立することができます。

2 つ目の登録方法は、SiteMinder FCC とフォーム認証モデルを利用する方法です。登録にキーワード FORM= が含まれる場合、以降の URL はフォームリダイレクトとして扱われます。SiteMinder には、フォームを使って Passport 登録ページを作成するためのテンプレートとして利用できる passport.fcc ファイルが用意されています。

SiteMinder r12.0 SP3 には、Passport 認証方式テンプレートが用意されています。ライブラリ名は smauthmspp です。この認証方式は、Windows 版、UNIX 版のいずれのポリシー サーバでも使用できます。

ポリシー サーバでの Passport 認証のサポート

ポリシー サーバと SiteMinder Web エージェントは、Passport 認証の以下の実装をサポートしています。

- 共通の登録 URL を通じて確立された Passport 認証
- 登録に SiteMinder HTML フォームを使用する Passport 認証
- Passport 識別情報が存在しない場合の匿名ログイン

社内での Passport 認証の展開方法については、<http://www.microsoft.com> を参照してください。

Passport 認証の保護レベルの設定

Passport 認証では、識別情報を確立するプロセスに参加サイトやリソースへのアクセスの許可が含まれないため、Passport 認証方式には、比較的到低い保護レベルを設定する必要があります。Passport 認証はパーソナライズに利用し、機密性の高いリソースには高い保護レベルを設定した認証方式を実行することをお勧めします。たとえば、Passport ユーザを認証でき、そのユーザの識別情報が SiteMinder 保護レベル 1 を使用して確立されるとします。ユーザが財務に関する機密情報を要求した場合は、保護レベル 10 の HTML フォーム認証方式を使用してそのユーザを強制的に再認証するように設定することができます。

Passport 認証方式の前提条件

Passport 認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- Passport SDK (WinNT ではバージョン 1.4、Win2000 ではバージョン 2.1) が設定されていること。
- Passport パートナー サイトが設定されていること。
- SiteMinder 5.x QMR1 または QMR v2 Web エージェントが、Windows システムの Internet Information Server (IIS) 上にインストールされ、稼働していること(使用可能な場合)。
- ポリシー サーバのバージョン 5.5 がインストールされていること。

MS Passport 認証方式の設定

参加している .NET Passport Web サイトを介してユーザを認証するには、MS Passport 認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[Microsoft Passport テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスに、設定情報を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

Passport 認証の検索仕様のマッピング

SiteMinder が Microsoft® .NET Passport によって確立された識別情報を認識できるように、検索仕様を設定します。検索仕様によって、SiteMinder はユーザーディレクトリで Passport Unique ID (PUID)を見つけることができます。

次のネームスペースについて検索仕様を設定することができます。

- LDAP
- ODBC
- カスタム

注: SiteMinder は、Windows NT ユーザー ディレクトリを使用した Passport 認証をサポートしていません。

ネームスペースの検索仕様をマッピングする方法

1. 目的の MS Passport 認証方式を開きます。
2. [方式のセットアップ]グループ ボックスの対応するディレクトリ フィールドに、マッピングを入力します。

マッピングは次のように設定します。

- LDAP:
`altSecurityIdentities=Kerberos:%s@company.local`
- ODBC:
`PassportUID=PUID:%s@company.local`
- カスタム:
`PassportUID=PUID:%s@company.local`

注: 1つの検索仕様を1つ以上のディレクトリタイプに適用するように認証方式を設定することができます。

3. [サブミット]をクリックします。

検索仕様が保存されます。

RADIUS CHAP/PAP 認証方式

RADIUS プロトコルは CHAP または PAP ベース認証の実行に使用できます。

PAP の概要

PAP (パスワード認証プロトコル) は、2 方向ハンドシェイクを使用してユーザを認証する簡単な方法です。PAP がこの処理を実行するのは、認証を行うサーバに初めてリンクするときのみです。この方式では、ID とパスワードのペアがユーザのマシンから認証を行うサーバに繰り返し送信されます。この送信は、認証が肯定応答されるか、接続が終了するまで続きます。

この認証方式が最も適しているのは、リモートホストで擬似的にログインするために、クリア テキスト パスワードが利用できない場合です。このような場合、この方式ではリモートホストでの通常ログインと同等レベルのセキュリティを提供します。

CHAP の概要

CHAP (チャレンジ ハンドシェイク式認証プロトコル) は、PAP よりも安全性の高い認証方式です。CHAP 認証方式では、ユーザを識別するために以下の処理を行います。

1. ユーザのマシンと認証を行うサーバ間でリンクが確立したら、サーバが接続リクエストにチャレンジ メッセージを送信します。リクエストは一方方向性ハッシュ関数を使用して、取得した値でこれにレスポンスします。
2. サーバは、リクエストから返された値とサーバで計算した予測ハッシュ値を比較して確認します。
3. 値が一致すると認証されます。不一致の場合は、通常、接続が切断されます。

サーバは接続しているパーティに対して、新しいチャレンジ メッセージの送信をいつでもリクエストできます。CHAP 識別子が頻繁に変更される点、およびサーバが認証をいつでもリクエストできる点から、CHAP は PAP よりも高い安全性を提供します。

RADIUS CHAP/PAP 方式の概要

RADIUS CHAP/PAP 方式では、ユーザのパスワードのダイジェストを計算してから、そのパスワードと RADIUS パケットの CHAP パスワードを比較してユーザを認証します。ダイジェストはユーザのハッシュ化パスワードで構成されます。これは、RADIUS CHAP/PAP 認証方式の設定時に指定したディレクトリ属性を使用して計算されます。

RADIUS CHAP/PAP 認証方式の前提条件

RADIUS CHAP/PAP 認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- クリア テキスト パスワード用に指定されているユーザ ディレクトリ内のフィールドに値が入っていること。
- ポリシー サーバが FIPS 専用モードで動作していないこと。ポリシー サーバが FIPS 専用モードで動作していると、RADIUS CHAP/PAP 認証方式はサポートされません。

RADIUS CHAP/PAP 認証方式の設定

RADIUS プロトコルを使用している場合、RADIUS CHAP/PAP 認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[RADIUS CHAP/PAP テンプレート]を選択します。

認証方式固有の設定が開きます。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスで平文のパスワードを指定します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレلمに割り当てられます。

RADIUS サーバ認証方式

SiteMinder では、ポリシー サーバを RADIUS サーバとして、NAS クライアントを RADIUS クライアントとして使用することにより RADIUS プロトコルに対応しています。RADIUS エージェントは、ポリシー サーバと NAS クライアントデバイスの通信を可能にします。RADIUS サーバ認証方式では、ポリシー サーバは SiteMinder で保護されたネットワークに接続している RADIUS サーバとして機能します。

この方式では、ユーザ名およびパスワードをユーザ証明として受け付けます。また、複数のインスタンスを定義できます。この方式では、RADIUS サーバが認証レスポンスで返す RADIUS 属性を解釈しません。

SiteMinder での RADIUS サーバ認証の詳細については、「ポリシー サーバ管理ガイド」の RADIUS としてのポリシー サーバの使用についての内容を参照してください。

RADIUS サーバ認証方式の前提条件

RADIUS サーバ認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- RADIUS サーバが、ポリシー サーバがアクセスできるネットワーク上にあること。
- ポリシー サーバが FIPS 専用モードで動作していないこと。ポリシー サーバが FIPS 専用モードで動作していると、RADIUS サーバ認証方式はサポートされません。

RADIUS サーバ認証方式の設定

RADIUS サーバとしてポリシー サーバを使用しており、RADIUS クライアントとして NAS クライアントを使用している場合は、RADIUS サーバ認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。

3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[RADIUS サーバ テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスで、RADIUS サーバ IP アドレス、ポート番号、および共有秘密キーを入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

SafeWord サーバ認証方式

この認証方式では SafeWord サーバに照合してユーザを認証します。認証するユーザには、SafeWord 社のハードウェアトークンを介してログインするユーザも含まれます。この方式のインスタンスは複数定義できます。SafeWord サーバの正確な設定パラメータは SafeWord 設定ファイルに指定されています。

注: SafeWord 認証方式、smauthenigma および smauthenigmahtml は、6.0 SP3 CR03 リリース以降は Windows および Solaris のプラットフォームでのみ対応しています。

SafeWord サーバ認証方式の前提条件

SafeWord 認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- ポリシー サーバがアクセスできるネットワーク上に SafeWord サーバがインストールされていること。
- SafeWord サーバの正確な場所が SafeWord 設定ファイルに指定されていること。

SafeWord サーバ認証方式の設定

SafeWord サーバに対してユーザ (SafeWord ハードウェアトークンを介してログインしているユーザを含む) を認証する必要がある場合に、SafeWord サーバ認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。

3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[カスタム SafeWord テンプレート]を選択します。
認証方式固有のフィールドとコントロールが開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスで **SafeWord** サーバ設定ファイルの場所を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

SafeWord サーバ認証方式と HTML フォーム認証方式

この認証方式では、カスタム HTML フォームを使用しながら **SafeWord** サーバに照合してユーザを認証します。認証するユーザには、**SafeWord** 社のハードウェアトークンを介してログインするユーザも含まれます。この方式のインスタンスは複数定義できます。**SafeWord** サーバの正確な設定パラメータは **SafeWord** 設定ファイルに指定されています。

SafeWord サーバ認証方式と HTML フォーム認証方式の前提条件

SafeWord サーバおよび HTML フォーム認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- ポリシー サーバがアクセスできるネットワーク上に **SafeWord** サーバがインストールされていること。
- **SafeWord** サーバの正確な場所が **SafeWord** 設定ファイルに指定されていること。
- カスタマイズした **.fcc** ファイルが、HTML フォーム認証を実行する **cookie** ドメイン内の **Web** エージェントのサーバ上にあること。サンプルの **.fcc** ファイルは、**Web** エージェントをインストールした **Samples/Forms** サブディレクトリの下にあります。
- カスタマイズされた **.unauth** ファイルが、**Web** エージェントのサーバ上にあること。

注: **.unauth** ファイルは、**.fcc** ファイルが **smerrorpage** ディレクティブを使用する場合は必要ありません。

- ポリシー サーバとユーザ ディレクトリの上にディレクトリ接続が確立されていること。

- デフォルトの HTML フォームライブラリがインストールされていること。HTML フォームライブラリが HTML フォーム認証を処理します。
 - Windows では SmAuthHTML.dll。
 - Solaris では smauthhtml.so。これらのファイルは Web エージェントの設定時に自動的にインストールされます。
- (Sun Java Systems) Sun Java Systems の Web サーバを使用している場合は、magnus.conf ファイル内の StackSize パラメータの値を 131072 より大きい値に増加してください。値を変更しないと、Web サーバはそのコアをダンプし、SiteMinder がフォームを使用して認証リクエストを行うたびに再起動します。

SafeWord サーバ認証方式と HTML フォーム認証方式の設定

SafeWord サーバおよびカスタムの HTML フォームに対してユーザ (SafeWord ハードウェアトークンを介してログインしているユーザを含む) を認証する必要がある場合に、SafeWord サーバおよびカスタム HTML フォーム認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[SafeWord HTML フォーム テンプレート]を選択します。
認証方式固有のフィールドとコントロールが開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスにサーバ名、SafeWord 設定ファイルの場所、およびサーバおよびターゲットの情報を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

SecurID 認証方式

RSA Ace/SecureID 認証方式では、ACE 証明書を使用してログインするユーザを認証します。ACE 証明書にはユーザ名、PIN、TOKENCODE が含まれています。ACE ユーザ名とパスワードは、ACE/Server ストアに保存され、ACE/Server の管理者以外は変更できません。1 回のみ使用可能な TOKENCODE は、SecureID によって生成されます。

SiteMinder は、以下の SecurID 認証方式に対応します。

SecurID 認証

この認証方式は、RSA SecureID ハードウェアトークンを介してログインするユーザを認証します。この方式では、ユーザ名およびパスワードを受け付けます。パスワードは、ユーザのトークン PIN にダイナミックコードが付いたものです。

注: Ace Server のユーザの userid へのマッピングに「uid」という名前のユーザ ディレクトリ属性を使用しなかった場合、SecurID テンプレート認証方式はユーザを認証できません。uid 以外の LDAP ディレクトリ属性を Ace Server の userid にマッピングするときに、SecurID HTML フォームテンプレートを使用します。

HTML フォーム対応の SecurID 認証

SiteMinder は、HTML フォームに対応の SecurID 認証の方式に対応します。HTML フォームを使用して、ユーザは自分自身の身元を証明できます。また、PIN またはトークン情報を誤って入力したために無効になったアカウントを、再び有効にすることもできます。

RSA ACE/SecurID 方式では、ACE PIN を変更する権限が与えられていないユーザを認証します。ただし、PIN の変更権限を与えられているユーザや PIN を変更する必要があるユーザは、HTML フォームに対応した RSA ACE/SecurID 方式を介して認証されます。いずれの方式も ACE/Server - Ace/Agent モデルに基づいており、RSA/SecurID (トークン)と RSA/ACE (サーバソフトウェア)製品が必要です。

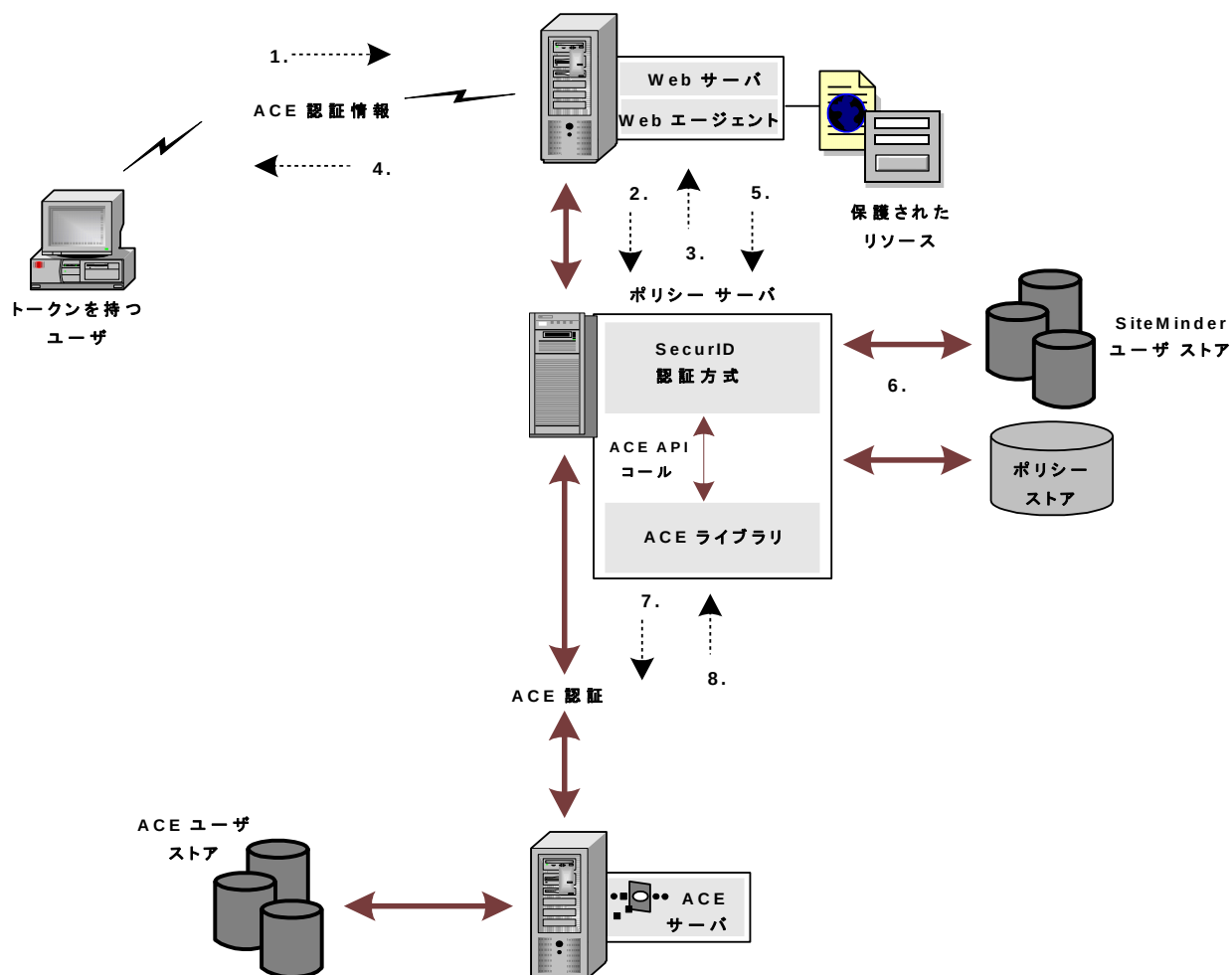
RSA ACE/Server は RSA SecurID トークンと連携し、有効な RSA Ace/Agent を通じてユーザの識別情報を認証します。また、RSA は Web エージェントとカスタムエージェントもサポートしています。SiteMinder SecurID 認証方式は、カスタム Ace/Agents として機能し、ACE/SDK とライブラリを使用します。

ただし、ACE/Server は、独自のポリシーに基づいてユーザの認証情報を処理し、この情報を ACE ユーザストアに格納します。このポリシーには、ACE 管理者によってユーザごとに設定される異なる要件と制限が含まれ、ACE 管理者はいつでもこれらの要件や制限を変更できます。管理者は PIN または TOKENCODE を使用して認証を行うようにユーザを設定します。PIN による認証が設定された場合には、TOKENCODE は不要です。TOKENCODE による認証が設定された場合には、TOKENCODE と PIN の両方が必要になります。また、認証時に新しい PIN の設定を求めるようにすることもできます。この新しい PIN モードでは、以前の PIN と新しい PIN が必要になります。

ユーザを認証するために、SiteMinder SecureID 認証方式では ACE ユーザと SiteMinder ユーザをマッピングする必要があります。このマッピングは、SiteMinder ポリシーストアで認証方式オブジェクト属性として表されます。

5.5 SP1 では、HTML フォームに対応した RSA ACE/SecurID 方式の拡張が新しい PIN モードに影響します。

下図は、RSA ACE/Server がどのように SiteMinder と対話するかを示します。



1. ユーザが Web ページなどのリソースを要求します。
2. SiteMinder Web エージェントがリソースが保護されているかどうかを確認します。
3. ポリシー サーバが、要求されたリソースが HTML フォームに対応した RSA ACE/SecurID 認証方式で保護されていることを通知します。
4. Web エージェントがユーザに認証情報を求めます。
5. ユーザが認証情報を入力すると、エージェントからポリシー サーバに情報が送信されます。
6. ポリシー サーバがユーザの明確化を行い、SiteMinder ユーザ名を RSA/ACE ユーザ名にマッピングします。この時点では、ポリシー サーバは SiteMinder パスワード ポリシーを適用しません。

7. ポリシー サーバは一連の ACE API 呼び出しを通じて認証リクエストを ACE/Server に送信します。このとき、ユーザの認証情報も ACE/Server に送られます。
8. ACE/Server はユーザが PIN を変更する必要があることを知らせ、制御権をポリシー サーバに戻します。
9. ポリシー サーバが制御権を Web エージェントに戻すと、Web エージェントはユーザを CGI または JSP にリダイレクトします。
10. CGI または JSP は適切な HTML フォームを生成し、そのフォームをユーザに表示して、ユーザ名と以前の PIN を入力し、新しい PIN を確認するようにユーザに求めます。
11. Web エージェントは、新しい認証情報をポリシー サーバに送ります。
12. ポリシー サーバは再び、ACE/Server に対して一連の API 呼び出しを行います。
13. 新しい PIN が受け入れられると、ACE API 呼び出しから成功が返されます。ここで、ユーザは新しい PIN を使用してログインするように求められ、認証プロセスは完了します。
14. 新しい PIN が拒否されると、手順 10 から手順 12 が繰り返されます。

次のような場合には、PIN が拒否されます。

- 長すぎる場合
- 短すぎる場合
- 英語の文字が入っている場合

上図に示す認証プロセスでは、HTML フォームにバックエンド PIN ポリシーに関連したメッセージが表示されます。新しい PIN を ACE/Server に送る前に、SecurID 認証方式によって PN ポリシーが確認されます。ACE SDK には、次の PIN 属性を検索できる一連の機能があります。

- 最大長
- 最小長
- 英数字または数字のみ

PIN はこの情報に基づいて確認され、該当するエラーメッセージがユーザに表示されます。確認は次の順序で行われます。

- PIN が最大文字数を超えていない
- PIN が最小文字数を下回らない
- PIN には無効な文字はありません。

5.5 SP1 の場合、これらのエラーメッセージではポリシーの関連する部分のみがユーザに表示されます。次に新しいメッセージの例を示します。

サンプル ユーザ Joe の PIN が 5 ～ 8 文字である場合に、新しい PIN として「poem」と入力すると、以下のようなエラー メッセージが表示されます。

新しい PIN が短すぎます。PIN には最低でも 5 文字が必要です。

次に入力した PIN が「novel」の場合には、以下のようなエラー メッセージが表示されます。

新しい PIN に英数字が含まれています。

次に入力した PIN が「123412341234」の場合には、以下のようなエラー メッセージが表示されます。

新しい PIN が長すぎます。PIN は 8 文字以下である必要があります。

SecureID 認証方式で PIN の確認が完了しても、ACE/Server によって新しい PIN が拒否されることがあります。この場合には、新しい認証情報が求められますが、その理由は表示されません。また、拡張された SecurID 認証方式では、PIN の変更が正常に完了すると、ターゲット Web ページへのアクセスが自動的に許可されます。

SecurID 認証方式の前提条件

SecureID サーバ認証方式を設定するには、以下の前提条件を満たしておく必要があります。

- Windows ポリシー サーバの場合、RSA ACE/クライアントソフトウェアがポリシー サーバと同じマシンにインストールされていること。RSA ACE/クライアントのサポートされるバージョンについては、「[CA Support](#)」サイトの「Platform Support Matrix」を参照してください。

- 以下に該当する場合は、ACE パスが securid ファイルの場所を指しているように設定する必要があります。
 - ACE 環境で ACE クライアント 7.0 以降を使用している。
 - ACE 環境で Node Secret を使用していない。
 - 以下のいずれか 1 つに該当：
 - ACE で、SiteMinder で保護しない別のアプリケーションを保護している。
 - ACE で、SiteMinder 以外の製品を保護している。

ACE パスを設定すると、ポリシー サーバから ACE サーバに送信された認証リクエストが失敗するのを回避できます。

注: ACE サーバにフェイルオーバーするように設定するための SM_ACE_FAILOVER_ATTEMPTS 環境変数は削除されました。

SecurID 認証方式の設定

SecurID 認証方式を使用して、ACE クレデンシャルでログインするユーザを認証します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。

3. [OK]をクリックします。
認証方式の設定が開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[カスタム SecurID テンプレート]を選択します。
認証方式固有のフィールドとコントロールが開きます。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに ACE ユーザ ID 属性を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

HTML フォームに対応した SecurID 認証方式の前提条件

SecureID に HTML 認証方式を設定する前に、以下の必要条件を満たしていることを確認します。

- ポリシー サーバと同じマシンに、RSA ACE/クライアントソフトウェアがインストールされていること。
- Windows のポリシー サーバでは、ACE 設定情報ファイル(sdconf.rec)が winnt\system32 ディレクトリに配置されていること。VAR_ACE 変数および USR_ACE 変数がそれぞれ適切な ACE エージェント データおよび prog ディレクトリを指していること。
- UNIX のポリシー サーバでは、sdconf.rec ファイルが *<policy server installation dir>/bin* ディレクトリに配置されていること。さらに、VAR_ACE 変数および USR_ACE 変数が *<policy server installation dir>/bin* を指しており、ポリシー サーバが ACE エージェントではなく SiteMinder API ライブラリを使用できること。
- カスタマイズした .fcc ファイルが、HTML フォーム認証を実行する cookie ドメイン内の Web エージェントのサーバ上にあること。サンプルの .fcc ファイルは、Web エージェントをインストールした Samples/Forms サブディレクトリの下にあります。
- カスタマイズした.unauth ファイルが、Web エージェントのサーバ上にあること。

注: .fcc ファイルが smerrorpage 命令を使用する場合は、.unauth ファイルは不要です。

- ポリシー サーバとユーザ ディレクトリの間でディレクトリ接続が確立されていること。
 - デフォルトの HTML フォーム ライブラリがインストールされていること。このライブラリが HTML フォーム 認証を処理します。
 - Windows では SmAuthHTML.dll。
 - Solaris では smauthhtml.so。
- これらのファイルは Web エージェントの設定時に自動的にインストールされます。

SecurID 認証方式と HTML フォーム 認証方式の設定

SecurID と HTML フォームの認証方式を使用して、ACE クレデンシャルでログインするユーザを認証するカスタム HTML フォームを使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。
認証方式の設定が開きます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [認証タイプのスタイル]リストから[SecurID HTML フォーム テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスでサーバ、ターゲット、および ACE 属性情報を入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

SecurID ユーザの再アクティブ化および確認に使用するフォームのサポート

SecurID と HTML フォームの認証方式によってレلمを保護する場合、ログインが不適切なために一時停止されたユーザは、**SiteMinder** に用意されているカスタマイズ可能な HTML フォームをいくつか使用して、アカウントのアクティブ化を試みることができます。これらのフォームのレイアウトや用語を修正することはできますが、ユーザ情報を収集するタグを修正することはできません。

SiteMinder に用意されているフォームを、以下に示します。

PWLogin.template

このフォームには、ユーザがログインに使用するユーザ名とパスワードを入力します。

PWNNextToken.template

このテンプレートでは、ユーザが有効な SecurID トークンを所有していることを確認するために、複数のトークンコードが要求されます。

新規ユーザアカウントをアクティブにするためのフォーム

以下のフォームは、管理者が新しいユーザ アカウントを作成し、そのユーザがログインするときに、**SiteMinder** で使用されます。フォームを使用して、ユーザは PIN を作成するか、または **SiteMinder** にランダムな PIN を生成させます。

PWSystemPIN.template

新規ユーザや、無効なログインを何度も繰り返したためにアカウントを一時停止されたユーザに対し、このテンプレートはユーザに新しい PIN の取得を求めるプロンプトを表示します。このテンプレートでは、ユーザの元のユーザ名とパスワードを使用できますが、保護されたリソースへのアクセス権が付与されるのではなく、ユーザは別のフォームにリダイレクトされます。そのフォームで、ユーザは新しい PIN を受け入れるか、作成することができます。

PWNewPINSelect.template

このテンプレートでは、システムで新しい PIN を生成するか、ユーザ自身が新しい PIN を入力するかを、ユーザが指定できます。

PWUserPIN.template

このテンプレートでは、ユーザが新しい PIN を入力できます。そのほかに、有効なユーザ名とパスワードも入力する必要があります。このテンプレートの \$USRMSG\$ は、新しい PIN 番号の作成手順に置き換わります。以下に例を示します。

PIN の長さは 4 ～ 8 文字で指定してください。

PWPINAccept.template

このテンプレートは、システム生成の新しい PIN が作成されたことを示します。このテンプレートの \$USRMSG\$ は、システムが生成した PIN に置き換わります。

[続行]をクリックすると、新しい PIN を使用したログイン画面が表示されます。

X.509 クライアント証明書認証方式

X.509 クライアント証明書は、ユーザの身元を示す暗号化された証明を提供します。証明書ベンダーから提供されるユーザ証明書は固有のもので、保護されたリソースにアクセスしようとするユーザの識別に使用できます。

クライアント証明書には以下の情報が含まれます。

- 対象名
- 発行者名:
- シリアル番号
- バージョン
- 検証期間
- 署名 (アルゴリズム ID とパラメータ)
- 対象の公開鍵 (および関連するアルゴリズム ID)
- X.509 v3 拡張 (任意)

SiteMinder は、X.509 クライアント証明書の認証方式を使用して、証明書認証を実行します。X.509 クライアント証明書認証を使用するには、お使いの環境が SSL 通信に対応している必要があります。つまり、クライアントブラウザ、Web サーバ、およびすべてのユーザ証明書で証明書認証を受諾および実行できるように設定されている必要があります。これらの設定は、SiteMinder 環境設定のスコープ外で行われます。

必要な SSL コンポーネントが正しくセットアップされたら、**SiteMinder X.509** 認証方式を設定できます。**SiteMinder** 設定タスクでは、以下の手順を実行する必要があります。

- **SiteMinder Web** エージェント設定ウィザードの実行時に、SSL 認証方式の **Advanced** を選択します。

Web エージェント設定ウィザードの詳細については、「**SiteMinder Web** エージェント インストール ガイド」を参照してください。

- 本ガイドの説明に従って、管理 UI を使用して **X.509** 認証方式の 1 つを設定します。

SiteMinder X.509 クライアント証明書認証方式は以下タスクを実行します。

- クライアント証明書情報を収集します。
- ユーザ証明書から情報に基づいてディレクトリ内のユーザを特定します。
SiteMinder では、このプロセスは「証明書マッピング」と呼ばれます。
- (任意) 証明書破棄リスト (CRL) またはオンライン証明書ステータス プロトコル (OCSP) を使用して、証明書が有効かどうかを確認します。

詳細情報:

[X.509 クライアント認証方式の証明書マッピング](#) (P. 405)

証明書認証の証明書の抽出

SiteMinder で保護されたリソースをユーザが要求した場合、**Web** エージェントはまずポリシー サーバにアクセスし、リソースを保護している認証方式を確認します。**X.509** 認証方式がリソースを保護している場合、**Web** エージェントは、設定された認証方式に対応する **SiteMinder** クレデンシャル コレクタへユーザのブラウザをリダイレクトします。クレデンシャル コレクタへのパスは認証方式の設定に定義されています。

クレデンシャル コレクタへの接続は SSL 対応の接続で、**Web** サーバはクライアント証明書が必要とされるよう設定されています。そのため、ブラウザは、認証用のクライアント証明書をサブミットする必要があります。クレデンシャル コレクタ URL の末尾のリソース名と拡張子によって、**Web** サーバからユーザ証明書を抽出するように **Web** エージェントに指示されます。**Web** エージェントは、認証方式で使用するための証明書をポリシー サーバに渡します。

詳細情報:

[SSL を介した認証](#) (P. 317)

SiteMinder でユーザの識別に証明書データを使用する方法

Web エージェントが証明書情報を収集したら、確認のためポリシー サーバへ渡されます。ポリシー サーバでは証明書マッピングを実行します。証明書マッピングの目的は、ユーザ証明書内の対象名に基づいて **SiteMinder** ユーザを特定することです。

まず、ポリシー サーバでは、証明書の発行者 **DN** を使用して、ポリシー ストア内の適切な証明書マッピングを調べます。発行者 **DN** は証明書マッピング設定の一部です。ポリシー サーバでマッピングを検出したら、証明書の対象名に基づいてマッピングを適用し、ユーザ ディレクトリ内のユーザ エントリを検索します。

ポリシー サーバは、以下のリポジトリに格納されたユーザ証明書にアクセスできます。

- LDAP/AD ユーザーディレクトリ
- ODBC ストア

重要: X.509 クライアント証明書認証方式の場合は常に証明書マッピングを設定する必要があります。

詳細情報:

[X.509 クライアント認証方式の証明書マッピング](#) (P. 405)

X.509 クライアント証明書認証方式の前提条件

X.509 クライアント証明書認証方式を設定するには、以下の前提条件を満たす必要があります。

- X.509 サーバ証明書を **SSL Web** サーバ上にインストールします。証明書が期限切れになっていないことを確認してください。

注: ポリシー サーバが **FIPS** モードで動作している場合は、証明書が **FIPS** 承認アルゴリズムのみを使用して生成されていることを確認してください。

- ネットワークでクライアントブラウザへの **SSL 接続 (HTTPS プロトコル)** がサポートされていることを確認します。

- X.509 クライアント証明書がクライアントブラウザにインストールされていることを確認します。証明書が期限切れになっていないことを確認してください。

X.509 証明書認証方式の設定

証明書認証を設定するには、SSL 環境のセットアップに加えて、以下の手順を実行します。

1. お使いの環境が SSL 通信に対応するよう設定します。クライアント ブラウザ、Web サーバ、およびすべてのユーザ証明書で証明書認証を受諾および実行できるよう設定します。
2. SiteMinder Web エージェントのインストール時に、SSL 認証を処理できるよう設定します。
3. 管理 UI で SiteMinder X.509 認証方式を設定します。
4. 証明書マッピングを定義して、クライアント証明書内の情報に基づいてディレクトリのユーザを識別できるようにします。
5. (任意) CRL または OCSP を使用して証明書の検証を設定します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。

3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[X.509 クライアント証明書テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスに、SSL クレデンシヤル コレクタのサーバ名とターゲットの情報を入力します。

7. [サブミット]をクリックします。

認証方式が保存され、レルムに割り当て可能になります。

X.509 証明書認証方式が 管理 UI に設定されました。次に、[証明書マッピング](#) (P. 405)を設定します。

X.509 クライアント証明書および基本認証方式

X.509 クライアント証明書および基本認証方式は、基本認証方式と X.509 クライアント証明書認証方式を組み合わせたものです。この認証方式は、重要なリソースのセキュリティを追加のレイヤを提供します。

ユーザが認証されるには、以下の 2 つのイベントが発生する必要があります。

- ユーザの X.509 クライアント証明書が確認されていること。
および
- ユーザが有効なユーザ名とパスワードを指定していること。

SiteMinder は以下の手順で X.509 クライアント証明書の認証を行います。

1. ユーザを SSL サーバにリダイレクトし、ユーザの証明書をサーバにマップするように、ポリシー サーバから SiteMinder Web エージェントに指示が出されます。
2. SiteMinder はユーザの存在を確認します。
3. SiteMinder はユーザの基本認証情報(ユーザ名とパスワード)を確認します。
4. SiteMinder は証明書の認証情報と基本認証情報が同じユーザを示していることを確認します。

X.509 クライアント証明書および基本認証方式の前提条件

X.509 クライアント証明書および基本の認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- X.509 サーバ証明書が **SSL Web** サーバ上にインストールされていること。
注: ポリシー サーバが **FIPS** モードで動作している場合は、証明書が **FIPS** 承認アルゴリズムのみを使用して生成されていることを確認してください。
- ネットワークがクライアント ブラウザへの **SSL** 接続をサポートしていること (**HTTPS** プロトコル)。
- X.509 クライアント証明書がクライアント ブラウザにインストールされていること。
- クライアント証明書とサーバ証明書の間で信頼関係が確立されていること。
- 証明書が、有効で信頼できる認証機関 (**CA**) から発行されていること。
- 発行する **CA** の公開キーが発行者のデジタル署名を検証すること。
- クライアント証明書とサーバ証明書の有効期限が切れていないこと。
- ユーザの公開キーがユーザのデジタル署名を検証すること。
- クライアントのユーザ名とパスワード情報がユーザ ディレクトリにあること。
- ポリシー サーバとユーザ ディレクトリの間ディレクトリ接続が確立されていること。

注: Apache Web サーバで証明書が必須またはオプションに設定されている場合は、`httpd.conf` ファイルで **SSL Verify Depth 10** の行をコメント解除する必要があります。

X.509 証明書および基本認証方式の設定

証明書認証と基本認証を組み合わせるには、**X.509** 証明書および基本認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [認証タイプのスタイル]リストから[X.509 クライアント証明書および基本テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに、SSL クレデンシヤル コレクタのサーバ名とターゲットの情報を入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

X.509 クライアント証明書または基本認証方式

X.509 クライアント証明書または基本認証方式では、基本認証方式または X.509 クライアント証明書認証方式のいずれかを使用してユーザを識別できます。ユーザが認証されるためには、以下のいずれかのイベントが発生する必要があります。

- ユーザの X.509 クライアント証明書が確認されていること。
または
- ユーザが有効なユーザ名とパスワードを指定していること。

この方式では、ユーザが保護されたリソースをリクエストすると、**Web** エージェントがユーザのブラウザに証明書の提示を要求します。ユーザが証明書を保有していないか、証明書を提示しないよう選択([キャンセル]をクリック)した場合、**Web** エージェントは **HTTP** プロトコルでユーザにユーザ名とパスワードを要求します。

この方式は **X.509** 証明書を段階的に展開していく必要がある場合に有用です。たとえば、**50,000** 人のユーザを抱える企業においては、**50,000** 人分の証明書を同時に発行して展開することは極めて困難です。ところが、この認証方式を利用すれば適切と思われる数(1 度に **500** または **5,000**)の証明書を発行していくことができます。証明書を展開していく過渡期においては、すでに証明書を保有しているユーザに対しては証明書を使ってリソースを保護し、その他の権限を持ったユーザに対してはディレクトリのユーザ名とパスワードに基づいてリソースへのアクセスを許可するということができます。

この方式には、**SSL** 接続の要求に基本認証を設定するオプションもあります。

注: **X.509** 証明書や基本認証など、複数の証明書ベースの認証方式を実装すると、ブラウザのキャッシュがいっぱいになって予期しない動作が起こる可能性があります。ユーザが証明書または基本認証方式で保護されているレルムにあるリソースへのアクセスに証明書ベースの認証を使用しないと選択した場合、ブラウザ(**IE** と **Netscape**)は自動的に証明書を送信しないという選択内容をキャッシュに保存します。このユーザが同じブラウザ セッションの中で、**X.509** 証明書、**X.509** 証明書および基本、または **X.509** 証明書およびフォームなどの必須証明書が指定された認証方式で保護されているリソースにアクセスしようとすると、「**Forbidden**」エラーメッセージが表示されます。

この場合、最初のリソースへのアクセス時にユーザが証明書ベースの認証の証明書を送信しないと選択し、ブラウザはその選択内容をキャッシュに保存しているため、証明書が必要なレルムにアクセスすると、ユーザのアクセスは自動的に拒否されます。

X.509 証明書または基本認証方式などの証明書ベースの認証方式で保護されているレルムと、認証を受けるために証明書を送信することが必須とされる証明書ベースの方式で保護されているレルムが混在する環境では、有効な証明書を保有しているユーザには、リソースのアクセス時にできるだけ証明書を使用するように奨励してください。

詳細情報:

[基本認証方式](#) (P. 314)

[X.509 クライアント証明書認証方式](#) (P. 315)

X.509 クライアント証明書または基本認証方式の前提条件

X.509 クライアント証明書または基本の認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- X.509 サーバ証明書が **SSL Web** サーバ上にインストールされていること。
注: ポリシー サーバが **FIPS** モードで動作している場合は、証明書が **FIPS** 承認アルゴリズムのみを使用して生成されていることを確認してください。
- ネットワークが、**HTTPS** プロトコルを使用したクライアント ブラウザへの **SSL** 接続に対応していること。
- X.509 クライアント証明書がクライアント ブラウザにインストールされていること。
- クライアント証明書とサーバ証明書の間で信頼関係が確立されていること。
- 証明書が、有効で信頼できる認証機関 (**CA**) から発行されていること。
- 発行する **CA** の公開キーが発行者のデジタル署名を検証すること。
- クライアント証明書とサーバ証明書の有効期限が切れていないこと。
- ユーザの公開キーがユーザのデジタル署名を検証すること。
- クライアントのユーザ名とパスワード情報がユーザ ディレクトリにあること。
- ポリシー サーバとユーザ ディレクトリの間でディレクトリ接続が確立されていること。

注: Apache Web サーバで証明書が必須またはオプションに設定されている場合は、`httpd.conf` ファイルで **SSL Verify Depth 10** の行をコメント解除する必要があります。

X.509 証明書または基本認証方式の設定

証明書認証または基本認証、あるいはその両方の認証を実装するには、X.509 証明書または基本認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。

3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[X.509 クライアント証明書または基本テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに、SSL クレデンシヤル コレクタのサーバおよびターゲットの情報を入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

X.509 クライアント証明書および HTML フォーム 認証方式

X.509 クライアント証明書およびフォーム認証方式は、フォーム認証方式と X.509 クライアント証明書認証方式を組み合わせたものです。この認証方式は、重要なリソースのセキュリティを追加のレイヤを提供します。ユーザが認証されるには、以下の 2 つのイベントが発生する必要があります。

- ユーザの X.509 クライアント証明書が確認されていること。
および
- ユーザが HTML フォームで要求される証明書を指定していること。

SiteMinder は以下の手順で X.509 クライアント証明書の認証を行います。

1. ユーザを SSL 対応の Web サーバの FCC にリダイレクトするように、ポリシーサーバから SiteMinder Web エージェントに指示が出されます。
2. Web エージェントにより、フォームが表示されます。
3. FCC が証明書とフォームをポリシーサーバに戻します。
4. ポリシーサーバによって、ユーザが証明書マッピングに存在することが確認されます。
5. ポリシーサーバによって、ユーザの HTML フォームの認証情報が確認されます。
6. SiteMinder は証明書の認証情報と HTML フォーム認証情報が同じユーザを示していることを確認します。

X.509 クライアント証明書および HTML フォームの認証方式の前提条件

X.509 クライアント証明書および HTML フォームの認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- X.509 サーバ証明書が SSL Web サーバ上にインストールされていること。
注: ポリシーサーバが FIPS モードで動作している場合は、証明書が FIPS 承認アルゴリズムのみを使用して生成されていることを確認してください。
- ネットワークがクライアントブラウザへの SSL 接続をサポートしていること (HTTPS プロトコル)。
- X.509 クライアント証明書がクライアントブラウザにインストールされていること。
- クライアント証明書とサーバ証明書の間で信頼関係が確立されていること。
- 証明書が、有効で信頼できる認証機関 (CA) から発行されていること。
- 発行する CA の公開キーが発行者のデジタル署名を検証すること。
- クライアント証明書とサーバ証明書の有効期限が切れていないこと。
- ユーザの公開キーがユーザのデジタル署名を検証すること。
- フォーム クレデンシャルがユーザ ディレクトリにあること。

- ポリシー サーバとユーザ ディレクトリの間でディレクトリ接続が確立されていること。
- (Sun Java Systems) Sun Java Systems の Web サーバを使用している場合は、`magnus.conf` ファイル内の `StackSize` パラメータの値を 131072 より大きい値に増加してください。値を変更しないと、Web サーバはそのコアをダンプし、SiteMinder がフォームを使用して認証リクエストを行うたびに再起動します。

注: Apache Web サーバで証明書が必須またはオプションに設定されている場合は、`httpd.conf` ファイルで `SSL Verify Depth 10` の行をコメント解除する必要があります。

収集された証明書とフォームのデータは一緒にポリシー サーバに渡されます。

状況	結果
証明書がない場合	SiteMinder がエラー 500 を発行した場合
証明書とフォームのクレデンシャルが拒否された場合	SiteMinder がエラー 500 を発行した場合

エージェント API のサポート

X.509 Client Certificate および HTML の Forms は `Sm_AuthApi_Cred_SSLRequired` および `Sm_AuthApi_Cred_FormRequired` ビットを使用します。

X.509 証明書および HTML フォーム認証方式の設定

証明書認証と HTML フォーム ベースの認証を組み合わせるには、X.509 証明書および HTML 認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。
認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [認証タイプのスタイル]リストから[X.509 クライアント証明書およびフォーム テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに、SSL クレデンシヤル コレクタのサーバ名とターゲットの情報を入力します。
7. [サブミット]をクリックします。
認証方式は保存され、場合によってはレルムに割り当てられます。

X.509 クライアント証明書または HTML フォーム認証方式

X.509 クライアント証明書またはフォーム認証方式では、フォーム認証方式または X.509 クライアント証明書認証方式のいずれかを使用してユーザを識別できます。ユーザが認証されるためには、以下のいずれかのイベントが発生する必要があります。

- ユーザの X.509 クライアント証明書が確認されていること。
または
- ユーザが HTML フォームで要求される証明書を指定していること。

この方式では、ユーザが保護されたリソースをリクエストすると、Web エージェントがユーザのブラウザに証明書の提示を要求します。この後、以下のような処理が行われます。

状況	結果
証明書が提示された場合	SiteMinder によって証明書が処理されます。
証明書が拒否された場合	SiteMinder がエラー 500 を発行した場合
証明書の提示がない場合	SiteMinder によってフォームが表示されます。
フォームが拒否された場合	SiteMinder によってフォームへの入力が再度求められます。

この方式は X.509 証明書を段階的に展開していく必要がある場合に有用です。たとえば、50,000 人のユーザを抱える企業においては、50,000 人分の証明書を同時に発行して展開することは極めて困難です。ところが、この認証方式を利用すれば適切と思われる数 (1 度に 500 または 5,000) の証明書を発行していくことができます。証明書を展開していく過渡期においては、すでに証明書を保有しているユーザに対しては証明書を使ってリソースを保護し、その他の権限を持ったユーザに対しては HTML フォームの認証情報に基づいてリソースへのアクセスを許可するということができます。

注: X509 証明書やフォーム認証など、複数の証明書ベースの認証方式を実装すると、ブラウザのキャッシュがいっぱいになって予期しない動作が起こる可能性があります。ユーザが証明書または基本認証方式で保護されているレルムにあるリソースへのアクセスに証明書ベースの認証を使用しないと選択した場合、ブラウザ (IE と Netscape) は自動的に証明書を送信しないという選択内容をキャッシュに保存します。このユーザが同じブラウザ セッションの中で、X509 証明書、X509 証明書および基本、または X509 証明書およびフォームなどの必須証明書が指定された認証方式で保護されているリソースにアクセスしようとすると、「Forbidden」エラーメッセージが表示されます。

この場合、最初のリソースへのアクセス時にユーザが証明書ベースの認証の証明書を送信しないと選択し、ブラウザはその選択内容をキャッシュに保存しているため、証明書が必要なレルムにアクセスすると、ユーザのアクセスは自動的に拒否されます。

X.509 証明書またはフォーム認証方式などの証明書ベースの認証方式で保護されているレルムと、認証を受けるために証明書を送信することが必須とされる証明書ベースの認証方式で保護されているレルムが混在する環境では、有効な証明書を保有しているユーザには、リソースのアクセス時にできるだけ証明書を使用するように奨励してください。

X.509 クライアント証明書または HTML フォームの認証方式の前提条件

X.509 クライアント証明書または HTML フォームの認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- X.509 サーバ証明書が SSL Web サーバ上にインストールされていること。
注: ポリシー サーバが FIPS モードで動作している場合は、証明書が FIPS 承認アルゴリズムのみを使用して生成されていることを確認してください。
- ネットワークがクライアントブラウザへの SSL 接続をサポートしていること (HTTPS プロトコル)。
- X.509 クライアント証明書がクライアントブラウザにインストールされていること。
- クライアント証明書とサーバ証明書の間で信頼関係が確立されていること。
- 証明書が、有効で信頼できる認証機関 (CA) から発行されていること。
- 発行する CA の公開キーが発行者のデジタル署名を検証すること。
- クライアント証明書とサーバ証明書の有効期限が切れていないこと。
- ユーザの公開キーがユーザのデジタル署名を検証すること。
- HTML フォームで要求されたユーザ属性がユーザ ディレクトリにあること。
- ポリシー サーバとユーザ ディレクトリの間でディレクトリ接続が確立されていること。
- (Sun Java Systems) Sun Java Systems の Web サーバを使用している場合は、magnus.conf ファイル内の StackSize パラメータの値を 131072 より大きい値に増加してください。値を変更しないと、Web サーバはそのコアをダンプし、SiteMinder がフォームを使用して認証リクエストを行うたびに再起動します。

エージェント API のサポート

エージェント API では、列挙型 `Sm_Api_Credentials_t` に `Sm_Api_Cred_CertOrForm` という値が追加されました。
`Sm_Api_Credentials_t` は、構造体 `Sm_AgentApi_Realm_t` で参照されるレルムにユーザがアクセスするために必要な証明書(存在する場合)を指定します。この列挙型は、構造体の `nRealmCredentials` フィールドに適用されます。

この新しい値は、ユーザ認証に X.509 証明書またはフォームベースの認証方式が必要であることを指定します。

X.509 証明書または HTML フォーム認証方式の設定

証明書認証または HTML フォーム ベースの認証、あるいはその両方の認証を実装するには、X.509 証明書または HTML フォーム認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[X.509 クライアント証明書またはフォーム テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスに、サーバおよびターゲットの情報を
を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

注: Apache Web サーバで証明書が必須またはオプションに設定されている場合は、`httpd.conf` ファイルで `SSL Verify Depth 10` の行をコメント解除してください。

匿名認証方式

匿名認証方式を使用すると、ネットワークで認証されていないユーザに対して SiteMinder からアクセス権限を付与できます。匿名認証方式をレルムに割り当ててもアクセスを制御できませんが、SiteMinder を使ってユーザに対するコンテンツをパーソナライズできます。

匿名認証方式を適用するレルム内のリソースにユーザがアクセスすると、ポリシー サーバは GUID (グローバル固有識別子) を割り当てます。この GUID はユーザのブラウザに保存され、匿名ユーザを識別する手段となります。

匿名認証方式を設定する場合、ゲストの DN (識別名) を指定する必要があります。このゲスト DN に対して、ポリシーをバインドしてパーソナライズしたコンテンツを提供できます。

注: 匿名認証方式で保護されたレルム内のパーソナライズされたコンテンツは、ユーザの GUID ではなく、ゲスト DN に基づきます。匿名ユーザに対しては、ゲスト DN を含むポリシーに対応したコンテンツが表示されます。一方、識別されたユーザは異なる DN を保有しているため、匿名認証方式で保護された同じリソースにアクセスした場合、ゲスト DN ではなく固有の DN に基づいたリソースのコンテンツが表示されます。

匿名認証方式の前提条件

匿名認証方式を設定する前に、以下の前提条件を満たしていることを確認します。

- ユーザ ディレクトリに、匿名ユーザのためのゲスト DN があること。
- ポリシー サーバとユーザ ディレクトリ間にディレクトリ接続が確立されていること。
- 匿名認証で割り当てた GUID を基にユーザを追跡する場合は、[SiteMinder グローバル設定]ウィンドウでユーザ追跡を有効にする必要があります。

注: ユーザトラッキングの有効化についての詳細は、「ポリシー サーバ管理 ガイド」を参照してください。

匿名認証方式の設定

登録されていないユーザに特定の Web コンテンツへのアクセス権を付与するには、匿名認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[匿名テンプレート]を選択します。
認証方式固有の設定が開きます。
5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. [方式のセットアップ]グループ ボックスに、ユーザの DN を入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

カスタム認証方式

SiteMinder に用意されていない認証方式を使用する場合は、カスタム認証方式を作成できます。カスタム認証方式を作成したら、その方式を SiteMinder [認証]ウィンドウで設定する必要があります。

注: CA シングル サインオンから SiteMinder のシングル サインオンを有効にするのに必要な smauthetsso カスタム認証方式の設定についての詳細は、「[CA SSO/WAC Integration](#) (P. 793)」を参照してください。

注: ソフトウェア開発キットをインストールしてある場合は、「*API Reference Guide for C*」でカスタム認証方式の作成についての詳細を参照してください。

カスタム認証方式の前提条件

CA の API を使用してカスタム認証方式を作成する場合は、その方式の前提条件が判別されます。前提条件は認証方式によって異なります。

カスタム認証方式の設定

SiteMinder が提供しない方式を使用する必要がある場合に、カスタムの認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
 2. [認証方式]-[認証方式の作成]をクリックします。
- [認証方式の作成]ペインが開きます。

3. [OK]をクリックします。

認証方式の設定が開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. 認証タイプのスタイルリストからカスタム テンプレートを選択します。

認証方式固有の設定が開きます。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。

6. 認証方式のクレデンシャルを処理するライブラリおよびライブラリに渡すパラメータを、[方式のセットアップ]グループ ボックスに入力します。

7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

Federation セキュリティ サービス認証方式

Federation セキュリティ サービスを購入およびインストール済みの場合は、既存の SiteMinder 方式に以下の認証方式が追加されます。

- SAML アーチファクト テンプレート
- SAML POST テンプレート
- SAML 2.0 テンプレート
- WS-Federation テンプレート

注:これらの認証方式についての詳細は、「*Federation Security Services Guide*」を参照してください。

SOA Security Manager 認証方式

SOA Security Manager を購入およびインストール済みの場合は、既存の SiteMinder 方式に以下の認証方式が追加されます。

- XMLドキュメント認証情報コレクタ (XML DCC)
- XML デジタル署名 (XML-DSIG)

- SAML セッションチケット
- WS-Security

これらの認証方式の詳細については、「*CA SOA Security Manager Policy Configuration Guide*」を参照してください。

偽装認証方式

一連のポリシー サーバ オブジェクトを設定することで、権限を保有するユーザが他のユーザを装う(偽装する)ことができます。この機能は、ヘルプデスクやカスタマーサービスのスタッフが顧客に代わって問題のトラブルシューティングを行う場合や、社員が外出している場合などに便利です。偽装プロセスの一部では、権限を保有するユーザが偽装プロセスを開始して、偽装対象のユーザを識別し、偽装セッションを確立することを可能にする偽装認証方式が必要になります。認証方式としては、HTML フォーム認証方式に近いものです。

インパーソネーション認証方式の前提条件

インパーソネーション認証方式を設定するには、事前に以下の前提条件を満たしておく必要があります。

- カスタマイズした .fcc ファイルが、インパーソネーションを実行する cookie ドメイン内の Web エージェントのサーバ上にあること。サンプルの .fcc ファイルは、Web エージェントをインストールした /forms サブディレクトリの下にあります。

.fcc ファイルの作成方法の詳細については、「[SiteMinder FCC Files \(P. 330\)](#)」を参照してください。インパーソネーションの具体的な .fcc ファイル要件については、「[.fcc ファイルによるインパーソネーションの有効化 \(P. 705\)](#)」を参照してください。

- ポリシー サーバと、偽装者ユーザと偽装対象ユーザが格納されているユーザ ディレクトリの間に、ディレクトリ接続が確立されていること。

- デフォルトの HTML フォーム ライブラリがインストールされていること。このライブラリが、認証を処理します。
 - Windows では `smauthimpersonate.dll`。
 - Solaris では `smauthimpersonate.so`。

これらのファイルは Web エージェントの設定時に自動的にインストールされます。

注: ディレクトリ マッピングでは、インパーソネーションはサポートされていません。インパーソネーションの対象ユーザは、ドメインに関連付けられている認証ディレクトリ内で一意である必要があります。そうでない場合、インパーソネーションは失敗します。

注: パスワード ポリシーが、インパーソネーションの対象ユーザに適用され、有効になっていると、インパーソネーション認証方式は失敗します。

インパーソネーション認証方式の設定

権限を持つユーザに他のユーザの代理をさせるには、インパーソネーション認証方式を使用します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成]ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル]リストから[インパーソネーション テンプレート]を選択します。

認証方式固有のフィールドとコントロールが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

5. [一般]グループ ボックスに、名前と保護レベルを入力します。
6. [方式のセットアップ]グループ ボックスに、サーバ名とターゲットの情報を☐入力します。
7. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

第 10 章: X.509 証明書の証明書マッピングおよび有効性チェック

このセクションには、以下のトピックが含まれています。

[X.509 クライアント認証方式の証明書マッピング](#) (P. 405)

[証明書の有効性チェック\(任意\)](#) (P. 413)

X.509 クライアント認証方式の証明書マッピング

ポリシー サーバでユーザの識別に証明書を使用するためには、証明書情報をユーザ ディレクトリのユーザ レコードと比較する必要があります。証明書マッピングは、ポリシー サーバでユーザ証明書の「対象名」を使用してユーザ ディレクトリ内の SiteMinder ユーザを特定し、そのユーザを認証する方法を定義します。

Microsoft SQL Server、Oracle、または LDAP ユーザ ディレクトリに認証情報が保存されているユーザに対して、証明書マッピングを設定できます。

詳細情報:

[ユーザ ディレクトリ](#) (P. 165)

証明書マッピングの設定

証明書マッピングを設定することにより、SiteMinder でユーザ証明書情報と、ユーザ ディレクトリに保存された情報を照合できるようになります。

証明書マッピングを設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [証明書マッピング]-[証明書マッピングの作成]をクリックします。

[証明書マッピングの作成] ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [発行者 DN]フィールドに証明書発行者 DNを入力します。発行者 DN は、証明書に表示される通りに正確に入力します。スペースや文字は一切追加しないでください。

DNを入力する際、予約された特殊文字は円記号(\)でエスケープします。特殊文字には以下のものが含まれます。

- セミコロン(;)
- 引用符(")
- バックスラッシュ(\)
- プラス(+)
- より大きい(>)
- より小さい(<)

DN用に予約された特殊文字の詳細については、
<http://www.faqs.org/rfcs/rfc2253.html> を参照してください。

注: リレーショナル データベースがポリシー ストアとして使用される場合、発行者 DN は 255 文字を超えることはできません。LDAP ディレクトリ サーバがポリシー ストアとして使用される場合は、発行者 DN は 1000 文字を超えることはできません。

4. 証明書がマップされるディレクトリ タイプを選択します。

LDAP ディレクトリの場合のみ、ポリシー サーバで、ユーザが提供した証明書がユーザ ディレクトリ内のユーザ レコードに格納されている証明書と一致するかどうかを確認されるよう設定することができます。[ディレクトリの証明書を要求する]チェック ボックスを使用すると、この確認が必須になります。

5. [マッピング]グループ ボックスで、X.509 ユーザ証明書情報を、ユーザ ディレクトリ内のユーザ エントリにマップする方法を指定します。ポリシー サーバは、単一の属性、カスタム マッピング式、またはユーザ証明書の対象名全体を使用してマッピングを適用し、正しいユーザ エントリを特定します。
6. (オプション)[証明書破棄リスト (CRL) のチェック]グループ ボックスで[CRL チェックの実行]を選択し、グループ ボックスに CRL 設定を指定します。

CRL を選択しない場合、OCSP を使用することもできます。

7. [サブミット]をクリックします。

証明書マッピングの作成タスクを処理できるよう送信します。

詳細情報:

[証明書の有効性チェック\(任意\)](#) (P. 413)

証明書マッピングのテスト

証明書マッピングをテストすると、ポリシー サーバがクライアント証明書をユーザ ディレクトリ属性にマップするときに使用する検索文字列が表示されます。

証明書マッピングをテストする方法

1. 証明書マッピングを開きます。
2. [マッピング]グループ ボックスで[テスト]をクリックします。
[証明書マップ テスト]グループ ボックスが表示されます。
3. [ディレクトリ]リストからユーザ ディレクトリ接続を選択します。

注: [ディレクトリ]リストには、証明書マッピング作成時に選択したタイプの既存のディレクトリ接続がすべて含まれています。

[ディレクトリ情報]グループ ボックスの内容は、ユーザ ディレクトリ接続のタイプによって異なります。Windows NT、ODBC、および OCI のユーザ ディレクトリ接続では、テスト対象となるディレクトリタイプがグループ ボックスに表示されます。LDAP ディレクトリ接続の場合、グループ ボックスには[ディレクトリタイプ]のほかに、[ユーザ DN 検索の先頭文字列]と[ユーザ DN 検索の終端文字列]が表示されます。この 2 つの値は、LDAP ディレクトリ内でユーザの DN を検索するときに使用されます。

ポリシー サーバは証明書マッピングをテストし、[証明書マップ テスト]グループ ボックスは結果を表示します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [閉じる]をクリックします。
[証明書マップテスト]ダイアログ ボックスが閉じます。

カスタム マッピング式

複数の属性の複雑なマッピングに対して、カスタムのマッピング式を使用できます。これにより、ユーザ DN から抽出される複数のユーザ属性を指定して、証明書マッピングを確立することができます。

注: SiteMinder テストツールを通じて証明書ベースの認証をシミュレートする場合にも、カスタム マッピング式が有効です。

カスタム マッピング式の構文は、マッピングの柔軟性を十分に高めるために設計された仕様になっています。この式は、証明書から取得する情報を識別し、その情報の適用先となるユーザ ディレクトリ内の場所を示します。基本的な構文は次のとおりです。

```
UserAttribute=%{CertificateAttribute},  
UserAttribute2=%{CertificateAttribute}
```

EnableCustomExprOnly レジストリ キー

LDAP ユーザ ディレクトリにカスタムの証明書マッピングを作成すると、その結果として作成される検索クエリ文字列には、[証明書マッピングの作成]ペインで指定したマッピング式に加えて、LDAP ユーザ DN 検索の先頭文字列およびユーザ DN 検索の終端文字列が含まれます。結果として作成されるクエリは、以下の例に示すように無効です。

LDAP ユーザ DN 検索の先頭文字列

```
(samAccountName=
```

LDAP ユーザ DN 検索の終端文字列

```
)
```

証明書マッピング式

```
(mail=%{E})
```

結果として作成される検索クエリ

```
(samAccountName=(mail=%{E}))
```

検索クエリからユーザ DN 検索の先頭文字列およびユーザ DN 検索の終端文字列を省略するには、\Netegrity\SiteMinder\CurrentVersion\PolicyServer\に移動して EnableCustomExprOnly レジストリ キーを 1 に設定します。以下の例に示す、結果として作成されるクエリ文字列が有効になります。

証明書マッピング式

```
mail=%{E}
```

結果として作成される検索クエリ

mail=%{E}

注: EnableCustomExprOnly レジストリ キーが 0 (デフォルト)、またはキーが存在しない場合、結果として作成される検索クエリにはユーザ DN 検索の先頭文字列とユーザ DN 検索の終端文字列が含まれます。

LegacyCertMapping レジストリ キーの有効化

LDAP 構文を使用して、論理演算子を含む検索フィルタを作成するには、LegacyCertMapping レジストリ キーを有効にする必要があります。レジストリ キーを有効にすると、証明書マッピングでレガシー動作が許可され、指定された LDAP 検索基準を使用してユーザを認証できるようになります。

LegacyCertMapping

KeyType: DWORD

値: 0 (無効) および 1 (有効)

デフォルト: 0

Windows でレジストリ キーを有効にする方法

1. HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\PolicyServer に移動し、LegacyCertMapping を開きます。
2. KeyType 値を REG_DWORD に編集します。
3. [値] を 1 の値に編集します。

注: 0x1 以外の値が設定されるか、またはレジストリ キーが存在しない場合、レジストリ キーは無効になります。

4. レジストリ キーを保存します。

LegacyCertMapping が有効になり、LDAP 検索フィルタ構文をカスタム マッピングで使えるようになります。

UNIX でレジストリ キーを有効にする方法

1. sm.registry ファイルを開きます。
2. ファイルに、以下の行を追加します。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\  
PolicyServer=XXXXX  
LegacyCertMapping=0X1 REG_DWORD
```

3. ファイルを保存します。

LegacyCertMapping が有効になり、LDAP 検索フィルタ構文をカスタム マッピングで使えるようになります。

カスタム マッピング: 例 1

ユーザの証明書に以下が含まれているとします。

SubjectDN: CN=John Smith, UID=JSMITH, OU=development, O=CompanyA

カスタムマッピングを次のように指定します。

CN=%{UID}, OU=%{OU}, O=%{O}

結果として得られる UserDN は次のとおりです。

CN=JSMITH, OU=development, O=CompanyA

カスタム マッピング: 例 2

以下の例に示すように、カスタム マッピング構文では、さらに複雑なマッピングを処理することもできます。

ユーザの証明書に以下が含まれているとします。

Subject DN: CN=John Smith + UID=jsmith +EMAIL=jsmith@companyA.com,
ou=development, o=companyA

カスタムマッピングを次のように指定します。

CN=%{CN.CN}+UID=%{CN.UID}, OU=%{O}

結果として得られる UserDN は次のとおりです。

CN=John Smith+UID=JSMITH, OU=companyA

この例では、CN に複数の属性が含まれています。この構文は、どのコンポーネントの CN を取得して、ユーザ DN の CN に適用するかを示しています。これは「CN.CN」または「CN.UID」を指定して実行されました。この構文は、カスタム式で CN の CN 部分と UID 部分の両方を使用することを示しています。

注:「+」演算子を使用して、ユーザ ディレクトリ内の複数の属性のあいまいさを解消することはできません。「+」演算子は、ユーザ ディレクトリに格納されているユーザのユーザ DN 内のその他の文字と同じように使用されます。

カスタム マッピング: 例 3

スタティックテキストを大かっこの外側に移動させてカスタム式に表示できます。次に例を示します。

ユーザの証明書には以下が含まれています。

Subject DN: CN=John Smith, UID=JSMITH, OU=development

マッピングを次のように指定します。

CN=%{UID}, OU=%{OU}, O=companyA

結果として得られる UserDN は次のとおりです。

CN=JSMITH, OU=development, O=CompanyA

詳細については、次のセクションを参照してください。

テンプレート文字列の使用

テンプレートの文字列は、テキストとカッコで囲まれた式(`%{...}`)で構成されています。カッコの外側のテキストはすべて変更されずに返されます。カッコで区切られた式は次のルールに従って評価されます。

- 識別できない変数名 (例: `DN`) は解決されてから返されます。
- 識別変数名 (たとえば、`DN.UID`) は、変数コンポーネントに解決されてから返されます。

証明書シリアル番号または発行者 DN へのマッピング

証明書マッピングでは、`subjectDN` の一部ではない `CertSerialNumber` 属性と `IssuerDN` 属性のマッピングをサポートしています。ユーザ証明書の `subjectDN` のこの 2 つの属性は、ユーザディレクトリの `UID` や `CN` などのデフォルトまたはカスタムユーザ属性にマッピングできます。

これらの属性をマッピングするには、[証明書マッピング] ウィンドウの [マッピング式] フィールドに以下の式を追加します。

- `CustomAttributeinLDAP1 = %{CertSerialNumber}`

同一タイプの複数属性でのカスタム証明書マッピング

証明書には、同じタイプの複数の属性がサブジェクト DN にある場合があります。`SiteMinder` では、カスタム マッピングを使用して、特定のタイプの属性について 1 番目の属性以外の属性を簡単に参照する方法をサポートしています。構文は以下のとおりです。

`%{attribute_name}` - 最初の `attribute_name` の場合

`%{attribute_nameN}` - N 番目の `attribute_name` の場合

証明書のサブジェクト DN が

`CN=user,ou=dev,sn=1234,sn=2345,sn=3456,o=company,c=us` である場合、すべての `sn` 属性にカスタム証明書マッピングを設定できます。たとえば、最初の `sn` にマップする場合は、カスタムマッピングとして「`%{sn}`」と入力します。2 番目の `sn` にマップする場合は、「`%{sn2}`」と入力します。

不必要な属性のマッピング

個人ごとにその証明書が多少異なっていることがあります。たとえば、2つのアカウント番号を持つユーザがいたり、1つしか持っていないユーザがいたりする場合があります。このような場合、2番目の属性があるときは2番目の番号を使用したいという可能性もあります。その場合は、次の表記法を使用します。

`%{attribute_name2/attribute_name}`

上記の例をもとにカスタム マッピングとして「`%{SN2/SN}`」と入力し、サブジェクト DN 内に 2 番目の番号がある場合はそれを使用することを指示できます。2 番目の番号がない場合は、1 番目のアカウント番号を使用します。

この表記法は、証明書マッピングに指定できる 2 種類の異なる属性を指定するときにも使用できます。たとえば、SN を使用するが SN が無い場合は CN を使用することを示す場合は、「`%{SN/CN}`」と入力します。

証明書の有効性チェック(任意)

証明書の有効性チェックは、X.509 クライアント証明書認証のオプション機能です。

ポリシー サーバでは、以下の方法を使用してユーザ証明書が有効かどうかを確認できます。

- 証明書破棄リスト(CRL)のチェック

ポリシー サーバで CRL を使用して証明書が取り消されていないかを判断します。管理 UI では、CRL を取得するために、CRL ディレクトリへのパスを指定するか、または CRL 配布ポイントを選択できます。

- OCSP (Online Certificate Status Protocol) のチェック

ポリシー サーバで OCSP レスポンダにリクエストを送信してユーザを参照します。OCSP レスポンダは、ユーザ証明書の取り消しステータスを判断し、レスポンスを送り返します。

ポリシー サーバは、証明書の検証にどの方法を使用するかを以下の方法で決定します。

- CRL チェックのみが設定された場合、ポリシー サーバは CRL を使用します。
- OCSP のみが設定された場合、ポリシー サーバは OCSP を使用します。
- CRL チェックおよび OCSP を設定し、フェイルオーバーが有効な場合、ポリシー サーバは最初指定されたプライマリ検証方法 (CRL または OCSP) をまず使用します。最初の検証方法に失敗した場合、2 番目の方法が使用されます。次のリクエストに対しては、最初の方法に戻ります。

ポリシー サーバでは、取得した最初の「有効」または「無効(取り消し)」のレスポンスが確実な情報として認識されます。最初の有効なレスポンスの後に、CRL または OCSP の後続のレスポンスを要求することはありません。また、ポリシー サーバでは、CRL および OCSP 検証の結果を集計してユーザ証明書の包括的なステータスを決定することはありません。

詳細情報:

[OCSP と CRL 間のフェイルオーバー](#) (P. 435)

有効性チェックを実行するための前提条件

ポリシー サーバでユーザ証明書の検証を行うには、X.509 クライアント証明書認証方式が設定され、保護されているリソースをユーザが要求したときにそのユーザを認証できる必要があります。

[X.509 クライアント証明書認証方式](#) (P. 381)を設定するための手順を確認します。

CRL と OCSP を設定するための手順は、次のセクション以降で説明します。

証明書破棄リスト(CRL)のチェック

証明書破棄リスト(CRL)は、失効した証明書のデジタル署名されたリストで、対応する証明書を発行した認証機関(CA)によって提供されるものです。CRL に対して証明書を照合することは、証明書が有効かどうかを判断する 1 つの方法です。

ポリシー サーバ証明書マッピングで設定された発行者 DN ごとに 1 つの CRL を使用することができます。

ポリシー サーバは、以下のいずれかの方法で **CRL** を取得します。

- **LDAP ディレクトリから CRL を取得**

ポリシー サーバは、**CRL** 設定で指定された **LDAP** ディレクトリへの接続を確立できます。そのディレクトリから **CRL** が取得されます。

LDAP ディレクトリには複数の **CRL** が存在している可能性があります。各証明書マッピングは、エントリポイントによって識別される 1 つの **CRL** のみを参照できます。そのため、各 **CRL** のエントリポイントはそれぞれ異なっている必要があります。

注: ポリシー サーバは、すべての理由コードが含まれている **CRL** のみを受け入れ、特定の理由コードのみを含む **CRL** は拒否します。

- **CRL 配布ポイント拡張情報 (CDP) によって指定された場所から CRL を取得**

ユーザ証明書には配布ポイント拡張情報を含めることができます。**CDP** 拡張情報は、**CRL** を取得できる場所を指しています。ポリシー サーバでは、配布ポイントについて、1 つの **CRL** への 1 つのエントリ、または異なる **CRL** への複数のポイントをサポートします。

配布ポイントには、**HTTP**、**HTTPS**、**LDAP** などの異なるソースを使用できます。**CDP** 拡張情報に、複数の値を持つ配布ポイント名を使用するエントリが含まれる場合、これらの値はすべて同じ **CRL** を指している必要があります。

ポリシー サーバで **CRL** を取得した後、必要なチェックを実行することができます。管理 UI で **CRL** のキャッシュが有効になっている場合、ポリシー サーバはメモリ内に **CRL** を格納できます。キャッシュが有効でない場合、ポリシー サーバは認証リクエストごとに **CRL** を取得する必要があります。

CRL の理由コード要件

ポリシー サーバでは、可能性のあるすべての理由コードの取り消し情報が含まれる **CRL** のみをサポートします。**CRL** に、一部の理由コードのみによって取り消された証明書が含まれる場合、ポリシー サーバはエラーを生成し、**CRL** を無効とみなします。ポリシー サーバでは無効な **CRL** を無視し、有効な **CRL** を見つけるまで利用可能な **CRL** の検索を続行します。

ポリシー サーバはデルタの **CRL** を無効な **CRL** を扱います。デルタ **CRL** には、**CA** が完全な **CRL** を発行した後に取り消しステータスが変更された証明書のみがリスト表示されます。ポリシー サーバではデルタ **CRL** を無視し、有効な **CRL** を見つけるまで利用可能な **CRL** の検索を続行します。

ポリシー サーバが利用可能なすべての **CRL** を検索しても有効なものが見つからなかった場合、そのユーザは認証されません。

CRL のサイズ制限

ポリシー サーバでは、デフォルト サイズとして 2MB までの CRL キャッシュ サイズを使用して証明書を検証できます。ただし、それより大きな CRL がある場合、MaxCrlBufferMB レジストリ キーを作成することにより、最大 1GB までキャッシュ サイズを増加させることができます。

レジストリ キーを作成する方法

1. ポリシー サーバにアクセスし、以下のいずれかを実行します。

Windows: レジストリ エディタを開き、次のキーまで移動します:

HKEY_LOCAL_MACHINE\Software\Netegrity\SiteMinder\CurrentVersion\PolicyServer

UNIX: sm.registry ファイルを開きます。このファイルのデフォルトの場所は、*siteminder_home/registry* です。

siteminder_home

ポリシー サーバのインストール パスを指定します。

2. MaxCrlBufferMB を追加し、レジストリ値をのタイプを REG_DWORD にします。

単位: メガバイト

ベース: 10 進数

デフォルト値: 2

最小値: 1

最大値: 1023

3. 以下のいずれかを行います。

Windows: レジストリ エディタを終了します。

UNIX: sm.registry ファイルを保存します。

4. ポリシー サーバを再起動します。

CRL 署名の検証

CRL 署名の検証は、CRL チェックのオプション機能です。

ポリシー サーバでは、CRL に対して証明書を比較する前に、LDAP ディレクトリに格納された CA 証明書で CRL の署名を検証します。ポリシー サーバは LDAP ユーザ ディレクトリの特定のエン트리から CA 証明書を取得します。これは、証明書内の発行者 DN または 管理 UI で証明書マッピングに設定した CRL ディレクトリの DN に基づいて特定されます。

CA 証明書は、ポリシー サーバがアクセスできる LDAP ディレクトリ内に格納します。LDAP ディレクトリでは、特定のディレクトリ エントリを `cacertificate` という名前の属性で設定します。`cacertificate` 属性は、複数の CA 証明書を格納できる多値属性です。CRL がパーティションに分割され、それぞれに対して異なる CA キーで署名された場合、複数の CA 証明書が必要になる可能性があります。ポリシー サーバでは、指定されたパーティションについて関連する CA 署名の証明書にアクセスできれば、そのパーティションの署名のみを検証します。

署名の検証について、ポリシー サーバでは以下のハッシュ アルゴリズムを使用できます。

- MD5
- MD2
- SHA1
- SHA2 アルゴリズム (SHA224、SHA256、SHA384、SHA512)

注: 使用されている署名アルゴリズムは CRL に指定されています。

CA 証明書が使用できないか、CRL の署名にサポートされていないアルゴリズムが使用されている場合、CRL 検証プロセスで署名チェックをオフにすることができます。

重要: 署名のチェックがオフにされた場合、CRL が格納されているリポジトリが適切に保護されていることを確認する必要があります。

CRL を特定するための CRL 配布ポイント

CRL 配布ポイント(CDP)は、CRL の場所を指している証明書拡張情報です。指定された場所から、ポリシー サーバは CRL を取得し、どの証明書が取り消されたかを確認できます。

ポリシー サーバは、CDP 拡張情報のいくつかのソースによって特定される CRL を取得できます。取得の別のオプションを使用すると、Web サーバが CRL を取得できることを確認するのに役立ちます。

ポリシー サーバは以下のソースから CRL を取得できます。

- LDAP
- HTTP
- HTTPS

HTTPS 配布ポイントの場合、ポリシー サーバは安全な接続を使用します。有効な CA 証明書ファイルまたは証明書バンドル(チェーンを形成する連結された証明書ファイル)は、ディレクトリ `policy_server_home/config` 内に存在する必要があります。そうでないと、HTTPS サーバへの接続に失敗します。

CA 証明書ファイルは PEM 形式(base64 エンコード)で、`cert.pem` という名前が付けられている必要があります。別の形式での証明書は OpenSSL コマンドラインユーティリティを使用して、pem 形式に変換できます。`cert.pem` ファイルには、CDP 拡張情報に設定された SSL Web サーバ用の発行者証明書と、各配布ポイント用の信頼された CA 証明書が含まれている必要があります。

注: OpenSSL ユーティリティの詳細については、OpenSSLドキュメントを参照してください。

CDP 拡張に複数のエントリが含まれている場合、ポリシー サーバでは証明書を検証するために、すべての理由コード含む CRL で最初を取得された CRL を使用します。CRL が取得される順序は、エントリが証明書内にリスト表示されているのと同じ順序です。ポリシー サーバが CDP リスト内の最初の CRL を取得できない場合、次の CRL の取得を試行し、成功するまで続行します。

ポリシー サーバがどのソースからも有効な CRL を取得できない場合、認証は失敗し、ユーザはアクセスを拒否されます。CRL と OCSP の間のフェイルオーバーを有効にすることは、この動作の唯一の例外です。CRL チェックがプライマリ検証方法で、それに失敗した場合、ポリシー サーバでは 2 番目の方法として OCSP にフェイルオーバーします。

注: OCSP の設定ファイルで[フェイルオーバー](#) (P. 435)を有効にします。

CDP 配布ポイントは、[証明書マッピング]ダイアログ ボックスで[CRL チェック](#) (P. 419)設定の一部として設定します。

CRL パーティションの署名の検証

異なる CA キーで CRL の各パーティションに署名することができます。ポリシー サーバでは、CRL の各パーティションについて関連 CA で署名された証明書にアクセスできる限り、すべての CRL パーティションの署名を確認できます。

CRL を見つけるために証明書配布ポイントを使用する場合は、CRL パーティションの使用が関与します。拡張情報には別の CRL への複数のリンクが含まれている場合があります、そのすべての署名に検証が必要です。

ポリシー サーバは、LDAP ディレクトリに格納された CA 証明書で CRL の署名を確認します。この LDAP ディレクトリでは、特定のエントリを **cacertificate** という名前前の属性で設定します。この属性は複数の値を持つことができます。複数の CA 証明書は、異なる CA キーによって署名された CRL パーティションを検証するために必要です。

証明書破棄リスト(CRL)のチェックの設定

CRL チェックを設定することにより、ユーザ証明書が取り消されたかどうかを確認することができます。これは、無効なクライアント証明書を持つユーザが保護されているリソースにアクセスできないよう徹底するのに役立ちます。

CRL は、LDAP ディレクトリ、または CDP によって指定された場所から取得できます。ポリシー サーバが特定の LDAP ディレクトリから CRL を取得する場合は、管理 UI の[ユーザ ディレクトリ]セクションで、そのディレクトリへの接続を必ず設定してください。この LDAP ディレクトリは、ユーザ ストアおよび CRL ストアとして機能します。CRL チェックを設定する前または CRL 設定中に、このディレクトリを設定します。

CRL チェックを設定する方法

1. 管理 UI にログインします。
2. [インフラストラクチャ]-[ディレクトリ]を選択します。
3. [証明書マッピング]オプションを展開します。
4. [証明書マッピングの作成]または[証明書マッピングの変更]を選択します。

[証明書マッピング] ダイアログボックスが表示されます。

5. [証明書破棄リスト(CRL)のチェック]グループ ボックスで[CRL チェックの実行]を選択します。

CRL 固有のフィールドおよびコントロールが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. [CRL ディレクトリ]フィールドで、ポリシー サーバが CRL を取得する LDAP ディレクトリの名前を選択します。このディレクトリ名は、管理 UI の[ユーザ ディレクトリ]セクションでディレクトリを設定するときに割り当てた名前です。リストにユーザ ディレクトリがない場合、[作成]をクリックしてディレクトリ接続を追加します。

[CRL ディレクトリ]フィールドに LDAP ディレクトリを指定しない場合は、ポリシー サーバが CRL を抽出する方法として[配布ポイントの使用]を選択します。

注: [CRL ディレクトリ]フィールドに対して、「証明書の拡張子から取得します」というオプションのテキスト文字列値が存在します。このオプションは、CRL の取得に配布ポイントを使用する場合のみ選択してください。

7. [CRL ディレクトリ]フィールドにユーザ ディレクトリを指定した場合は、[CRL ディレクトリ]フィールドのエントリ ポイント DN の値を入力します。

このフィールドのエントリは、ポリシー サーバが CRL を取得するために CRL ディレクトリを検索する DN になります。このフィールドは、[CRL ディレクトリ]フィールドが LDAP ディレクトリに設定されている場合のみ有効です。CRL の取得に配布ポイントを使用する場合は、このフィールドを空白のままにします。

8. (任意)CRL の署名を確認するため[署名の確認]を選択します。

ポリシー サーバでは、アクセス可能な LDAP ホストに対して、CRL の署名を確認するのに必要な証明書を取得することを要求します。管理 UI で LDAP ホストがユーザ ディレクトリ接続として設定されていることを確認してください。

ポリシー サーバは、CRL を特定するために CRL 配布ポイントを使用できます。その配布ポイントが LDAP URI である場合、ポリシー サーバは CRL の署名を確認できます。配布ポイントが HTTP URI である場合は、証明書を取得する LDAP ホストが利用可能ではないため、[署名の確認]オプションは選択しないでください。

9. (任意)CRL の取得に CDP 拡張情報を使用するには、[配布ポイントの使用]を選択します。このオプションは、CRL ディレクトリを指定する代わりに使用できます。

[配布ポイントの使用]を選択して[CRL ディレクトリ]フィールドにディレクトリを入力した場合、ポリシー サーバでは、CRL の検索に配布ポイントのみを使用します。配布ポイントは、CRL ディレクトリより優先されるため、CRL ディレクトリのエントリは無視されます。

10. 必要に応じて残りの設定を完了し、[サブミット]をクリックします。

証明書破棄リストのチェックが有効になりました。

HTTP プロキシを介した CRL へのアクセス

CRL リクエストは、HTTP 接続を使用して送信できます。この場合、HTTP GET リクエストが証明書検証用の CRL を取得する必要があります。

企業が使用する多くの環境では、HTTP トラフィックは HTTP プロキシを経由します。ポリシー サーバが HTTP プロキシ経由で CRL を取得するには、ポリシー サーバが存在するシステム上で `http_proxy` 環境変数を設定する必要があります。例:

```
set http_proxy=http://username:password@proxy.example.org:8080
export http_proxy
```

ポート番号を指定しない場合、SiteMinder ではデフォルトでポート 1080 が使用されます。

username

プロキシ サーバのログイン ユーザ名。この名前はプロキシ サーバ設定の有効なユーザである必要があります。

password

プロキシ サーバのログイン パスワード。このパスワードはプロキシ ユーザ設定の有効なエントリである必要があります。

注: この環境変数は、プロキシを介して OCSP レスポンダにアクセスするために使用することはできません。

オンライン証明書状態プロトコル チェック(OCSP)

オンライン証明書状態プロトコル(OCSP)は、ユーザ証明書が有効かどうかを確認することによって、環境全体にわたってセキュリティを管理するのに役立ちます。OCSP では、OCSP レスポンダを使用して、X.509 クライアント証明書の取り消しステータスを判断します。OCSP レスポンダは、特定の証明書について証明書検証データを集計し、OCSP リクエストに応答することによって、リアルタイムに検証を実行します。

OCSP の利点の 1 つと言えるのが、最新の証明書ステータス情報を保持するためにクライアント側で常に CRL をダウンロードしておく必要がないことです。CRL は非常に大容量である場合もあります。

OCSP チェックを実行するために、ポリシー サーバでは、**SMocsp.conf** ファイルという名前のテキストベースの設定ファイルを使用します。このファイルは、ディレクトリ **policy_server_home/config** にあり、SiteMinder OCSP 機能を使用するため必要となります。

SMocsp.conf ファイルには、1 つ以上の OCSP レスポンダの操作を定義する設定が含まれます。ユーザ証明書が有効かどうか確認する際、ポリシー サーバでは **SMocsp.conf** ファイル内の発行者 DN を検索します。発行者 DN が見つかった場合、その発行者 DN と関連付けられた指定の OCSP レスポンダを使用して証明書ステータスの確認が行われます。発行者 DN が見つからなかった場合、ポリシー サーバは OCSP チェックを実行せず、証明書は有効であるとみなされます。OCSP 検証が必要でない場合は、発行者 DN がなくても証明書が有効とみなされます。

[OCSP リクエストに署名](#) (P. 433)することはできますが、この署名は任意です。

OCSP レスポンダがポリシー サーバにレスポンスを返した場合、ポリシー サーバのデフォルトの動作では署名されたレスポンスを検証します。**SMocsp.conf** ファイル内のいくつかの設定では、レスポンス検証を有効にすることが必要とされます。

注: CRL チェックが管理 UI で有効になっている場合、ポリシー サーバでは、**SMocsp.conf** ファイルが存在するかどうかにかかわらず、デフォルトで CRL チェックを使用します。フェイルオーバーを有効にし、OCSP をプライマリ検証方法に設定した場合のみ、OCSP が CRL チェックより優先されます。フェイルオーバーは OCSP 設定ファイルで設定されます。

OCSP に関する要件

証明書検証に OCSP を使用するには、以下のコンポーネントをセットアップする必要があります。

- 認証機関(CA)環境を確立します。
- OCSP レスポンダをセットアップします。
- LDAP ディレクトリを設定して、返された OCSP レスポンスの署名を検証する OCSP の信頼されたレスポンド証明書が格納されるようにします。OCSP の信頼されたレスポンド証明書は、1 つの信頼された検証証明書または証明書の集合になります。

これらの証明書は、OCSP トランザクションとは別の通信で取得します。

証明書の信頼された証明書または証明書の集合を格納するには、管理 UI の[ユーザ ディレクトリ]セクションに LDAP ディレクトリを設定します。ユーザ ディレクトリの存在によって、ポリシー サーバはディレクトリに接続し、レスポンスの署名を検証するための証明書またはその集合を特定できるようになります。証明書の集合が格納されている場合は、すべての証明書を格納するため、複数値のバイナリ属性を必ず使用してください。

OCSP レスポンダには、レスポンスと共に署名検証の証明書を含めることができます。この場合、ポリシー サーバは証明書を検証し、さらに LDAP ディレクトリ内の信頼された証明書でレスポンスの署名を検証します。

署名検証の証明書がレスポンスにない場合、ポリシー サーバは LDAP ディレクトリ内の証明書のまたはその集合でレスポンスの署名を検証します。

OCSP を設定する際は、証明書の場合またはその集合を `SMocsp.conf` ファイルの `ResponderCertEP` 設定に指定します。

ポリシー サーバは、SHA-1 および SHA-2 アルゴリズム (SHA224、SHA256、SHA384、SHA512) を使用して署名されたすべての OCSP レスポンスを使用することができます。

- ユーザ証明書を発行した CA 証明書を LDAP ディレクトリに格納します。この CA 証明書はユーザ証明書を検証します。この証明書は OCSP の信頼されたレスポンス証明書を格納するのと同じ LDAP ディレクトリに格納することも、別の LDAP ディレクトリに格納することもできます。
- SiteMinder OCSP 設定ファイル (`SMocsp.conf`) を設定します。サンプルの設定ファイルである `SMocsp.Sample.conf` がポリシー サーバと併せてインストールされます。このファイルをコピーして名前を `SMocsp.conf` に変更し、必要に応じて内容を変更します。

UNIX オペレーティング プラットフォームの場合は、ファイル名で大文字と小文字が区別されます。

- 必要に応じて、ポリシー サーバが OCSP リクエストに署名するために使用する秘密キー/証明書のペアがポリシー サーバで利用可能であることを確認します。この鍵/証明書のペアは `smkeydatabase` に格納します。

OCSP 設定ファイルの作成

ポリシー サーバは、SMocsp.conf という名前のファイルを使用して OCSP チェックを行います。SMocsp.conf ファイルは 1 つ以上の OCSPResponder レコードが含まれる ASCII ファイルです。

SMocsp.conf ファイルはディレクトリ *policy_server_home/config* 内に存在する必要があります。設定を容易にするため、SMocsp.Sample.conf という名前のサンプルファイルがポリシー サーバと併せて config フォルダにインストールされます。お使いの環境用に OCSP を設定するには、サンプルファイルをコピーして名前を SMocsp.conf に変更します。

注: UNIX プラットフォームの場合、ファイル名で大文字と小文字が区別されます。

以下は、1 つの OCSPResponder エントリを含む SMocsp.conf ファイルの例です。

注: サンプル ファイルには利用可能なすべての設定が含まれていますが、すべての設定が必要となるわけではありません。

```
[
OCSPResponder
IssuerDN C=US,ST=Massachusetts,L=Boston,O=,OU=QA,CN=Issuer,E=user@domain.com
CACertDir 10.1.22.2:389
CACertEP uid=caroot,dc=systest,dc=com
ResponderCertDir 10.2.11.1:389
ResponderCertEP cn=OCSP,ou=PKI,ou=Engineering,o=ExampleInc,c=US
ResponderCertAttr cacertificate
ResponderLocation http://10.12.2.4:389
AIAExtension NO
HttpProxyEnabled YES
HttpProxyLocation http://10.11.2.5:80
HttpProxyUserName proxyuser1
HttpProxyPassword letmein
SignRequestEnabled YES
SignDigest SHA256
Alias defaultenterpriseprivatekey
IgnoreNonceExtension NO
PrimaryValidationMethod OCSP
EnableFailover YES
]
```

SMocsp.conf ファイルを変更する際のガイドラインを以下に示します。

- 設定の名前はすべて大文字と小文字が区別されるわけではありません。エントリの大文字と小文字が区別されるかどうかは特定の設定によって決まります。
- ファイル内で空白のままになっている設定がある場合、ポリシー サーバはエラー メッセージを送信し、エントリが無効で設定が無視されることを通知します。その設定が意図的に空白になっている場合は、メッセージを無視してかまいません。
- 設定の名前の先頭にスペースを含めることはしないでください。

このファイルには以下の設定項目があります。

OCSPResponder

必須です。エントリが **OCSP** レスポンダレコードであることを示します。各 **OCSP** レスポンダ レコードは、**OCSPResponder** で始まる必要があります。

IssuerDN

必須です。証明書発行者の **DN** を指定します。この値は、ファイル内の各 **OCSP** レスポンダ レコードのラベルとなります。

エントリ: 証明書内の発行者 **DN** 値。

AlternateIssuerDN

任意です。セカンダリ **IssuerDN** または逆の **DN** を指定します。

CACertDir

必須です。ユーザ証明書を発行する **CA** 証明書を保持する **CA** 証明書ディレクトリの名前を指定します。

このディレクトリは、ポリシー サーバが接続できるように、管理 UI で **SiteMinder** ユーザ ディレクトリとして設定する必要があります。

ユーザ ディレクトリの有効な **IP** アドレスおよびポート番号を入力します。

CACertEP

必須です。**CA** 証明書が存在する **CA** 証明書ディレクトリのエントリ ポイントを指定します。

証明書ディレクトリのエントリ ポイントを表す文字列を入力します。

ResponderCertDir

必須です。レスポンド証明書が格納される LDAP ディレクトリを指定します。

このディレクトリは、ポリシー サーバが接続できるように、管理 UI で SiteMinder ユーザ ディレクトリとして設定する必要があります。

ディレクトリの有効な IP アドレスおよびポート番号を入力します。

ResponderCertEP

必須です。レスポンド証明書が存在する LDAP ディレクトリのエントリ ポイントを指定します。レスポンド証明書ディレクトリは ResponderCertDir 設定で指定されます。

署名検証の証明書は、レスポンスの署名または中間の証明書の集合を直接検証する証明書です。

OCSP レスポンドには、レスポンスと共に署名検証の証明書を含めることができます。この場合、ポリシー サーバは、LDAP ディレクトリ内の信頼された証明書を使用してレスポンスの署名および証明書を検証します。署名検証の証明書がレスポンスにない場合、ポリシー サーバは LDAP ディレクトリ内の証明書のまたはその集合のみでレスポンスの署名を検証します。

証明書またはその集合が存在するディレクトリのエントリ ポイントを表す文字列を入力します。

ResponderCertAttr

必須です。ポリシー サーバがレスポンド証明書ディレクトリ (ResponderCertDir 設定で指定) でレスポンド証明書の検索に使用する LDAP ディレクトリ属性を指定します。

ResponderLocation

任意です。OCSP レスポンダ サーバの場所を示します。

ResponderLocation 設定または AIAExtension 設定を使用できますが、以下の点に注意してください。

- ResponderLocation 設定が空白のままか、または SMocsp.conf ファイルにない場合、AIAExtension 設定は YES に設定される必要があります。また、AIA 拡張が証明書に含まれている必要があります。
- ResponderLocation 設定に値があり、AIAExtension が YES に設定されている場合、ポリシー サーバは検証に ResponderLocation を使用します。ResponderLocation 設定は AIAExtension より優先されます。
- この設定に指定された OCSP レスポンダがダウンしており、AIAExtension が YES に設定されている場合、認証は失敗します。ポリシー サーバは、この証明書の AIA 拡張で指定されたレスポンダの使用は試しません。

場所を入力する際は、*responder_server_url:port_number* の形式で入力します。

レスポンダ サーバの URL およびポート番号を入力します。

AIAExtension

任意です。ポリシー サーバが検証情報を見つけるために証明書の Authority Information Access (AIA) 拡張を使用するかどうかを指定します。

AIAExtension または ResponderLocation の設定を使用できますが、以下の点に注意してください。

- AIAExtension が YES に設定され、ResponderLocation が設定されていない場合、AIA 拡張が証明書内にあれば、ポリシー サーバは検証に証明書の AIA Extension を使用します。
- AIAExtension が YES に設定され、ResponderLocation 設定にも値がある場合、ポリシー サーバは検証に ResponderLocation を使用します。ResponderLocation 設定は AIAExtension より優先されます。
- AIAExtension が NO に設定されている場合、証明書に AIA 拡張が含まれているかどうかにかかわらず、ポリシー サーバは ResponderLocation を使用します。

「YES」または「NO」を入力します。

デフォルト: NO

HttpProxyEnabled

任意です。Web サーバではなくプロキシ サーバに OCSP リクエストを送信するようにポリシー サーバに指示します。

「YES」または「NO」を入力します。

デフォルト: NO

HttpProxyLocation

任意です。プロキシ サーバの URL を指定します。この値は、HttpProxyEnabled が YES に設定されている場合のみ必須です。

http://で始まる URL を入力します。

注: https://で始まる URL は入力しないでください。

HttpProxyUserName

任意です。プロキシ サーバに対するログイン クレデンシャルのユーザ名を指定します。このユーザ名はプロキシ サーバの有効なユーザの名前である必要があります。この値は、HttpProxyEnabled が YES に設定されている場合のみ必須です。

英数字の文字列を入力します。

HttpProxyPassword

任意です。プロキシ サーバユーザ名に対応するパスワードを指定します。この値はクリア テキストで表示されます。この値は、HttpProxyEnabled が YES に設定されている場合のみ必須です。

英数字の文字列を入力します。

SignRequestEnabled

任意です。生成された OCSP リクエストに署名するようポリシー サーバに指示します。署名機能を使用する場合はこの値を Yes に設定します。

この値は、ユーザ証明書の署名とは関係がなく、OCSP リクエストにのみ関係があります。

注: この設定は、OCSP レスポンダで署名されたリクエストが必要とされる場合のみ必須です。

「YES」または「NO」を入力します。

デフォルト: NO

SignDigest

任意です。OCSP リクエストを署名するときにポリシー サーバが使用するアルゴリズムを指定します。この設定では大文字と小文字は区別されません。この設定は、SignRequestEnabled 設定が YES に設定された場合のみ必須です。

次のいずれかのオプションを入力します: SHA1、SHA224、SHA256、SHA384、SHA512

デフォルト: SHA1

Alias

任意です。リクエストに署名したキー/証明書のペアをポリシー サーバが検索するためのエイリアス名です。このキー/証明書ペアは SiteMinder smkeydatabase 内にある必要があります。

注: このエイリアスは、SignRequestEnabled 設定が YES に設定された場合のみ必須です。

小文字の ASCII 英数文字を使用してエイリアスを入力します。

IgnoreNonceExtension

任意です。OCSP リクエストに乱数を含めないようにポリシー サーバに指示します。乱数(一度使用される数字)は、レスポンスの再利用を防ぐため認証リクエスト内に含まれることがある一意の数です。このパラメータを Yes に設定すると、OCSP リクエストに乱数が含まれないようになります。

「YES」または「NO」を入力します。

デフォルト: NO

PrimaryValidationMethod

任意です。ポリシー サーバで証明書の検証に使用するプライマリ検証方法が OCSP または CRL であるかを示します。この設定は、EnableFailover 設定が YES に設定された場合のみ必須です。

「OCSP」または「CRL」を入力します。

デフォルト: OCSP

EnableFailover

OCSP と CRL の証明書検証方法間でフェイルオーバーするようポリシー サーバに指示します。

「YES」または「NO」を入力します。

デフォルト: NO

OCSP チェックの設定

OCSP チェックを設定して、無効なクライアント証明書を持つユーザが保護されているリソースにアクセスできないようにします。

注: OCSP チェックを有効にする前に、[証明書認証](#) (P. 381)用の環境をセットアップする必要があります。

ポリシー サーバは、SHA-1 および SHA-2 アルゴリズム (SHA224、SHA256、SHA384、SHA512) を使用して署名されたすべての OCSP レスポンスを使用することができます。

OCSP チェックを設定する方法(フェイルオーバーなし)

1. `policy_server_home/config` に移動します。
SMocsp.Sample.conf という名前のサンプル ファイルがポリシー サーバと共に config ディレクトリにインストールされます。
2. サンプルの設定ファイルをコピーして名前を SMocsp.conf に変更します。
UNIX プラットフォームの場合、ファイル名で大文字と小文字が区別されます。
Windows プラットフォームの場合は区別されません。
3. テキスト エディタでファイルを開きます。
4. 証明書マッピングで指定された IssuerDN と一致する IssuerDN ごとに、ファイル内に一意の OCSPResponder エントリを追加します。
重要: 発行者 DN ごとにレスポンス レコードを設定しない場合、ポリシー サーバでは証明書の有効性を検証せずにユーザを認証します。
5. ファイルを保存します。
6. ポリシー サーバを再起動します。
7. 管理 UI にログオンします。
8. [インフラストラクチャ]-[ディレクトリ]を選択します。
9. [証明書マッピング]オプションを展開します。
10. [証明書マッピングの作成]または[証明書マッピングの変更]を選択します。
[証明書マッピング]ダイアログ ボックスが表示されます。

11. OSCP が使用する予定の唯一の有効性チェック方法である場合は、[CRL チェックの実行]チェック ボックスをオフにします。フェイルオーバーを使用する場合は、CRL チェックを無効にしないでください。

ユーザ証明書の発行者が証明書マッピングと一致し、CRL チェックが有効になっている場合、ポリシー サーバは OSCP ではなく CRL チェックを使用します。CRL が OSCP より優先される唯一の例外は、フェイルオーバーが有効にされた場合です。[フェイルオーバーを有効](#) (P. 435)にした場合、ポリシー サーバは設定されたプライマリ検証方法を使用します。

12. 変更を保存して 管理 UI を終了します。
13. (任意) [OCSP リクエストに署名](#) (P. 433)するようポリシー サーバを設定します。

OCSP が有効になりました。OCSP を無効にするには、SMocsp.conf ファイルの名前を変更します。

HTTP プロキシを介した OCSP レスポンドへのアクセス

OCSP リクエストは HTTP 接続を介して送信されます。この場合、証明書検証のため OCSP レスポンドへのリクエストに対する HTTP GET が必要になります。

企業が使用する多くの環境では、HTTP トラフィックは HTTP プロキシを経由します。ポリシー サーバが HTTP プロキシを介して OCSP リクエストを送信できるようにするには、SMocsp.conf ファイルにプロキシ情報を設定します。

プロキシを介した OCSP レスポンドへのアクセスを設定する方法

1. 既存の SMocsp.conf ファイルを編集するか、またはポリシー サーバの config ディレクトリ (*policy_server_home/config*) にファイルを作成します。
2. 以下のように設定を編集します。

- HttpProxyEnabled YES
- HttpProxyLocation *proxy_server_URL*
- HttpProxyUserName *user_login_name*
- HttpProxyPassword *password_for_login*

それぞれの値の設定方法については、[OCSP 設定ファイルの作成](#) (P. 425) 方法を確認してください。

3. ポリシー サーバを再起動します。

OCSP リクエストの署名(オプション)

SiteMinder 証明書認証方式を使用している場合、ポリシー サーバで OCSP リクエストに署名することができます。リクエストの署名は、OCSP レスポンダが署名されたリクエストを必要とする場合のみ必須となります。

OCSP リクエストに署名するための前提条件

OCSP の署名を設定する前に、以下の前提条件タスクを完了する必要があります。

1. [OCSP チェック](#) (P. 422)を設定します。
2. `smkeydatabase` という名前の SiteMinder キー データベースをセットアップします。

[smkeytool ユーティリティ](#) (P. 653)を使用してデータベースを作成します。

3. リクエストに署名するための署名キー/証明書のペアを SiteMinder キー データベースに追加します。

キー/証明書ペアを追加する際は、エイリアスを指定して、ポリシー サーバがキー データベース内の証明書エントリを識別できるようにします。エイリアスを指定する際は以下の制限に注意します。

- 1つの証明書は、1つのエイリアスでのみ格納します。同じ証明書を別のエイリアスで格納しようとする場合、失敗します。複数のレスポンダで同じ署名証明書を使用する場合は、それらのレスポンダに同じエイリアスを使用します。
- 証明書のエイリアス名には任意の名前を付けることができますが、最初のエイリアスは `defaultenterpriseprivatekey` である必要があります。
- エイリアスはすべて小文字の USA ASCII 文字である必要があります。英数文字を使用する必要があります。

注: 秘密キーは RSA キーである必要があります。

OCSP リクエストの署名の設定

SiteMinder 証明書認証方式を使用している場合、ポリシー サーバではリクエストに署名し、レスポンスを検証することができます。

OCSP リクエストの署名を設定する方法

1. テキスト エディタで `SMocsp.conf` ファイルを開きます。このファイルは、`policy_server_home/config` ディレクトリにあります。
2. 各レスポンスについて `SMocsp.conf` ファイルに以下のエントリを追加します。

SignRequestEnabled

OCSP リクエストに署名するようポリシー サーバに指示します。

有効値: Yes または No

署名機能を使用する場合はこの値を **Yes** に設定します。

署名を無効にするにはデフォルトの **No** のままにします。

デフォルト: No。SignRequestEnabled エントリがレスポンス レコードにならない場合、ポリシー サーバでは OCSP リクエストに署名できません。

SignDigest

OCSP リクエストの署名に使用するアルゴリズムを示します。

オプション

- SHA1
- SHA224
- SHA256
- SHA384
- SHA512

SignDigest では大文字小文字が区別されません。

デフォルト: SHA1

Alias

ポリシー サーバが署名用のキー/証明書ペアを取得するために使用するエイリアスを示します。エイリアスは小文字の USA ASCII 英数文字に制限されます。

3. SMocsp.conf ファイルを保存します。
4. ポリシー サーバを再起動します。

OCSP リクエストの署名が有効になりました。

OCSP と CRL 間のフェイルオーバー

ポリシー サーバでは証明書の検証方法として OCSP と CRL を使用できます。両方の方法を設定した場合、両者の間でフェイルオーバーするようポリシー サーバを設定することが可能です。フェイルオーバーを有効にすると、1 方の証明書検証方法が使用できない場合に認証が失敗するのを回避することができます。

証明書チェックのフェイルオーバーを実装するには、OCSP 設定ファイル (SMocsp.conf) 内にプライマリ検証方法を指定します。プライマリ検証方法が利用できない場合、ポリシー サーバは、リクエストを完了するために 2 番目の検証を使用します。次のリクエストについては、プライマリ検証方法を再び使用し、失敗が発生しない限りその方法を使用します。

プライマリ検証方法としての OCSP

プライマリ検証方法が OCSP で、OCSP レスポンドが利用できない場合、ポリシー サーバは、代わりに CRL を使用して証明書の検証を実行します。

OCSP レスポンドが「good」または「revoked」のレスポンス インジケータを返している限り、OCSP 機能が無視してフェイルオーバーされることはありません。レスポンス インジケータが「unknown」である場合は、CRL チェックへのフェイルオーバーが発生します。

プライマリ検証方法として **OCSP** を設定した場合、ポリシー サーバは以下のよう
に動作します。

OCSP 証明書の検証結果	フェイルオーバー無効	フェイルオーバー有効
有効	ユーザは認証されます	OCSP の結果のみに基づいてユーザが 認証されます。CRL チェックは必要あり ません。
取り消し	ユーザは認証されません	ユーザは認証されません。CRL チェック は必要ありません。
不明またはレスポンスなし	ユーザは認証されません	CRL チェックが実行されます

プライマリ検証方法としての **CRL** チェック

プライマリ検証方法が **CRL** チェックで、ポリシー サーバが **CRL** を取得できない場
合、**OCSP** レスポンダにフェイルオーバーします。この場合、ポリシー サーバで
は **CRL** ディレクトリ サーバへの接続が利用可能でない場合のみ **OCSP** を使用し
ます。**CRL** が有効なレスポンスを返す場合、**OCSP** は使用されません。

注: フェイルオーバーが有効になっていない場合、**CRL** チェックと **OCSP** の両方
が設定されていれば、ポリシー サーバでは証明書の検証に **CRL** チェックのみを
使用します。

プライマリ検証方法として **CRL** を設定した場合、ポリシー サーバは以下のよう
に動作します。

CRL 証明書の検証結果	フェイルオーバー無効	フェイルオーバー有効
有効	ユーザは認証されます	最初の有効な CRL に基づいて チェックされます。それ以上の CRL チェックは必要ありません。
取り消し	ユーザは認証されません	それ以上の CRL または OCSP チェックは必要ありません。

CRL 証明書の検証結果	フェイルオーバー無効	フェイルオーバー有効
レスポンスなし/取得失敗	CDP 拡張が利用可能な場合、ポリシー サーバでは CRL の取得に成功するまで順序どおりに各配布ポイントを試します。証明書のステータスが有効または取り消しである場合、それらの状態の説明を参照します。 すべての理由コードを含む CRL が取得されない場合、ポリシー サーバはデフォルトで「Not Authenticated」になります。	CDP 拡張が利用可能な場合、ポリシー サーバでは CRL の取得に成功するまで順序どおりに各配布ポイントを試します。証明書のステータスが有効または取り消しである場合、それらの状態の説明を参照します。 すべての理由コードを含む CRL が取得されない場合、OCSP を使用します。

OCSP と CRL 間のフェイルオーバーの設定

ポリシー サーバでは OCSP と CRL を証明書の検証方法として使用し、両者の間でフェイルオーバーを有効にすることができます。フェイルオーバーを有効にする前に、管理 UI で CRL チェックを設定し、SMocsp.conf ファイルを作成することによって OCSP を設定します。フェイルオーバーが機能するためには、CRL チェックと OCSP チェックが有効である必要があります。

CRL への OCSP のフェイルオーバーを有効にする方法

1. テキスト エディタで SMocsp.conf ファイルを開きます。このファイルは、*policy_server_home/config* ディレクトリにあります。
2. 各レスポンスレコードについて以下のエントリを追加または変更します。

EnableFailover

SiteMinder でプライマリ検証方法からセカンダリ検証方法にフェイルオーバーできるようにします。

この値を **Yes** に設定すると、フェイルオーバーが有効になります。

フェイルオーバーを設定しない場合は、デフォルトの **No** のままにします。フェイルオーバーを設定せず、OCSP レスポンダが検証を実行できない場合、トランザクションは失敗します。

有効値: Yes または No

デフォルト: No

PrimaryValidationMethod

別の方法を試行する前に、どちらの証明書検証方法がまず使用されるかが示されます。

EnableFailover が YES に設定され、この設定の値が OCSP である場合、ポリシー サーバではまず OCSP 検証を使用します。OCSP レスポンダからのレスポンスがないか、レスポンス インジケータが「unknown」である場合は、CRL にフェイルオーバーします。

この設定の値が CRL である場合、OCSP 検証が設定されている場合でも無視され、CRL が使用されます。ポリシー サーバで CRL を取得できないか、CRL が失効している場合、OCSP にフェイルオーバーします。

有効値: OCSP、CRL

デフォルト: OCSP

3. SMocsp.conf ファイルの変更を保存します。
4. ポリシー サーバを再起動します。

証明書検証のトラブルシューティング

X.509 証明書の認証および検証の問題を解決するのに役立つよう詳細なトレースログ記録が提供されます。

一般的な OCSP および CRL のメッセージに加え、フェイルオーバー イベントが発生した場合は、ポリシー サーバで証明書検証の失敗に特化した診断メッセージがログ記録され、続いてフェイルオーバーを示すメッセージが記録されます。このメッセージによって、OCSP にアクセスできず CRL を使用していること、または CRL フェッチに失敗して OCSP チェックにフェイルオーバーしていることが示されます。

OCSP および CRL のログ メッセージを参照するには、ポリシー サーバ管理コンソールのプロファイラを使用して、認証トレースログ記録を有効にします。

トレースログに含まれるコンポーネントおよびデータ フィールドは、デフォルトのテンプレートファイル smtracedefault.txt を変更することによって指定できます。

以下の `smtracedefault.txt` ファイルでは、証明書検証の診断用にトレース ログ ファイルに含めることが望ましいコンポーネントを示しています。

```
components: Login_Logout/Authentication, Login_Logout/Certificates,  
Login_Logout/Receive_Request, IsAuthorized/Policy_Evaluation,  
IsAuthorized/Receive_Request, Directory_Access, LDAP/Ldap_Error_Messages
```

```
data: Date, PreciseTime, SrcFile, Function, ReturnValue,  
Message, User, Directory, SearchKey, ErrorString, ErrorValue,  
AuthStatus, AuthReason, CertSerial, SubjectDN, IssuerDN,  
CertDistPt, UserDN, Data, HexadecimalData, CallDetail, Returns, Result
```

認証トレース ログ記録を有効にする手順については、「ポリシー サーバ管理ガイド」を参照してください。

OCSP の署名の場合のみ、署名の検証でトレース メッセージを有効にすることができます。

OCSP 署名に対してトレースを有効にする方法

1. `policy_server_home/config` に移動します。
`policy_server_home` はポリシー サーバをインストールしたディレクトリです。
2. テキスト エディタで `JVMOptions` ファイルを開きます。
3. 以下のとおり、`-DOCSP_PS_TRACE` 設定を追加し、`true` に設定します。
`-DOCSP_PS_TRACE=true`
4. ファイルを保存します。
5. ポリシー サーバを再起動します。

`OcspCertKeyRetriever.log` という名前のトレース ファイルがポリシー サーバの現在の作業ディレクトリに以下のとおり書き込まれます。

Windows: `system32`

Unix: `siteminder` または `siteminder/bin`

第 11 章：強い認証

このセクションには、以下のトピックが含まれています。

[クレデンシャル セレクタの概要](#) (P. 441)

[クレデンシャル セレクタのユースケース](#) (P. 442)

[このユースケースのクレデンシャル セレクタ ソリューション](#) (P. 444)

[フロントエンド認証方式の確立](#) (P. 445)

[失敗した認証の管理](#) (P. 455)

[バックエンド処理のセットアップ](#) (P. 456)

[サンプル アプリケーションの保護](#) (P. 464)

[クレデンシャル セレクタ ソリューションのテスト](#) (P. 469)

クレデンシャル セレクタの概要

SiteMinder クレデンシャル セレクタは、SiteMinder の強力な認証ソリューションの 1 つです。ユーザはクレデンシャル セレクタによって、保護されているリソースにアクセスするのに必要な認証クレデンシャルのタイプを選択できます。ポリシー サーバは、ユーザの認証コンテキストに基づいて許可の判断を行い、それから、同じシングル サインオン環境でユーザ レスポンスを生成できます。

クレデンシャル セレクタ機能はスタンドアロン コンポーネントとして実装されており、SiteMinder により保護されたすべてのアプリケーションによって使用できます。

クレデンシャル セレクタのユースケース

このユースケースでは、ユーザが保護されているリソースへのアクセスを要求した場合に、異なるレベルのアクセスを提供するさまざまなクレデンシャルを選択できます。ユーザが保護されているサンプル アプリケーションを要求すると、以下のログイン ダイアログ ボックスが表示されます。

パスワードおよび/または認証 ユーザ名: パスワード: ☐ X.509クライアント認証を使用 ログイン	SecurID 認証 ユーザ名: SecurID PIN: ログイン
Windows 認証 ログイン	SafeWord 認証 ユーザ名: ログイン

ダイアログ ボックス上のログイン ボタンはそれぞれ異なるクレデンシャルを送信します。その後の表示や操作は、ユーザが提供するクレデンシャルのタイプによって異なります。ユーザは以下の認証タイプから選択できます。

- パスワード/証明書認証
- Windows 認証
- SecureID 認証
- SafeWord 認証

ユーザが正常に認証および許可されると、サンプル アプリケーションへのアクセスを許可され、サンプル アプリケーションは、ユーザがログインするのに使用した認証レベルおよび認証方式のタイプを通知するメッセージを表示します。

パスワード/証明書の認証によるアクセスのリクエスト

ログイン ダイアログ ボックスのパスワード/証明書セクションでは、クレデンシャルの以下の組み合わせのうちの 1 つを選択して提供できます。

- HTML フォームに入力されるユーザ名とパスワードのみ
- X.509 クライアント証明書のみ
- ユーザ名/パスワードおよび X.509 クライアント証明書

ユーザが有効なユーザ名およびパスワードのみを提供した場合、以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 5
You have used username/password authentication
```

X.509 クライアント証明書のチェック ボックスのみを選択した場合、ブラウザで設定されたクライアント証明書のうちの 1 つを選択するように求められます。ポリシー サーバがこれを認識すると、以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 10.
You have used X.509 client certificate authentication
```

パスワード/証明書オプションでは、ユーザが提供するクレデンシャルに応じて異なる認証レベルを提供できます。**SiteMinder** の **X.509** 証明書またはフォームによる認証方式は、パスワード/証明書オプションに似ていますが、提供されるクレデンシャルのタイプを区別しません。そのため、保護レベルはユーザが提供するクレデンシャルに関係なく同じになります。

ユーザ名とパスワードの両方が提供され、**X.509** クライアント証明書チェック ボックスがオンの場合、ユーザはクライアント証明書を求められます。ポリシー サーバが証明書を認識し、提供されたユーザ名と証明書が一致する場合、以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 15
You have used X.509 client certificate and username/password authentication
```

Windows 認証によるアクセスのリクエスト

ユーザが **Windows** ドメインにログイン中の場合、保護されているリソースを要求すると以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 5
You have used the Windows domain authentication
```

Windows ドメインにログインしていない場合、**Windows** ドメイン クレデンシャルを要求されます。

SecurID 認証によるアクセスのリクエスト

SecurID 認証に有効なユーザ名 SecurID PIN を提供した場合、保護されているリソースを要求すると以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 20
You have used the SecurID authentication
```

SafeWord 認証によるアクセスのリクエスト

SafeWord 認証にユーザ名のみを提供した場合、2 段階の処理が発生します。SiteMinder がユーザ名を SafeWord サーバに渡し、サーバはユーザの認証に使用するクレデンシャルを決定します。SafeWord は 1 回のログインにつき 4 つまでの認証子に対応します。認証子は固定(パスワードを使用)または動的(トークン カード PIN を使用)のどちらでもかまいません。

正常にアクセスすると、以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 20
You have used the SafeWord authentication
```

このユースケースのクレデンシャル セレクタ ソリューション

このユースケースのためにクレデンシャル セレクタをセットアップするには、以下のコンポーネントを設定します。

- フロントエンド認証方式が使用し、保護されているサンプル アプリケーションをユーザが要求する場合に表示されるログイン ダイアログ ボックスをレンダリングするフォーム認証情報コレクタ (FCC)。selectlogin.fcc という名前のサンプル FCC が、SiteMinder Web エージェントのインストールに含まれています。
- 管理 UI には以下が含まれます。
 - クレデンシャル セレクタを使用するアプリケーションに公開されるフロントエンド認証方式。
 - 複数のバックエンド認証方式(ユーザが選択できるクレデンシャルの各タイプごとに 1 種類の認証方式)。バックエンド処理とは、クレデンシャル セレクタのみが対話する機能です。これらの機能はエンドユーザには認識されません。

- 特別に設定されたポリシードメインにはレルム、ルール、レスポンス、およびポリシーが含まれ、クレデンシャル セレクタのバックエンド処理を表します。
- バックエンド処理を表すポリシードメインのエントリ ポイントとなる **Web** エージェント。
- フロントエンド認証方式を使用するサンプル アプリケーション。このユースケースについては、サンプル アプリケーションはユーザにあいさつのメッセージを表示します。このメッセージは、ログイン時にユーザが選択するクレデンシャルに応じて変わります。

フロントエンド認証方式の確立

フロントエンド認証方式では、フォーム クレデンシャル コレクタ (FCC)、つまり `selectlogin.fcc` を使用して、保護されたリソースへのアクセスをリクエストする際に使用するログイン選択画面を生成します。FCC は **Web** エージェント用に FCC 命令文を動的に構築するため、エージェントは認証方式の選択に合わせてユーザをリダイレクトできます。

この `selectlogin.fcc` はクレデンシャル セレクタ用のサンプルである点に注意してください。認証の選択肢と HTML 形式の組み合わせは状況によって異なります。

注: FCC の詳細については、このガイドの認証方式に関する章、または「**Web** エージェント設定ガイド」を参照してください。

フォーム認証情報コレクタ(FCC)の使用

FCC 形式は、SiteMinder Web エージェントが使用する独自の形式であり、クレデンシャルを収集してポリシー サーバに渡します。FCC は、ヘッダと本文で構成されます。

FCC ヘッダの説明

FCC のヘッダは FCC ディレクティブのリストであり、1 行当たりディレクティブを 1 つ含みます。FCC ディレクティブの構文は以下のとおりです。

@<ディレクティブ> [= <値>]

値には置換文字 % が含まれている場合があります、この場合、**%parameter%** といった形式となります。このパラメータはクレデンシャルが送信される場合に、FCC のファイルと共に POST アクションで渡されます。

FCC ディレクティブのセット数は限られています。この例の場合、最も重要なディレクティブは以下のとおりです。

@target

Web エージェントがポリシー サーバに渡す必要のあるリソース URL

@username

Web エージェントがポリシー サーバに渡す必要のあるユーザ名

@password

Web エージェントがポリシー サーバに渡す必要のあるパスワード

@smagentname

Web エージェントがポリシー サーバに渡す必要のあるエージェント名。
エージェントの設定の **EncryptAgentName** パラメータが「はい」に設定されている場合、名前は暗号化されます。

ヘッダには FCC ディレクティブをすべて表示する必要はありません。以下のとおり、多くのディレクティブには暗黙のデフォルトがあります。

- @target=%target%
- @username=%username%
- @password=%password%
- @smagentname=%smagentname%

FCC 本文の説明

FCC の本文には、HTML または他の Web ブラウザが判読可能な形式が含まれます。ユーザが認証情報を要求される場合に Web ブラウザでレンダリングされます。

本文には置換文字が含まれている場合があります、この場合、`$$parameter$$` といった形式となります。パラメータ名は、GET アクションで FCC のファイルと共に渡される既知のパラメータの特定のセットに属する必要があります。

この例の場合、重要なディレクティブは以下のとおりです。

`$$target$$`

ユーザが要求したリソース URL

`$$smagentname$$`

Web エージェントの名前。エージェントの設定の `EncryptAgentName` パラメータが「はい」に設定されている場合、名前は暗号化されます。

フロント エンド認証のための `selectlogin.fcc` ファイルの設定

`selectlogin.fcc` ファイルは、サンプル FCC として Web エージェントのインストールに含まれています。このファイルはフロントエンド認証方式により使用され、保護されているリソースをユーザが要求する場合に表示されるログイン ダイアログボックスをレンダリングします。

ユーザがリソースを要求すると、Web エージェントは要求された URL をポリシーサーバに渡します。ほとんどの場合、Web エージェントが渡すリソース URL はユーザが要求するものと同じです。SiteMinder 認証方式のために定義された FCC ファイルは、以下のとおり `%target%` (POST パラメータ) として `$$target$$` (GET パラメータ) を渡し、`@target=%target%` の命令を使用することにより、要求された URL が送信されるようにします。

```
<!-- some HTML code -->
<form name="Login" method="POST">
<!-- some HTML code -->
<input type="hidden" name="target" value="$$target$$">
<!-- more HTML code -->
</form>
<!-- more HTML code -->
```

注: 省略されている場合は、`@target=%target%` 命令がデフォルトで使用されます。

この場合、`selectlogin.fcc` ファイルは `%target%` パラメータの値を以下の値に置き換えることによって動作します。

```
/path/redirect.ext?authtype=type&target=$$target$$
```

`redirect.ext`

ターゲット パラメータで提供される URL にリダイレクトする単純なスクリプト。例の値は、`redirect.asp` または `redirect.jsp` です。また、リダイレクト スクリプトの URL が提供されるクレデンシャルに依存する限り同じ物理ファイルを公開する異なるリダイレクト スクリプト ファイルまたは仮想ディレクトリを使用することもできます。

`type`

ユーザのクレデンシャルの選択によって決定される文字列。

異なるリダイレクト URL が異なる認証方式によって保護される場合、FCC によって収集されたクレデンシャルは選択された認証方式によって処理されます。これは、ユーザの認証レベルを含むユーザのセッションを確立する認証方式です。ユーザがリダイレクト スクリプト リソースを認証および許可されると、最初に要求したリソースにリダイレクトされます。

注: シングル サインオンが有効であり、ユーザの保護レベルがフロントエンド認証方式の保護レベルと同等か、またはこれより高い場合、元のリソースに対してユーザのセッションが認証されます。ユーザが許可されるかどうかは、ユーザの認証コンテキストを確認するポリシーの設定に依存します。たとえば、特定のリソースにアクセスするには、最小限の保護レベルまたは特定の条件が必要な場合があります。

Selectlogin.fcc の設定の詳細

`selectlogin.fcc` ファイルでは、さまざまな認証方式を設定できます。次に、一部の方式のための設定の詳細を示します。

- 証明書とフォーム方式は、SSL 接続によるポスティングを必要とします。フロントエンド認証方式が SSL 接続を使用しない場合は、Web エージェントは GET リクエストで取得したのと同じ `selectlogin.fcc` URL を POST できません。

以下の JavaScript コードを使用して、URL を SSL URL に変換できます。

```
arr = document.URL.split("://");  
document.Login.action = "https://" + arr[1];
```

- SafeWord の 2 段階の認証に対応するには、最初のフォームの認証要求で取得したユーザ名を、2 回目の認証要求までクライアント側で記憶しておく必要があります。Web エージェントのインストールにも含まれる `safeword.fcc` ファイルは、`@smtransient FCC` の命令を使用してユーザ名を一時的 Cookie に保持します。selectlogin.fcc ファイルでも同じ命令を使用できますが、ユーザが異なるクレデンシャルを選択した場合でも Cookie が作成されます。以下のとおり、`safeword.fcc` ファイルへの POST の action 引数を変更することをお勧めします。

```
document.Login.action = "safeword.fcc";
```

- Windows 認証方式は FCC ファイルを使用しません。特定の擬似リソース URL を使用します。この URL は Web サーバには存在しませんが、SiteMinder Web エージェントによって認識されます。Windows 認証を動作させるには、以下のとおり、この同じ擬似リソース URL に action 引数を設定します。

```
document.Login.action = "/siteminderagent/ntlm/creds.ntc";
```

- SecurID 認証方式は FCC を使用しませんが、エージェントは selectlogin.fcc ファイルに SecurID クレデンシャルを直接 POST できます。action 引数の変更は必要ありません。
- アクション URL は別の Web サーバによってホストされる場合がありますが、この Web サーバも、SiteMinder 固有のリソース URL (FCC、SCC、および NTC) が認識されて正常に処理されるよう、SiteMinder Web エージェントにより保護される必要があります。

注: SCC の擬似リソース URL は証明書のための認証に使用されます。

サンプル selectlogin.fcc ファイル

簡略化された selectlogin.fcc ファイル (HTML のフォーマットなし) を次に示します。smquerydata および postpreservationdata の非表示の入力フィールドが含まれます。これらのフィールドは、それぞれ GET および POST のパラメータを渡すのに必要です。

smauthreason パラメータは、ポリシー サーバが提供する理由コードおよび認証要求を保持します。

サンプル selectlogin.fcc ファイルは以下のとおりです。

```
@username=%USER%
@smretries=0

<html>
<head>
  <script language="JavaScript">
    function submitForm(form)
    {
      authtype = "none";

      if (form == 1)
      {
        document.Login.USER.value      = document.Login.USER1.value;
        document.Login.PASSWORD.value = document.Login.PASSWORD1.value;

        if (!document.Login.UseCert.checked)
        {
          // username/password only
          authtype = "form";
        }
        else if (document.Login.USER.value == "" &&
                 document.Login.PASSWORD.value == "")
        {
          // certificate only
          authtype = "cert";
        }
        else
        {
          // username/password and certificate
          authtype = "certform";

          // This option requires posting over SSL.
          arr = document.URL.split("://");
          document.Login.action = "https://" + arr[1];
        }
      }
    }
  </script>
</head>
<body>
```

```
else if (form == 2)
{
    // SecurID authentication
    authtype = "securid";
    document.Login.USER.value = document.Login.USER2.value;
    document.Login.PASSWORD.value = document.Login.PASSWORD2.value;
}
else if (form == 3)
{
    // Safeword authentication
    authtype = "safeword";
    document.Login.USER.value = document.Login.USER3.value;
    document.Login.PASSWORD.value = "";

    // POST to safeword.fcc, for additional processing.
    // NOTE: This forces the web agent to POST to safeword.fcc
    // even if the authentication scheme's URL parameter
    // is set to selectlogin.fcc for redirection purposes.
    document.Login.action = "safeword.fcc";
}
else if (form == 4)
{
    // Authenticate with the current Windows login credentials
    authtype = "windows";
    document.Login.USER.value = "";
    document.Login.PASSWORD.value = "";

    // POST to creds.ntc (required by the Windows authentication scheme).
    document.Login.action = "/siteminderagent/ntlm/creds.ntc";
}
// Generate the target, depending on the user's choice of credentials.
// This sample uses redirect.asp, but it could also be redirect.jsp, redirect.pl,
etc.
// This sample uses the following format:
/auth/redirect.asp?authtype=<choice>&target=<original target>
// Other formats are also possible, e.g.:
/auth-<choice>/redirect.asp?target=<original
    target>
// The helper realms' resource filters must be defined accordingly (see the tech
note).
```

```
// Check if the target is not already in the same format. The user may
// have been redirected back to selectlogin.fcc upon authentication failure,
// if the authentication scheme's URL parameter is set to selectlogin.fcc.
if ("$$target$$.indexOf("/auth/redirect.asp?authtype=") == 0 &&
    "$$target$$.indexOf("&target=") > 0)
{
    // This must be a redirect. Extract the original target, but not
    // the authtype parameter, because the user may have made a different
    // choice of credentials this time.
    trgarr = "$$target$$.split("&target=");
    document.Login.target.value = "/auth/redirect.asp?authtype=" + authtype +
"&target=" + trgarr[1];
}
else
{
    // This is not a redirect. Pass $$target$$ as a URL query parameter.
    document.Login.target.value = "/auth/redirect.asp?authtype=" + authtype +
"&target=$$target$$";
}

document.Login.submit();
}
function resetCredFields()
{
    document.Login.PASSWORD.value = "";
    document.Login.PASSWORD1.value = "";
    document.Login.PASSWORD2.value = "";
}
</script>
</head>

<body onLoad="resetCredFields();">
<center>
<form name="Login" method="POST">
    <input type="hidden" name="USER">
    <input type="hidden" name="PASSWORD">
    <input type="hidden" name="smagentname" value="$$smagentname$$">
    <input type="hidden" name="smauthreason" value="$$smauthreason$$">
    <input type="hidden" name="smquerydata" value="$$smquerydata$$">
    <input type="hidden" name="postpreservationdata"
value="$$postpreservationdata$$">
    <input type="hidden" name="target">
```



```
<!-- Some table formatting throughout -->
<!-- Authentication Choice: Password And/Or Certificate -->
  <input type="text"      name="USER1">
  <input type="password" name="PASSWORD1">
  <input type="button"   value="Login" onClick="submitForm(1);">

<!-- Authentication Choice: Windows Authentication -->
  <input type="button"   value="Login" onClick="submitForm(4);">

<!-- Authentication Choice: SecurID Authentication -->
  <input type="text"      name="USER2">
  <input type="password" name="PASSWORD2">
  <input type="button"   value="Login" onClick="submitForm(2);">

<!-- Authentication Choice: SafeWord Authentication -->
  <input type="text"      name="USER3">
  <input type="button"   value="Login" onClick="submitForm(3);">
<!-- More table formatting -->
</form>
</center>
</body>
</html>
```

フロントエンド認証方式の設定

グリーティングを生成するサンプル アプリケーションを保護するフロントエンド認証方式を設定する必要があります。これは、**AuthChannel** 認証方式を設定することで解決できます。

AuthChannel 認証方式の例については、以下の図を参照してください。

一般	
名前: 認証チャンネル	説明: 認証方式:HTMLフォーム
各方式共通セットアップ	
認証方式タイプ: HTML フォーム テンプレート 保護レベル: 1 [1	
☒ この認証方式に対してパスワード ポリシーを有効にする	
方式のセットアップ	
☐ 関連目標を使用 Web サーバ名: auth.sample.com ポート: 6565 ☐ SSL 接続を使用 ターゲット: /siteminderagent/forms/login.fcc ☐ この認証方式にクレデンシャルの保存を許可	

AuthChannel 認証方式は以下のように設定されます。

認証タイプのスタイル

HTML フォーム テンプレート

保護レベル

1

フロントエンド認証方式は、構成内の他の方式よりも保護レベルを低くする必要があるので、AuthChannel 認証方式の保護レベルは 1 に設定します。ユーザの実際の保護レベルは、保護されたリソースにアクセスしようとログインする際に選択する認証方式によって決まります。フロントエンド認証方式の保護レベルは低いため、最初にリクエストしたリソースにリダイレクトされるときにユーザが再度確認されることはありません。

Web サーバ名

auth.sample.com

これは、サンプル アプリケーションが置かれている Web サーバです。

ターゲット

`/siteminderagent/forms/selectlogin.fcc`

このターゲットは `selectlogin.fcc` ファイルを指しています。`selectlogin.fcc` ファイルは Web エージェントのインストールに付属しているサンプル ファイルです。

失敗した認証の管理

認証方式がユーザを拒否すると、このユーザは、その認証方式のターゲット パラメータで指定された URL にリダイレクトされます (この認証方式でこのパラメータを使用できる場合)。

以下の中から、ユーザの状況に最適な動作を設定します。

- 認証方式の選択に戻るのではなく、ユーザに同じ認証方式のクレデンシアルを再度送信するよう要求します。
- ユーザが元のログイン画面に戻って再度選択できるようにします。これを行うには、各認証方式のターゲット パラメータとして `selectlogin.fcc` を必要に応じて設定する必要があります。

クレデンシアルの特定の選択で `selectlogin.fcc` ファイル以外の FCC のファイルにクレデンシアルをポストする必要がある場合は、`selectlogin.fcc` で HTML フォームのアクション パラメータを希望する FCC のファイルの URL に設定します。たとえば、以下のコマンドはアクション パラメータを `safeword.fcc` ファイルに設定します。

```
document.Login.action = "safeword.fcc";
```

基本認証方式のような、ターゲット パラメータのない方式を使用している場合、正常な認証でのユーザの操作は同じです。ただし、ユーザが再度認証を求められる場合は、この再認証は基本認証方式に基づきます。つまり、HTML フォームではなくプロンプトダイアログ ボックスを使用します。

注: SafeWord の基本方式は SafeWord HTML フォーム方式と異なり、2 段階の認証および複数の認証子に対応しません。

バックエンド処理のセットアップ

クレデンシアル セレクタを使用するには、バックエンド処理用に以下のコンポーネントが必要です。

- ログイン クレデンシアルの選択肢ごとに 1 つの認証方式
- バックエンド コンポーネントを含むポリシードメイン
- 各認証方式を使用するレルム
- レルムごとのルール
- 認証コンテキストを設定し、リソース アクセスを認可するポリシー

バックエンド処理で使用する認証方式のセットアップ

ユーザが使用できるように選択したクレデンシアルごとに 1 つずつ、バックエンド処理用の認証方式をセットアップする必要があります。これら複数の方式をセットアップすることで、ユーザは保護されたリソースにアクセスする際に、使用するクレデンシアルのタイプを選択できるようになります。

クレデンシアルのタイプごとに、認証方式は異なります。

- HTML フォーム認証方式
- X.509 クライアント証明書認証方式
- X.509 クライアント証明書およびフォーム認証方式
- SecurID HTML フォーム認証方式
- SafeWord HTML フォーム認証方式
- Windows 認証方式

注: 基本認証方式は、ポリシー サーバのインストールの一部としてセットアップされているデフォルトの方式です。

バックエンド処理のためのポリシードメインのセットアップ

クレデンシアル セレクタのバックエンド処理は、ポリシードメインによって表されます。このソリューションでは、ポリシードメインの名前は **BackendAuth** です。

バックエンド ポリシードメインのレルムの設定

設定されているバックエンド認証方式ごとにレルムが 1 つずつあります。各レルムには、以下のようにリソースフィルタを定義する必要があります。

`/auth/redirect.asp?authtype=type&target=`
`type`

以下のいずれかを指定できます。

`form` ユーザ名/パスワード認証
`cert` 証明書認証
`certform` 証明書とフォーム認証
`securid` SecurID 認証
`safeword` SafeWord 認証
`windows` Windows 認証

注: これらのタイプは、このユースケースで選択されるものです。これら以外の値もありますが、このタイプは、`selectlogin.fcc` ファイル、または `selectlogin.fcc` テンプレートに基づいたその他の FCC ファイルにある認証タイプの値と一致している必要があります。また、レルムのリソースフィルタは FCC ファイルのリダイレクトターゲットとも一致している必要があります。

以下のペインにレルムが一覧されます。

ドメインの表示: 認証バックエンド

一般	レルム	ポリシー	レスポンス	ルール グループ	変数
レルム					
名前		説明			
	SafeWord	 SafeWord			
	フォーム	 フォーム			
	Windows	 Windows			
	証明書	 証明書			
	SecurID	 SecurID			
	証明およびフォーム	 証明およびフォーム			

レルムを保護する Web エージェントは、1 つの場合と複数の場合があります。このソリューションでは 1 つの Web エージェントを使用しますが、複数の Web エージェントを使用する場合はそれらのエージェントが特定の要件を満たしている必要があります。

クレデンシャル セレクタ機能の一部としてレルムを保護する Web エージェントに必要な要件は、以下のとおりです。

- レルムを保護するすべての Web エージェント間に、シングル サインオンを設定する必要があります。つまり、すべてのエージェントが同じ cookie ドメインにある、または同じ cookie プロバイダを使用していることを意味します。

注: シングル サインオンを設定するには、「Web エージェント設定ガイド」を参照してください。

- FCC ファイルの `@smagentname` ディレクティブの値は、最初にリクエストされたリソースを保護する Web エージェントの少なくとも 1 つの予測値と一致する必要があります。

この予測値は、Web エージェントの `AgentName` パラメータ、または `DefaultAgentName` パラメータ (`AgentName` パラメータに値が割り当てられていない場合) の値です。エージェントの設定の `EncryptAgentName` パラメータが「はい」に設定されている場合、値は暗号化する必要があります。

`@smagentname` ディレクティブを設定する方法の 1 つに、同じネーミングプロパティで各 Web エージェントを設定する方法があります。これらのエージェントは、さらに同じエージェント設定オブジェクトも共有できます。名前を暗号化しないのであれば、FCC ファイルで `@smagentname` ディレクティブをプログラマ的に設定する方法もあります。

重要: `@smagentname` ディレクティブを間違えて設定すると、ポリシー サーバログに「リクエストで受信したレルムはありません」という内容のエラーメッセージが表示されることがあります。

- `FCCForcelsProtected` パラメータを `yes` に設定し、`selectlogin.fcc` ファイルによって生成された新しいターゲットに対して 2 つ目の `IsProtected` コールが必ず作成されるようにします。 `IgnoreQueryData` パラメータは `no` に設定し、Web エージェントが URL クエリ パラメータを無視しないようにする必要があります。

バックエンド ポリシー ドメインのルール作成

BackendAuth ポリシー ドメインとして設定する各レルムには、2 つのルールを設定する必要があります。

- OnAuthAccept ルール
- Web エージェント アクション ルール (この例では、GET アクション ルールを使用します)

どちらのルールも、[リソース]フィールドの値はアスタリスク(*)になります。

たとえば、BackendAuth ポリシー ドメインのフォーム レルムの場合、ルールは以下のようになります。

OnAuthAccept

Resource: /auth/redirect.asp?authtype=form&target=*

Action: OnAuthAccept

GetRule

Resource: /auth/redirect.asp?authtype=form&target=*

Action: Get

バックエンド ポリシー ドメインの AuthContext レスポンスの設定

AuthContext レスポンスは BackendAuth ドメインの認証方式ごとに設定します。これらの各レスポンスには、AuthContext レスポンス属性が含まれており、OnAuthAccept イベントでのみ評価されます。その値は SM_AUTHENTICATIONCONTEXT ユーザ属性の値として SiteMinder セッション チケットに追加されます。ただし、ユーザ レスポンスとしてクライアントに返されることはありません。

この例の場合、レスポンスのリストは以下のようになります。

エージェントのタイプ 説明		
名前		
Form	Web エージェント	ユーザ名/パスワード認証用の AuthContext
Certificate	Web エージェント	証明書認証用の AuthContext
CertandForm	Web エージェント	証明書とフォーム認証用の AuthContext

SecurID	Web エージェント	SecurID 認証用の AuthContext
SafeWord	Web エージェント	SafeWord 認証用の AuthContext
Windows	Web エージェント	Windows 認証用の AuthContext

注: レスポンス属性値は、長さが 80 バイトになるように切り捨てられます。

AuthContext レスポンス属性を設定するには、
WebAgent-OnAuthAccept-Session-AuthContext レスポンス属性タイプを選択します。

以下の図は、WebAgent-OnAuthAccept-Session-AuthContext 属性タイプを使用した AuthContext レスポンス属性の作成を示します。

属性タイプ

属性 WebAgent-OnAuthAccept-Session-AuthContext

値タイプ Text

属性のセットアップ

属性の種類

☒ スタティック
☐ ユーザ属性
☐ DN 属性
☐ アクティブ レスポンス
☐ 変数定義
ネストされたグループの許可 ☐

属性フィールド

● 変数名
● 変数値
HTTP 変数名 HTTP_

詳細

スクリプト

図にあるように、AuthContext レスポンス属性タイプはスタティックです。
Federation セキュリティ サービスが使用されている場合、スタティック属性を指定して定数またはリテラル値を定義すればカプセル化を高めることができます。
定数値には文字列も含まれます。

SiteMinder 変数とアクティブな式により、AuthContext レスポンス属性を設定する際の柔軟性が増します。また、これらには認証タイムスタンプや SAML アサーションのハッシュ値を含めることもできます。

以下のグループボックスは、このソリューションに設定されている結果レスポンスの 1 つを示します。これはフォーム レスポンスの属性です。

ドメイン 認証バックエンド

属性リスト		
	エージェント タイプ 属性名	値
	WebAgent-OnAuthAccept-Session-AuthContext	user / password 

バックエンド クレデンシャル選択のポリシーの設定

この例では、BackendAuth ドメインに以下の 2 つのポリシーがあります。

- ユーザの認証コンテキストを設定するポリシー
- Web エージェントアクションのポリシー

以下の図は 2 つのポリシーを示します。

一般
レルム
ポリシー
レスポンス
ルール グループ
変数

ポリシー

名前	説明
 ポリシー取得	要求されるリソースへの認証ポリシー 
 認証コンテキストポリシー	認証コンテキストポリシーの設定 

作成

認証コンテキスト ポリシーの作成

AuthContext ポリシーは SiteMinder セッション チケットに認証コンテキストを設定します。これは認証と検証に使用されます。このポリシーでは、各レルムの OnAuthAccept ルールを対応するレスポンスとペアにして、保護されたリソースへのアクセスを許可します。たとえば、フォーム レルムの OnAuthAccept ルールはフォーム レスポンスとペアになり、SafeWord レルムの OnAuthAccept ルールは SafeWord レスポンスとペアになります。

ユーザ認証とユーザ検証は OnAuthAccept イベントであるため、セッション チケットの認証コンテキストは検証のたびに上書きされる場合があります。認証コンテキスト属性を更新する機能は、その属性の値にカウンタが含まれている場合などは非常に便利です。ただし、クレデンシャル セレクタを使用するこのソリューションでは、AuthContext ポリシーは認証コンテキストが空の場合にのみ起動するように設定されています。これにより、セッション チケットは上書きされず、ユーザが選択したクレデンシャルを記憶しておくことができます。

認証コンテキストは上書きから保護する必要があります。保護するには、セッション チケットから SM_AUTHENTICATIONCONTEXT 属性を取得するアクティブな式を AuthContext ポリシーで記述します。

Federation セキュリティ サービスが使用されている場合、AuthContext と呼ばれるユーザ コンテキスト変数を作成します。この変数を AuthContext ポリシーで使用するれば、セッション チケットから SM_AUTHENTICATIONCONTEXT 属性を取得するアクティブな式を定義できます。

一般	
名前:	認証コンテキスト
変数のタイプ:	ユーザ コンテキスト
説明:	

定義	
戻り型	文字列
項目	ユーザ プロパティ
プロパティ	SM-AUTHENTICATIONCONTEXT
バッファ サイズ	80

AuthContext ポリシーで、AuthContext 変数を使ってアクティブな式を定義します。

一般

ユーザ

ルール

式

式

式 (AuthContext=="")

編集

クリア

保護ポリシーの作成

リクエストされたリソースに対して認証済みユーザを許可するには、BackendAuth ドメインにもう 1 つポリシーが必要になります。このポリシーはリソースの保護を目的とし、このドメインの認証コンテキスト ポリシーに新たに加えられるものです。

この保護ポリシーは、リダイレクション ターゲットに対して認証済みユーザを許可します。これは、redirect.asp ファイルの GET アクションで実行します。ポリシーに GetPolicy という名前を付けます。

GetPolicy には、以下の関連ルールが含まれています。

ルール	レルム
GetRule	Form
GetRule	Certificate
GetRule	CertandForm
GetRule	SecureID
GetRule	SafeWord
GetRule	Windows

サンプル アプリケーションの保護

クレデンシャル セレクタを使用するには、SiteMinder アプリケーションはコンポーネントのフロントエンド認証方式で保護されているレلمを設定する必要があります。アプリケーションを保護するポリシーには、ユーザの認証レベルおよび認証コンテキストに基づく制限がある場合があります。

このソリューションでは、サンプル アプリケーションは認証および許可されたユーザ向けのあいさつメッセージを生成します。

このメッセージを生成するファイルには以下のコードが含まれます。

```
<html>
<head></head>
<body>

<h3>
<p>Greetings, <%=Request.ServerVariables("HTTP_USERNAME") %>!
<p>Your authentication level is <%=Request.ServerVariables("HTTP_AUTHLEVEL") %>
<p>You have used <%=Request.ServerVariables("HTTP_AUTHCONTEXT") %>
authentication
</h3>

</body>
</html>
```

ログイン ダイアログ ボックスでの認証オプションによって、以下の各アクセスレベルおよびメッセージが使用されます。

```
Greetings, SampleUser!
Your authentication level is 5
You have used username/password authentication
```

```
Greetings, SampleUser!
Your authentication level is 10.
You have used X.509 client certificate authentication
```

```
Greetings, SampleUser!
Your authentication level is 5
You have used Windows domain authentication
```

サンプル アプリケーションには、異なるレلمに含まれる 4 種類のリソースがあります。それぞれのレلمにフロントエンド認証方式を設定する必要があります。

サンプル アプリケーションのレルムとルール

このサンプルでは、サンプル アプリケーションを構成するさまざまなリソースを以下の 4 つのレルムで保護しています。

- パブリックリソースを含むレルム
- 少なくともレベル 5 の認証を必要とするリソースを含むレルム
- 少なくともレベル 15 の認証を必要とするリソースを含むレルム
- SafeWord で認証されたユーザにのみ利用可能なリソースを含むレルム

レルムは以下ようになります。

ドメインの変更: サンプルアプリ

一般		
レルム		
ポリシー		
レスポンス		
ルール グループ		
変数		
レルム		
名前		説明
サンプルアプリ		
制限		SafeWord認証ユーザの制限
保護		認証レベル5以上
パブリック		保護されていないリソース

以下に示すように、保護されたレルムは AuthChannel フロントエンド認証方式に設定されます。

一般	
名前:	保護済み
ドメイン:	サンプルアプリ

リソース	
エージェント	sampleagent 2
リソース フィルタ	/protected/
有効なリソース	sampleagent/protected/
デフォルト リソース保護	☒ 保護 ☐ 保護なし
認証方式	AuthChannel

SampleAgentGroup は、サンプル アプリケーションへのエントリ ポイントを提供する Web エージェントを含めることができます。

クレデンシャル セレクタ機能の一部としてレルムを保護する Web エージェントに必要な要件は、以下のとおりです。

- レルムを保護するすべての Web エージェント間に、シングル サインオンを設定する必要があります。つまり、すべてのエージェントが同じ cookie ドメインにある、または同じ cookie プロバイダを使用していることを意味します。

注: シングル サインオンを設定するには、「Web エージェント設定ガイド」を参照してください。

- FCC ファイルの `@smagentname` ディレクティブの値は、最初にリクエストされたリソースを保護する Web エージェントの少なくとも 1 つの予測値と一致する必要があります。

この予測値は、Web エージェントの `AgentName` パラメータ、または `DefaultAgentName` パラメータ (`AgentName` パラメータに値が割り当てられていない場合) の値です。エージェントの設定の `EncryptAgentName` パラメータが「はい」に設定されている場合、値は暗号化する必要があります。

`@smagentname` ディレクティブを設定する方法の 1 つに、同じネーミングプロパティで各 Web エージェントを設定する方法があります。これらのエージェントは、さらに同じエージェント設定オブジェクトも共有できます。名前を暗号化しないのであれば、FCC ファイルで `@smagentname` ディレクティブをプログラマ的に設定する方法もあります。

重要: `@smagentname` ディレクティブを間違えて設定すると、ポリシー サーバログに「リクエストで受信したレルムはありません」という内容のエラーメッセージが表示されることがあります。

- `FCCForcelsProtected` パラメータを `yes` に設定し、`selectlogin.fcc` ファイルによって生成された新しいターゲットに対して 2 つ目の `IsProtected` コールが必ず作成されるようにします。 `IgnoreQueryData` パラメータは `no` に設定し、Web エージェントが URL クエリパラメータを無視しないようにする必要があります。

サンプル アプリケーションを保護するポリシーのルール

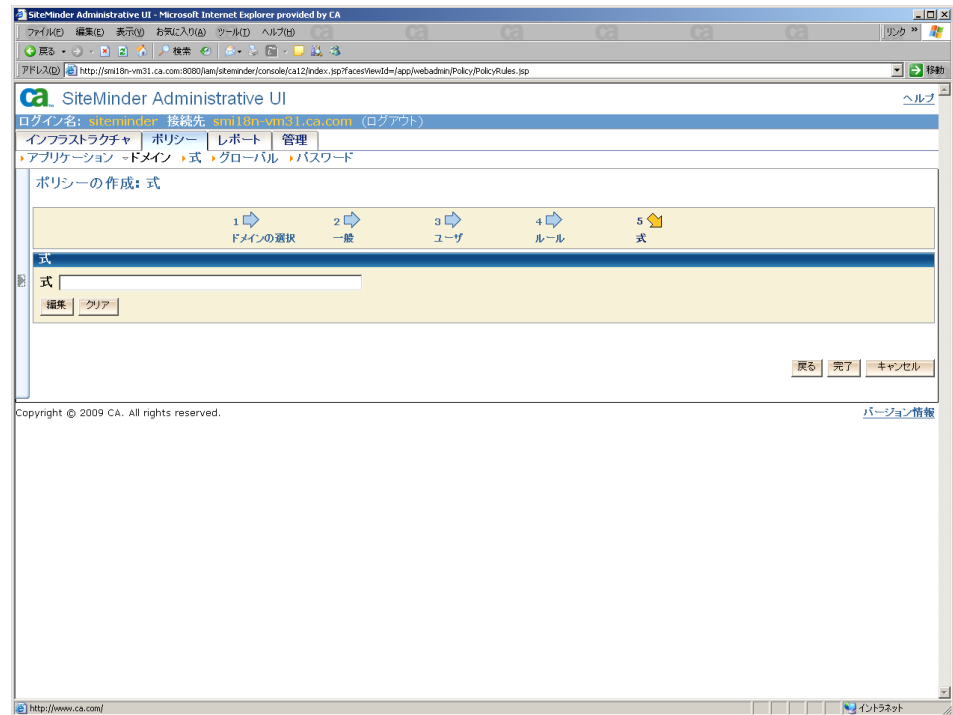
サンプル アプリケーションリソースを含むレルムには、どのようなタイプのルールでも設定できます。

サンプル アプリケーションを保護するポリシーの設定

`GetProtected` ポリシーでは、保護されたリソースへのアクセスに 5 以上の保護レベルを必要とします。この保護レベル制限を適用するには、`GetProtected` ポリシーで、`SiteMinder` セッション チケットから `SM_AUTHENTICATIONLEVEL` 属性を取得するアクティブな式を記述します。

注: この認証レベル制限は、レベル 1 のパスワード認証しかサポートしていないカスタム Web エージェントからアプリケーションを保護できるように設計されています。

Federation セキュリティ サービスが使用されている場合、AuthLevel と呼ばれるユーザ コンテキスト変数を作成します。この変数を GetProtected ポリシーで使用すれば、セッション チケットから SM_AUTHENTICATIONLEVEL 属性を取得するアクティブな式を定義できます。



サンプル アプリケーションのレスポンスの設定

この例で、挨拶メッセージを表示するサンプル アプリケーションは 3 つの HTTP ヘッダ変数を使用します。

- HTTP_USERNAME
- HTTP_AUTHLEVEL
- HTTP_AUTHCONTEXT

これらのヘッダは、以下のレスポンスと一緒にクライアントに返されます。

一般	
名前: サンプルレスポンス	説明: ユーザ名、認証レベル、認証
ドメイン: マイドメイン1	

属性	
エージェント タイプ	
☐ RADIUS ☒ SiteMinder エージェント タイプ Web Agent	

属性リスト		
Agent Type	Attribute Name	Value
WebAgent-HTTP-Header-Variable	UserName=	<%userattr="SM_USERSESSIONUNIVID"%>
WebAgent-HTTP-Header-Variable	AuthContext=	<%userattr="SM_AUTHENTICATIONCONTEXT"%>
WebAgent-HTTP-Header-Variable	AUTHLEVEL=	<%userattr="SM_AUTHENTICATIONLEVEL"%>

属性値は[レスポンス属性]ペインで指定します。

クレデンシャル セレクタ ソリューションのテスト

このユース ケースで説明されているさまざまなコンポーネントをセットアップしてから、クレデンシャル セレクタの機能をテストできます。このテストでは、ユーザが指定するクレデンシャルに基づいてさまざまな挨拶が表示されます。

クレデンシャル セレクタをテストする方法

1. リンクをクリックする方法で、サンプル アプリケーションにアクセスしてみます。

selectlogin.fcc ファイルで定義されているログイン画面が表示されます。

2. 1 つのタイプのクレデンシャルを入力してログインします。

入力したクレデンシャルに合わせて初期画面が表示されます。たとえば、ユーザ名と SecurID PIN を入力した場合は、以下のメッセージが表示されます。

```
Greetings, SampleUser!
Your authentication level is 20
You have used the SecurID authentication
```

3. アプリケーションを終了し、再度サンプル アプリケーションにアクセスしてみます。
4. 前の手順で入力したのとは違うクレデンシャルのセットを入力します。
指定したクレデンシャルの挨拶メッセージが表示されます。

クレデンシャル セレクタのテストが問題なく終了します。

第 12 章: ドメイン

このセクションには、以下のトピックが含まれています。

[ポリシードメインの概要](#) (P. 471)

[ドメインとユーザ メンバシップ](#) (P. 473)

[ポリシードメインの設定方法](#) (P. 473)

[ドメインのグローバル ポリシー処理の無効化](#) (P. 478)

[アフィリエイトドメイン](#) (P. 479)

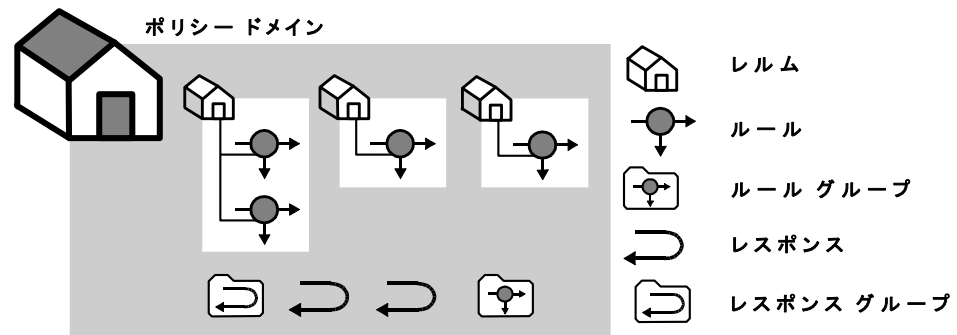
[ドメインの変更](#) (P. 479)

[ドメインの削除](#) (P. 479)

ポリシードメインの概要

ポリシードメインは、1 つ以上のユーザ ディレクトリに関連したリソースの論理グループです。またポリシードメインでは、そのポリシードメイン内のオブジェクトを変更できる管理者アカウントが 1 つ以上必要です。ポリシードメインには、レルム、ルール、レスポンス、ポリシー（および必要に応じてルールグループとレスポンスグループ）が含まれています。適切な権限を持つ管理者が、1 人以上の管理者にポリシードメインを割り当てます。管理者権限の詳細については、「ポリシー サーバ管理」を参照してください。

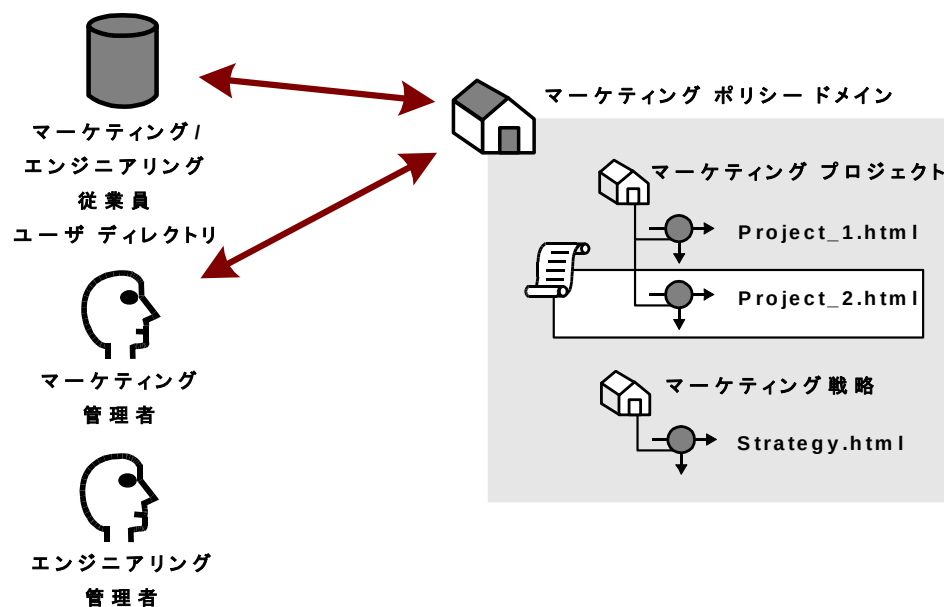
ポリシードメイン内のリソースは、1 つ以上のレルムにグループ化することができます。レルムとは、共通のセキュリティ(認証)要件を持つリソースのセットです。リソースへのアクセスは、そのリソースを含むレルムに関連するルールによって制御されています。次の図は、レルムとそれに関連したルール、およびルールグループ、レスポンスグループ、レスポンスのペアを含む小規模なポリシードメインを示しています。



レールムとルールをポリシードメインにグループ化することで、組織に安全なリソースのドメインを作ることが可能になります。ポリシーに関するセクションで、ポリシードメイン内にポリシーを作成して、ポリシードメインのリソースへのアクセスを制御する方法について説明します。

次のサンプル図に、マーケティング ポリシードメインで指定されたマーケティング ポリシー管理者が、マーケティング戦略とマーケティング プロジェクトのレールムを管理する場合を示します。このポリシードメインでは、マーケティング ポリシードメインの管理権限を持たないエンジニアリング管理者が、マーケティング部に属するリソースを制御することはできません。ただし、マーケティング ポリシードメインはエンジニアリング ユーザを含むユーザ ディレクトリに関連しています。

マーケティング部の管理者が、エンジニアリング スタッフがリソース **Project 2.html** にアクセスできるようなポリシーをマーケティング ポリシードメイン内に作成すると、エンジニアリング ユーザはそのリソースにアクセスできるようになります。



ドメインとユーザ メンバシップ

ポリシードメインは、ドメイン オブジェクトの格納場所として機能する以外に、ユーザ ディレクトリへの接続も行います。ポリシー サーバは、ターゲットリソースがあるレلمの要件に基づいてユーザを認証します。ユーザを認証するには、ユーザが定義されているユーザディレクトリをポリシー サーバで見つける必要があります。ポリシー サーバは、レلمが属するポリシードメインを探してユーザディレクトリを見つけます。ポリシー サーバは、ポリシードメインから、ポリシードメインの検索順に従って指定されたユーザ ディレクトリを照会します。

検索順は、ポリシードメインにユーザディレクトリ接続を追加するときに定義します。ディレクトリ接続を追加する順序によって、ポリシー サーバがユーザを検索する順序が決定されます。たとえば、Windows NT ディレクトリから LDAP ディレクトリにユーザのデータを移行する企業のポリシードメインを設定した場合に、ポリシー サーバでの検索を、最初に新しい LDAP ディレクトリ、次に Windows NT ユーザディレクトリという順序で行うようにするには、まず LDAP ディレクトリ接続をポリシードメインに追加し、その後 Windows NT ユーザディレクトリ接続を追加します。

ポリシードメインの設定方法

ドメインを設定して、1 つ以上のユーザ ディレクトリを含むリソースの論理グループを作成します。ドメインの設定では以下の操作が必要です。

- ユーザ認証用にユーザ ディレクトリを 1 つ以上割り当てます。
- レلمを 1 つ以上作成し、セキュリティポリシーに基づいてリソースをグループ化します。

注: 将来、レلمを追加する必要がある場合は、ポリシードメインのプロパティを編集できます。

次に、新しいポリシードメインを設定するための手順を示します。

1. ポリシードメインの設定
2. ユーザ ディレクトリの割り当て
3. レلمの作成

ポリシードメインの設定

ポリシードメインを作成することで、リソースの論理的なグループを保護することができます。

ポリシードメインを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ドメイン]-[ドメインの作成]をクリックします。

[ドメインの作成]ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. [一般]グループ ボックスの各フィールドに、ポリシーの名前と説明を入力します。
4. ユーザ ディレクトリとレルムを追加します。
5. [サブミット]をクリックします。

ドメインの作成タスクが、処理のためにサブミットされます。

ユーザ ディレクトリの割り当て

ポリシードメインには、1 つまたは複数のユーザ ディレクトリを追加できます。ポリシー サーバは、ユーザ ディレクトリに格納されるクレデンシャルにユーザが入力するクレデンシャルを比較することにより、ユーザを認証します。ポリシー サーバは、ポリシードメインで示されるのと同じ順序でユーザ ディレクトリを検索します。

ポリシードメインにユーザ ディレクトリを追加する方法

1. [ドメイン]ウィンドウの[ユーザ ディレクトリ]で、[追加/削除]をクリックします。

[ユーザ ディレクトリの選択]ウィンドウが開きます。

2. [使用可能なメンバー]リストから 1 つまたは複数のユーザ ディレクトリを選択し、右向きの矢印をクリックします。

[使用可能なメンバー]リストからユーザ ディレクトリが削除され、[メンバーの選択]リストに追加されます。

注: 一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。

3. [OK]をクリックします。

選択したユーザ ディレクトリがドメインに追加されます。

注: 新規ユーザ ディレクトリを作成してドメインに追加するには、[ドメイン]ウィンドウの[ユーザ ディレクトリ]グループ ボックスで[新規...]をクリックします。

レルムの作成

レルムは、ドメイン内に作成され、**Web** エージェントに関連付けられます。レルムは、リソースフィルタを使用して、同様のセキュリティ要件を持ち、共通の認証方式を共有するリソースをグループ化します。

SiteMinder Web エージェントのレルムの設定

ドメインを作成するときに、ドメインに 1 つ以上のレルムを作成し、それらのレルムを **SiteMinderWeb** エージェントまたはエージェント グループに関連付けることができます。レルムによって、同様のセキュリティ要件を持ち、共通の認証方式を共有するリソースのグループ化を行います。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ドメインにレルムを作成して SiteMinder Web エージェントまたはエージェント グループに関連付ける方法

1. [ポリシー]-[ドメイン]-[レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択]ペインが開きます。
2. ドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]が開きます。
3. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
4. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。

5. SiteMinder Web エージェントまたはエージェント グループを選択し、[OK] をクリックします。
6. [リソース]グループ ボックスで、残りのリソース プロパティを指定します。
7. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
8. 新しいサブ レルムを作成するか、または[サブ レルム]グループ ボックス上の既存のサブ レルムを削除します。
9. [セッション]グループ ボックスで、セッション プロパティを指定します。
10. [詳細]グループ ボックスで、レルムが処理するイベントの登録方式、許可 ディレクトリ マッピング、およびタイプを指定します。
11. [完了]をクリックします。

レルムの作成タスクが、処理のためにサブミットされます。

RADIUS エージェントのレルムの設定

ドメインを作成するときに、ドメインに 1 つ以上のレルムを作成し、それらのレルムを RADIUS エージェントまたはエージェント グループに関連付けることができます。レルムによって、同様のセキュリティ要件を持ち、共通の認証方式を共有するリソースのグループ化を行います。

注: 管理 UI を使用して、RADIUS エージェントで保護されたレルムを設定することができます。これらのレルムでは、SiteMinder Web エージェントのレルムに必要な情報と同じ情報がすべてが必要とは限りません。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ドメインにレルムを作成して RADIUS エージェントまたはエージェント グループに関連付ける方法

1. [ポリシー]-[ドメイン]-[レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択]ペインが開きます。
2. ドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]ペインが開きます。
3. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。

4. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。
5. **RADIUS** エージェントまたはエージェント グループを選択し、[OK]をクリックします。
6. [リソース]グループ ボックスで、残りのリソース プロパティを指定します。
7. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
8. [セッション]グループ ボックスで、セッション プロパティを指定します。
9. [完了]をクリックします。
レルムの作成タスクが、処理のためにサブミットされます。

4.x アフィリエイト エージェントのレルムの設定

ドメインを作成するときに、ドメインに 1 つ以上のレルムを作成し、それらのレルムをアフィリエイト エージェントまたはエージェント グループに関連付けることができます。レルムによって、同様のセキュリティ要件を持ち、共通の認証方式を共有するリソースのグループ化を行います。

注: アフィリエイト エージェントは、**Federation** セキュリティ サービスの管理ユーザ インターフェースを使用して作成する必要があります。アフィリエイト エージェントの詳細については、「*Federation Security Services Guide*」を参照してください。

注: 4.x アフィリエイト エージェントと **SAML** アフィリエイト エージェントを混同しないでください。**Web** アプリケーションがユーザの操作性を **Federation** セキュリティ サービスの一部としてパーソナライズできるように、**SAML** アフィリエイト エージェントはユーザ情報を **Web** サーバに渡します。4.x アフィリエイト エージェントはリソースを保護します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ドメインにレルムを作成してアフィリエイト エージェントまたはエージェント グループに関連付ける方法

1. [ポリシー]-[ドメイン]-[レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択] ペインが開きます。
2. ドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]が開きます。
3. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
4. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。
5. アフィリエイト エージェントまたはエージェント グループを選択し、[OK]をクリックします。

注: Web サーバでアフィリエイト エージェントが設定されている場合は、[リソース]グループ ボックス上の残りのプロパティが指定されます。詳細については、「Web エージェント設定ガイド」を参照してください。
6. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
7. [完了]をクリックします。
レルムの作成タスクが、処理のためにサブミットされます。

ドメインのグローバル ポリシー処理の無効化

グローバル ポリシーを使用することで、すべてのドメイン全体でレスポンスを特定のリソースとイベントに関連付けることができます。デフォルトでは、グローバルポリシーはポリシードメイン内のすべてのリソースに適用されます。

特定のドメインのグローバル ポリシーを無効にする方法

1. ドメインを開きます。
2. [グローバル ポリシーを適用]チェック ボックスをオフにし、[サブミット]をクリックします。

これで、グローバル ポリシーはこのドメイン内のリソースに適用されなくなりました。

アフィリエイトドメイン

SiteMinder Federation セキュリティ サービスを購入済みの場合、Federation セキュリティ サービス管理ユーザ インターフェースを使用できます。アフィリエイトドメインは SAML 1.x Consumers または、1 つ以上のユーザ ディレクトリに関連付けられた SAML 2.0 Service Providers の論理グループです。

注: 詳細については、「*Federation Security Services Guide*」を参照してください。

ドメインの変更

ポリシーまたはアフィリエイトドメインに関連する名前、説明、ユーザディレクトリ接続、管理者は、変更することができます。この他のドメインの設定項目はすべて、他のオブジェクトでの設定が反映された結果であるため、変更はできません。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects](#) (P. 61)」を参照してください。

ドメインの削除

重要: ドメインを削除すると、ドメイン ユーザ ディレクトリおよび管理者の接続とオブジェクト(ルール、ルールグループ、レルム、レスポンス、レスポンスグループ、およびポリシー、またはドメインに含まれているアフィリエイト)がすべて破棄されます。

削除されたオブジェクトがすべてキャッシュからクリアされるまで、少し時間がかかる場合があります。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects](#) (P. 61)」を参照してください。

第 13 章: レルム

このセクションには、以下のトピックが含まれています。

[レルムの概要](#) (P. 481)

[エージェント、レルム、およびルールによるリソースの識別](#) (P. 482)

[ネストされたレルム](#) (P. 485)

[リクエスト処理におけるレルム](#) (P. 487)

[レルムの設定](#) (P. 487)

[レルムの変更](#) (P. 491)

[レルムの削除](#) (P. 491)

[ネストされたレルムの設定](#) (P. 492)

[リソース キャッシュからのレルムのクリア](#) (P. 493)

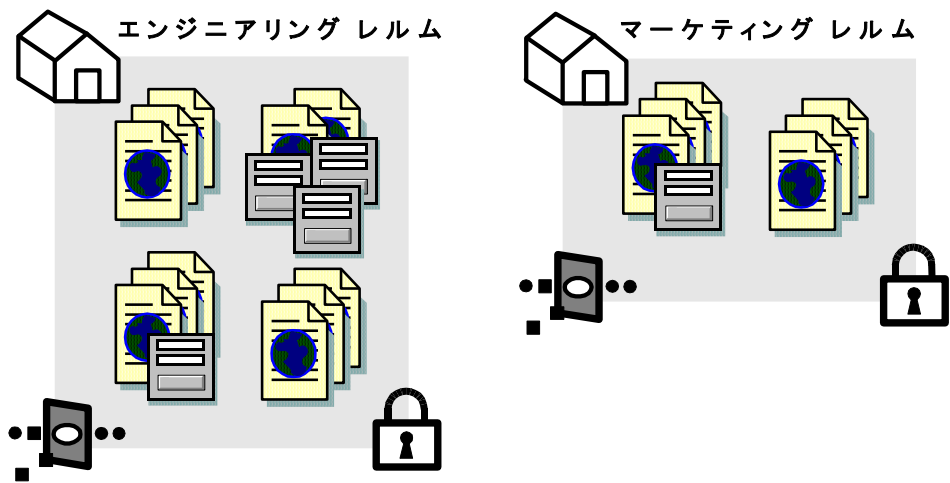
レルムの概要

セキュリティ ポリシーを作成できるように、複雑なリソース セットは論理的にグループ化する必要があります。SiteMinder のリソースの基本的なグループ化は、レルムです。

レルムとは、セキュリティ要件に従ってグループ化された、ポリシードメイン内のリソースのクラスターです。通常は、ネットワーク上の共通の場所に存在するリソースに対して定義されます。たとえば、ネットワーク上の `/marketing` ディレクトリにあるマーケティング情報は、マーケティング組織の管理者が制御するポリシードメイン内のレルムとして設定することができます。

レルムのコンテンツは、エージェントによって保護されます。ユーザがレルム内のリソースを要求すると、関連するエージェントがユーザの認証と許可を処理します。認証方法はレルムによって決まります。

次の図に、2 つのレルムの内容を示します。



それぞれのレルムには、HTML ファイル、フォーム、アプリケーションなどのリソースが含まれます。また、各レルムはエージェントおよび認証方式に関連付けられています。

エージェント、レルム、およびルールによるリソースの識別

SiteMinder で保護されているリソースは、以下で識別されます。

[エージェント][レルムリソースフィルタ][ルールリソース]

エージェント

エージェントは、保護されたリソースのレルムを 1 つ以上含むサーバを監視します。

レルム リソース フィルタ

ディレクトリの相対パスなど、レルムでカバーされるリソースを指定する文字列。最上位レルムの場合には、ファイルまたはアプリケーションのホストサーバに対して相対的なリソースを指定します。 ネストされたレルムの場合には、親レルムに対して相対的なリソースを指定します。

たとえば、次のような Web サーバのドキュメントルートの直下にあるディレクトリの内容がレルムに含まれているとします。

```
<document_root>/HR
```

この場合には、次のようにレルムリソースフィルタを指定します。

```
/HR
```

ルール リソース

ルールが適用されるリソースを指定する文字列または正規表現。リソースが含まれるレルムに対して相対的なリソースを指定します。たとえば、レルムリソースフィルタがディレクトリ名で終わる場合、ルールリソースにはレルムディレクトリのサブディレクトリだけでなく、そのサブディレクトリに含まれる次のようなファイルの名前も含まれることがあります。

`/Managers/PayRanges.html`

ワイルドカードを使用してルールの指定範囲を拡大できます。以下に例を示します。

`/Managers/*`

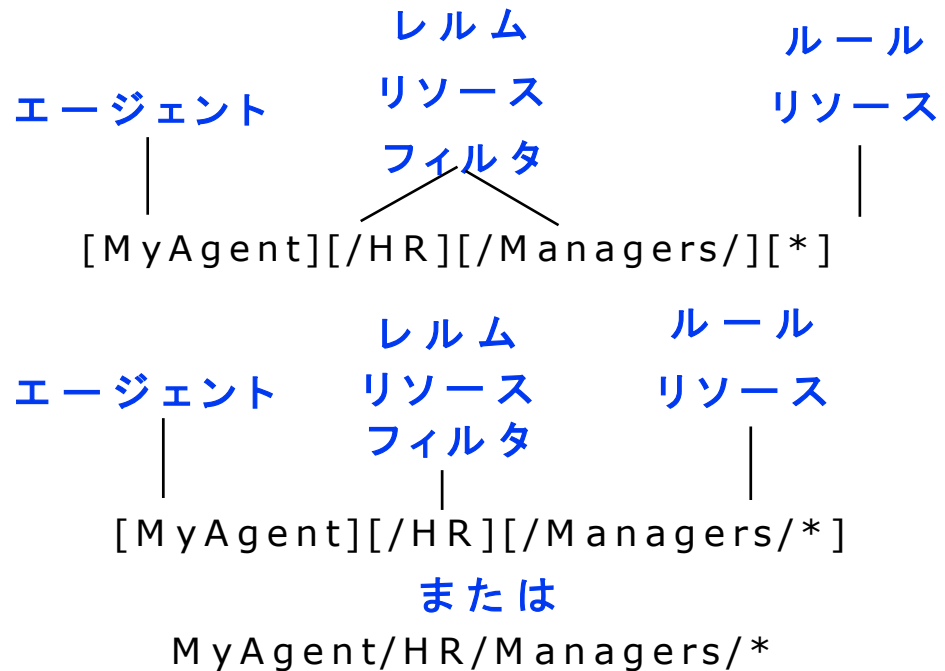
この 3 つの要素を使用して、次のような例を考えてみましょう。

- **MyAgent** というエージェントが、**myorg.org** というドメイン内で **MyHost** というホスト上の **Web** サーバを保護しているとします。
- このレルムに次のような **Web** サーバディレクトリを含めるとします。

`<document_root>/HR`

- **HR** ディレクトリには、**Managers** というサブディレクトリがあり、このサブディレクトリ内のすべてのファイルを保護するとします。

ポリシー サーバに対して有効なリソースを、以下の図に示します。



Managers というディレクトリを **/HR** レルム下にネストされたディレクトリとして設定することができます。

Managers サブディレクトリ下の保護されたページ **PayRanges.html** にアクセスするには、ユーザは以下の手順に従う必要があります。

1. 次のようにリソースを指定します。

`http://MyHost.myorg.org/HR/Managers/PayRanges.html`

2. リソースへのアクセスが許可されているユーザのクレデンシャルを入力します。管理者はポリシーを使用して、リソースへのアクセスが許可されるユーザを指定します。

保護されていないレルム、ルールおよびポリシー

デフォルトでは、レルムは保護状態で作成されます。ほとんどの場合は、レルムを保護なしの状態に変更しないで、保護されたレルムを使用してください。保護されたレルムでは、すべてのリソースがアクセスから保護されています。アクセスを許可するには、ルールを定義してからポリシーに設定する必要があります。

保護なしの状態でレルムを作成した場合は、ルールを設定してから **SiteMinder** でレルム内のリソースを保護する必要があります。保護されていないレルムのリソースにルールを作成すると、指定したリソースのみが保護されます。リソースを保護したら、ポリシーにルールを追加して、ユーザにリソースへのアクセスを許可する必要があります。レルム内のリソースのあるサブセットのみを無許可のアクセスから保護する必要がある場合には、保護されていないレルムを使用することができます。

次の表に、保護なしのレルムを設定する際に必要なアクションの例を示します。

アクション	保護の状態
[リソースフィルタ]を /dir に設定して、 Realm1 という保護されていないレルムを作成します。	/dir 自身と、その下位のリソースとサブディレクトリは保護されません。
Realm1 の次のリソースに Rule1 を作成します。 file.html .	ファイル /dir/file.html は保護されますが、 /dir のその他の内容は保護されません。

Policy1 を作成し、Rule1 と User1 をそのポリシーにバインドします。 User1 は /dir/file.html にアクセスすることができます。他のすべてのユーザは、保護されたファイルにアクセスできません。

注: 匿名認証方式を使用してレルムのユーザを追跡する場合、そのレルムは保護されている必要があります。匿名認証方式の詳細については、「[匿名認証方式](#) (P. 397)」を参照してください。

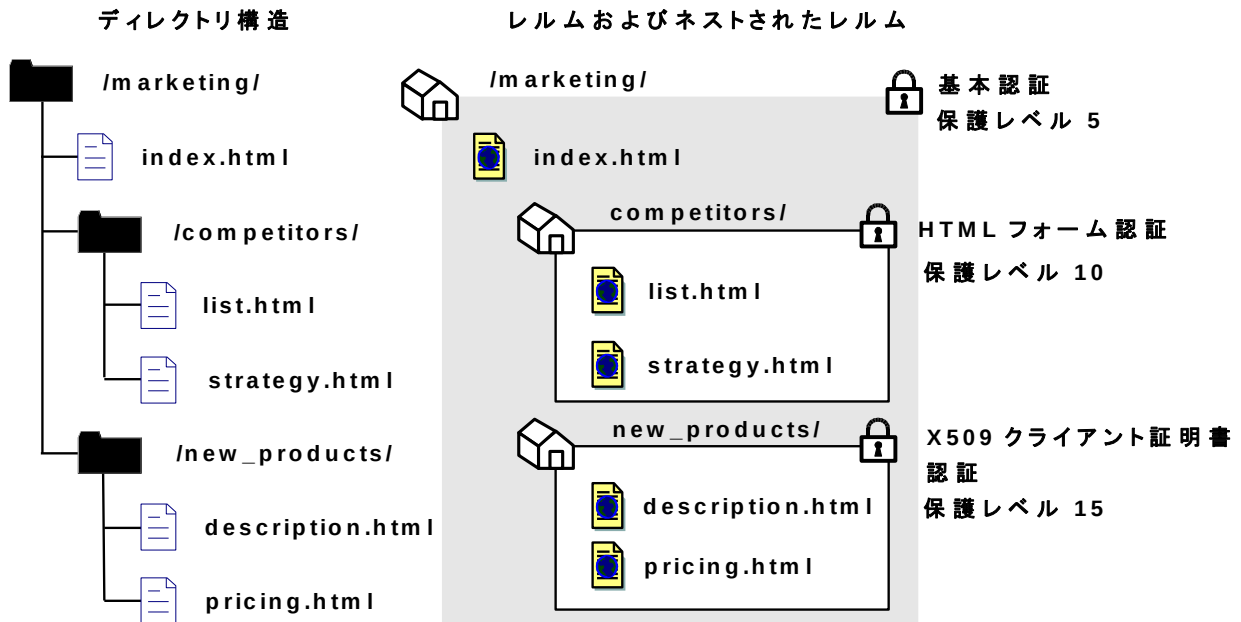
ネストされたレルム

SiteMinder では、ファイルやフォルダのディレクトリがファイル システムの内容を表すのと同様に、レルムはリソースのグループを表します。レルムをネストすると、ディレクトリ ツリーの下位レベルにあるリソースの保護レベルを高めることができます。ネストされたレルムとは、既存の任意のレルム内に作成できます。この後で、より高い保護レベルが設定された認証方式をネストされたレルムに割り当てます。

デフォルトでは、子レルム内のリソースにアクセスするには、ユーザは親レルムと子レルムの両方のリソースについて認証を受ける必要があります。ポリシー サーバのデフォルト動作はグローバルに変更できます。これにより、親レルムまたは子レルムのどちらかについて認証されたユーザは常に子レルムのリソースへのアクセスが許可されるように設定できます。ただし、AND ロジックを OR ロジックに変更すると、セキュリティが低下するため、ロジックの変更はお勧めできません。OR ロジックに変更するには、[ネストされたセキュリティを有効にする]チェックボックスをオフにします。

注: 最上位レルムを含め、ネスト構造のレルムには匿名認証レルムを割り当てないでください。匿名認証方式で保護されたリソースについて特定のユーザを認証することはできないため、AND ロジックは失敗します。

次の例は、ネストされたレルムを使用してセキュリティを強化する方法を示します。



上図を見ればわかるように、レルムはリソースのファイル構造に似ています。ネストされた各レルムでは、親レルムとは異なる認証方式が使用されています。各子レルムの認証方式の保護レベルは、親レルムよりの保護レベルよりも高いため、ユーザがツリーの下位レベルにあるリソースにアクセスするには、再度認証を受けなければなりません。この例を実装するには、レルムごとにルールを作成する必要があります。その上で、各ポリシーにルールを含めて、対応するポリシーを作成します。こうすることで、子レルム内のリソースにアクセスする必要があるユーザは、親レルム内のリソースにもアクセスできるようになります。

注: システムとドメイン オブジェクトの管理権限を持つ管理者のみが、レルムの作成、編集、削除を行えます。ただし、ドメイン オブジェクトの管理権限を持つ管理者は、それぞれのポリシードメイン内にあるレルムの下で、ネストされたレルムの作成、編集、削除を行うことができます。

リクエスト処理におけるレルム

ユーザがリソースを要求すると、ポリシー サーバは、一致する最長のレルムを使用して、リソースが保護されているかどうかを判別し、保護されている場合はユーザの識別情報を確立するために使用する必要がある認証方式を決定します。一致する最も長いレルムは、要求されたリソースのパスに一致するネストされたレルムのグループのうち、最も深いレベルに存在するリソースフィルタで構成されます（ネストされたレルムが使用されていない場合は単体のレルム）。

例

前のセクションの例を使用すると、`/marketing/competitors/list2.html` のファイル `list2.html` は、ネストされたレルム `/marketing/competitors/` と一致します。ポリシー サーバが `list2.html` の認証を処理する場合、ユーザの認証は HTML フォームを介して行われます。これは、HTML フォームが、`/marketing/competitors/ realm` に関連付けられている認証方式であるためです。

同じ例で、`current_budget.html` というファイルは、`/marketing/budgets/current_budget.html` にあります。`/budget` ディレクトリがネストされたレルムで明確に呼び出されることはないため、このリソースで一致する最も長いレルムは `/marketing/` になります。したがって、ユーザは基本認証方式のユーザ名とパスワードによって認証されます。

レルムの設定

レルムとは、ネットワーク上の特定の場所のリソースをグループ化したものです。レルムのコンテンツは、エージェントによって保護されます。ユーザがレルム内のリソースを要求すると、関連するエージェントがユーザの認証と許可を処理します。レルムは、認証方法を指定します。

アフィリエイト エージェントを含むすべてのタイプの SiteMinder エージェントについて、レルムを設定できます。

SiteMinder Web エージェントで保護されたレルムの設定

レルムを設定して、ユーザが **Web** サーバ経由でアクセスするリソースのグループを保護します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

レルムを設定する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択]ペインが表示されます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]ペインが表示されます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
5. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。
6. SiteMinder Web エージェントまたはエージェント グループを選択し、[OK]をクリックします。
7. [リソース]グループ ボックスで、残りのリソース プロパティを指定します。
8. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
9. 新しいサブレルムを作成するか、または[サブレルム]グループ ボックス上の既存のサブレルムを削除します。
10. [セッション]グループ ボックスで、セッション プロパティを指定します。
11. [詳細]グループ ボックスで、レルムが処理するイベントの登録方式、許可ディレクトリ マッピング、およびタイプを指定します。
12. [完了]をクリックします。
レルムの作成タスクが、処理のためにサブミットされます。

RADIUS エージェントにより保護されたレルムの設定

レルムを設定して、ユーザが **Web** サーバ経由でアクセスするリソースのグループを保護します。

注: 管理 UI を使用して、RADIUS エージェントで保護されたレルムを設定することができます。これらのレルムでは、SiteMinder Web エージェントのレルムに必要な情報と同じ情報がすべてが必要とは限りません。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

レルムを設定する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択]ペインが表示されます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]ペインが表示されます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
5. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。
6. RADIUS エージェントまたはエージェントグループを選択し、[OK]をクリックします。
7. [リソース]グループ ボックスで、残りのリソース プロパティを指定します。
8. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
9. [セッション]グループ ボックスで、セッション プロパティを指定します。
10. [完了]をクリックします。
レルムの作成タスクが、処理のためにサブミットされます。

4.x アフィリエイト エージェントのレルムの設定

このトピックはレガシー 4.x アフィリエイト エージェントについて説明しています。
4.x アフィリエイト エージェントには、アフィリエイトドメインへ追加できるアフィリエイト、サービスプロバイダ、またはリソース パートナーとの関係がありません。

4.x アフィリエイト エージェントは、アフィリエイトの **Web** サイトのリソースを保護しません。ただし、情報を 4.x アフィリエイト エージェントに渡すレスポンスを設定するには、レルムを設定する必要があります。4.x アフィリエイト エージェントのレルムの設定は **SiteMinder Web** エージェントを設定するより単純です。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

レルムを設定する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レルム]-[レルムの作成]をクリックします。
[レルムの作成: ドメインの選択]ペインが表示されます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[レルムの作成: レルムの定義]ペインが表示されます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
5. [リソース]グループ ボックス上の省略記号ボタンをクリックします。
[エージェントの選択]グループ ボックスが開きます。
6. アフィリエイト エージェントまたはエージェント グループを選択し、[OK]をクリックします。

注: Web サーバでアフィリエイト エージェントが設定されている場合は、[リソース]グループ ボックス上の残りのプロパティが指定されます。詳細については、「Web エージェント設定ガイド」を参照してください。

7. [ルール]グループ ボックスで新規ルールを作成するか、または既存のルールを削除します。
8. [完了]をクリックします。
レلمの作成タスクが、処理のためにサブミットされます。

レلمの変更

既存のレلمを変更する場合、以下の項目は変更できません。

- エージェント

既存のレلمが含まれているサーバを保護するエージェントは変更できません。エージェントを変更する必要がある場合は、レلمを削除し、新しいエージェントでレلمを作成し直してください。

- リソース フィルタ

既存のレلمのリソース フィルタは変更できません。リソース フィルタを変更する必要がある場合は、レلمを削除し、新しいリソース フィルタでレلمを作成し直してください。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

レلمの削除

レلمを削除すると、そのレلمに関連するネストされたレلمもすべて削除されます。さらに、削除されたレلمとそのネストされたレلمに関連付けられているルールもすべて削除されます。

削除されたオブジェクトがすべてキャッシュからクリアされるまで、少し時間がかかる場合があります。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

ネストされたレルムの設定

親レルムが管理者の範囲内のドメインに関連付けられている場合、ドメイン オブジェクトの管理権限を持つ管理者は親レルム内にネストされたレルムを作成できます。

注: SiteMinder Web エージェントが保護するレルムの下でのみ、ネストされたレルムを作成できます。

ネストされたレルムの作成方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レルム]-[レルムの変更]をクリックします。
[レルムの変更]ペインが開きます。
3. 検索条件を指定して[検索]をクリックします。
検索条件に一致するレルムのリストが表示されます。
4. レルムを選択し、[選択]をクリックします。
[レルムの変更: 名前]ウィンドウが開きます。
5. [サブレルム]グループ ボックスで、[サブレルム]をクリックします。
[レルムの作成]ウィンドウが開きます。
6. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[レルムの作成: 名前]ウィンドウが開きます。
7. [一般]グループ ボックスの各フィールドに、レルムの名前と説明を入力します。
8. [リソース]グループ ボックスの[リソース フィルタ]フィールドに、リソース フィルタのパスを入力します。
注: ネストされたレルムのリソース フィルタが、親レルムのリソース フィルタに追加されます。たとえば、親レルムのフィルタが `/marketing` であり、ネストされたレルムのフィルタが `/data` の場合、全体としてのフィルタは `<agent_of_the_parent_realm>/marketing/data` となります。
注: アスタリスク(*)および疑問符(?)記号は、リソース フィルタ内ではワイルドカードではなく文字列として扱われます。
9. [セッション]グループ ボックスでセッション プロパティを指定します。

10. [詳細]グループ ボックスで、以下の設定を指定します。

- 認可ディレクトリ マッピング
- イベントの処理

11. [サブミット]をクリックします。

レルムの作成タスクが、処理のためにサブミットされます。

リソース キャッシュからのレルムのクリア

SiteMinder では、ユーザが保護されたリソースにアクセスすると、レルムの情報がキャッシュされます。これにより、SiteMinder は最近使用されたリソースを追跡して、ネットワーク パフォーマンスを向上させることができます。しかし、セキュリティ要件やレルムの内容を変更した場合、SiteMinder のリソース キャッシュからレルムをクリアすることをお勧めします。

注: システムとドメイン オブジェクトの管理権限を持っている場合、[キャッシュ管理]ダイアログ ボックスを使用して、リソース キャッシュからすべてのレルムをクリアすることができます。より詳細な情報については、「ポリシー サーバ管理ガイド」を参照してください。

リソース キャッシュからレルムをクリアする方法

1. 管理 UI にログインします。
2. [ポリシー]タブをクリックします。
3. [ドメイン]-[レルム]-[レルムの変更]をクリックします。
[検索]ウィンドウが表示されます。
4. (オプション) ユーザの検索条件を絞り込むために検索フォームに入力します。
5. [検索]をクリックします。
レルムのリストが表示されます。
6. 希望するレルムの左側のラジオ ボタンをクリックし、次に、[選択]をクリックします。
[レルム: <レルム名>の変更]ウィンドウが表示されます。
7. [詳細]グループ ボックスで[クリア]をクリックします。
SiteMinder が、レルムをリソースキャッシュからクリアします。

第 14 章：ルール

このセクションには、以下のトピックが含まれています。

[ルールの概要](#) (P. 495)

[Web エージェントアクションのルールの設定](#) (P. 504)

[認証イベントアクション用のルールの設定](#) (P. 505)

[許可イベントアクションのルールの設定](#) (P. 506)

[インパーソネーション イベントアクションのルールの設定](#) (P. 508)

[アフィリエイト エージェントのルールの設定](#) (P. 509)

[リソース照合と正規表現](#) (P. 511)

[ルールの有効化および無効化](#) (P. 512)

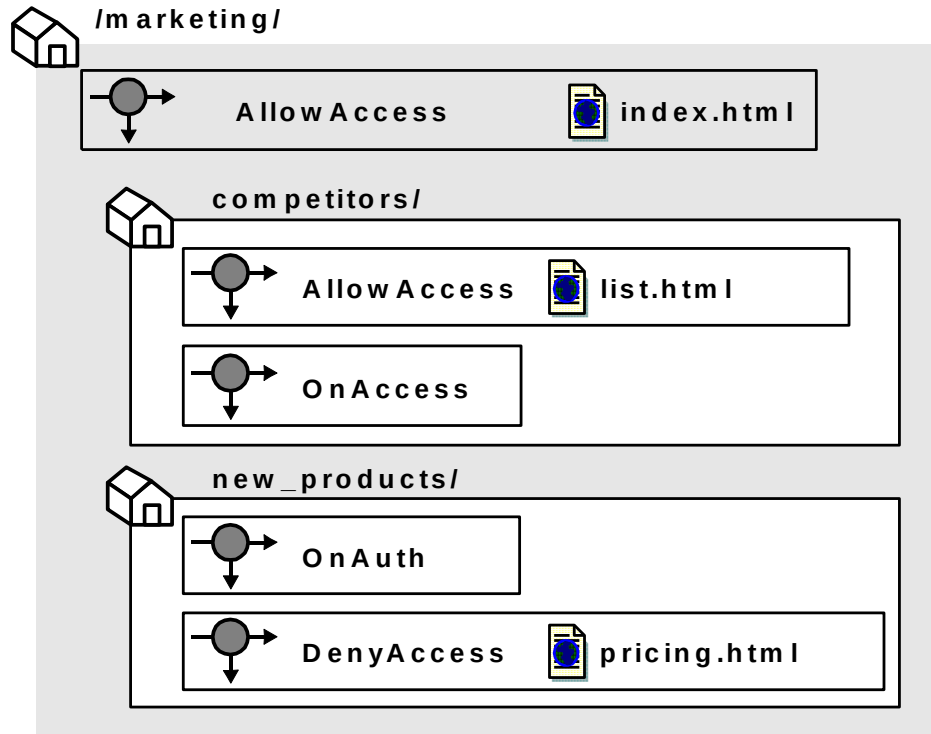
[高度なルール オプション](#) (P. 513)

[ルールの削除](#) (P. 515)

ルールの概要

SiteMinder ではルールを使用して、特定のリソースを識別したり、リソースに対するアクセスの許可/拒否を決定します。また、認証/許可イベントが発生したときに、レスポンスを発行することもできます。ルールを作成するには、そのルールを特定のレルムに関連付ける必要があります。

次の図は、複数のレルムとネストしたレルム、およびそれに関連したルールを示しています。



上の図では、個々のレルムやネストしたレルムに、それぞれのリソースに関連した特定のルールがあります。1つのルールを、レルム内にあるすべてのリソースまたはリソースのサブセットに関連付けることもできます。この処理には、リソース照合またはリソースを指定する正規表現を使用します。

ルールがポリシーの一部として動作する仕組み

ポリシーは、ルール、ユーザ、およびレスポンスを結び付けることで、リソースを保護します。ルールはポリシーの一部で、保護するリソースとルールを起動するアクションの種類を決定します。

たとえば、レルム内にある全 HTML ファイルをルールで指定し、そのファイルを GET アクションから保護することができます。Web サーバはこのルールを使って HTML ページに対するリクエストに応答します。ユーザがブラウザを介してリソースにアクセスしようとする、ルールが起動します。そのルールを含むポリシーによって、ユーザが選択したリソースを表示できるかどうかが決まります。

ポリシー サーバがルールを処理する方法

ポリシー サーバでは、ポリシーで定義されているユーザ、ルール、およびレスポンスの相互関係に従ってルールを評価します。ユーザが、保護されたリソースにアクセスする場合、ポリシー サーバは、ポリシーに含まれているルールを処理して、ユーザがそのリソースへのアクセスを許可されているか、認証イベントや許可イベントを処理する必要があるか、およびレスポンスを生成して SiteMinder エージェントに返す必要があるかを判別する必要があります。

ポリシー サーバは、許可イベントを処理する場合、保護されたリソースと一致する最長のリソースフィルタを使用してレلمを検索します。ポリシー サーバは、そのレلمに関連付けられたルールのみを起動します。この例で、ユーザはマネージャです。このマネージャは、以下の保護されたリソースにアクセスする必要があります。

`/company/employees/managers/performance/`

以下のレلمには、保護されたリソースと一致するリソース フィルタがあります。

レلم名	レلمの説明	リソース フィルタ
Company	顧客、従業員、ベンダー	<code>/company/</code>
Company Employees	すべての従業員	<code>/company/employees/</code>
Company Managers	すべてのマネージャ	<code>/company/employees/managers/</code>
Performance Management	マネージャとチーム メンバ	<code>/company/employees/managers/performance/</code>

一致するリソース フィルタが最長であるレلمは、Performance Management です。許可イベントに応じて、ポリシー サーバは、Performance Management レلمに関連付けられたすべてのルールを起動します。

レلمがネストされている場合には、ポリシー サーバは処理中に使用する一致する順番に並べたリストを保持します。一致するすべてのルールがリソースへのアクセスを拒否すると、処理は停止し、ポリシー サーバは、アクセス拒否ルールに関連付けられたレスポンスを SiteMinder エージェントに返します。

ポリシー サーバは、起動する一致ルールのすべてからレスポンスを収集します。ルールに基づいてレスポンスの収集を終了すると、重複しているレスポンスを削除します。

ネストされたレلمが使用されている環境では、ポリシー サーバは適合するルールに対するレスポンスをすべて集めたリストを作成します。**OnAuthAccept** ルールの場合、ポリシー サーバはレスポンスのリスト全体を **SiteMinder** エージェントに返します。**OnAuthReject** ルールの場合、ポリシー サーバは、最も深くネストされたレلم内のルールに関連付けられているレスポンスを **SiteMinder** エージェントに返します。**OnAuthReject** ルールは、ポリシーに関連付けられているユーザについてのみ起動します。

ルールとネストされたレلم

ネストされたレلمとは、既存のレلم内に作成されたレلمです。ネストされたレلمは親レلمか最上位レلمに属し、親レلمの子と見なされます。ネストされたレلمを作成すると、子レلم内のリソースを保護する別のルールも作成できます。また、既存のルールをコピーして、他のレلمにそのルールを適用したり、ルール名を変更したりできます。

ルール アクション

ルールのアクションにより、ルールが起動する条件が決定されます。ルールが起動するのは、ルールに指定されているアクションが発生したとポリシー サーバが判断したときです。ルールは、既存の有効なポリシーに含まれている必要があります。たとえば、ポリシーに **HTML** ページへのアクセスを許可するルールが含まれていて、そのポリシーが特定のディレクトリに存在するユーザを指定する場合に、そのディレクトリに一覧表示されているユーザの 1 人がリソースにアクセスしようすると、ポリシー サーバはリクエストを処理するためにルールを起動する必要があると判断します。

ルールが起動されると、ポリシー サーバは、ルールを含むポリシーの設定に基づいてルールで指定されたアクションを処理します。たとえば、ユーザがポリシーで指定されたグループに所属していないときは、**HTML** ページへの **HTTP Get** アクションを許可するルールにより、ユーザはリソースにアクセスできません。

Web エージェント アクション

Web エージェント アクションを含むルールは、ルールで指定された **HTTP** アクションのいずれかが発生したときに、ルールで指定されたリソースへのアクセスを許可または拒否します。

アクセス許可が指定されているルールが起動されたときに、ユーザの認証が成功すると、**SiteMinder** は、ユーザが指定リソースにアクセスすることを許可します。ルールでアクセス拒否が指定されている場合、**SiteMinder** は認証に成功したユーザのアクセスを拒否します。アクセス拒否ルールをポリシーに追加することで、リソースに対するアクセス権を持たない特定の個人やグループのアクセスを拒否することができ、更にセキュリティを高めることができます。アクセス許可がデフォルトです。

アクセス拒否ルールは、アクセス許可ルールより優先されます。ユーザがリソースにアクセスしようとしたときにアクセス拒否ルールとアクセス許可ルールが起動した場合には、アクセス拒否ルールが、すべてのアクセス許可ルールよりも優先されます。

Web エージェントのルール アクションは、以下のとおりです。

Get

HTTP を介して、参照するリソースを取得します。

Put

レガシー **HTTP** アクションをサポートします。

Post

ユーザによって提供された情報を、**HTTP** を介して通知します。

SOA エージェント アクション

CA SOA Security Manager を購入された場合は、2 つの追加の **Web** エージェントルール アクションが **SOA** エージェントで使用できるようになります。

ProcessSOAP

SOAP エンベロープでラップされた着信 **XML** メッセージをサポートします。

ProcessXML

SOAP エンベロープでラップされていない生の受信 XML メッセージをサポートします。

詳細については、「*CA SOA Security Manager Policy Configuration Guide*」を参照してください。

アフィリエイト エージェント アクション

アフィリエイト エージェントは、ポータル Web サイトの Web サーバにインストールされた Web エージェントと通信する SiteMinder エージェントです。

アフィリエイト エージェントのルールはとても単純です。なぜなら、アフィリエイト エージェントはアフィリエイト Web サイトのリソースは保護しないからです。アフィリエイト エージェントは、ポータルサイトから送信されたレスポンスを処理します。このため、アフィリエイト Web サイトのアプリケーションは、SiteMinder により保護されたポータル Web サイトで集められたユーザに関する情報を活用することができます。

アフィリエイト エージェント アクションは、アフィリエイト エージェントに関連付けられたレルム用の Web エージェント アクションの場所でのみ利用可能です。使用できるアクションは 1 つのみです。

訪問

SiteMinder ポータル サイトがアフィリエイト ウェブサイトと対話できるようにします。

注: アフィリエイト エージェントの詳細については、「Web エージェント設定ガイド」を参照してください。

認証イベント

認証イベントは、SiteMinder がユーザの識別情報を確立する際に発生するイベントです。ルール アクションの 1 つである認証イベントは、認証プロセスの特定の時点でルールを起動する役割を果たします。

認証イベントは、ユーザが On-Auth イベントを含むルールで保護されているリソースにアクセスするときに発生します。Web エージェントアクションや許可イベントとは異なり、認証イベントは常にレルム全体に適用されます。レルムの一部にのみ適用される On-Auth ルールを作成することはできません。

次のリストは、発生する可能性がある **On-Auth** イベントです。

OnAuthAccept

認証に成功した場合に発生します。このイベントは、認証が成功した後、ユーザをリダイレクトするときに使用されることもあります。

OnAuthReject

On-Auth-Reject ルールを含むポリシーにバインドされたユーザの認証に失敗した場合に発生します。このイベントは、認証が失敗した後、ユーザをリダイレクトするときに使用されることもあります。

OnAuthAccept イベントと **OnAuthReject** イベントは、いずれも認証時(ユーザがユーザ名とパスワードを入力したとき)および確認時(ユーザの **cookie** からユーザ情報が読み込まれたとき)に起動します。ただし、認証時にのみ発生する特殊なアクションもあります。

レルムタイムアウトの上書き(**EnforceRealmTimeouts** が使用された場合を除く)。

EnforceRealmTimeouts オプションをサポートしている Web エージェントを使用していて、そのオプションをオンにしている場合を除き、ユーザのアイドル タイムアウトと最大タイムアウト時間は、そのユーザが最後に認証されたレルムの設定値がそのまま適用されます。これらの値は、ユーザが認証情報を再度入力するまで変わりません。

EnforceRealmTimeouts の詳細については、「**SiteMinder 4.x Web and Affiliate Agent Quarterly Maintenance Release 4 Release Notes**」のセクション 3.3 を参照してください。

リダイレクト。

リダイレクトが認証時のみに許可される理由はいくつかありますが、最大の理由は、**OnAuth** リダイレクトを確認時にも許可すると、リダイレクトが無限ループに入る可能性が非常に高いことです。

ユーザのパスワードへのアクセス。

パスワードは **SMSESSION cookie** に保存されないため、ユーザが認証時に実際にパスワードを入力するまでわかりません。

注: On-Auth イベントの結果はレルム単位で発生します。たとえば、OnAuthAccept ヘッダが設定されているレルム A からユーザがレルム B に移動しても、レルム B にはこのヘッダが設定されていないので、ヘッダを使用できません。ユーザがレルム A に戻ると、このヘッダは再度設定されます。

OnAuthAttempt

SiteMinder がユーザを認識できず、そのユーザを拒否したときに発生します(たとえば、未登録ユーザを、最初にユーザ登録サイトにリダイレクトできます)。

OnAuthChallenge

カスタム チャレンジ/レスポンス認証方式がアクティブになったときに発生します(トークンコードなど)。

OnAuthUserNotFound

アクティブ レスポンスを発行するときにのみ使用します。このイベントは、アクティブレスポンス以外のレスポンスの発行に使用することはできません。

認証イベント アクションに対応するルールは、ポリシー内で **SiteMinder** レスポンスと結び付けられます。ユーザが認証または拒否されると、該当する **On-Auth** ルールに関連するレスポンスが認証を要求してきたエージェントに返されます。

注: SiteMinder のパフォーマンスを最適化し、Web エージェントがポリシー サーバから静的情報を取得しなければならない回数を制限するには、OnAuthAccept 認証イベントに基づいてルールを設定し、静的情報を返すレスポンスを作成します。ポリシー内でルールとレスポンスを結び付けると、ポリシー内に指定されたユーザに対してルールが起動され、問題なく認証されたユーザだけに静的レスポンスを返します。

許可イベント

許可イベントは、SiteMinder がユーザに対してリソースへのアクセスを許可するかどうか確認するときに発生するイベントです。ルールアクションの 1 つである許可イベントは、許可プロセスの特定の時点でルールを起動する役割を果たします。

以下のリストは、発生する可能性のある許可イベントです。

OnAccessAccept

ユーザが許可された場合に発生します。このイベントは、リソースへのアクセスを許可されたユーザをリダイレクトすることができます。

OnAccessReject

ユーザが許可されなかった場合に発生します。このイベントを使用して、リソースへのアクセスが許可されなかったユーザをリダイレクトすることができます。

許可イベントアクションに対応したルールは、ポリシーの中で **SiteMinder** レスポンスと結び付けられます。ユーザが許可または拒否されると、該当する **On-Access** ルールに関連するレスポンスが許可を要求してきたエージェントに返されます。

インパーソネーション イベント

インパーソネーション機能を使用すると、権限を持つユーザが、そのセッションを終了することなく、別のユーザのロールで動作することができます。リソースにアクセスするときに、インパーソネーション イベントを使用して、インパーソネーション セッションを開始します。

使用可能なインパーソネーション イベントは、以下のとおりです。

ImpersonationStart

適切なポリシーに追加すると、このイベントを含むルールによって、インパーソネーション セッションの開始が可能になります。

ImpersonationStartUser

適切なポリシーに追加することで、このイベントを含むルールによって、特定のユーザのインパーソネーションが可能になります。

高度なルール オプション

高度なオプションにより、以下の追加のルール設定を定義できます。

時間制限

ルールを起動する時間および起動しない時間を指定します。

アクティブ ルール

外部のビジネス ロジックに基づいて動的に許可できます。

Web エージェント アクションのルールの設定

指定された Web エージェント アクションに対して起動し、リソースへのアクセスを許可または拒否するルールを作成できます。このルールは、リソースを保護するよう設計されています。

注: CA SOA Security Manager XML Agent のためのルールも作成できます。このエージェント タイプのためのルールを作成する場合は、ProcessSOAP および ProcessXML の 2 つの追加ルール アクションを使用できます。詳細については、「CA SOA Security Manager Policy Configuration Guide」を参照してください。

ルールを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール]-[ルールの作成]をクリックします。
[ルールの作成: ドメインの選択] ペインが開きます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[ルールの作成: レルムの選択] ペインが開きます。
4. ルールで保護するリソースが含まれているレルムを選択し、[次へ]をクリックします。
[ルールの作成]のルール定義ペインが開きます。
注: 保護するリソースに対するレルムが存在しない場合、それらのリソースを保護するためのルールを作成することはできません。
5. [一般]グループ ボックスの各フィールドに、ルールの名前と説明を入力します。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
6. ルールに保護させるリソースを[リソース]フィールドに入力します。
[有効なリソース]が更新され、このリソースが含まれます。
7. [許可/拒否と有効/無効]グループ ボックスで、保護されたリソースへのアクセスをルールが許可するか拒否するかを指定します。
8. [アクション]グループ ボックスで、[Web エージェント アクション]ラジオ ボタンを選択します。
[アクションリスト]に HTTP アクションが読み込まれます。
9. [アクション]リストから 1 つ以上の HTTP アクションを選択します。

10. (オプション) [詳細] グループ ボックスで、時間制限、アクティブなルール、または両方を指定します。
11. [完了] をクリックします。
 ルールの作成タスクを処理できるよう送信します。

認証イベントアクション用のルールの設定

ユーザがリソースへのアクセスを取得するために認証する場合に発生するアクションを制御する認証イベントアクションのルールを設定します。

ルールを作成するレルムは、認証イベントを処理可能である必要があります。
 [レルム] ウィンドウの [詳細] グループ ボックスで [認証イベントの処理] が選択されていることを確認します。

ルールを作成する方法

1. [ポリシー]、[ドメイン] をクリックします。
2. [ルール]-[ルールの作成] をクリックします。
 [ルールの作成: ドメインの選択] ペインが開きます。
3. [ドメイン] リストからドメインを選択し、[次へ] をクリックします。
 [ルールの作成: レルムの選択] ペインが開きます。
4. ルールで保護するリソースが含まれているレルムを選択し、[次へ] をクリックします。
 [ルールの作成] のルール定義ペインが開きます。
注: 保護するリソースに対するレルムが存在しない場合、それらのリソースを保護するためのルールを作成することはできません。
5. [一般] グループ ボックスの各フィールドに、ルールの名前と説明を入力します。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ] をクリックします。

6. [アクション]グループ ボックスで[認証イベント]ラジオ ボタンを選択します。
[アクション リスト]に認証イベントが投入されます。

注: 認証イベントはレルム全体に適用されるため、[リソース]フィールドは無効になります。[アクセス許可]オプションおよび[アクセス拒否]オプションは認証イベントには適用されないため、これらのオプションも無効になります。

7. [アクション リスト]から認証イベントを 1 つ以上選択します。
8. (オプション) [詳細]グループ ボックスで、時間制限設定またはアクティブルール設定、またはこれらの両方を設定します。
9. [完了]をクリックします。

ルールは保存され、指定されたレルムおよびリソースに適用されます。

許可イベントアクションのルールを設定

ユーザが認証されると、許可イベントが発生します。許可のルールを設定すると、SiteMinder はリクエストされたリソースに対してユーザが許可されているかどうかに基づいてレスポンスを呼び出すことができます。各自の権限に基づいてユーザにアクセス権が付与または拒否されたときには、その状況に応じたイベントが発生します。

ルールを作成するレルムは、許可イベントを処理可能である必要があります。
[レルム]ウィンドウの[詳細]グループ ボックスで[許可イベントの処理]オプションが選択されていることを確認します。

ルールを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール]-[ルールの作成]をクリックします。
[ルールの作成: ドメインの選択]ペインが開きます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[ルールの作成: レルムの選択]ペインが開きます。

4. ルールで保護するリソースが含まれているレلمを選択し、[次へ]をクリックします。

[ルールの作成]のルール定義ペインが開きます。

注: 保護するリソースに対するレلمが存在しない場合、それらのリソースを保護するためのルールを作成することはできません。

5. [一般]グループ ボックスの各フィールドに、ルールの名前と説明を入力します。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. ルールに保護させるリソースを[リソース]フィールドに入力します。

[有効なリソース]が更新され、このリソースが含まれます。

7. [アクション]グループ ボックスで[許可イベント]ラジオ ボタンを選択します。

[アクションリスト]に許可イベントが投入されます。

注: [アクセス許可]オプションおよび[アクセス拒否]オプションは無効になります。これらのオプションは、許可イベントには適用されません。

8. [アクションリスト]から許可イベントを 1 つ以上選択します。

9. (オプション) [詳細]グループ ボックスで、時間制限設定またはアクティブルール設定、またはこれらの両方を設定します。

10. [サブミット]をクリックします。

ルールは保存され、指定されたレلمおよびリソースに適用されます。

OnAccessReject ルールのポリシーに関する考慮事項

OnAccessReject イベントを含むグローバルなルールを作成する場合は、ポリシー サーバがグローバルなポリシーを処理する方法と、OnAccessReject ルールによって生じる特別な状況を検討してください。

GET/POST ルールと同じポリシーに OnAccessReject ルールを含めると、OnAccessReject ルールは起動しません。ユーザの認証時に、SiteMinder はユーザの ID を解決します。したがって、OnAccessReject ルールと GET/POST ルールが同じポリシー内にあると、リソースへのアクセスが許可されるべきユーザと、OnAccessReject イベントにリダイレクトされるべきユーザが同一になります。ユーザはアクセスを許可されるので、拒否イベントが適用されることはないからです。

この矛盾を解消するには、**OnAccessReject** ルール用の別のポリシーを作成し、このルールを適用するユーザを指定します。このポリシーには他のイベントルールを含めることもできます。

たとえば、LDAP ユーザディレクトリで、**User1** にはリソースへのアクセスを許可し、グループ、**ou=People**、**o=company.com** のそれ以外のユーザはすべて **OnAccessReject** ページにリダイレクトするように設定するとします。この場合には、次の 2 つのポリシーを作成する必要があります。

Policy1

このポリシーには、**User1** のアクセスを許可する **GET/POST** ルールを含めます。

Policy2

OnAccessReject ルールとリダイレクトレスポンスを含めて、グループ **ou=People**、**o=company.com** を指定します。

User1 は許可されているため、**User1** がリソースにアクセスしても **OnAccessReject** ルールは起動しません。ただし、グループ、**ou=People**、**o=company.com** に含まれる他のすべてのユーザについては、リソースにアクセスしようとすると、**OnAccessReject** ルールが起動します。これらのユーザにはリソースへのアクセスが許可されていないからです。

インパーソネーション イベント アクションのルールの設定

リソースがアクセスされるときにインパーソネーション セッションを開始するよう、インパーソネーション イベントのルールを設定します。

ルールを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール]-[ルールの作成]をクリックします。
[ルールの作成: ドメインの選択] ペインが開きます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[ルールの作成: レルムの選択] ペインが開きます。

4. ルールで保護するリソースが含まれているレلمを選択し、[次へ]をクリックします。

[ルールの作成]のルール定義ペインが開きます。

注: 保護するリソースに対するレلمが存在しない場合、それらのリソースを保護するためのルールを作成することはできません。

5. [一般]グループ ボックスの各フィールドに、ルールの名前と説明を入力します。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. ルールに保護させるリソースを[リソース]フィールドに入力します。

[有効なリソース]が更新され、このリソースが含まれます。

7. [アクション]グループ ボックスで[インパーソネーション イベント]ラジオ ボタンを選択します。

[アクションリスト]にインパーソネーション イベントが投入されます。

注: [アクセス許可]オプションおよび[アクセス拒否]オプションは無効になります。これらのオプションは、インパーソネーション イベントには適用されません。

8. [アクションリスト]からインパーソネーション イベントを 1 つ以上選択します。

9. (オプション) [詳細]グループ ボックスで、時間制限設定またはアクティブルール設定、またはこれらの両方を設定します。

10. [完了]をクリックします。

ルールは保存され、指定されたレلمおよびリソースに適用されます。

アフィリエイト エージェントのルールの設定

4/X アフィリエイト エージェントは、ポータル Web サイトの Web サーバにインストールされた Web エージェントと通信する SiteMinder エージェントです。4.x アフィリエイト エージェントのルールはとても単純です。なぜなら、アフィリエイト エージェントは、アフィリエイト Web サイトのリソースは保護しないからです。4.x アフィリエイト エージェントは、ポータル サイトから送信されたレスポンスを処理します。このため、アフィリエイト Web サイトのアプリケーションは、SiteMinder により保護されたポータル Web サイトで集められたユーザに関する情報を活用することができます。

4.x アフィリエイト エージェントのルールで可能なアクションは、「Visit」のみです。これは 4.x アフィリエイト エージェント固有のアクションで、このアクションにより、SiteMinder ポータル サイトとアフィリエイト Web サイトが対話できるようになります。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ルールを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール]-[ルールの作成]をクリックします。
[ルールの作成: ドメインの選択]ペインが開きます。
3. [ドメイン]リストからドメインを選択し、[次へ]をクリックします。
[ルールの作成: レルムの選択]ペインが開きます。
4. ルールで保護するリソースが含まれているレルムを選択し、[次へ]をクリックします。
[ルールの作成]のルール定義ペインが開きます。

注: 保護するリソースに対するレルムが存在しない場合、それらのリソースを保護するためのルールを作成することはできません。

5. [一般]グループ ボックスの各フィールドに、ルールの名前と説明を入力します。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. [アクション]グループ ボックスで、アフィリエイト エージェントのアクション ラジオ ボタンで選択します。

[アクション リスト]に Visit が読み込まれます。

注: 残りのルール プロパティは無効になります。Web サーバ上でアフィリエイト エージェントを設定すると、アフィリエイト エージェントのルールで保護されたリソースを指定します。

7. [完了]をクリックします。

ルールは保存され、指定されたレルムおよびリソースに適用されます。

リソース照合と正規表現

ルールでは、リソース照合と正規表現照合を使用して、ルール内のリソースを指定することができます。

標準的なリソース照合

デフォルトでは、ルールのリソース照合はワイルドカードを使用して行います。

以下の表に、リソース照合でサポートされている文字を示します。

文字	使用方法
*	ワイルドカード(*)は、文字列内のすべての文字の照合に使用します。 <code>*.html</code> という文字列を使用すると、 <code>index.html</code> や <code>out.html</code> など、拡張子が <code>.html</code> のファイルすべてを検索できます。
?	疑問符(?)は、文字列内の 1 文字を照合します。 <code>lmn?p</code> という文字列を使用すると、 <code>lmnop</code> や <code>lmnep</code> などのサブストリングを検索できます。

リソース照合のための正規表現

正規表現を使用すると、リソース照合での柔軟性が高まります。正規表現を使用した照合を有効にするには、[SiteMinder ルールダイアログ]の[正規表現]チェックボックスをオンにします。

正規表現は、文字列照合に使用するテキストパターンです。正規表現で使用する構文の例を以下のテーブルで示します。

文字	結果
¥	メタキャラクタ(「*」など)を引用する場合に使用
¥¥	1 つの「¥」文字と一致
(A)	副次式のグループ化 (パターンの評価順序に影響)
[abc]	単純な文字クラス (文字 <code>abc...</code> の中の任意の 1 文字に一致)

文字	結果
[a-zA-Z]	範囲指定のある文字クラス(a から z、A から Z の範囲内にある 1 文字に一致)
[^abc]	文字 abc... 以外の任意の 1 文字に一致
.	改行以外の任意の 1 文字と一致
^	行の先頭に一致
\$	行の末尾に一致
A*	A が 0 回以上繰り返すものに一致 (greedy)
A+	A が 1 回以上繰り返すものに一致 (greedy)
A?	A が 1 回または 0 回現れるものに一致 (greedy)
A{n}	A が正確に n 回繰り返すものに一致 (greedy)
A{n,}	A が少なくとも n 階繰り返すものに一致 (greedy)
A{n,m}	A が n 回以上、m 回以下繰り返すものに一致 (greedy)
A*?	A が 0 回以上繰り返すものに一致 (reluctant)
A+?	A が 1 回以上繰り返すものに一致 (reluctant)
A??	A が 0 回または 1 回現れるものに一致 (reluctant)
AB	A の後に B が続くものに一致
A B	A または B のいずれかが現れるものに一致
¥1	1 番最初のかっこで囲まれた副次式への後方参照
¥n	n 番目のかっこで囲まれた副次式への後方参照

制限: 各正規表現に含むことができる副次式は、その正規表現自体を含めて 10 個までです。副次式の数、正規表現に含まれる左側かっこの数に 1 を加えたものと同等です。

ルールの有効化および無効化

ルールを有効にして、SiteMinder が指定されたリソースを保護できるようにします。ルールを無効にして、SiteMinder が指定されたリソースを保護しないようにします。

ルールが有効なときは、そのルールを含むポリシーが作成されない限り、またリソースにアクセスしようとするユーザがポリシーで指定されたグループの一員でない限り、保護されたリソースにアクセスすることはできません。ポリシーが作成される前にリソースへのアクセスを許可するには、ルールを無効にします。

ルールを有効または無効にする方法

1. ルールを開きます。
2. ルールを有効にするには[有効]チェック ボックスを選択し、無効にするにはクリアします。
3. [サブミット]をクリックします。
ルールが保存されます。

高度なルール オプション

[ルール]ウィンドウの[詳細]グループボックスでは、ルールの設定を追加で定義することができます。このグループ ボックスでは、時間制限およびアクティブルールを設定できます。時間制限とアクティブなルールについては、次のセクションで説明します。

ルールへの時間制限の追加

時間制限を設定し、**SiteMinder** でルールを起動する時間を指定します。

たとえば、月曜～金曜、午前 9 時～午後 5 時の時間制限を設定すると、指定した時間帯のみルールを起動するように **SiteMinder** が指定されます。ユーザがリソースにアクセスできるのは、ルールが起動するように設定されている時間帯です。指定した時間帯以外は、リソースにアクセスできません。

注: **SiteMinder** が複数のタイム ゾーンで時間を処理する方法の詳細については、「[Web エージェントとポリシー サーバの時間の計算方法 \(P. 103\)](#)」にあります。

時間制限を設定する方法

1. [時間制限]グループ ボックスの「設定」をクリックします。

[時間制限]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. 開始日および有効期限を指定します。
3. 毎時間制限テーブルで時間制限を指定します。

注: 各チェック ボックスは 1 時間に相当します。チェック ボックスで選択した時間帯は、ルールが起動し、指定されているリソースに適用されます。チェック ボックスで選択していない時間帯は、ルールが起動しないため、指定されているリソースに適用されません。

4. [OK]をクリックします。

時間制限が保存され、ルール設定が表示されます。

アクティブ ルールの設定

外部のビジネス ロジックに基づいて動的に許可を行うアクティブ ルールを設定します。このときポリシー サーバでは、ユーザ自ら作成する共有ライブラリの関数を呼び出します。この共有ライブラリは許可 API で指定されたインターフェースに適合する必要があります。この API はソフトウェア開発キットで使用できます。

注: 共有ライブラリの詳細については、「*Programming Guide for C*」を参照してください。

アクティブ ルールを設定する方法

1. [アクティブ ルール]グループ ボックスのフィールドに、ライブラリ名、関数名、関数パラメータを指定します。

アクティブ ルール文字列が[アクティブ ルール]フィールドに表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [サブミット]をクリックします。

アクティブ ルールが保存されます。

ルールの削除

ルールを削除すると、そのルールを含むポリシーからもルールが自動的に削除されます。ただし、ポリシーはシステム上に残ります。ルールの削除後もポリシーが機能するかどうかを確認してください。

注: ポリシーには、少なくとも 1 つのルールが必要です。

ルールグループ内に含まれているルールを削除する場合、削除対象のルールがルールグループから削除されるまでに数秒かかることがあります。削除対象のオブジェクトをすべてキャッシュから削除する場合も、少し時間がかかることがあります。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

第 15 章：ルール グループ

このセクションには、以下のトピックが含まれています。

[ルール グループの概要](#) (P. 517)

[ルール グループの作成](#) (P. 518)

[ルール グループへのルールの追加](#) (P. 519)

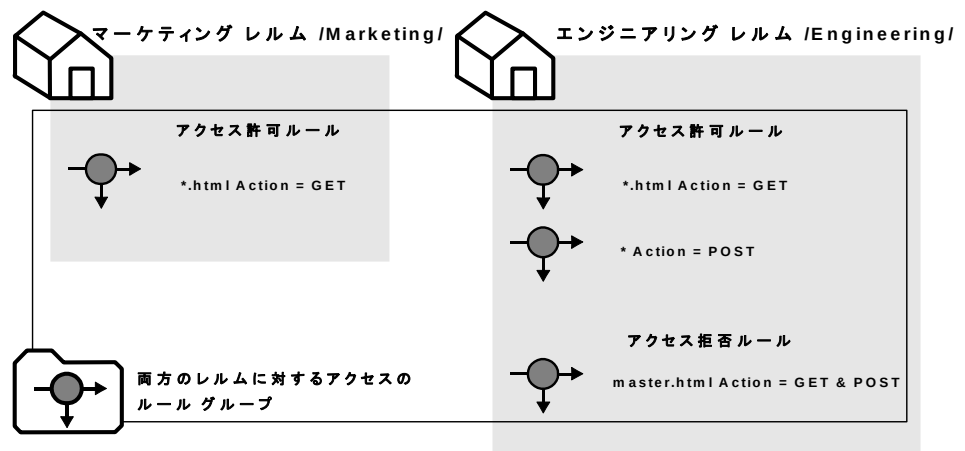
[ルール グループの変更](#) (P. 520)

[ルール グループの削除](#) (P. 520)

ルール グループの概要

ルール グループとは、SiteMinder ポリシーに結び付けることができるルールのセットです。ルール グループを使用して、同じポリシーに適用するルールのグループを組み合わせることができます。たとえば、Web サイトの異なるリソースに対する GET アクションを許可するルールが複数ある場合、すべてのリソースを含むルールグループを作成できます。ルールを含むポリシーを設定するときには、すべてのルールを別々にポリシーに追加するのではなく、1 つのルールグループとしてそれらのルールをポリシーに追加できます。

ポリシーにルール グループを含めると、グループ内の各ルールが評価され、グループ内の他のルールに関係なく単独で適用されます。



前述の図は、マーケティングレalmとエンジニアリングレalmの両方に対するルールを含むルールグループを示しています。この場合、4 つのルールを別々に設定するのではなく、ルールグループとしてポリシーで使用できます。

ルール グループの作成

ルール グループを作成してドメインに追加できます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ルール グループの作成方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール グループ]-[ルール グループの作成]をクリックします。
[ルール グループの作成]ウィンドウが開きます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[ルール グループの作成:ドメインの選択]ウィンドウが開きます。
4. ドロップダウンリストからドメイン名を選択し、[次へ]をクリックします。
[ルール グループの作成: ルール グループの定義]ウィンドウが開きます。
5. [一般]グループ ボックスのフィールドにルール グループの名前および説明を入力します。
6. [属性]グループ ボックスで **RADIUS** または **SiteMinder**、およびエージェント タイプを選択します。
7. [グループ メンバ]グループ ボックスで[追加/削除]をクリックします。
[ルールの選択]ウィンドウが開きます。

[使用可能なメンバー]列は、指定されたドメインおよび指定されたエージェントタイプに関連付けられたレムで定義されたルールをすべて表示します。エージェントタイプが **Generic RADIUS** である場合、[使用可能なメンバー]列は **RADIUS** エージェントが対応しているルールをすべて表示します。

8. [使用可能なメンバー]リストから 1 つまたは複数のルールを選択し、右向きの矢印をクリックします。

[使用可能なメンバー]リストからルールが削除され、[メンバーの選択]リストに追加されます。

一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。

9. [OK]をクリックします。
選択したルールがルール グループに追加されます。
10. [完了]をクリックします。
ルール グループの作成タスクを処理できるよう送信します。

ルール グループへのルールの追加

同じドメインにある同じエージェント タイプのルール グループにルールを追加できます。

ルール グループへのルールの追加方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [ルール グループ]-[ルール グループの変更]をクリックします。
[ルール グループの変更]ウィンドウが開きます。
3. 検索条件を指定して[検索]をクリックします。
ルール グループのリストが開きます。
4. ルール グループを選択し、[選択]をクリックします。
[ルール グループの変更: 名前]ウィンドウが開きます。
5. [グループ メンバ]グループ ボックスで[追加/削除]をクリックします。
[ルールの選択]ウィンドウが開きます。

注: [使用可能なメンバー]列は、指定されたドメインおよび指定されたエージェント タイプに関連付けられたレلمで定義されたルールをすべて表示します。エージェント タイプが **Generic RADIUS** である場合、[使用可能なメンバー]列は **RADIUS** エージェントが対応しているルールをすべて表示します。

6. [使用可能なメンバー]リストから 1 つまたは複数のルールを選択し、右向きの矢印をクリックします。
[使用可能なメンバー]リストからルールが削除され、[メンバーの選択]リストに追加されます。

注: 一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。

7. [OK]をクリックします。
選択したルールがルールグループに追加されます。
8. [サブミット]をクリックします。
ルールグループの変更タスクを処理できるよう送信します。

ルールグループの変更

SiteMinder エージェントのエージェントタイプおよび RADIUS エージェントのベンダータイプ以外のすべてのルールグループのプロパティは変更可能です。エージェントタイプまたはベンダータイプを変更するには、ルールグループを削除して、新しいレスポンスグループを作成してください。

注: ポリシー サーバオブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

ルールグループの削除

ルールグループを削除すると、グループ化が削除されます。グループ内に含まれているルールは削除されません。

注: ポリシー サーバオブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

第 16 章: レスポンスとレスポンス グループ

このセクションには、以下のトピックが含まれています。

[レスポンス](#) (P. 521)

[SiteMinder がレスポンスを処理する方法](#) (P. 522)

[レスポンス グループ](#) (P. 544)

レスポンス

レスポンスは、ポリシー サーバから **SiteMinder** エージェントに、スタティック テキスト、ユーザ属性、DN 属性、カスタマイズされたアクティブ レスポンス、定義された変数のランタイム値を渡します。サーブレット、**Web** アプリケーション、その他のカスタム アプリケーションはレスポンスを使用して、カスタマイズされたコンテンツを表示したり、**SiteMinder** の設定を変更したり、ユーザを別のリソースにリダイレクトしたりできます。**Web** アプリケーションと共に動作する場合には、きめの細かいアクセス制御を行うための権限もしくは権限付与の手段としてレスポンスを利用することができます。

ポリシーには、ユーザとユーザ グループにバインドされるルールとレスポンスが含まれています。ポリシー内では、レスポンスは特定のルールまたはルール グループに結び付けられています。ルールが起動すると、関連するレスポンスは、**SiteMinder** エージェントに情報を返します。

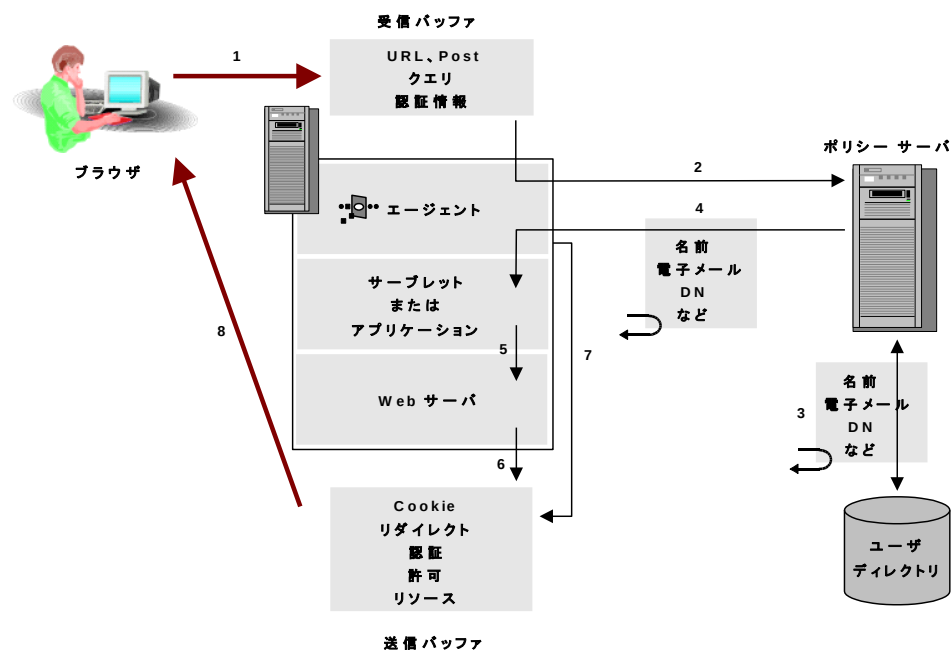
レスポンスの形式は、名前/値のペアになっています。ルールがトリガされると、ポリシー サーバは、名前/値がペアになっているレスポンスを **SiteMinder** エージェントに返します。

たとえば、保護された **Web** ページにアクセスしようとしたユーザが、そのページのコンテンツを表示する許可を受けていない場合、レスポンスにより、ユーザがアクセス権を持っていないことを示す **HTML** ページにユーザをリダイレクトして、システム管理者に問い合わせるための詳細情報を表示させることができます。

Web エージェントの場合、**SiteMinder** は、HTTP ヘッダ変数や HTTP cookie 変数にレスポンス属性を追加して、ルールに指定された **Web** リソースやアプリケーションでレスポンスを使用できるようにします。**RADIUS** 環境では、レスポンスは **RADIUS** クライアントに返されます。

SiteMinder がレスポンスを処理する方法

以下の図は、SiteMinder がどのようにレスポンスを使用して、リソースのユーザーリクエストを処理するかを示します。



前述の図で、SiteMinder は、以下の手順に従ってレスポンスを処理します。

1. ユーザが、SiteMinder エージェントによって保護されているリソースを要求します。
「受信バッファ」は Web サーバのバッファです。このバッファには、要求された URL、Post データ、またはクエリ文字列が Web サーバの処理中に格納されます。
2. SiteMinder エージェントが、保護されているリソースへのリクエストをインターセプトし、ポリシー サーバと通信してユーザの認証と許可を行います。
認証プロセスの過程で、ユーザをユーザディレクトリ内のレコードにバインドします。
3. ポリシー サーバが、このバインドを使用して、ユーザ ディレクトリ内のユーザのエントリから、SiteMinder レスポンスで指定された属性を取得します。
4. ポリシー サーバは、レスポンスで指定されたユーザ属性を Web エージェントに返します。

- レスポンスで指定された属性を使用できるようにカスタマイズされたサーブレットやアプリケーションは、**Web** エージェントに返された属性を使用できます。

サーブレットまたはアプリケーションはプロセスを実行して、**Web** サーバにその結果を渡します。

- Web** サーバの「送信バッファ」に、ユーザに返す必要がある結果情報が格納されます。
- Web** エージェントは、SiteMinder 固有の情報を **Web** サーバの送信バッファに追加します。

Web エージェントは、SiteMinder の cookie、リダイレクト用の URL、および認証または許可の成功/失敗などの情報を送信バッファに渡すことができます。

- Web** サーバは、送信バッファの内容をユーザに渡します。

レスポンス タイプ

レスポンスには、1 つ以上のレスポンス属性が入っています。ポリシー サーバがレスポンスを処理した後に、SiteMinder エージェントがこのレスポンス属性を受け取ります。使用可能なレスポンス属性は、レスポンスのタイプによって異なります。

SiteMinder レスポンスには、以下の 3 つのタイプがあります。

- **Web** エージェントレスポンス
- アフィリエイト エージェントレスポンス
- RADIUS レスポンス

注: SiteMinder API を使用して、カスタム エージェントおよびレスポンス属性用のレスポンス タイプを作成できます。SiteMinder API は、ソフトウェア開発キットとは別売です。詳細については、「API Reference Guide for C」を参照してください。

Web エージェント レスポンス

Web エージェントレスポンスは、SiteMinder **Web** エージェントが使用可能な名前/値のペアを提供する SiteMinder レスポンスです。**Web** エージェントレスポンスに含めることができるのは、HTTP ヘッダ変数、cookie 変数用のリダイレクト用の URL です。

アフィリエイト エージェント レスポンス

アフィリエイト エージェント レスポンスは、SiteMinder アフィリエイト エージェントが使用可能な名前/値のペアを提供する SiteMinder レスポンスです。アフィリエイト エージェント レスポンスに含めることができる属性は、HTTP ヘッダ変数または cookie 変数のための属性です。

RADIUS レスポンス

RADIUS レスポンスは、RADIUS エージェントが使用できる値を提供する SiteMinder レスポンスです。RADIUS レスポンスにはサポートされているすべての RADIUS レスポンス属性を含めることができます。

レスポンス属性

各 SiteMinder レスポンスには、1 つ以上のレスポンス属性が含まれています。このレスポンス属性は、レスポンスのタイプによって異なります。次のセクションでは、各レスポンスタイプで使用可能なレスポンス属性について説明します。

Web エージェント レスポンス属性

Web エージェント レスポンス属性は、SiteMinder Web エージェントが解釈して、他のアプリケーションに渡すことのできるレスポンス属性です。使用可能な一般的な Web エージェントレスポンス属性は、次のとおりです。

WebAgent-HTTP-Authorization-Variable

属性が定義されており、将来 SiteMinder で使用するために予約されていることを示します。

WebAgent-HTTP-Cookie-Variable

SetCookie ヘッダーを作成し、Web ブラウザに非永続的な cookie を設定します。この cookie は、Web エージェントが設定された cookie ドメインにだけ存在します。複数の WebAgent-HTTP-Cookie-Variable を入力することができます。

制限: 承諾または拒否レスポンスで使用します。この属性は、レスポンスごとに複数のインスタンスが許可されています。

WebAgent-HTTP-Header Variable

Web アプリケーションが使用する任意の動的な名前/値ペアを指定します。複数の WebAgent-HTTPHeader-Variable を入力することができます。

Web エージェントが Web ブラウザに返信するレスポンスには、ヘッダ変数が含まれていません。その代わり、ポリシー サーバによって生成されるこれらのレスポンスは、Web サーバのリクエスト ヘッダに配置されます。

したがって、ポリシー サーバ管理コンソールで有効化するデバッグログでは、ヘッダ変数を確認できません。

制限: 承諾または拒否レスポンスで使用します。この属性は、レスポンスごとに複数のインスタンスが許可されています。

WebAgent-OnAccept-Redirect

この属性が使用されるレスポンスのタイプに応じて、以下のいずれかを定義します。

- 許可レスポンスでは、この属性はリソースへのアクセスを許可された場合のユーザのリダイレクト先 URL を定義します。
- 認証レスポンスでは、この属性はセキュリティレルムに対して認証された場合のユーザのリダイレクト先 URL を定義します。

これが許可レスポンスと認証レスポンスのどちらであるかを決定するには、OnAuthAccept または OnAccessAccept のイベント アクションを指定するルールを持つポリシーに、この属性を含めます。

制限: 承諾のレスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

WebAgent-OnAccept-Text

許可または認証に成功した後でユーザをリダイレクトするときに、Web エージェントが HTTP_ONACCEPT_TEXT 環境変数に挿入するテキストを指定します。

制限: 承諾のレスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

注: Web エージェントの OnAcceptText レスポンスを設定する場合は、Web エージェントに対応する FCC 互換モード パラメータ (fcccompatmode) を yes に設定します。これで、ユーザ認証が Web エージェントで実行されて、レスポンス内のテキストをユーザのブラウザに表示できるようになります。FCC 互換モード パラメータ (fcccompatmode) が no に設定された場合、ユーザ認証はフォーム クレデンシャル コレクタ (FCC) で実行されます。この場合、レスポンスはトリガされますが、レスポンス内のテキストは失われます。

WebAgent-OnAuthAccept-Session-Idle-Timeout

ユーザセッションのアイドル状態の制限時間(秒単位)を上書きします。制限時間に達すると、ユーザは再認証を要求されます。このレスポンスは、**OnAuthAccept** 認証イベントを設定したルールと関連付けてください。

制限: 承諾のレスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

WebAgent-OnAuthAccept-Session-Max-Timeout

ユーザセッションのアクティブ状態の制限時間(秒単位)を上書きします。制限時間に達すると、ユーザセッションが終了してユーザは再認証を要求されます。このレスポンスは、**OnAuthAccept** 認証イベントを設定したルールと関連付けてください。

制限: 承諾のレスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

WebAgent-OnAuthAccept-Session-AuthContext

認証方式用の **AuthContext** レスポンス属性を指定します。このレスポンス属性の値は **SM_AUTHENTICATIONCONTEXT** ユーザ属性の値として SiteMinder セッション チケットに追加されます。この値が、ユーザレスポンスとしてクライアントに返されることはありません。

注: レスポンス属性値は、長さが 80 バイトになるように切り捨てられます。

制限: 受諾レスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

WebAgent-OnReject-Redirect

この属性が使用されるレスポンスのタイプに応じて、以下のいずれかを定義します。

- 許可レスポンスでは、この属性はリソースへのアクセスを拒否された場合のユーザのリダイレクト先 URL を定義します。
- 認証レスポンスでは、この属性はセキュリティレルムに対する認証に失敗した場合のユーザのリダイレクト先 URL を定義します。

これが許可レスポンスと認証レスポンスのどちらであるかを決定するには、**OnAuthReject** または **OnAccessReject** のイベントアクションを指定するルールを持つポリシーに、この属性を含めます。

制限: 拒否レスポンスで使用します。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

WebAgent-OnReject-Text

許可または認証に失敗した後でユーザをリダイレクトするときに、Web エージェントが HTTP_ONREJECT_TEXT 環境変数に挿入するテキストを指定します。

制限: 拒否レスポンスで使います。1 つのレスポンスで許可されるこの属性のインスタンスは 1 つだけです。

アフィリエイト エージェント レスポンス属性

アフィリエイト エージェント レスポンス属性は、SiteMinder アフィリエイト エージェントが解読して、アフィリエイト Web サイトの他のアプリケーションに渡すことができるレスポンス属性です。

使用可能なアフィリエイト エージェント レスポンス属性は、次のとおりです。

- AffiliateAgent-HTTP-Header-Variable
- AffiliateAgent-HTTP-Cookie-Variable

注: レスポンス属性の詳細については、「Web エージェント設定ガイド」を参照してください。

RADIUS エージェント レスポンス属性

RADIUS エージェント レスポンス属性は、RADIUS エージェントが解読できるレスポンス属性です。SiteMinder でサポートされるレスポンス属性はすべて、RFC (Request for Comments) 2138 に明記されている属性に対応しています。RFC 2138 では、RADIUS プロトコルによってサポートされる属性が説明されています。

レスポンスとディレクトリ マッピング

ディレクトリ マッピングでは、レルムに対して別個の許可ユーザ ディレクトリを指定できます。別個の許可ディレクトリを定義した場合、ユーザの許可時に使用される情報を含むディレクトリと、ユーザの認証時に使用される情報を含むディレクトリは異なります。

作成したレスポンスを認証 (OnAuth) イベントに関連付けた場合、取得先がユーザ ディレクトリである情報は、すべて認証ディレクトリから取得されます。許可 (OnAccess) イベントを作成した場合、取得先がユーザ ディレクトリである情報は、すべて許可ディレクトリから取得されます。

レスポンスの設定

エージェント タイプおよび属性リストを指定してレスポンスを作成できます。レスポンスには指定された属性が含まれ、指定されたエージェントに送信されます。

レスポンスを作成する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レスポンス]、[レスポンスの作成]をクリックします。
[レスポンスの作成: ドメインの選択] ペインが表示されます。
3. ドメインを選択し、[次へ]をクリックします。
[レスポンスの作成: レスポンスの定義] ペインが表示されます。
4. [一般] ボックスのフィールドにレスポンスの名前および説明を入力します。
5. [属性] グループ ボックスで **RADIUS** または **SiteMinder**、およびエージェント タイプを選択します。
6. (オプション) [レスポンス属性の作成] をクリックしてレスポンス属性を作成し、属性リストに追加します。
[レスポンス属性の作成] ペインが表示されます。
7. [完了] をクリックします。
レスポンスの作成タスクが処理のためにサブミットされます。

レスポンス属性

SiteMinder では、レスポンスごとに 1 つ以上のレスポンス属性が含まれています。レスポンス属性によって、ポリシー サーバから SiteMinder エージェントに渡される情報が識別されます。SiteMinder では、エージェント タイプごとに異なるレスポンス属性を設定できます。

注: CA シングル サインオンからのシングル サインオンを有効にするのに必要な smetssockie Web エージェントのアクティブ レスポンス属性の設定についての詳細は、「[Web エージェントレスポンス属性の設定 \(P. 808\)](#)」を参照してください。

レスポンス属性のタイプ

SiteMinder は、さまざまなタイプのレスポンス属性に対応しています。レスポンス属性のタイプによって、ポリシー サーバがレスポンス属性の該当する値を見つける場所が決まります。

SiteMinder レスポンスにレスポンス属性を追加する場合、指定できるレスポンス属性のタイプは次のとおりです。

スタティック

一定で変化しないデータを返します。

SiteMinder レスポンスの一部として文字列を返す場合は、スタティック属性を使用します。スタティックを使用すると、Web アプリケーションに情報を提供できます。たとえば、ユーザグループが Web サイト上に特定のカスタマイズされたコンテンツを持っている場合は、`show_button = yes` という静的なレスポンス属性がアプリケーションに渡されます。

ユーザ属性

ユーザ ディレクトリのユーザの入力からプロフィール情報を返します。

ユーザ属性を使用して、ディレクトリ内のユーザに関連する情報を返します。ユーザ属性は、LDAP、Windows NT、Microsoft SQL Server、Oracle の各ユーザディレクトリから取得できます。

注: ポリシー サーバがユーザ ディレクトリ属性の値をレスポンス値として返すには、SiteMinder の[ユーザ ディレクトリ]ペインでユーザ ディレクトリを設定しておく必要があります。

DN 属性

LDAP、Microsoft SQL Server または Oracle ユーザ ディレクトリのディレクトリオブジェクトからプロフィール情報を返します。

DN 属性を使用して、ユーザが関連付けられているディレクトリオブジェクトに関する情報を返します。ユーザが所属するグループ、ユーザ DN の一部をなす組織 (OU) などは、属性を DN 属性として扱うことのできるディレクトリオブジェクトの例です。

たとえば、DN 属性を使用すると、ユーザのメンバシップに基づいて、ユーザの部署を返すことができます。

注: ポリシー サーバが DN 属性の値をレスポンス値として返すには、SiteMinder の[ユーザ ディレクトリ]ペインでユーザ ディレクトリを設定しておく必要があります。

アクティブ レスポンス

SiteMinder の許可 API に基づいた、顧客作成のライブラリから値を返します。

アクティブ レスポンスは、外部ソースから情報を返す場合に使用します。ポリシー サーバを使ってユーザ作成の共有ライブラリの関数を呼び出すと、アクティブ レスポンスが生成されます。この共有ライブラリは、許可 API (別売のソフトウェア開発キットに付属、詳細は「*API Reference Guide for C*」を参照) で指定されたインタフェースに準拠している必要があります。

注: アクティブ レスポンスによって返される値が有効かどうかは、ユーザ自身で確認してください。たとえば、アクティブ レスポンスが数値型を返す場合、ライブラリや関数から返される文字列は数字である必要があります。レスポンス属性を設定する際、レスポンス属性の正しい値タイプが[レスポンス属性] ペインに表示されます。

変数定義

指定された変数の値をランタイムに返します。

定義済み変数のリストから変数を選択して使用するには、[変数定義] を選択します。

Web エージェント レスポンス属性の設定

[レスポンス] ウィンドウの[属性] グループ ボックスで SiteMinder および Web エージェントを選択して、SiteMinder Web エージェントのレスポンス属性を作成できます。Web エージェント レスポンス属性は、HTTP ヘッダ変数、cookie 変数、他のリソースへのリダイレクト、テキスト、タイムアウト値をサポートしています。

注: SOA Security Manager を購入およびインストール済みの場合、WebAgent-SAML-Session-Ticket-Variable レスポンス属性を作成できます。詳細については、「*CA SOA Security Manager Policy Configuration Guide*」を参照してください。

レスポンス属性を作成する方法

1. [レスポンス] ペインの[属性リスト] グループ ボックスで、[レスポンス属性の作成] をクリックします。

[レスポンス属性の作成] ペインが表示されます。

2. ドロップダウンリストからレスポンス属性を選択します。

注: レスポンス属性の詳細については、「Web エージェント設定ガイド」を参照してください。

3. [属性の種類]グループ ボックスで属性タイプを選択します。
[属性フィールド]グループ ボックスの各フィールドが、指定した属性タイプと一致するように更新されます。
4. [属性フィールド]グループ ボックスの各フィールドに入力します。
注: レスポンスで利用できる、自動的に生成された SiteMinder ユーザ属性の一覧については、「[SiteMinder が生成するユーザ属性 \(P. 538\)](#)」を参照してください。
5. (省略可) [詳細]グループ ボックスの[スクリプト]フィールドで属性を編集します。
注: [詳細]グループ ボックスで属性を編集すると、[属性のセットアップ]グループ ボックスが閉じます。
6. [属性キャッシング]グループ ボックスで、[キャッシュ値]または[値の再計算間隔]を指定します。
7. [サブミット]をクリックします。
レスポンス属性の作成タスクが、処理のためにサブミットされて、[レスポンス]ペインの[属性リスト]にレスポンス属性が追加されます。

RADIUS レスポンス属性の設定

[レスポンス]ウィンドウの[属性]グループ ボックスで **RADIUS** およびエージェントを選択して、**RADIUS** ベンダーのレスポンス属性を作成できます。**RADIUS** レスポンス属性は、**RADIUS** プロトコルが対応する任意の属性です。

レスポンス属性を作成する方法

1. [レスポンス]ペインの[属性リスト]グループ ボックスで、[レスポンス属性の作成]をクリックします。
[レスポンス属性の作成]ペインが表示されます。
2. ドロップダウンリストからレスポンス属性を選択します。
注: レスポンス属性の詳細については、「**Web** エージェント設定ガイド」を参照してください。
3. [属性の種類]グループ ボックスで属性タイプを選択します。
[属性フィールド]グループ ボックスの各フィールドが、指定した属性タイプと一致するように更新されます。

4. [属性フィールド]グループ ボックスの各フィールドに入力します。

注: レスポンスで利用できる、自動的に生成された SiteMinder ユーザ属性の一覧については、「[SiteMinder が生成するユーザ属性 \(P. 538\)](#)」を参照してください。

5. (省略可) [詳細]グループ ボックスの[スクリプト]フィールドで属性を編集します。

注: [詳細]グループ ボックスで属性を編集すると、[属性のセットアップ]グループ ボックスが閉じます。

6. [属性キャッシング]グループ ボックスで、[キャッシュ値]または[値の再計算間隔]を指定します。
7. [サブミット]をクリックします。

レスポンス属性の作成タスクが、処理のためにサブミットされて、[レスポンス]ペインの[属性リスト]にレスポンス属性が追加されます。

アフィリエイト エージェント レスポンス属性の設定

[レスポンス]ウィンドウの[属性]グループ ボックスで SiteMinder およびアフィリエイト エージェントを選択して、SiteMinder アフィリエイト エージェントのレスポンス属性を作成できます。アフィリエイト エージェントレスポンス属性は、HTTP ヘッダ変数および cookie 変数に対応しています。エージェントタイプについての詳細は、「[Web エージェント設定ガイド](#)」を参照してください。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「[ポリシー サーバ オブジェクトの複製](#)」を参照してください。

レスポンス属性を作成する方法

1. [レスポンス]ペインの[属性リスト]グループ ボックスで、[レスポンス属性の作成]をクリックします。

[レスポンス属性の作成]ペインが表示されます。

2. ドロップダウンリストからレスポンス属性を選択します。

注: レスポンス属性の詳細については、「[Web エージェント設定ガイド](#)」を参照してください。

3. [属性の種類]グループ ボックスで属性タイプを選択します。

[属性フィールド]グループ ボックスの各フィールドが、指定した属性タイプと一致するように更新されます。

4. [属性フィールド]グループ ボックスの各フィールドに入力します。
注: レスポンスで利用できる、自動的に生成された SiteMinder ユーザ属性の一覧については、「[SiteMinder が生成するユーザ属性 \(P. 538\)](#)」を参照してください。
5. (省略可) [詳細]グループ ボックスの[スクリプト]フィールドで属性を編集します。
注: [詳細]グループ ボックスで属性を編集すると、[属性のセットアップ]グループ ボックスが閉じます。
6. [属性キャッシング]グループ ボックスで、[キャッシュ値]または[値の再計算間隔]を指定します。
7. [サブミット]をクリックします。
レスポンス属性の作成タスクが、処理のためにサブミットされて、[レスポンス]ペインの[属性リスト]にレスポンス属性が追加されます。

レスポンスにおける変数オブジェクト

レスポンス属性に変数オブジェクトを組み込むことにより、変数オブジェクトを含むレスポンスを作成できます。変数オブジェクトをレスポンス属性に使用すると、要求の許可時に評価される動的情報を含めることができます。

注: レスポンスに含められた変数オブジェクトは、要求の許可時にのみ評価され、認証プロセスでは評価されません。変数が含まれるレスポンスは、許可イベントにのみ適用できます。

レスポンスにはレスポンス属性をいくつでも含めることができます。各レスポンス属性には、変数オブジェクトが 1 つずつ含まれています。HTTP ヘッダや Cookie 変数と同様に、SiteMinder 変数オブジェクトは名前と値の組み合わせです。SiteMinder 変数オブジェクトは HTTP ヘッダおよび Cookie 変数とは異なります。しかし、変数オブジェクト名を使用して、実行時に変数オブジェクト値を調べます。その後、レスポンス属性の場合は、その結果の名前と値の組み合わせを HTTP ヘッダまたは Cookie 変数で返すことができます。

変数を含むレスポンス属性の設定

1 つのレスポンスには、1 つ以上のレスポンス属性値を含めることができます。この値は、変数オブジェクトによって決まります。各レスポンス属性には、変数オブジェクトが 1 つずつ含まれています。各変数オブジェクトは、名前と値のペアです。変数オブジェクトの名前は、実行時に変数オブジェクトの値を調べるのに使用します。SiteMinder は、その結果の名前と値のペアを Web エージェントに渡します。

変数を含むレスポンス属性の設定方法

1. 「[レスポンスの設定](#) (P. 528)」の説明に従って、レスポンスを作成します。
2. [属性]グループ ボックスで [エージェント タイプ]に[SiteMinder]および [Web エージェント]を選択します。
3. [属性リスト]グループ ボックスの[レスポンス属性の作成]をクリックします。
[レスポンス属性の作成]ペインが表示されます。
4. [属性タイプ]グループ ボックスのドロップダウンリストからレスポンス属性を選択します。
5. [属性の種類]グループ ボックスでレスポンス属性のタイプを選択します。
6. [属性フィールド]グループ ボックスの[変数名]フィールドに変数オブジェクトの名前を入力します。

注: このフィールドが必須の場合、SiteMinder はこの名前を名前と値のペアの形式で Web エージェントに渡します。

7. 選択したレスポンス属性タイプについて、[属性フィールド]グループ ボックスで以下のフィールドに入力します。

スタティック

[変数値]フィールド、でスタティック変数の値を指定します。

ユーザ属性

[属性名]フィールドで、ユーザ属性の名前を指定します。

DN 属性

[DN 仕様]フィールドでユーザまたはユーザグループの DN を、[属性名]フィールドでユーザ属性の名前を指定します。

(オプション)[検索]をクリックして、特定のユーザ ディレクトリ内のユーザまたはユーザグループのセットを検索および選択します。

(オプション)[ネストされたグループの許可]チェック ボックスを選択します。

アクティブ レスポンス

[ライブラリ名]および[関数名]フィールドにライブラリの名前およびライブラリ関数の名前を指定します。また、オプションとして、[パラメータ]フィールドにパラメータの名前を指定します。

注: ユーザのライブラリは SiteMinder Authorization API に基づいている必要があります。

変数定義

[検索]をクリックして、[変数]フィールドに入力する既存の変数オブジェクトを選択します。

注: SiteMinder は、[属性フィールド]グループ ボックスのフィールドに入力した情報を使用して、Web エージェントに名前と値のペアの形式で渡す値を決定します。

8. [OK]をクリックします。

レスポンス属性が保存されます。

レスポンス属性に含めるユーザの選択

[ユーザの検索]ペインでは、ユーザ ディレクトリを 1 つ選択し、そのディレクトリのユーザとユーザグループのリストを検索して、レスポンス属性に含める一連のユーザ、またはユーザグループを選択できます。

レスポンス属性に含めるユーザを選択する方法

1. [属性のセットアップ]グループ ボックスの[属性の種類]で[DN 属性]を選択します。

[属性フィールド]グループ ボックスが拡張され、[DN 仕様]フィールドが表示されます。

2. [属性フィールド]グループ ボックスの[検索]をクリックします。

[ユーザの検索]ペインが表示されます。

3. リストから 1 つのユーザ ディレクトリの名前を選択し、[検索]をクリックします。

[ユーザ検索]ペインが表示されます。

4. (オプション)[検索タイプ]を選択し、[実行]をクリックします。

属性/値

[ユーザ/グループ]ダイアログのフィールドで属性名と値を指定します。

式

[ユーザ/グループ]ダイアログの[式]フィールドで検索式を指定します。

注: [リセット]をクリックすると、検索結果をクリアできます。

5. リストから一連のユーザ、またはユーザ グループを選択し、[OK]をクリックします。

[ユーザの検索]ペインが表示されます。

6. [OK]をクリックします。

[レスポンス属性]ペインが再度表示され、選択した一連のユーザ、またはユーザ グループが[属性フィールド]グループ ボックスの[DN 仕様]フィールドに追加されます。

変数の検索の使用による変数の選択

[変数の選択]ウィンドウにより、既存の変数オブジェクトのリストから変数オブジェクトを 1 つ選択できます。

変数の検索の使用による変数の選択方法

1. [属性のセットアップ]グループ ボックスで、[属性の種類]として[変数定義]を選択します。

2. [属性フィールド]グループ ボックスの[検索]をクリックします。

[変数の選択]ウィンドウが開きます。

3. リストから変数オブジェクトを 1 つ選択し、[OK]をクリックします。

[レスポンス属性の作成]ウィンドウがもう一度開いて、[属性フィールド]グループ ボックスの[変数]フィールドに変数オブジェクトの名前が表示されます。

レスポンス属性キャッシングの設定

レスポンスは、リクエストを行ったエージェントに値を返します。エージェントに返されるデータは固定値であったり、時間とともに変化したりします。SiteMinder エージェントを使用してリソースを保護すると、エージェントは固定データの値をキャッシュに保存できるので、関連するポリシーが適用されるたびに値を計算し直す必要がありません。

たとえば、顧客の口座番号は固定値ですが、顧客の残高は取引のたびに変わります。口座番号を一度取得したら、キャッシュに保存しておくのが効率的です。しかし、残高の場合は、定期的に計算し直して最新の情報を確認する必要があります。

注: SiteMinder は、RADIUS レスポンス属性をキャッシュに保存しません。

レスポンス属性キャッシングを設定する方法

1. レスポンスを開きます。
関連するレスポンス属性が、[属性リスト]グループ ボックスに表示されます。
2. 希望するレスポンス属性の左側の編集アイコンをクリックします。
[レスポンス属性の変更]ウィンドウが開きます。
3. [属性キャッシング]グループ ボックスでキャッシュ設定を指定します。
注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
4. [サブミット]をクリックします。
キャッシュ設定が保存されます。

レスポンスの編集

レスポンスのプロパティは、エージェントタイプを除き、すべて変更できます。エージェントタイプを変更する場合は、レスポンスを削除して、新しいレスポンスを作成してください。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects](#) (P. 61)」を参照してください。

レスポンスの削除

レスポンスを削除すると、関連付けられたすべてのポリシーからのレスポンスを削除します。

削除されたオブジェクトがすべてキャッシュからクリアされるまで、少し時間がかかる場合があります。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

SiteMinder が生成するユーザ属性

SiteMinder が自動的に生成するユーザ属性は、以下のとおりです。これらの属性は、Web エージェントレスポンスのレスポンス属性として指定できます。

%SM_USER

Web エージェントは、すべてのリクエストに対して **SM_USER** http ヘッダ変数にユーザ名を配置します。証明書に基づく許可など、ユーザがユーザ名を入力する必要がない場合、またはユーザ名が不明の場合には、**SM_USER** ヘッダ変数の値は設定されません。

%SM_USERDN

認証されたユーザ の場合、Web エージェントはこの **http** ヘッダ変数に、ポリシー サーバによって決定される **DN** を設定します。証明書ベースの認証の場合には、この属性を使用してユーザの身元を確認できます。

%SM_USERNAME

認証されたユーザの場合、この属性には **SiteMinder** によって明確にされるユーザ **DN** が格納されます。認証されていないユーザの場合、この属性にはログイン時にユーザが入力するユーザ **ID** が格納されます。

%SM_USERIMPERSONATORNAME

認証方式でインパーソネーションが実行される場合、この属性には **SiteMinder** によって認証されるユーザ **DN** が格納されます。

%SM_USERLOGINNAME

この属性にはログイン時にユーザが指定するユーザ **ID** が格納されます。

%SM_USERIPADDRESS

この属性には認証時または許可時のユーザの **IP** アドレスが格納されます。

%SM_USERPATH

認証されたユーザの場合、この属性には(いずれもユーザ ディレクトリ定義で指定される)ディレクトリ ネームスペースとディレクトリ サーバ、およびユーザ DN (SiteMinder によって明確にされる)が格納されます。例:

"LDAP://123.123.0.1/uid=scarter,ou=people,o=airius.com"

認証されていないユーザの場合は、SM_USERNAME と同じです。

%SM_USERPASSWORD

この属性にはログイン時にユーザが指定するパスワードが格納されます。この属性は、OnAuthAccept を通じて認証が成功した場合にのみ使用できます。値は認証時時にのみ返され、許可時には返されません。

%SM_TRANSACTIONID

この属性には、エージェントによって生成されるトランザクション ID が格納されます。

%SM_USERSESSIONSPEC

ユーザのセッション チケットです。

%SM_USERSESSIONID

この属性には、既に認証済みユーザのセッション ID または認証が成功したときにユーザに割り当てられる予定のセッション ID が格納されます。

%SM_USERSESSIONIP

この属性には、最初のユーザの認証中(セッションの確立時)に使用された IP アドレスが格納されます。

%SM_USERSESSIONUNIVID

この属性には、ユーザのユニバーサル ID が格納されます。ユーザ ディレクトリ定義にユニバーサル ID ディレクトリ属性の指定がない場合には、この値はデフォルトでユーザの DN に設定されます。

%SM_USERSESSIONDIRNAME

この属性には、ポリシー サーバが使用するように設定されているユーザ ディレクトリの名前が格納されます。

%SM_USERSESSIONDIROID

この属性には、ポリシー サーバが使用するように設定されているユーザ ディレクトリのオブジェクト ID が格納されます。

%SM_USERSESSIONTYPE

この属性には、ユーザのセッション タイプが格納されます。値は次のいずれかです。

- 2 - セッション
- 1 - ID

%SM_USERLASTLOGINTIME

この属性には、ユーザが前回ログインし、認証された時刻が GMT 時で格納されます。このレスポンス属性は、OnAuthAccept 認証イベントに対してのみ使用可能です。この属性を設定するには、次の 2 つの条件を満たす必要があります。

- パスワード サービスが有効になっていること
- ユーザが少なくとも 1 回は SiteMinder を通じてログインしていること

%SM_USERPREVIOUSLOGINTIME

この属性は、最後のログイン (SM_USERLASTLOGINTIME により表される) の前の正常なログインの時間を GMT を使用して格納します。このレスポンス属性は、OnAuthAccept 認証イベントに対してのみ使用可能です。この属性を投入するには、パスワード サービスが有効である必要があります。

%SM_USERGROUPS

この属性には、ユーザが属するグループが格納されます。ユーザがネストされたグループに属する場合には、この属性には階層の最下位にあるグループが含まれます。ユーザが属するすべてのネストされたグループについては、SM_USERNESTEDGROUPS を使用してください。

たとえば、ユーザ JSmith が、Accounting グループに含まれる Accounts Payable グループに属する場合、For only the group furthest down in the hierarchy, use SM_OUSERNESTEDGROUPS[には Accounts Payable グループが含まれます。Accounting グループと Accounts Payable グループの両方を使用するには、SM_USERNESTEDGROUPS を使用します。

%SM_USERNESTEDGROUPS

この属性には、ユーザが属するネストされたグループが格納されます。階層の最下位にあるグループについてのみ、SM_OUSERNESTEDGROUPS[を使用してください。

たとえば、ユーザ JSmith が、Accounting グループに含まれる Accounts Payable グループに属する場合、SM_USERNESTEDGROUPS には Accounting グループと Accounts Payable グループが含まれます。Accounting グループのみを使用する場合には、SM_OUSERNESTEDGROUPS[を使用します。

%SM_USERSCHEMAATTRIBUTES

この属性には、DN に関連付けられたユーザ属性、またはユーザに関連付けられたプロパティが格納されます。ユーザディレクトリが SQL データベースの場合、SM_USERSCHEMAATTRIBUTES には、ユーザデータが格納されているテーブルの列名が格納されます。たとえば、SmSampleUsers スキーマを使用している場合には、SM_USERSCHEMAATTRIBUTES には、SmUser テーブルの列名が格納されます。

%SM_USERPOLICIES

ユーザがリソースへのアクセスを許可されている場合、この属性にはユーザにアクセス許可を付与するポリシーの名前が格納されます。たとえば、商品を購入できるのは Buyer ポリシーに関連付けられたユーザに限定されているとします。ポリシー サーバがあるユーザに商品の購入を許可した場合、SM_USERPOLICIES には Buyer が含まれます。

%SM_USERPRIVS

ユーザがリソースへのアクセスを許可または認証されている場合、SM_USERPRIVS 属性には、そのユーザに適用されるすべてのポリシードメイン内のすべてのポリシーのレスポンス属性がすべて格納されます。

%SM_USERREALMPRIVS

ユーザがレルムにあるリソースへのアクセスを許可または認証されている場合、SM_USERREALMPRIVS 属性には、そのレルムのすべてのルールすべてのレスポンス属性が格納されます。

たとえば、「Equipment Purchasing」と呼ばれるレルムがあると仮定します。このレルムには、CheckCredit ルールがあります。さらに、この CheckCredit ルールには、購入者のクレジット上限 (limit = \$15000 など) をレスポンス属性として返すレスポンスが関連付けられています。購入者が 5000 ドル相当の装置を購入しようすると、CheckCredit ルールが起動します。この場合、SM_USERREALMPRIVS には、Equipment Purchasing レルムの下にあるすべてのルールのレスポンス属性がすべて格納されます。

%SM_AUTHENTICATIONLEVEL

ユーザがリソースに対して認証されると、この属性はユーザが認証された認証方式の保護レベルを表す整数値 (0 ～ 1000) を格納します。

%SM_USERDISABLEDSTATE

この属性に格納される 10 進数値は、ユーザが無効化された理由を表すビットマスクです。このビットは、SDK の一部である

Sm_Api_DisabledReason_t データ構造下の SmApi.h に定義されます。

たとえば、ユーザが一定の時間何も操作を行わなかった場合には、Sm_Api_Disabled_Inactivity という理由で無効化されます。

Sm_Api_DisabledReason_t では、理由 Sm_Api_Disabled_Inactivity は、値 0x00000004 に対応します。したがって、この場合には SM_USERDISABLEDSTATE は 4 になります。

ユーザが無効化されるには、他にもさまざまな理由があります。

詳細情報:

[Web エージェントレスポンス属性の設定 \(P. 530\)](#)

SiteMinder で生成されるレスポンス属性の可用性

以下の表は、認証、許可、インパーソネーションの各イベントの発生時に SiteMinder で生成されるレスポンス属性の使用の可否をまとめたものです。

レスポンス属性	認証イベントと許可イベント					インパーソ ネーション(偽 装) イベント
	GET/PUT	On Auth Accept	On Auth Reject	On Access Accept	On Access Reject	
SM_USERNAME	はい	はい	はい	はい	はい	いいえ
SM_USERPATH	はい	はい	はい	はい	はい	いいえ
SM_USERIPADDRESS	はい	はい	はい	はい	はい	いいえ
SM_USERPASSWORD	いいえ	はい	はい	いいえ	いいえ	いいえ
SM_TRANSACTIONID	はい	いいえ	いいえ	はい	はい	いいえ
SM_USERSESSIONID	はい	はい	いいえ	はい	はい	いいえ
SM_USERSESSIONSPEC	はい	いいえ	いいえ	はい	はい	いいえ
SM_USERSESSIONIP	はい	はい	はい	はい	はい	いいえ
SM_USERSESSIONUNIVID	はい	はい	いいえ	はい	はい	いいえ
SM_USERSESSIONDIRNAME	はい	はい	いいえ	はい	はい	いいえ
SM_USERSESSIONDIROID	はい	はい	いいえ	はい	はい	いいえ
SM_USERSESSIONTYPE	はい	はい	いいえ	はい	はい	いいえ
SM_USERLASTLOGINTIME	いいえ	はい	いいえ	いいえ	いいえ	いいえ
SM_OUSERNESTEDGROUPS[	はい	はい	いいえ	はい	はい	いいえ
SM_USERNESTEDGROUPS	はい	はい	いいえ	はい	はい	いいえ
SM_USERSCHEMAATTRIBUTES	はい	はい	はい	はい	はい	いいえ
SM_USERLOGINNAME	いいえ	はい	はい	いいえ	いいえ	いいえ
SM_USERIMPERSONATORNAME	いいえ	いいえ	いいえ	いいえ	いいえ	はい
SM_USERDISABLEDSTATE	はい	はい	いいえ	はい	はい	いいえ

レスポンス属性	認証イベントと許可イベント					インパーソ ネーション(偽 装) イベント
	GET/PUT	On Auth Accept	On Auth Reject	On Access Accept	On Access Reject	
SM_USERPOLICIES	いいえ	いいえ	いいえ	はい	いいえ	いいえ
SM_USERREALMPRIVS	はい	いいえ	いいえ	いいえ	いいえ	いいえ
SM_USERPRIVS	はい	いいえ	いいえ	いいえ	いいえ	いいえ

レスポンス グループ

レスポンス グループは、論理的にグループ化されたレスポンスの集合で、ポリシー内の 1 つのルールに適用することができます。レスポンス グループとペアになっているルールが起動されると、レスポンス グループ内の関連する全レスポンスが発行されます。

レスポンス グループを使用して、1 つのオブジェクトに複数のレスポンスを結合できます。ポリシーを作成すると、ポリシー内の 1 つのルールに複数のレスポンスを簡単に関連付けることができます。

レスポンス グループの設定

レスポンスのセットをポリシー内のルールに適用するレスポンス グループを作成できます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

レスポンス グループの設定方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レスポンス グループ]-[レスポンス グループの作成]をクリックします。
[レスポンス グループの作成]ウィンドウが開きます。

3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

[レスポンスの作成:ドメインの選択]ウィンドウが開きます。

4. ドロップダウンリストからドメイン名を選択し、[次へ]をクリックします。

[レスポンス グループの作成:レスポンス グループの定義]ウィンドウが開きます。

5. [一般]グループ ボックスのフィールドにレスポンス グループの名前および説明を入力します。

6. [属性]グループ ボックスで **RADIUS** または **SiteMinder**、およびエージェントタイプを選択します。

注: 指定されたエージェントタイプは、グループ内のレスポンスのエージェントタイプと一致させる必要があります。指定されたエージェントタイプのレスポンスのみをグループに含めることができます。

7. [グループ メンバ]グループ ボックスで[追加/削除]をクリックします。

[レスポンスの選択]ウィンドウが開きます。

注: [使用可能なメンバー]列は、指定されたエージェントタイプについて指定されたドメインで定義されたレスポンスをすべて表示します。エージェントタイプが **Generic RADIUS** である場合、[使用可能なメンバー]列は **RADIUS** エージェントが対応しているレスポンスをすべて表示します。

8. [使用可能なメンバー]リストから 1 つまたは複数のレスポンスを選択し、右向きの矢印をクリックします。

[使用可能なメンバー]リストからレスポンスが削除され、[メンバーの選択]リストに追加されます。

注: 一度に複数のメンバを選択するには、**Ctrl** キーを押しながら追加のメンバをクリックします。メンバを範囲で選択するには最初のメンバをクリックし、その後、**Shift** キーを押しながら範囲の最後のメンバをクリックします。

9. [OK]をクリックします。

選択されたレスポンスがレスポンス グループに追加されます。

10. [サブミット]をクリックします。

レスポンス グループの作成タスクを処理できるよう送信します。

レスポンス グループへのレスポンスの追加

レスポンス グループには、同じエージェントタイプのレスポンスを追加することができます。レスポンスはすべて、同じドメインに存在する必要があります。

レスポンス グループにレスポンスを追加する方法

1. レスポンス グループを開きます。
2. [グループのメンバー]グループ ボックスの[追加/削除]をクリックします。
[レスポンスの選択]グループ ボックスが開きます。[使用可能なメンバー]列に、選択されたドメインにある、指定されたエージェントタイプまたは RADIUS ベンダー タイプの使用可能なレスポンスが表示されます。
注: Generic RADIUS を指定した場合、[使用可能なメンバー]列には、RADIUS エージェントでサポートされるすべてのレスポンスが表示されます。
3. レスポンスを[メンバーの選択]列に移動し、グループにそれらのレスポンスを追加して、[OK]をクリックします。
[レスポンス グループ]ペインが開きます。選択したルールが、[グループのメンバー]グループ ボックスに開きます。
4. [サブミット]をクリックします。
レスポンス グループが保存されます。

レスポンス グループの変更

レスポンス グループのプロパティは、エージェントタイプ以外はすべて変更できます。

エージェントタイプを変更するには、レスポンス グループを削除して、新しいレスポンスグループを作成してください。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

レスポンス グループの削除

レスポンス グループの削除はグループ化を削除するだけであり、グループに含まれる個々のレスポンスは削除しません。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

第 17 章: ポリシー

このセクションには、以下のトピックが含まれています。

[ポリシーの概要](#) (P. 549)

[ポリシーの設定方法](#) (P. 553)

[ユーザまたはグループをポリシーから除外](#) (P. 559)

[ネストされたグループをポリシーに許可](#) (P. 560)

[\[AND ユーザ/グループ\] チェック ボックス](#) (P. 561)

[ユーザ/グループ間の AND/OR 関係の指定](#) (P. 563)

[手動設定によるユーザの追加](#) (P. 564)

[ポリシー サーバの LDAP 許可パフォーマンスの向上](#) (P. 566)

[ポリシーへの LDAP 式の追加](#) (P. 567)

[ポリシーの有効化および無効化](#) (P. 568)

[ポリシーの詳細オプション](#) (P. 569)

[ポリシー バインドの確立](#) (P. 574)

[ポリシーの削除](#) (P. 587)

[SQL クエリへのポリシーのバインド](#) (P. 587)

[ポリシーの処理](#) (P. 588)

ポリシーの概要

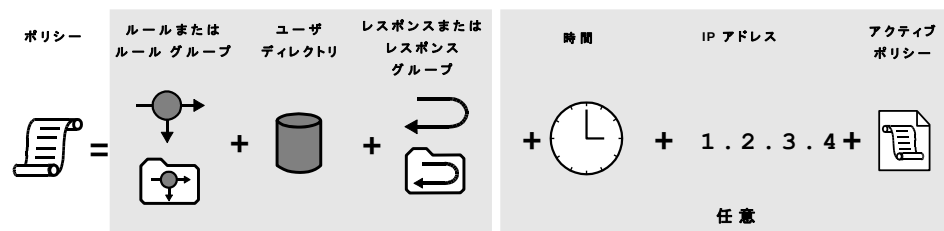
ポリシーは、ユーザとリソースがどのように関連付けられるかを定義します。管理 UI でポリシーを作成すると、ユーザ、リソース、リソースに関連するアクションを識別するさまざまなオブジェクトが結び付けられます (バインドされます)。

ポリシーはポリシー ドメイン内に格納されます。ポリシーを設定する際は、ポリシー ドメイン内で使用可能なユーザ ディレクトリからユーザとユーザグループを選択できます。

SiteMinder はルールによってリソースを識別します。ポリシーを作成する際は、ルールを選択して、ポリシーに含めるリソースを指定できます。

ポリシー内のユーザとリソースを識別して、アクションを指定できます。このアクションは、ユーザが指定のリソースにアクセスした場合に発生します。アクションは、レスポンスの形式で実行されます。ポリシーにレスポンスを追加して、リソースへのアクセスの許可/拒否、ユーザのセッション時間のカスタマイズ、他のリソースへのユーザのリダイレクト、ユーザ ディレクトリに含まれる属性に基づいたユーザの受信内容のカスタマイズができます。

ポリシーを構成するすべての要素を次の図に示します。これらの要素については、図の後で簡単に説明されていますが、この章の後の部分でも詳細に説明されています。



ルール/ルールグループ

ポリシーには少なくとも 1 つのルールまたはルールグループを指定する必要があります。ルールによって、ポリシーに含まれる 1 つまたは複数の特定のリソースが識別されます。

ユーザ

ポリシーには、ポリシーが影響するユーザまたはユーザのグループを指定する必要があります。ユーザまたはユーザグループへの接続は、SiteMinder [ユーザディレクトリ] ウィンドウで設定します。ポリシーへの関連付けが可能なユーザまたはユーザグループは、ポリシーが存在するポリシードメインに含まれるディレクトリのユーザまたはユーザグループだけです。

レスポンス

ルールに指定されたリソースにユーザがアクセスしたときに発生するアクションを定義します。レスポンスによって、ユーザディレクトリから属性を返して他のアプリケーションが使用できるようにしたり、リソースの表示方法をカスタマイズしたりできます。また、認証イベントや許可イベントに基づいてアクションを起動することもできます。

(オプション) IP アドレス

特定のユーザ IP アドレスに限定して、ポリシーを適用できます。IP アドレス制限をポリシーに追加すると、ポリシーに指定されていない IP アドレスからユーザがリソースにアクセスしようとした場合、そのユーザに対してポリシーは起動されません。したがって、アクセスの許可/拒否が行われず、レスポンスも処理されません。

(オプション) 時間制限

特定の日または時間範囲に限定して、ポリシーを適用できます。ポリシーに時間制限が指定されている場合、指定時間以外にポリシーは起動しません。したがって、保護されたリソースへのアクセスの許可/拒否は行われず、レスポンスも処理されません。

(オプション) アクティブなポリシー

SiteMinder の外部のビジネス ロジックをポリシーの定義に取り入れることができます。アクティブなポリシーによって、SiteMinder API を使って作成されたカスタムソフトウェアと SiteMinder が対話できるようになります。

ポリシーの説明

ポリシーは、他のポリシー サーバオブジェクトを結び付け、それを 1 つの論理グループにします。この論理グループによって、オブジェクトがどのように相互に作用するかが決定されます。ディレクトリ接続でアクセスできるユーザ、特定のリソースを指定するルール、およびアクションを定義するレスポンスを結び付けることによって、ポリシーは、リソースへのアクセスを許可されるユーザを定義します。また、ポリシーに含まれるレスポンスによって、ユーザのリソースアクセス時にディレクトリ属性を取得してパーソナライゼーションを提供することもできます。

ポリシーに指定されたユーザの 1 人が、ポリシーのルールのいずれかで識別されるリソースへのアクセスを試みると、ポリシー サーバは、ポリシー内の情報を使用して、そのユーザがリソースにアクセスできるかどうか、およびパーソナライズを実行する必要があるかどうかを解決します。

より詳細な設定を行えば、特定の時間範囲やユーザ IP アドレスに限定してポリシーを適用することができます。このように詳細な設定を行うことで、リソースグループの管理者は管理対象リソースをきめ細かく管理できます。

ポリシー バインド

ポリシー バインドは、ユーザをポリシーに結び付けるために使用される方法です。ポリシー サーバは、ポリシーに含まれているユーザまたはグループによって作成されたポリシー バインドの一部であるユーザに対するポリシーのみを解決します。

ポリシー サーバが、保護されたリソースへのユーザのアクセスを解決するには、事前にユーザが認証されていることが必要です。SiteMinder は、ユーザの認証時にそのユーザのコンテキストを確立します。ユーザ コンテキストの情報によって、ユーザが特定され、ユーザに許可されているリソース アクセス権限がわかります。

たとえば、ユーザ ディレクトリ内のグループ Employees に属するユーザを認証するときに、ポリシー サーバは、グループ Employees 内にそのユーザのメンバシップに対するポリシー バインドを作成します。Employees グループ メンバのアクセスを許可するポリシー内のルールで保護されたリソースにユーザがアクセスを試みると、ユーザのポリシー バインドによって SiteMinder はユーザを許可します。

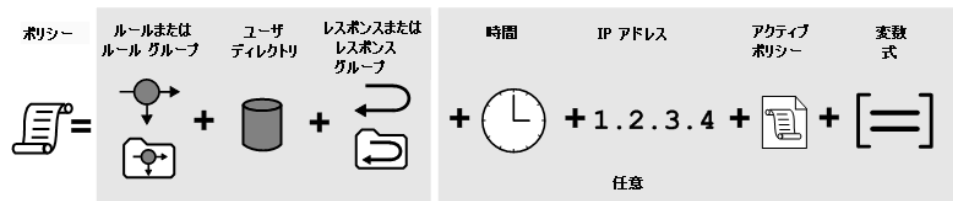
ポリシーにおける式

eTelligent ルールは、変数のセットをポリシー式で使用可能にします。

式を使用すれば、ポリシーを拡張して、実行時に評価される動的な情報を含めることができます。変数オブジェクトを式に含めて、ポリシーで保護されたリソースの権限付与を判断するブール値の条件を作成することができます。

アクティブなポリシー式に変数オブジェクトを使用するには、[式]ダイアログボックスでポリシー オブジェクトを設定し、適切なブール式を作成する必要があります。インターフェースは、「[ポリシーへの LDAP 条件式の追加 \(P. 567\)](#)」で示す LDAP 検索式エディタに似ています。

注: 下図に示すように、ポリシー オブジェクトが対応する他のデータに式を追加できます。



注: アクティブな式と名前付き式は同じではありません。どちらのタイプの式もランタイムに評価されるのは同じですが、以下の点で異なります。

- アクティブな式はブール式ですが、名前付き式は文字列、数値、ブール値を返すことができます。
- アクティブな式はそのまま参照されるため、使用するたびに再入力する必要がありますが、名前付き式は名前で参照されるため、どこからでも参照でき、再利用できます。

ポリシーの設定方法

次に、ポリシーを設定するための手順を示します。

注: ポリシーの作成には、**Scripting Interface for Perl**も利用できます。詳細については、「*Programming Guide for Perl*」を参照してください。

ポリシーの設定方法

1. [ポリシーを作成](#) (P. 553)します。
2. [ポリシーにユーザを追加](#) (P. 555)します。
3. [ポリシーに 1 つ以上のルールを追加](#) (P. 556)します。
4. (オプション) [レスポンスまたはレスポンス グループをルールに関連付け](#) (P. 556)ます。
5. (オプション) [グローバル レスポンスにルールに関連付け](#) (P. 557)ます。
6. (オプション) [ポリシーの詳細オプションを設定](#) (P. 569)します。

ポリシーの作成

新規または既存のドメインにポリシーを追加することによって、ポリシーを作成することができます。ポリシーでは、ユーザとリソースの関係を定義します。

ポリシーを作成して既存のドメインに追加する方法

1. [ポリシー]-[ドメイン]をクリックします。
2. [ドメイン]-[ドメインの変更]をクリックします。
[ドメインの変更]ペインが開きます。

3. 検索条件を指定して[検索]をクリックします。
検索条件と一致するドメインのリストが開きます。
4. ドメインを選択し、[選択]をクリックします。
[ドメインの変更: 名前]ペインが開きます。
5. [ドメイン]ペインの[ポリシー]タブをクリックします。
[ポリシー]グループ ボックスが開きます。
6. [作成]をクリックします。
[ポリシーの作成]ペインが開きます。
7. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[ポリシーの作成: 名前]ペインが開きます。
8. [一般]グループ ボックスの各フィールドに、ポリシーの名前と説明を入力します。
9. [ユーザ]タブをクリックします。
[ユーザ ディレクトリ]グループ ボックスが開きます。
10. ポリシーに、ユーザ、ユーザ グループ、またはその両方を追加し、[サブミット]をクリックします。
[ドメインの変更: 名前]メインが再び開きます。
11. [サブミット]をクリックします。
ドメインの変更タスクが、処理のためにサブミットされます。

ポリシーにユーザを追加します。

ポリシーに個別のユーザ、ユーザ グループ、または両方を追加し、追加されたユーザとポリシーの間のポリシー バインドを作成できます。保護されたリソースにユーザがアクセスしようとした場合、ポリシーは、ユーザがポリシー バインドの一部であることを確認した後でポリシーに指定されたルールを起動し、ユーザがそのリソースにアクセス可能かどうかを調べます。

ポリシーにユーザを追加する方法

1. [ポリシー] ペインの[ユーザ] タブをクリックします。
[ユーザ ディレクトリ] ウィンドウが開きます。このウィンドウには、ポリシー ドメインに関連付けられた各ユーザ ディレクトリのグループ ボックスが含まれています。
2. ユーザ ディレクトリからポリシーにユーザまたはグループを追加します。
各ユーザ ディレクトリ グループ ボックスから、[メンバーの追加]、[エントリの追加]、[すべて追加]を選択できます。ユーザをポリシー追加するのに使用する方法によって、ユーザを追加できるダイアログ ボックスが開きます。
注：[メンバーの追加]を選択すると、[ユーザ/グループ] ウィンドウ開きます。個々のユーザは、自動的に表示されません。検索ユーティリティを使用して、ディレクトリの 1 つに含まれる特定のユーザを見つけることができます。
右向き矢印(>)またはマイナス記号(-)のクリックにより、ユーザまたはグループをそれぞれ編集または削除できます。
3. 好きな方法で個別のユーザ、ユーザ グループ、または両方を選択して [OK] をクリックします。
[ユーザ ディレクトリ] ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、ポリシーの新規ユーザが表示されています。

ポリシーにユーザをバインドするタスクが完了しました。

ポリシーへのルールの追加

ルールには、ポリシーに含まれる特定のリソースと、ルール起動時のそのリソースへのアクセスの可否が示されています。レスポンスは、ルール起動時に発生するアクションを示します。

注: ポリシーには、少なくとも 1 つのルールまたはルール グループを追加しなければなりません。

ルールまたはルール グループをポリシーに追加する方法

1. [ポリシー]ウィンドウで[ルール]タブをクリックします。
[ルール]グループ ボックスが表示されます。
2. [ルールの追加]をクリックします。
[使用可能なルール]ウィンドウが開きます。
3. ポリシーに追加する個別のルール、ルール グループ、または両方を選択して[OK]をクリックします。
[ルール]グループ ボックスが、追加されたルールおよびグループを表示します。
4. (オプション)レスポンス、またはレスポンス グループとルールを関連付けます。

注: ポリシーからルールまたはルール グループを削除するには、[ルール]グループ ボックスでルールの右側のマイナス記号(-)をクリックします。新規ルールを作成するには、[使用可能なルール]ウィンドウで[新規ルール]をクリックします。

レスポンスまたはレスポンス グループへのルールの関連付け

レスポンスまたはレスポンス グループをポリシー内のグループに関連付けられます。ルールが起動すると、関連するレスポンスも起動します。

ルールをレスポンスまたはレスポンス グループに関連付ける方法

1. レスポンスに関連付けるルールまたはルール グループの[レスポンスの追加]をクリックします。

[使用可能なレスポンス]ウィンドウが開いて、ポリシードメインに設定されたレスポンスおよびレスポンス グループを表示します。

2. レスポンスまたはレスポンス グループを選択して、[OK]をクリックします。

[ルール]グループ ボックス内でレスポンスが開かれ、それぞれのルールに関連付けられます。

注: 必要なレスポンスが存在しない場合は、新規作成をクリックしてレスポンスを作成します。

グローバル レスポンスへのルールの関連付け

ルールを既存のグローバル レスポンスに関連付けることができます。

ルールをグローバル レスポンスに関連付ける方法

1. [ポリシー]ウィンドウで[ルール]タブをクリックします。

[ルール]グループ ボックスが表示されます。

2. 変更するルールの横にある[レスポンスの追加]ボタンをクリックします。

[使用可能なレスポンス]ウィンドウが開きます。

注: グローバル レスポンス、レスポンス、およびグループ レスポンスは、この順序で[使用可能なレスポンス]ウィンドウに表示されています。

3. グローバル レスポンスを選択して、[OK]をクリックします。

[ルール]グループ ボックスがもう一度開きます。選択されたレスポンスがルールに追加されています。

4. [サブミット]をクリックします。

ポリシーの変更タスクを処理できるよう送信します。

ポリシーへの式の追加

ブール式を作成してポリシーに追加できます。ブール式は変数によって演算し、ポリシーが処理された時点の変数の値が処理の結果に影響します。したがって、ブール式はポリシーの決定に影響を及ぼします。

ポリシーへの式の追加方法

1. [ポリシー]ウィンドウで[式]タブをクリックします。
[式]グループ ボックスが開きます。
2. [編集]をクリックします。
[ポリシー式]ウィンドウが開きます。
3. [条件]グループ ボックスのフィールドに変数名を入力するか、または[変数の検索]をクリックしてドロップダウンリストから演算子を選択し、[追加]をクリックします。
[infix 形式]グループボックスに条件が追加されます。
注: 複数の条件を作成するには、このステップを繰り返します。
4. 条件を選択して[Infix 形式]グループ ボックスのボタンをクリックして式を作成します。
5. [OK]をクリックします。
[式]グループ ボックスがもう一度開いて、グループ ボックスのフィールドに式が表示されます。
6. [サブミット]をクリックします。
ポリシーの変更タスクを処理できるよう送信します。

ユーザまたはグループをポリシーから除外

管理 UI では、ユーザまたはユーザのグループをポリシーから除外できます。この機能を使用すると、ポリシーに指定するユーザグループの規模が大きい場合でも、グループの小さなサブセットをポリシーから簡単に除外できます。

ユーザまたはグループをポリシーから除外する方法

1. [ポリシー] ペインの[ユーザ] タブをクリックします。
[ユーザ ディレクトリ] ウィンドウが開きます。
2. [ユーザ ディレクトリ] グループ ボックスで、以下のいずれかをクリックします。
 - メンバーの追加
 - エントリの追加
 - すべて追加
3. ステップ 2 でクリックした項目に対応する以下のリストからタスクを選択します。
 - [メンバーの追加] をクリックした場合は、[ユーザ/グループ] ウィンドウの中で表示される[現在のメンバ] リストから検索し、希望するユーザまたはグループのチェック ボックスをオンにします。
 - [エントリの追加] をクリックした場合は、[ユーザ ディレクトリ検索式エディタ] を使用して、除外する項目を見つける検索式を作成します。
 - [すべて追加] をクリックした場合は、[ユーザ ディレクトリ] グループ ボックスにグループ全体が表示されます。手順 5 に進みます。
4. [OK] をクリックします。
[ユーザ ディレクトリ] ウィンドウがもう一度開いて、選択したユーザまたはグループおよび[除外] ボタンを表示します。
5. 選択したユーザまたはグループを除外するには、[除外] をクリックします。
[現在のメンバー] リストのユーザまたはグループの右側にチェック マークが表示され、そのユーザまたはグループがポリシーから除外されたことを示します。[除外] ボタンの代わりに[追加] ボタンが表示されます。

ポリシーからグループを除外するということは、除外されたグループのメンバー（つまり、除外されたユーザ）でポリシーに含まれるものはすべて、そのポリシーに含まれなくなるということです。たとえば、グループ **Employees** はあるが、グループ **Marketing** は除外されているポリシーの場合、**Employees** グループのメンバーで **Marketing** グループに属さないものは、すべてそのポリシーに含まれます。

6. [サブミット]をクリックします。

変更がサブミットされます。ユーザまたはグループがポリシーから除外されます。

ネストされたグループをポリシーに許可

LDAP ユーザ ディレクトリは、別のグループを含むグループを持つことができます。非常に複雑なディレクトリの場合、大量のユーザ情報を組織化する 1 つの方法として、ネストされたグループを階層にします。

各 LDAP ユーザ ディレクトリに対し、ポリシーがネストされたグループを許可するよう指定できます。LDAP ディレクトリ内にネストされたグループを許可すると、ポリシーの処理時に、ディレクトリの各ユーザグループおよびすべてのサブグループがそれぞれ検索されます。ネストされたグループを許可しない場合は、ディレクトリの各ユーザグループはされますが、ポリシーの処理時にサブグループは検索できません。

ネストされたグループを LDAP ユーザ ディレクトリを含むポリシーに許可する方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。

[ユーザ ディレクトリ]ウィンドウが開きます。このウィンドウには、ポリシードメインに関連付けられた各ユーザ ディレクトリに対応するグループ ボックスが含まれています。

2. ネストされたグループを含む各ユーザ ディレクトリについて[ネストされたグループの許可]チェック ボックスをオンにし、[サブミット]をクリックします。

ポリシーの変更タスクを処理できるよう送信します。また、指定された LDAP ユーザ ディレクトリではネストされたグループが許可されます。

[AND ユーザ/グループ]チェック ボックス

[AND ユーザ/グループ]チェック ボックスでは、複数のユーザ グループのメンバであるユーザに許可を制限できるほか、1 つ、または複数のユーザ グループのメンバである特定のユーザに許可を制限できます。ユーザ ディレクトリの個人ユーザおよびユーザ グループをポリシーに追加する際、チェック ボックスの選択によりそれらの間に AND 関係を指定できます。また、チェック ボックスをオフにすることで、それらの間の OR 関係を指定することもできます。

AND 関係を指定し、結果として作成されるポリシーをユーザに適用する場合、そのユーザが許可されるには以下の要件を満たしている必要があります。

- ユーザはポリシーにバインドされている各ユーザ グループのメンバである必要があります。
- ポリシーにバインドされているのが個人ユーザの場合、ユーザはその個人ユーザである必要があります。

注: ポリシーから除外されたユーザ、またはポリシーから除外されたグループのメンバであるユーザは、許可することができません。

例: User1、Group1、および Group2 がすべてポリシーにバインドされ、AND 関係が指定されているとします。この場合、test_user を許可するには、test_user は User1 であり、かつ Group1 と Group2 のメンバである必要があります。

例: User1、User2、および Group1 がすべてポリシーにバインドされ、AND 関係が指定されているとします。この場合、test_user が User1 と User2 の両方になることはできません。したがって、test_user は許可できません。

重要: ポリシーに 2 人以上の個人ユーザを追加して AND 関係を指定しないでください。1 人のユーザが複数の個人になることは不可能なので、ポリシーは常に失敗します。

AND 関係と OR 関係の両方を指定するには、以下のいずれかの設定を選択します。

- 1 つのポリシーと複数のユーザ ディレクトリ

この設定では、1 つのポリシーに 2 つ以上のユーザ ディレクトリを使用することができます。1 つのディレクトリ内での個人のユーザとユーザ グループ間の関係は、AND または OR に設定できます。異なる複数のディレクトリ内での個人のユーザとユーザ グループ間の関係は、常に OR です。

例: 2つのユーザ ディレクトリと 1つのポリシーがあります。各ディレクトリには、2つのユーザ グループがあり、AND 関係が指定されています。Directory1 には Group1 と Group2 があり、Directory2 には Group3 と Group4 があるとします。この場合、test_user を許可するには、test_user は Group1 と Group2 のメンバであるか、Group3 と Group4 のメンバである必要があります。

この状況は、以下のように論理的に表すことができます。

Directory1(Group1 AND Group2) OR Directory2(Group3 and Group4)

ユース ケース: 2つのユーザ ディレクトリと 1つのポリシーがあります。Directory1 には、Facilities と Human_Resources のユーザ グループがあり、AND 関係が指定されています。Directory2 には、Marketing と Sales のユーザ グループがあり、OR 関係が指定されています。この場合、ユーザを許可するには、ユーザは Facilities と Human_Resources のメンバであるか、Marketing のメンバまたは Sales のメンバである必要があります。この状況は、以下のように論理的に表すことができます。

Directory1(Facilities AND Human_Resources) OR Directory2(Marketing OR Sales)

- 複数のポリシーと 1つのユーザ ディレクトリ

この設定では、共有ドメイン内の 2 つ以上のポリシーが、1 つのユーザ ディレクトリへのアクセス権を持ちます。ユーザ ディレクトリ内の個人のユーザとユーザ グループ間の関係は、一方のポリシーで AND に設定し、もう一方のポリシーで OR に設定することができます。共有ドメイン内の異なるポリシー間の関係は、常に OR です。

例: 2つのポリシーと 1つのユーザ ディレクトリがあります。ユーザ ディレクトリには 4 つのユーザ グループがあります。Group1 と Group2 は Policy1 にバインドされ、Group3 と Group4 は Policy2 にバインドされているとします。どちらのポリシーでもユーザ グループ間には AND 関係が指定されています。この場合、test_user は、Policy1 または Policy2 のアプリケーションによって許可することができます。この状況は、以下のように論理的に表すことができます。

Policy1(Group1 AND Group2) OR Policy2(Group3 AND Group4)

ユースケース: 2つのポリシーと1つのユーザディレクトリがあります。ユーザグループ Human_Resources、Marketing、および Sales は Policy1 にバインドされ、OR 関係が指定されています。ユーザグループ Facilities と Human_Resources は Policy2 にバインドされ、AND 関係が指定されています。この場合、ユーザを許可するには、ユーザは Human_Resources、Marketing、または Sales のメンバーであるか、Facilities と Human_Resources のメンバーである必要があります。2つ目のポリシーは、Human_Resources のメンバーでもある Facilities のメンバーのみを許可します。

この状況は、以下のように論理的に表すことができます。

Policy1(Human_Resources OR Marketing OR Sales) OR Policy2(Facilities AND Human_Resources)

ユーザ/グループ間の AND/OR 関係の指定

[AND ユーザ/グループ]チェックボックスでは、複数のユーザグループのメンバーであるユーザに許可を制限できるほか、1つ、または複数のユーザグループのメンバーである特定のユーザに許可を制限できます。ユーザディレクトリの個人ユーザおよびユーザグループをポリシーに追加する際、チェックボックスの選択によりそれらの間に AND 関係を指定できます。また、チェックボックスの選択を解除すれば OR 関係を指定できます。

AND 関係を指定し、結果として作成されるポリシーをユーザに適用する場合、そのユーザが許可されるには以下の要件を満たしている必要があります。

- ユーザはポリシーにバインドされている各ユーザグループのメンバーである必要があります。
- ポリシーにバインドされているのが個人ユーザの場合、ユーザはその個人ユーザである必要があります。

重要: ポリシーに2人以上の個人ユーザを追加して AND 関係を指定しないでください。1人のユーザが複数の個人になることは不可能なので、ポリシーは常に失敗します。

ユーザと1つ以上のユーザグループ間、または1つのユーザディレクトリ内の複数のユーザグループ間に AND 関係を指定する方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザディレクトリ]ペインが表示され、各ユーザディレクトリは別々のグループボックスに表示されます。
2. AND 関係を指定する各ユーザディレクトリに対応する[AND ユーザ/グループ]チェックボックスを選択します。
3. [サブミット]をクリックします。
タスクは処理のためにサブミットされます。

手動設定によるユーザの追加

ポリシーの[ユーザ/グループ]ダイアログボックスの[使用可能なメンバー]リストを使用してポリシーに含めるユーザおよびグループを指定するのに加えて、[手動設定]グループボックスでユーザまたは検索文字列を指定することもできます。

手動設定によるユーザまたはグループの追加

1. [ポリシー]タブをクリックし、次に、[ドメイン]-[ポリシーの変更]をクリックします。
[検索]ウィンドウが表示されます。
2. (オプション) ユーザの検索条件を絞り込むために検索フォームに入力します。
3. [検索]をクリックします。
ポリシーのリストが表示されます。
4. 希望するポリシーの左側のオプション ボタンをクリックし、次に、[選択]をクリックします。
[ポリシーの変更: <名前>]ウィンドウが表示されます。
5. [ユーザ]タブをクリックします。
ドメインに関連付けられたユーザディレクトリが[ユーザディレクトリ]グループボックスに表示されます。

6. ポリシーの[ユーザ/グループ]ダイアログボックスで、以下のいずれかを実行します。

- **Active Directory ユーザ ディレクトリ**については、以下の設定を指定します。

[手動設定]フィールド

Active Directory ユーザ ディレクトリ用の検索フィルタを指定します。

[エントリの検証]チェック ボックス

エントリを Active Directory ユーザ ディレクトリに追加する前に検索フィルタを検証するかどうかを指定します。

注: Active Directory 検索フィルタの検証が失敗する場合は、このチェック ボックスをオフにしてください。

デフォルト: オン

- **LDAP ディレクトリ**については、[検索する場所]ドロップダウンリストから、以下の検索タイプのうちの 1 つを選択します。

DN の確認

ディレクトリで DN を見つけます。

ユーザの検索

ユーザ エントリ内の一致のみを検索します。

グループの検索

グループ エントリ内の一致のみを検索します。

組織の検索

組織エントリ内の一致のみを検索します。

エントリの検索

.WinNT ディレクトリのユーザ、グループ、および組織エントリの一致を検索します。

- **Microsoft SQL Server、Oracle、および Windows NT ディレクトリ**の場合は、[手動設定]フィールドにユーザ名を入力します。

注: Microsoft SQL Server と Oracle については、[手動設定]フィールドにユーザ名の代わりに SQL クエリを入力できます。

例: SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';

ポリシー サーバは、ユーザ ディレクトリの[証明と接続]タブの[ユーザ名]フィールドに指定されたデータベース ユーザとしてクエリを実行します。[手動設定]フィールドに入力する SQL 文を作成する前に、ユーザ ディレクトリのデータベース スキーマに関する知識が必要です。たとえば、SmSampleUsers スキーマを使用している場合に特定のユーザを追加するには、SmUser テーブルからユーザを選択します。

注: LDAP ディレクトリの場合は、[手動設定]フィールドに「all」と入力して、ポリシーを LDAP ディレクトリ全体にバインドできます。

7. 現在のメンバーへの追加を選択します。
管理 UI は、[現在のメンバー]リストにユーザまたはクエリを追加します。
8. [OK]をクリックし、変更内容を保存して[ポリシーの変更: <名前>]ウィンドウに戻ります。

ポリシー サーバの LDAP 許可パフォーマンスの向上

以下のように、ロールに基づく許可をユーザのロールではなく特定のユーザレコードに制限することにより、LDAP ユーザ ディレクトリに格納されたユーザに対するポリシー サーバの許可パフォーマンスを向上できます。

ポリシー サーバのパフォーマンスを增強する方法

1. [ポリシーの変更]ウィンドウで[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウが開きます。このウィンドウには、ポリシー ドメインに関連付けられた各ユーザ ディレクトリに対応するグループ ボックスが含まれています。
2. 許可のパフォーマンスを向上させたいディレクトリすでにグループ ボックスに表示されている場合は、ステップ 8に進みます。
3. 希望するディレクトリが表示されない場合は、ディレクトリのグループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開いて、選択されたユーザ ディレクトリのユーザおよびグループを表示します。

4. ドロップダウンリストから検索タイプを選択します。

属性/値

ユーザ属性名および値のペアを指定します。

式

SiteMinder 式を指定します。

5. [ユーザ/グループ]グループ ボックスの[属性]フィールドおよび[値]フィールドに、許可に必要なユーザ属性名および値を入力します。
6. [実行]をクリックしてディレクトリを検索します。
ディレクトリのリストが表示されます。
7. 追加するディレクトリのチェック ボックスをオンにし、次に、[OK]をクリックします。
[ユーザ/グループ]ウィンドウが閉じて、[ユーザ ディレクトリ]ウィンドウが表示されます。選択したディレクトリがグループ ボックスに表示されます。
8. ディレクトリの左側の[編集] (矢印)アイコンをクリックします。
[ユーザ ディレクトリ検索式エディタ]が表示されます。
9. [DN の確認]が[検索する場所]ドロップダウンリストに表示されていることを確認してから、[OK]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が閉じます。ポリシー サーバの LDAP 検索は現在のユーザのコンテキストでのみ実行され、LDAP サーバのベース DN では実行されません。この最適化は LDAP サーバおよびポリシー サーバの負荷を減少させ、これにより、より迅速な許可レスポンスが可能になります。

ポリシーへの LDAP 式の追加

LDAP ユーザ ディレクトリへの接続を持つポリシードメインにポリシーを作成する場合は、ユーザ ディレクトリ検索式エディタを使用して、LDAP 検索式をポリシーにバインドできます。検索式は、ユーザプロファイル、グループ プロファイル、および組織プロファイルにある属性に基づいて、ユーザをポリシーに結び付けることができます。

ポリシーに LDAP 式を追加する方法

1. [ポリシー]タブをクリックし、次に、[ドメイン]-[ポリシーの変更]をクリックします。
[検索]ウィンドウが表示されます。
2. (オプション) ユーザの検索条件を絞り込むために検索フォームに入力します。
3. [検索]をクリックします。
ポリシーのリストが表示されます。
4. 希望するポリシーの左側のラジオ ボタンをクリックし、次に、[選択]をクリックします。
[ポリシーの変更: <名前>]ウィンドウが表示されます。
5. [ユーザ]タブをクリックします。
ドメインに関連付けられたユーザ ディレクトリが[ユーザ ディレクトリ]グループ ボックスに表示されます。
6. LDAP 検索式を適用するユーザ ディレクトリについて[エントリの追加]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が表示されます。
7. 特定のユーザ、グループまたは組織属性をユーザのポリシーにバインドする LDAP 式を作成します。
注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
8. [OK]をクリックします。
ユーザ ディレクトリ テーブルに式が表示されます。

ポリシーの有効化および無効化

管理 UI では、ポリシーを有効にしたり、無効にしたりすることができます。デフォルトでは、ポリシーを作成するとそのポリシーは有効になっています。ポリシーが有効な場合、ルールに指定されたリソースにユーザがアクセスしようとする、ポリシーに含まれるルールが起動します。

ポリシーを無効にした場合、ポリシーに含まれるルールは起動しますが、ポリシーはいかなるユーザのアクセスも許可しません。また、リソースについては、ポリシーに含まれるルールに指定されたものはすべて保護されます。ポリシーを有効にするまでは、ポリシーに指定されたルールに関連付けられたリソースにユーザはアクセスできません。ただし、別の有効なポリシーによって、無効なポリシー内のリソースへのアクセスが許可される場合は、有効なポリシーに関連付けられているユーザであれば、無効なポリシー内のリソースにアクセスできます。

ポリシーを有効または無効にする方法

1. ポリシーを開きます。
2. [有効]チェックボックスをオンまたはオフにします。

このチェックボックスをオンにすると、ポリシーは有効になります。このチェックボックスをオンにすると、ポリシーは無効になります。無効なポリシーは起動しません。

3. [サブミット]をクリックします。
ポリシーが保存されます。

ポリシーの詳細オプション

管理 UI には、ポリシーの設定時に使用できる高度な機能が多数用意されています。この機能には、以下が含まれます。

■ IP アドレス

任意の IP アドレスを指定できます。ユーザは、このアドレスを使用してポリシーを起動させる必要があります。

■ 時間制限

ポリシーが起動する時間の範囲を指定できます。ポリシーに時間制限を追加すると、指定した時間以外はポリシーは無効になります。

■ アクティブなポリシー

SiteMinder API で作成された共有ライブラリで関数コールを呼び出すことができます。この関数コールによって、ポリシーを起動するかどうかが決まります。

ポリシーが使用できる IP アドレス

以下の特定の項目からポリシーのリソースにアクセスするユーザーに対してのみ、ポリシーが起動するよう指定します。

- IP アドレス
- ホスト名
- サブネット マスク
- IP アドレスの範囲

たとえば、ポリシーにルールを追加してリソースへのアクセスを許可した後で、IP アドレスの範囲を指定した場合、指定した範囲の IP アドレスからログインしたユーザーだけが、保護されたリソースにアクセスできます。

単一 IP アドレスの指定

単一 IP アドレスを指定して、指定した IP アドレスを使ってポリシーのリソースにアクセスするユーザーに対してのみ、ポリシーを起動するようにします。

単一の IP アドレスを指定する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。

IP アドレスの設定が表示されます。

3. [単一ホスト]ラジオ ボタンを選択します。
単一ホストに固有の設定が表示されます。
4. IP アドレスを入力し、[OK]をクリックします。

[IP アドレス]グループ ボックスに、その IP アドレスが表示されます。

注： IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。

5. [サブミット]をクリックします。
ポリシーが保存されます。

ホスト名の指定

ホスト名を指定して、指定したホストからポリシーのリソースにアクセスするユーザに対してのみ、ポリシーを起動するようにします。

ホスト名を指定する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。
IP アドレスの設定が表示されます。
3. [ホスト名]ラジオ ボタンを選択します。
ホスト名に固有の設定が表示されます。
4. ホスト名を入力し、[OK]をクリックします。
ホスト名が[IP アドレス]グループ ボックスに表示されます。
5. [サブミット]をクリックします。
ポリシーが保存されます。

サブネット マスクの追加

サブネットマスクを指定して、指定したサブネットマスクからポリシーのリソースにアクセスするユーザに対してのみ、ポリシーを起動するようにします。

サブネット マスクを追加する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。
IP アドレスの設定が表示されます。
3. [サブネット マスク]ラジオ ボタンを選択します。
サブネット マスクに固有の設定が表示されます。
4. [IP アドレス]フィールドに IP アドレスを入力します。
注: IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。
5. [サブネット マスク]フィールドに、サブネット マスクを入力します。

6. [OK]をクリックします。
サブネット マスクが[IP アドレス]グループ ボックスに表示されます。
7. [サブミット]をクリックします。
ポリシーが保存されます。

IP アドレスの範囲の追加

IP アドレスの範囲を指定し、アドレスの範囲に該当する IP アドレスを使ってポリシーのリソースにアクセスしたユーザにのみポリシーが起動するようにします。

IP アドレスの範囲を追加する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。
IP アドレスの設定が表示されます。
3. [範囲]ラジオボタンを選択します。
IP アドレスの範囲に固有の設定が表示されます。
4. [最小値]フィールドに、開始 IP アドレスを入力します。
注: IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。
5. [最大値]フィールドに、終了 IP アドレスを入力します。
6. [OK]をクリックします。
[IP アドレス]グループ ボックスに、IP アドレスの範囲が表示されます。
7. [サブミット]をクリックします。
ポリシーが保存されます。

ポリシーの時間制限

管理 UI では、ポリシーに時間制限を追加できます。時間制限を追加すると、時間制限が指定した時間の範囲内でポリシーが起動します。時間制限で指定していない時間帯にユーザがリソースにアクセスしても、ポリシーは起動しません。

たとえば、リソースへのアクセスを確保するポリシーの時間制限を作成し、ポリシーが月曜日から金曜日の午前 9 時から午後 5 時のみ起動すると指定します。ユーザは、時間制限で示された時間の間のみ認証および許可されます。ポリシーによって保護されたリソースは、指定された時間外には使用できません。

注: 時間制限は、ポリシー サーバがインストールされたサーバのシステム クロックに基づいています。

ルールの時間制限とポリシーの時間制限の関係

ポリシーに時間制限を指定するときに、そのポリシーのルールに時間制限がある場合は、2 つの制限の論理積に当たる時間にポリシーが起動します。

たとえば、ポリシーの時間制限が午前 9 時から午後 5 時で、ルールの時眼制限が月曜日から金曜日の場合は、月曜日から金曜日の午前 9 時から午後 5 時のみ、ポリシーが起動します。

ポリシーへの時間制限の追加

ポリシーが特定の時間にのみ起動するように、ポリシーに時間制限を追加します。

ポリシーに時間制限を追加する方法

1. ポリシーを開きます。
2. [時間]グループ ボックスで[設定]をクリックします。

[時間制限]ペインが表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. 開始日および有効期限を指定します。
4. 毎時間制限テーブルで時間制限を指定します。

注: 各チェック ボックスは 1 時間に相当します。チェック ボックスで選択した時間帯は、ルールが起動し、指定されているリソースに適用されます。チェック ボックスで選択していない時間帯は、ルールが起動しないため、指定されているリソースに適用されません。

5. [OK]をクリックします。
時間制限が保存されます。

アクティブなポリシーの設定

アクティブなポリシーを使用して、外部のビジネスロジックに基づく動的な許可を行います。ポリシー サーバがユーザ作成の共有ライブラリの関数を呼び出すと、アクティブなポリシーが許可の決定に追加されます。

この共有ライブラリは、許可 API (別売のソフトウェア開発キットに付属) で指定されたインタフェースに準拠する必要があります。

注: 詳細については、「*API Reference Guide for C*」を参照してください。

アクティブなポリシーを設定する方法

1. グローバル ポリシーを開きます。
2. [詳細]グループ ボックスで、[アクティブなポリシーの編集]チェック ボックスをオンにします。
アクティブなポリシーの設定が表示されます。
3. [ライブラリ名]フィールドに、共有ライブラリの名前を入力します。
4. 共有ライブラリに、アクティブなポリシーを実装する関数の名前を入力します。
5. [サブミット]をクリックします。
ポリシーが保存されます。

ポリシー バインドの確立

以降のセクションでは、さまざまなタイプのポリシー バインドの確立方法について説明します。ユーザ情報が存在するユーザ ディレクトリの種類によって、ポリシーのバインド方法が異なります。

LDAP ディレクトリのポリシー バインド

SiteMinder は、ユーザの認証時にそのユーザ コンテキストを確立します。その結果、アクセス制御ポリシーの決定は、次の表に示す条件のいずれかに一致するユーザ コンテキストに基づくことになります。

ユーザ ネームスペース	説明
ユーザ	ユーザの識別名 (DN) は、ポリシーに指定された DN と一致している必要があります。
ユーザ属性	ユーザ属性に関連した条件を指定する検索式は、真である必要があります。
ユーザ グループ	ユーザの DN は、ポリシーに指定されたユーザグループのメンバーである必要があります。
グループ属性	グループ属性に関連した条件を指定する検索式は、真である必要があります。
組織ロール	ユーザは、ポリシーに指定された組織ロールを持っている必要があります。
組織単位	ユーザは、ポリシーに指定された組織単位のメンバーである必要があります。組織単位は、ユーザの DN、グループ、またはロールの一部である必要があります (デフォルトでは、グループおよびロールは使用されません)。
組織	ユーザは、ポリシーに指定された組織のメンバーである必要があります。組織は、ユーザの DN、グループ、またはロールの一部である必要があります (デフォルトでは、グループおよびロールは使用されません)。
組織属性	組織属性に関連した条件を指定する検索式は、真である必要があります。
カスタム オブジェクト クラス	SiteMinder は、ポリシーをカスタム ディレクトリ オブジェクトに関連付けるように設定することができます。

通常、SiteMinder の[ポリシー]ペインでユーザまたはユーザ属性をポリシーにバインドするには、使用可能なディレクトリ エントリのリストからエントリを選択します。使用可能なディレクトリのリストでは、個々のユーザは表示されません。ただし、ディレクトリ内の特定ユーザの検索や、ユーザをポリシーに直接追加することはできます。

手動設定フィールドでのユーザへのポリシーのバインド

個々のユーザをポリシーに結び付けるには、2 つの方法があります。1 つは、SiteMinder ポリシーの[ユーザ/グループ]ダイアログ ボックスの[手動設定]フィールドを使用する方法です。もう 1 つは、SiteMinder ポリシーの[ユーザ/グループ]ダイアログ ボックスの検索機能を使用する方法です。

[手動設定]フィールドでユーザをポリシーにバインドする方法

1. SiteMinder [ポリシー]ダイアログ ボックス[ユーザ]タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が開きます。

2. [検索する場所]ドロップダウン リストをクリックし、次に、[ユーザの検索]を選択します。
3. [手動設定]フィールドに、ユーザ DN を指定します。

たとえば、「uid=JSmith, ou=people, o=myorg.org」と指定します。

注: ユーザ DN を指定する場合は、ユーザの識別名と同じにしてください。ただし、手動設定を使用しても、ユーザの DN に含まれる情報のサブセットとは一致しません。

4. [OK]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が閉じて、入力したユーザ DN がディレクトリのグループ ボックスに表示されます。
5. [サブミット]をクリックして、変更を保存します。

検索機能によるユーザへのポリシーのバインド

[ユーザ ディレクトリ]ウィンドウで個々のユーザをポリシーにバインドするには、2つの方法があります。[ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックし、[ユーザ/グループ]ウィンドウで属性/値機能を使用します。または、[ユーザ ディレクトリ]グループ ボックスで[エントリの追加]をクリックし、[ユーザ ディレクトリ検索式エディタ]を使用します。

検索機能を使用して個別のユーザをポリシーにバインドする方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開きます。
3. 検索条件を指定し、[実行]をクリックします。
検索条件に一致するユーザのリストが表示されます。
4. 必要なユーザを選択し、[OK]をクリックします。
[ユーザ ディレクトリ]ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、選択したユーザが表示されています。
5. [サブミット]をクリックします。
作成タスクまたはポリシーの変更タスクを処理できるよう送信します。

ユーザ属性へのポリシーのバインド

ポリシーをユーザ属性に結び付けるには、LDAP 検索式を指定します。この式でグループ属性に関する条件を定義し、条件が真になるようにします。たとえば、所在先(*l*)が **westcoast** である、またはメールアドレス(*mail*)の終わりの文字列が **string.com** であるユーザ全員にポリシーをバインドするには、[手動設定]フィールドに以下の検索式を挿入します(先頭にパイプ(*|*)を使用します)。

```
(|(l=westcoast)(mail=*string.com))
```

ユーザ グループへのポリシーのバインド

[ユーザ/グループ]ウィンドウで[ユーザ グループ]が開きます。

ユーザ グループにポリシーをバインドする方法

1. [ユーザ]タブをクリックします。
[ユーザ ディレクトリ]グループ ボックスで、ドメインに関連付けられたユーザ ディレクトリが開きます。
2. [メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開きます。
3. ユーザ グループを選択します。
4. [OK]をクリックします。
[ユーザ ディレクトリ]グループ ボックスが開きます。それぞれのユーザ ディレクトリ テーブルが、ポリシーを適用するユーザ グループを表示します。

組織ロールへのポリシーのバインド

SiteMinder では、ポリシーを組織ロールにバインドできます。ポリシーを組織ロールにバインドする場合、ポリシーを起動するには、ユーザがそのロールのメンバーである必要があります。

[手動設定]フィールドで組織ロールをポリシーにバインドする方法

1. SiteMinder [ポリシー]ダイアログ ボックス[ユーザ]タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が開きます。
2. [検索する場所]ドロップダウン リストをクリックし、次に、[ユーザの検索]を選択します。
3. [手動設定]フィールドで、組織のロールを指定します。
4. [OK]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が閉じて、入力した組織ロールがディレクトリのグループ ボックスに表示されます。
5. [サブミット]をクリックして、変更を保存します。
組織ロールがポリシーにバインドされます。

グループ属性へのポリシーのバインド

ポリシーをグループ属性に結び付けるには、LDAP 検索式を指定します。この式でグループ属性に関する条件を定義し、条件が真になるようにします。

グループ属性へのポリシーのバインド方法

1. SiteMinder [ポリシー] ダイアログ ボックス[ユーザ] タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加] をクリックします。
[ユーザ ディレクトリ検索式エディタ]が開きます。
2. [検索する場所] ドロップダウン リストをクリックし、次に、[ユーザの検索] を選択します。
3. [手動設定] フィールドで、グループを指定します。たとえば、アメリカ合衆国のマサチューセッツ州にあるすべてのグループにポリシーをバインドするには、[手動設定] フィールドに以下の検索式を挿入します。
`(&(c=USA)(s=Massachusetts))`
4. [OK] をクリックします。
[ユーザ ディレクトリ検索式エディタ] が閉じて、入力したグループがディレクトリのグループ ボックスに表示されます。
5. [サブミット] をクリックして、変更を保存します。

組織単位へのポリシーのバインド

ポリシーを組織単位にバインドするには、組織単位を定義する LDAP 検索式を指定します。

[手動設定] フィールドで組織単位をポリシーにバインドする方法

1. SiteMinder [ポリシー] ダイアログ ボックス[ユーザ] タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加] をクリックします。
[ユーザ ディレクトリ検索式エディタ]が開きます。
2. [検索する場所] ドロップダウン リストをクリックし、次に、[組織の検索] を選択します。

3. [手動設定]フィールドで、組織単位を指定します。たとえば、組織単位 (ou) が **marketing** であるユーザ全員にポリシーをバインドするには、[手動設定]フィールドに以下の検索式を挿入します。

ou=marketing

4. [OK]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が閉じて、入力したユーザ **DN** がディレクトリのグループ ボックスに表示されます。
5. [サブミット]をクリックして、変更を保存します。
組織単位がポリシーにバインドされます。

組織へのポリシーのバインド

ポリシーを組織にバインドするには、組織を定義する LDAP 検索式を指定します。

[手動設定]フィールドで組織をポリシーにバインドする方法

1. SiteMinder [ポリシー]ダイアログ ボックス[ユーザ]タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が開きます。

2. [検索する場所]ドロップダウン リストをクリックし、次に、[組織の検索]を選択します。

3. [手動設定]フィールドで、組織を指定します。

たとえば、組織 (**o**) が **myorg.org** であるユーザ全員にポリシーをバインドするには、[手動設定]フィールドに以下の検索式を挿入します。

o=myorg.org

4. [OK]をクリックします。
[ユーザ ディレクトリ検索式エディタ]が閉じて、入力した組織がディレクトリのグループ ボックスに表示されます。
5. [サブミット]をクリックして、変更を保存します。
組織単位がポリシーにバインドされます。

組織属性へのポリシーのバインド

ポリシーを組織属性にバインドするには、LDAP 検索式を指定します。この式でグループ属性に関する条件を定義し、条件が真になるようにします。

[手動設定]フィールドでユーザをポリシーにバインドする方法

1. SiteMinder [ポリシー]ダイアログ ボックス[ユーザ]タブから、検索するユーザ ディレクトリを含むグループ ボックスを見つけて、次に、[エントリの追加]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が開きます。

2. [検索する場所]ドロップダウンリストをクリックし、次に、[組織の検索]を選択します。

3. [手動設定]フィールドで、組織属性を指定します。たとえば、アメリカ合衆国のマサチューセッツ州にあるすべての組織ポリシーをバインドするには、[手動設定]フィールドに以下の検索式を挿入します。

`(&(c=USA)(s=Massachusetts))`

4. [OK]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が閉じて、入力した組織属性がディレクトリのグループ ボックスに表示されます。

5. [サブミット]をクリックして、変更を保存します。

ポリシーが組織属性にバインドされます。

カスタム オブジェクト クラスへのポリシーのバインド

SiteMinder では、ポリシーをカスタム オブジェクト クラスにバインドするように設定できます。ソフトウェア開発キットをインストールしてある場合、詳細については「API Reference Guide for C」を参照してください。

ポリシーのバインド (Windows NT ユーザディレクトリ用)

SiteMinder は、ユーザの認証時にそのユーザのコンテキストを確立します。その結果、アクセス制御ポリシーの決定は、次に示す条件のうちのいずれかに一致するユーザコンテキストに基づくことになります。

ユーザネームスペース	説明
ユーザ	ユーザのユーザ名は、ポリシーに指定したユーザ名と一致する必要があります。
ユーザ グループ	ユーザは、ポリシーに指定されたユーザグループのメンバーである必要があります。

通常、[ポリシー] ペインでユーザをポリシーにバインドするには、使用可能なディレクトリ エントリのリストからエントリを選択します。ただし、使用可能なディレクトリのリストには、個々のユーザは表示されません。

手動設定フィールドでのユーザへのポリシーのバインド

[ユーザ ディレクトリ] ウィンドウで個々のユーザをポリシーにバインドするには、2 つの方法があります。[ユーザ ディレクトリ] グループ ボックスで [メンバーの追加] をクリックし、[ユーザ/グループ] ウィンドウで属性/値検索機能を使用します。または、[ユーザ ディレクトリ] グループ ボックスで [エントリの追加] をクリックし、[ユーザ ディレクトリ検索式エディタ] を使用します。

[手動設定] フィールドで個々のユーザをポリシーにバインドする方法

- [ポリシー] ペインの [ユーザ] タブをクリックします。
[ユーザ ディレクトリ] ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
- [ユーザ ディレクトリ] グループ ボックスで [エントリの追加] をクリックします。
[ユーザ ディレクトリ検索式エディタ] ウィンドウが開きます。
- [条件] および [infix 形式] グループ ボックスでユーザ DN を指定します。

4. [OK]をクリックします。
[ユーザ ディレクトリ]ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、指定したユーザが表示されています。
5. [サブミット]をクリックします。
作成タスクまたはポリシーの変更タスクを処理できるよう送信します。

検索機能によるユーザへのポリシーのバインド

[ユーザ ディレクトリ]ウィンドウで個々のユーザをポリシーにバインドするには、2つの方法があります。[ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックし、[ユーザ/グループ]ウィンドウで属性/値機能を使用します。または、[ユーザ ディレクトリ]グループ ボックスで[エントリの追加]をクリックし、[ユーザ ディレクトリ検索式エディタ]を使用します。

検索機能を使用して個別のユーザをポリシーにバインドする方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開きます。
3. 検索条件を指定し、[実行]をクリックします。
検索条件に一致するユーザのリストが表示されます。
4. 必要なユーザを選択し、[OK]をクリックします。
[ユーザ ディレクトリ]ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、選択したユーザが表示されています。
5. [サブミット]をクリックします。
作成タスクまたはポリシーの変更タスクを処理できるよう送信します。

ユーザ グループへのポリシーのバインド

ユーザ グループにポリシーをバインドできます。

ユーザ グループにポリシーをバインドする方法

1. [ポリシー] ペインの[ユーザ] タブをクリックします。
[ユーザ ディレクトリ] ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ] グループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ] ウィンドウが開きます。
3. ユーザ グループを選択します。
4. [OK]をクリックします。
[ユーザ ディレクトリ] ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、選択したユーザ グループが表示されています。

Microsoft SQL Server または Oracle ユーザ ディレクトリのポリシー バインド

SiteMinder は、ユーザの認証時にそのユーザのコンテキストを確立します。その結果、アクセス制御ポリシーの決定は、次に示す条件のうちのいずれかに一致するユーザコンテキストに基づくことになります。

ユーザネームスペース	説明
ユーザ	ユーザの名前は、ポリシーに指定したユーザ名と一致する必要があります。
ユーザ グループ	ユーザは、ポリシーに指定されたユーザグループのメンバーである必要があります。
ユーザ属性	ユーザ属性に関連した条件を指定する検索式は、真である必要があります。
SQL クエリ	ユーザに関連した条件を指定する SQL クエリは、真である必要があります。

通常、ポリシーの[ユーザ/グループ]ペインでユーザまたはユーザ属性をポリシーにバインドするには、使用可能なディレクトリ エントリのリストからエントリを選択します。ただし、個々のユーザは、使用可能なディレクトリのリストには表示されないことがあります(ユーザディレクトリの SQL クエリ方式でクエリの列挙をどのように設定したかによります)。

手動設定フィールドでのユーザへのポリシーのバインド

[ユーザ ディレクトリ]ウィンドウで個々のユーザをポリシーにバインドするには、2つの方法があります。[ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックし、[ユーザ/グループ]ウィンドウで属性/値検索機能を使用します。または、[ユーザ ディレクトリ]グループ ボックスで[エントリの追加]をクリックし、[ユーザ ディレクトリ検索式エディタ]を使用します。

[手動設定]フィールドで個々のユーザをポリシーにバインドする方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ]グループ ボックスで[エントリの追加]をクリックします。
[ユーザ ディレクトリ検索式エディタ]ウィンドウが開きます。
3. [条件]および[infix 形式]グループ ボックスでユーザ DN を指定します。
4. [OK]をクリックします。
[ユーザ ディレクトリ]ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、指定したユーザが表示されています。
5. [サブミット]をクリックします。
作成タスクまたはポリシーの変更タスクを処理できるよう送信します。

検索機能によるユーザへのポリシーのバインド

[ユーザ ディレクトリ]ウィンドウで個々のユーザをポリシーにバインドするには、2つの方法があります。[ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックし、[ユーザ/グループ]ウィンドウで属性/値機能を使用します。または、[ユーザ ディレクトリ]グループ ボックスで[エントリの追加]をクリックし、[ユーザ ディレクトリ検索式エディタ]を使用します。

検索機能を使用して個別のユーザをポリシーにバインドする方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開きます。
3. 検索条件を指定し、[実行]をクリックします。
検索条件に一致するユーザのリストが表示されます。
4. 必要なユーザを選択し、[OK]をクリックします。
[ユーザ ディレクトリ]ウィンドウがもう一度開きます。ユーザ ディレクトリのグループ ボックスに、選択したユーザが表示されています。
5. [サブミット]をクリックします。
作成タスクまたはポリシーの変更タスクを処理できるよう送信します。

ユーザ グループへのポリシーのバインド

ユーザ グループにポリシーをバインドできます。

ユーザ グループにポリシーをバインドする方法

1. [ポリシー]ペインの[ユーザ]タブをクリックします。
[ユーザ ディレクトリ]ウィンドウで、ドメインに関連付けられているユーザ ディレクトリのリストが開きます。
2. [ユーザ ディレクトリ]グループ ボックスで[メンバーの追加]をクリックします。
[ユーザ/グループ]ウィンドウが開きます。

3. ユーザグループを選択します。

4. [OK]をクリックします。

[ユーザディレクトリ]ウィンドウがもう一度開きます。ユーザディレクトリのグループボックスに、選択したユーザグループが表示されています。

ユーザ属性へのポリシーのバインド

ポリシーをユーザ属性にバインドするには、検索式を指定します。この式でユーザ属性に関する条件を定義し、条件が真になるようにします。

たとえば、市外局番が 555 であるユーザ全員にポリシーをバインドするには、[手動設定]フィールドに以下の検索式 (`areacode='555'`) を挿入します。

ポリシーの削除

ポリシーを削除した場合、ポリシー内の要素はすべて削除されないことに注意してください。ポリシーを削除しても、ポリシーの要素のグループ化(またはバインド)が解除されるだけです。

注: ポリシー サーバ オブジェクトの変更および削除の詳細については、[「Manage Policy Server Objects \(P. 61\)」](#)を参照してください。

SQL クエリへのポリシーのバインド

SiteMinder では、[手動設定]フィールドを使って SQL クエリを指定し、ポリシーをユーザにバインドすることができます。

[手動設定]フィールドを使ってポリシーを SQL クエリにバインドする方法

1. SiteMinder [ポリシー]ダイアログ ボックス[ユーザ]タブから、検索するユーザディレクトリを含むグループボックスを見つけて、次に、[エントリの追加]をクリックします。

[ユーザディレクトリ検索式エディタ]が開きます。

2. [検索する場所]ドロップダウンリストをクリックし、次に、[ユーザの検索]を選択します。

3. [手動設定]フィールドに、ユーザ DN を指定します。

たとえば、「uid=JSmith, ou=people, o=myorg.org」と指定します。

注: ユーザ DN を指定する場合は、ユーザの識別名と同じにしてください。
ただし、手動設定を使用しても、ユーザの DN に含まれる情報のサブセットとは一致しません。

4. [OK]をクリックします。

[ユーザ ディレクトリ検索式エディタ]が閉じて、入力したユーザ DN がディレクトリのグループ ボックスに表示されます。

5. [サブミット]をクリックして、変更を保存します。

たとえば、預金残高が 1000ドルを超えるユーザ全員にポリシーをバインドする場合は、以下のようにクエリを指定します。

```
SELECT name FROM customers WHERE balance, 1000
```

SQL クエリーの内容は、使用しているデータベーススキーマによって異なります。

ポリシーの処理

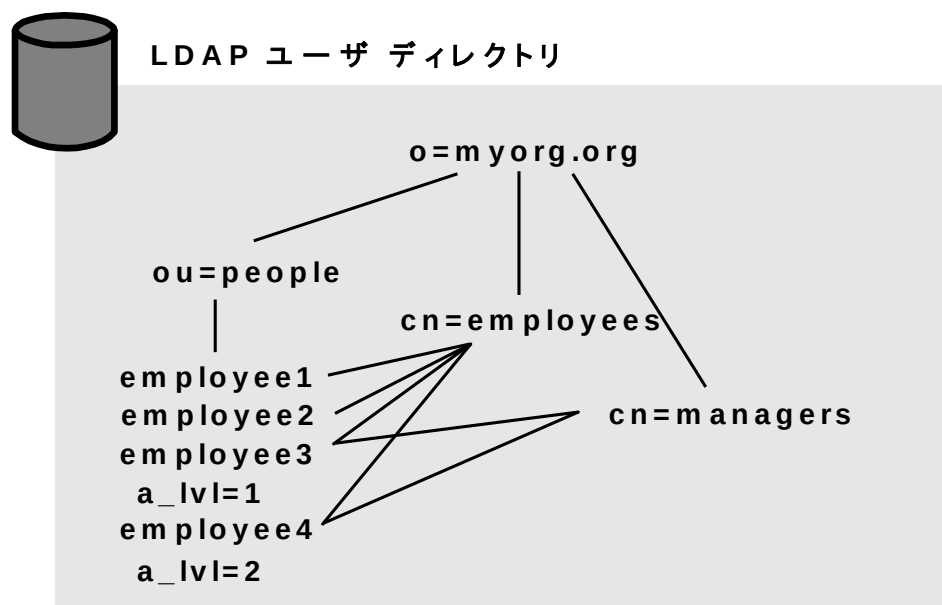
ポリシー サーバでは、ポリシーを使用して、リソースへのアクセスを許可したり、許可されたユーザに権限を付与したりします。通常、ポリシーによって保護されたリソースは、SiteMinder の階層型レルム、つまり「ネストされたレルム」のセキュリティ要件、組織要件、ビジネス要件を模倣した階層に格納され、SiteMinder レルムを利用します。この章ではこれ以降、SiteMinder がポリシーを処理したり階層構造の中から権限付与情報を収集したりする方法を説明します。

ネストされたレルムを持つ SiteMinder を設定するときの例

このセクションで説明する設定の例は、この章の残りの部分でも使用します。

ユーザ ディレクトリ

ポリシーやその他の関連するポリシー サーバ オブジェクトを持つポリシー ドメインに、次の図のような LDAP ユーザ ディレクトリへの接続が含まれるものとします。



例示したユーザ ディレクトリに存在するものは、次のとおりです。

o=myorg.org

これは組織です。

ou=people

これは組織単位です。ここに、すべての従業員の情報が含まれます。

employee<n>

これらは、各従業員のディレクトリ エントリです。**a_lvl** は、アクセスレベルを示すユーザ属性です。このセクションの例では、**employee1** と **employee2** のアクセスレベルを **0 (a_lvl=0)** とします。

cn=employees

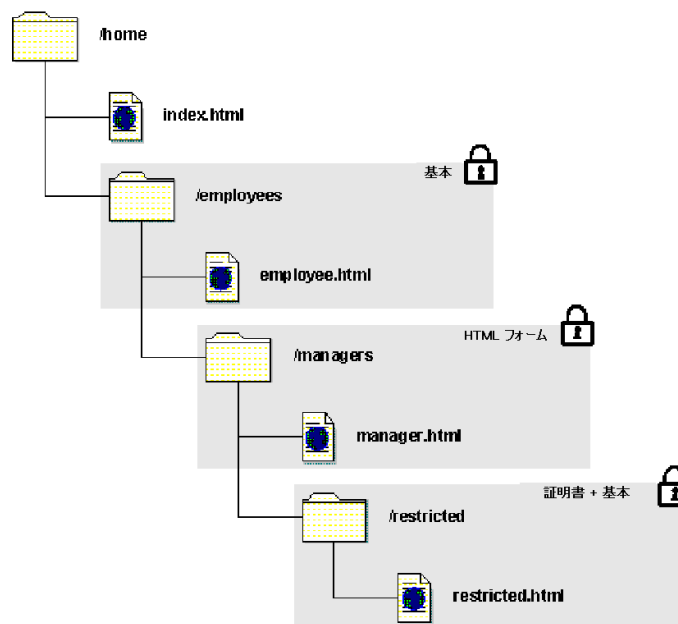
これは、すべての従業員がメンバとして格納されているグループです。

cn=managers

これは、マネージャという役職を持つすべての従業員がメンバとして格納されているグループです。このグループには **employee3** と **employee4** だけが所属します。従業員のアクセスレベルは **0** より大きくなっています。

ネストされたレルムとリソース

このセクションの例で使用するネストされたレルムとリソースの構造は、次の図のとおりです。グレー表示されたレルムは、特定の認証方式で保護されているレルムです。



ネストされたレルムを構成しているディレクトリとリソースは、次のとおりです。

/home

これは、サンプルのネストされたレルムの最上位ディレクトリです。このレルムの場合、リソースフィルタは **/home/** であり、レルム内には **index.html** という保護されていないリソースが存在します。この **index.html** ファイルは、認証方式で保護されていません。

/employees

このディレクトリは **/home** ディレクトリに含まれます。クレデンシャルとしてユーザ名とパスワードを必要とする基本認証方式で保護されています。このディレクトリに関連付けられたレルムの場合、リソースフィルタは **employees/** であり、レルム内には **employee.html** ファイルが存在します。このレルムの全体的なリソースフィルタは、**/home/employees/** です。

`/managers`

このディレクトリは `/employees` ディレクトリに含まれます。クレデンシャルとしてユーザ名、パスワード、および追加の個人情報を必要とする HTML フォーム認証方式によって保護されています。このディレクトリに関連付けられたレルムの場合、リソースフィルタは `managers/` であり、レルム内には `manager.html` ファイルが存在します。このレルムのリソースフィルタは、全体として `/home/employees/managers/` となります。

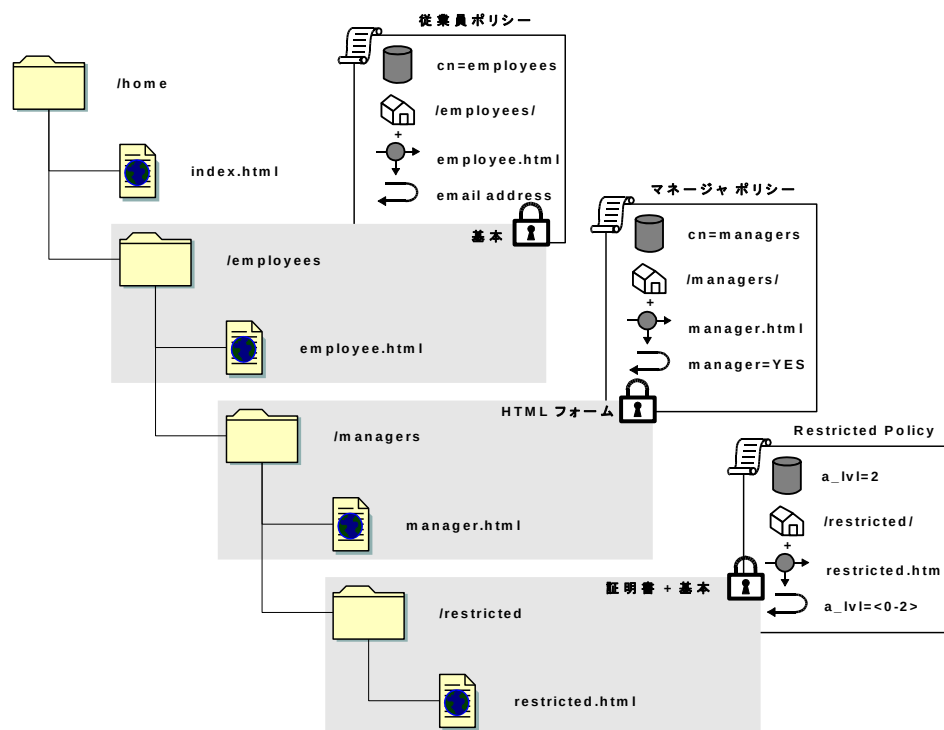
`/restricted`

このディレクトリは `/managers` ディレクトリに含まれます。クレデンシャルとしてユーザ名、パスワード、および有効な X.509 証明書を必要とする、X.509 証明書と基本認証方式によって保護されています。このディレクトリに関連付けられたレルムの場合、リソースフィルタは `restricted/` であり、レルム内には `restricted.html` ファイルが存在します。このレルムのリソースフィルタは、全体として `/home/employees/managers/restricted/` となります。

このセクションの例では、`/employees` レルムの保護レベルが最も低く、`/managers` レルム、`/restricted` レルムの順に保護レベルが高くなります。

ポリシーとレスポンス

この章の後半の例で使用するポリシーとレスポンスを、以下の図に示して説明します。



例示したポリシーとそれぞれのポリシーに含まれるオブジェクトについて説明します。

Employee ポリシー

Get ルールを含むポリシーです。employee.html リソースを保護します。このリソースは、/employees レルムにあります。このポリシーによって、ユーザグループの cn=employees がバインドされます。したがって、LDAP ディレクトリ内の全従業員は、正常に認証されればリソースにアクセスできます。認証されたユーザがこのポリシーによって許可されると、SiteMinder は、そのユーザの電子メール アドレスのレスポンスを返します。たとえば、employee1 が

/home/employees/employee.html へのアクセスを試みて、認証に成功すると、ポリシー サーバは employee1 がリソースにアクセスすることを許可し、電子メール アドレス

employee1@myorg.org を返します。

Web アプリケーションでは、他の社内リソースにアクセスする場合、このレスポンスを使って、そのリソースをカスタマイズできます。

Manager ポリシー

Get ルールを含むポリシーです。manager.html リソースを保護します。このリソースは、/managers レルムにあります。このポリシーによって、ユーザグループの cn=managers がバインドされます。したがって、cn=managers グループ内の従業員だけが、正常に認証されればリソースにアクセスできます。認証されたマネージャがこのポリシーによって許可されると、SiteMinder はスタティックレスポンスを返します。この例では、employee3 が /home/employees/managers/manager.html にアクセスしようとして正常に認証された場合、employee3 はリソースへのアクセスを許可され、ポリシーサーバは次のレスポンスを返します。

manager=YES

アプリケーションでは、会社のマネージャだけが使用可能な機能を、このレスポンスを使用して有効にすることができます。

Restricted ポリシー

Get ルールを含むポリシーです。restricted.html リソースを保護します。このリソースは、

/restricted レルムにあります。このポリシーによってバインドされるディレクトリ内の従業員は、アクセスレベルを示すユーザ属性が 2 (a_lvl=2) の従業員だけです。正しいアクセスレベルを持つマネージャは、正常に認証されるとリソースにアクセスできます。ユーザが restricted.html リソースへのアクセスを試みると、SiteMinder は _lvl=<0-2> のレスポンスを返します。たとえば、employee4 が

/home/employees/managers/restricted/restricted.html へのアクセスを試みて、認証に成功すると、ポリシー サーバは employee4 がリソースにアクセスすることを許可し、以下のレスポンスを返します。

a_lvl=2

アプリケーションは、アクセスレベル 2 を持つ従業員だけが使用可能な機能を、このレスポンスを使用して有効にすることができます。

階層構造のポリシーに対する認証処理

ポリシーにはルールが必要です。ルールには、認証イベントと許可イベントを指定できます。ポリシー サーバがユーザの認証情報に基づいてユーザを識別する場合、ルールの設定内容に応じて、次の 4 つの認証イベントのいずれかが発生します。

OnAuthAccept

正常に認証された場合。

OnAuthAttempt

ポリシードメインの検索順序に指定されたディレクトリを検索してもユーザが見つからなかったため、認証に失敗した場合。

OnAuthReject

ユーザがディレクトリ内に見つかりましたが、ユーザから提供された認証情報が正しくないため、認証に失敗した場合。このイベントが発生した場合、ポリシー サーバは、ポリシードメインのディレクトリ検索順序で指定された次のディレクトリを検索します。検索順序に指定されたディレクトリではユーザの証明を確認できない場合、ポリシー サーバは **OnAuthReject** イベントを処理します。

OnAuthReject ルールが起動するには、ディレクトリでユーザが検出される必要があります。どのディレクトリからもユーザが見つからない場合には、**OnAuthReject** ルールは起動しません。

OnAuthUserNotFound

認証サーバに有効なセッション チケットがあるにもかかわらず、ユーザ ディレクトリが見つからない場合。この状態は、シングル サインオン環境で複数ディレクトリを使用する場合にのみ発生します。ただし、アイドル状態が長いためにユーザがエージェントのキャッシュから削除され、さらにディレクトリからも削除された場合にも発生する可能性があります。このイベントが発生した場合、ポリシー サーバは、ユーザがディレクトリに存在するかどうかを評価し、セッションチケットには頼りません。

ポリシー サーバは、一致する最も長いレルムに基づいてユーザを認証しようとします。たとえば、ユーザが `/home/employees/managers/manager.html` にアクセスしようすると、ポリシー サーバは、`/managers` レルムを使用して必要な認証報を決定します。前出の図の例では、ユーザは、ブラウザベースのフォームに必要な項目を入力する必要があります。このフォームは、レルムに関連付けられている **HTML フォーム認証方式**で要求されるものです。

注：一致する最も長いレルムによって、ユーザのセッションのタイムアウトも決定されます。タイムアウトがレルムに関連付けられていて、そのレルム内でユーザが正常に認証された場合、そのタイムアウトが使用されます。レスポンスを使用して、特定のリソースまたはリソース グループのレルムのタイムアウト値を上書きすることができます。

成功した認証

OnAuthAccept ルールは、次のすべての条件が満たされたときに起動します。

- ユーザが入力する証明情報が、一致する最も長いレルムの要件を満たす。
- ポリシー サーバによって、ユーザディレクトリ内にユーザが認識される。
- ポリシー サーバによって、ユーザのクレデンシャルが確認される。

この時点で、一致する最も長いレルムや、ネストされたレルムの階層構造の中でそのレルムより上位にあるすべてのレルムから、ポリシー サーバは **OnAuthAccept** 権限付与情報を収集します。前述の図の例では、リソース `/home/employees/managers/manager.html` に対するユーザの認証が成功すると、ポリシー サーバは、`/managers` レルム、`/employees` レルム、`/home` レルムの順に、すべての **OnAuthAccept** 資格を収集します。

拒否された認証

ポリシードメインは、ディレクトリの検索順序に従って設定されます。ポリシー サーバがユーザを認証しようとする場合、検索順序に従ってユーザディレクトリ内でユーザを検索し、入力された証明情報を確認します。ポリシー サーバがディレクトリ内にユーザを特定できても、ユーザが入力した証明と一致しない場合は、検索順序に指定された次のディレクトリを検索します。ポリシー サーバがどのディレクトリを検索しても一致するユーザが見つからないと、要求されたリソースが含まれているレルムのコンテキストで、ユーザの認証は失敗します。

たとえば、ユーザが `/home/employees/managers/manager.html` へのアクセスを試みたときに、そのユーザがユーザ ディレクトリに格納されているのに、検索順序に指定されている任意のディレクトリに対して有効なクレデンシャルを入力しなかった場合、`/managers` レルムでの認証イベントは失敗します。次に、ポリシー サーバは、そのレルムで拒否された認証試行に対するイベント (**OnAuthReject**) をすべて処理します。

失敗した認証

ディレクトリ検索順序に指定されているディレクトリをポリシー サーバが検索しても、認証を試みているユーザが見つからない場合は、要求されたリソースが含まれるレルムのコンテキストにおいて、認証が失敗します。次に、ポリシー サーバは、失敗した認証試行に対するイベント (**OnAuthAttempt**) をすべて処理します。

階層構造のポリシーに対する許可処理

ポリシーにルールを指定して、リソースへのアクセスを許可することができます。また、**SiteMinder** イベントを起動するルールを追加することもできます。可能な許可イベントは、次のとおりです。

OnAccessAccept

ユーザが正常に許可されたときに発生するイベントです。

OnAccessReject

認証されたユーザがリソースへのアクセスを拒否されたときに発生するイベントです。

ルールに許可イベントが指定されていない場合は、ルールが実施することは、リソースへのアクセスを許可または拒否することです。

許可されたユーザに対するポリシーの処理

ネストされたレルム内のリソースへのアクセスをポリシー サーバがユーザに許可するには、階層の最上位からリソースを含むレルムまでの各レルムで、ユーザが許可されている必要があります。ユーザが認証されると、ポリシー サーバは、リソースの上位レルムに適用されたポリシーをチェックし、ユーザが許可されていることを確認します。前述の例では、**employee3** が `/home/employees/managers/manager.html` にアクセスしようすると、ポリシー サーバは `/home`、`/employees`、および `/managers` の順にレルムをチェックして、**employee3** が許可されているかどうかを確認します。

ポリシー サーバは、ユーザが許可されていると判断できる場合に、ネストされた各レルムからそのユーザの権限付与情報を収集します。次に、レルムの階層の最上位から順に、一致する最も長いレルムまで、**OnAccessAccept** イベントを処理します。前述の例では、ポリシー サーバから返されるレスポンスには、**manager=YES** というスタティック レスポンスと、**employee3** の電子メール アドレスが含まれます。

無許可のユーザに対するポリシーの処理

無許可のユーザについては、そのユーザのアクセスをポリシーで明示的に拒否しているかどうか、アクセスを許可するポリシーから単にユーザが除外されているかどうかによって、ポリシーの処理が変わることはありません。無許可のユーザについて収集されてきた権限付与情報はすべて失われます。ポリシー サーバは、ユーザアクセスを拒否するレルム、つまり、要求されたリソースが存在するレルムのコンテキストにおいて、**OnAccessReject** イベントを処理します。

前述の例では、**cn=managers** グループのメンバではない **employee1** が `/home/employees/managers/manager.html` へのアクセスを試みると、ポリシー サーバは、`/managers` レルムの **OnAccessReject** イベントのみを処理します。

階層構造のポリシーでのディレクトリ マッピング

SiteMinder の設定で、ネストされたレルムがある場合は、レルムごとに、認証ディレクトリとは異なる許可ディレクトリを指定するディレクトリ マッピングがある可能性があります。ディレクトリ マッピングによって、企業内に複数の許可ディレクトリが配置された状態で、単一のディレクトリを使用してユーザを認証できます。ネストされたレルムのグループで資格情報を処理する場合、ポリシー サーバは階層内のレルムごとにディレクトリ マッピングを調べます。ディレクトリ マッピングがなければ、ポリシー サーバは認証ディレクトリを許可ディレクトリとして使用します。

第 18 章：エンタープライズ ポリシー管理 (EPM)

このセクションには、以下のトピックが含まれています。

[EPM によるアプリケーションの保護 \(P. 599\)](#)

[アプリケーション セキュリティ ポリシーを作成する方法 \(P. 600\)](#)

[アプリケーション セキュリティ ポリシーを作成するための管理権限 \(P. 601\)](#)

[アプリケーションを保護する場合の EPM ユース ケース \(P. 602\)](#)

[アプリケーションの詳細なポリシー コンポーネント \(P. 625\)](#)

EPM によるアプリケーションの保護

エンタープライズ ポリシー管理 (EPM) は、SiteMinder に固有の概念やコンポーネントに関する詳しい知識を必要とせずにビジネス アプリケーションを保護できるアクセス管理モデルです。

EPM は、アプリケーションを保護するためのポリシー設定を示します。アプリケーションを保護するには、デフォルトが指定されていない設定にデータを指定するだけです。他の設定の変更はオプションです。このため、簡単にポリシー設定ができるようになっています。SiteMinder には、その他にもアプリケーションの保護をきめ細かく定義できる設定がありますが、必須ではありません。

SiteMinder についてすでに詳しい知識を持つ管理者であれば、アプリケーション指向の概念と SiteMinder の基本コンポーネント間の関係も考慮できます。この関係は 管理 UI に反映されます。以下の表は、この関係を示しています。

アプリケーション ダイアログとグループ ボックス	SiteMinder の基本コンポーネント
一般設定	ポリシードメインを定義する
コンポーネント	レルムを定義する
リソース	ルールを指定する
アプリケーション ロール	ユーザ ディレクトリ検索の機能を置き換える

EPM にはアプリケーション ロールが導入されています。アプリケーション ロールでは、あるリソースまたはリソース グループにアクセスできる一連のユーザを定義します。この一連のユーザは、名前付き、または名前なし式で識別されます。アプリケーション ロールでは、アプリケーションへのアクセスをリクエストするユーザの権限を定義できます。

EPM には以下の利点があります。

- アプリケーション中心のアプローチ

ほとんどのビジネスで、アプリケーションを中心に据えることとアクセス管理に対する見解には密接な関係があります。

- 一貫したセキュリティ エンフォースメント モデル

EPM のセキュリティ エンフォースメント モデルは、SiteMinder を中心にしたモデルで実装される場合と変わった点はありませんが、SiteMinder 固有のコンポーネントは設定から隠蔽されます。

- セキュリティの簡素化

リソースの保護が簡素化されており、保護が必要なアプリケーション、アプリケーションリソース、アクセスを許可するアプリケーション ロールを指定するだけで保護できます。セキュリティ ポリシーを設定するために、コンポーネントのすべての側面を調査、または変更する必要はありません。

- 強化された委任機能

SiteMinder 管理者は、SiteMinder の専門知識がなくてもアプリケーションへのアクセス権を与えることができます。このため、上級セキュリティ管理者は、他の管理者にアクセス管理の責任を委任できます。

アプリケーション セキュリティ ポリシーを作成する方法

組織内のアプリケーションを保護するには、アプリケーション セキュリティ ポリシーを作成します。これらのポリシーでは、保護するリソースを定義し、保護されたアプリケーションへのアクセスを許可するユーザを指定します。

アプリケーション セキュリティ ポリシーを作成するには、以下の手順に従います。

1. 保護するアプリケーションを定義します。
2. 新しいユーザ ディレクトリを作成します。またはアプリケーションと既存のユーザ ディレクトリを関連付けます。

3. 保護するリソース(アプリケーションの一部や Web ページなど)を指定します。
4. 保護されたリソースにアクセスできるユーザを識別するアプリケーション ロールを作成します。アプリケーション ロールは、アプリケーション ロールのメンバシップ基準を満たすユーザがいないかどうかユーザ ディレクトリを検索する式で定義します。
5. (オプション)レスポンスを設定してアプリケーションをカスタマイズします。
6. (オプション)アプリケーションに関するメタデータを提供するカスタム属性を指定します。
7. すべてのリソースとロールがすべて定義されるまで、手順 4 および 5 を繰り返します。
8. アプリケーション ロールをリソースに関連付けてポリシーを作成します。

一連の[ユース ケース](#) (P. 602)では、アプリケーションを保護する詳細な設定手順を示します。

注: ユース ケースに従うために、実際に 管理 UI を開いてもかまいません。

アプリケーション セキュリティ ポリシーを作成するための管理権限

SiteMinder では、管理モデルを使用して、表示および操作できる各種管理者を判別することができます。

アプリケーション セキュリティ ポリシーを実装するには、必要な管理権限を持っている必要があります。管理者には、アプリケーション関連の以下の権限を割り当てることができます。

- アプリケーション管理

アプリケーション管理権限では、アプリケーションとそのコンポーネントの作成、変更、および削除を行うことができます。

- ポリシー管理

ポリシー管理権限では、アプリケーションに関連付けられているリソース、ロール、およびポリシーの定義を行うことができます。

アプリケーションとポリシーの管理権限を割り当てる方法

1. [管理]-[管理者]-[管理者の作成]をクリックします。
[管理者の作成]ペインが開きます。
2. [名前]フィールドに管理者の名前を入力します。
3. [権限]グループ ボックスの[作成]をクリックします。
[権限]ダイアログ ボックスが開きます。
4. テーブルで、以下の 1 つ以上を選択します。
 - アプリケーション管理
 - ポリシー管理
5. [OK]をクリックします。

管理者に、アプリケーションとポリシーの管理権限が付与されました。

アプリケーションを保護する場合の EPM ユース ケース

以下のユース ケースを通して、EPM をアプリケーション セキュリティ ポリシーに適用する方法について説明します。

- [Web ポータルを保護するためのアプリケーション セキュリティ ポリシー](#) (P. 602)
- [ロールに基づくアプリケーション セキュリティ ポリシー](#) (P. 607)
- [ユーザ マッピングと名前付き式を使用したアプリケーション セキュリティ ポリシー](#) (P. 616)

Web ポータルを保護するアプリケーション セキュリティ ポリシー

このユース ケースのソフトウェア会社 (sample-software-company.com) は、会社概要と製品に関する情報を公開する Web ポータルを運営しています。

メイン ホーム ページと製品情報ページの販促資料やホワイト ペーパーには、だれでも制限なくアクセスできます。Web ポータルのこの領域には、セキュリティ ポリシーを必要としません。ただし、ソフトウェア ダウンロード領域へのアクセスは登録済みの顧客に制限します。各顧客にはユーザ名とパスワードが割り当てられ、LDAP ディレクトリ サーバに保存されます。

以下のユース ケースは、アプリケーション セキュリティ ポリシーで、登録済みの顧客のみがアクセスできるように制限してソフトウェア ダウンロード領域を保護する方法を示します。

例:

- SiteMinder 環境には、登録済みの顧客全員のユーザ名とパスワードが保存されている LDAP ディレクトリ サーバが 1 つあります。
- 顧客はユーザ名とパスワードで認証してからでないと、ソフトウェア ダウンロード領域にアクセスできません。

解決方法:

このユース ケースを解決する方法

1. 保護を必要とする Web ポータルを識別し、顧客情報を含むディレクトリを選択します。
2. ポータルのソフトウェア ダウンロード領域用のリソースを別に作成します。
3. 登録済みの顧客ロールを作成します。
4. 別に作成したリソースと登録済みの顧客ロールを関連付けて、アプリケーション セキュリティ ポリシーを作成します。

Web ポータルの識別とユーザ ディレクトリの選択

Web ポータルに対するアプリケーション セキュリティ ポリシーでは、保護するリソースの最上位のロケーション、およびリソースの使用を許可するユーザのディレクトリを指定する必要があります。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

Web ポータルを識別して、ディレクトリ サーバを選択する方法

1. [ポリシー]-[アプリケーション]-[アプリケーションの作成]をクリックします。
[アプリケーションの作成]ペインが開きます。
2. [一般]グループ ボックスのフィールドに値を入力します。以下の例に示すように、その目的または機能を思い出しやすい値を選択します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

名前

Sample Software Company Portal

説明

ダウンロード エリアを除く、ポータルのすべての部分へのアクセスを許可します。

3. [コンポーネント]グループ ボックスで、保護するリソースが格納されているディレクトリを指定します。この Web ポータルのユース ケースでは、以下のサンプルを使用します。

エージェントのタイプ

Web エージェント

エージェント

PortalAgent

リソース フィルタ

/downloads

4. その他の設定はデフォルトのままにします。
5. [ユーザ ディレクトリ]グループ ボックスで、[追加/削除]をクリックします。
[ユーザ ディレクトリの選択]ダイアログ ボックスが表示されます。
6. 関連するユーザが格納されているディレクトリを選択し、右向きの矢印をクリックして、[使用可能なメンバー]列から[メンバーの選択]列にそのディレクトリを移動します。
7. [OK]をクリックします。
[一般]タブに戻ります。
8. [サブミット]をクリックします。

Web ポータル アプリケーションが識別され、ディレクトリが選択されます。

Web ポータル リソースの作成

リソースの場所とユーザ ディレクトリを指定した後、Web ポータルのサブディレクトリにある保護するリソースを個別に指定する必要があります。

Web ポータル リソースを作成する方法

1. [リソース]タブをクリックします。
リソースの一覧が表示されます。
2. [作成]をクリックします。
[アプリケーションリソースの作成]ペインが表示されます。
3. [一般]グループ ボックスのフィールドに値を入力します。以下の例に示すように、その目的または機能を思い出しやすい値を選択します。

名前

ダウンロード領域

説明

ソフトウェアのダウンロードは登録済みの顧客に制限されている

リソース

*

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [有効なリソース]と保護するリソースが一致していることを確認します。このユース ケースで有効なリソースは以下のとおりです。

有効なリソース

/downloads/*

この文字列は、ダウンロード ディレクトリのすべてのリソースを保護することを指定します。

5. アクション グループ ボックスの Web エージェントのアクション ラジオ ボタンが選択されていることを確認し、アクションリストで以下の項目をクリックします。
 - Get
 - Post
6. [OK]をクリックします。

Web ポータル リソースが作成され、リソースリストに表示されます。

Web ポータル ロールの作成

Web ポータル リソースを指定してから、Web ポータルの登録済みの顧客用ロールを作成する必要があります。ロールはリソースとユーザ グループを関連付けます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

Web ポータル ロールを作成する方法

1. [ロール]タブをクリックします。
2. [作成]をクリックします。
3. [タイプ「ロール」のオブジェクトの作成]ボタンが選択されていることを確認し、[OK]をクリックします。
[ロールの作成]ペインが表示されます。
4. [一般]グループ ボックスのフィールドに値を入力します。以下の例に示すように、その目的または機能を思い出しやすい値を選択します。

名前

登録済みの顧客

説明

ソフトウェア ダウンロードへのアクセスが許可されている登録済みの顧客

式

TRUE

式エディタを使用して、このフィールドを完成させるか、式を入力できます。式エディタにアクセスするには、[編集]をクリックします。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

5. [OK]をクリックします。
登録済みの顧客のロールが作成されます。

Web ポータル ポリシーの作成

リソースとロールを作成してから、保護する Web ポータルのリソースを Web ポータルのリソースにアクセスするユーザのロールに関連付ける必要があります。これにより、アプリケーションを保護するポリシーが作成されます。

Web ポータルを保護するポリシーを作成する方法

1. [ポリシー]タブをクリックします。
2. ソフトウェアのダウンロード(Software Downloads)行で、「登録済み顧客」のロールを選択します。

このロールを選択することで、登録済みの顧客のみが Web ポータルのソフトウェア ダウンロード領域にアクセスできることを指定できます。

3. [サブミット]をクリックします。

確認の画面が表示されます。Web ポータルのアプリケーション セキュリティ ポリシーが作成されます。

リソースのリストの表示

リソースのリストの表示は、以下のいずれかのラジオ ボタンをクリックして並べ替えることができます。

名前

リソースを指定したときに設定した名前に従ってリソースを並べ替えます。

例: Software Downloads

フィルタ

保護されている実際のリソースに従ってリソースを並べ替えます。

例: * (アスタリスクはすべてのリソースを示します。)

ロールに基づくアプリケーション セキュリティ ポリシー

このユース ケースの金融サービス会社(acme-financial.com)では、手当と実績を管理する人事アプリケーションが使われています。このアプリケーションの手当の部分には、すべての従業員がアクセスする必要がありますが、実績管理の部分へのアクセスは管理者にのみ許可する必要があります。

以下の手順では、EPM モデルとアプリケーション ロールを組み合わせる人事アプリケーション用のセキュリティ ポリシーを作成する方法を詳しく示します。

例:

- SiteMinder 環境には、AcmeLDAP と呼ばれる 1 つのユーザ LDAP ディレクトリがあります。
- ユーザ ディレクトリは、すべての従業員、および管理者でもある従業員を識別します。これらについては、ディレクトリで以下のように定義します。
 - group:cn=employees、ou=Groups、o=acme-financial.com
 - group:cn=managers、ou=Groups、o=acme-financial.com
- 管理者を含むすべての従業員は、基本認証方式で認証する必要があります。

ロールに基づいたアプリケーション セキュリティの解決策は以下のとおりです。

このユース ケースを解決するには、以下の手順を完了します。

1. ユーザ ディレクトリに属性ディレクトリ マッピングを作成します。
2. アプリケーションを作成します。
3. ロール基準を満たすユーザを検索するユーザ ディレクトリを選択します。
4. メイン アプリケーションのサブコンポーネントであるリソースを指定します。
5. アプリケーションにアクセスできる 2 つのロールを定義します。
6. リソースとロールをアプリケーション ポリシーに組み合わせます。

保護を必要とするアプリケーションの識別

このユース ケースでは、人事アプリケーションの各部分に別々のアクセス権限を設定する必要があります。このためには、メイン アプリケーション下にあるディレクトリを識別して、適切なアクセス権を設定する必要があります。

サンプルの人事アプリケーションを保護する方法

1. [ポリシー]-[アプリケーション]をクリックします。
2. [アプリケーションの作成]をクリックします。
[アプリケーションの作成]ペインが開きます。
3. [一般]タブをクリックします。

4. [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは、以下のデータを指定します。

名前

HR アプリケーション

説明

社内の人事アプリケーションを識別します。

5. [コンポーネント]グループ ボックスの各フィールドに値を入力します。このユース ケースでは、以下のデータを指定します。

エージェントのタイプ

Web エージェント

エージェント

hrportal agent

リソース フィルタ

/benefits

デフォルト リソース保護

保護

認証方式

基本

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. 保護されているリソースに関連付けられているユーザ ディレクトリを指定します。このディレクトリで **SiteMinder** はロール基準を満たすユーザを探します。
 - a. [追加/削除]をクリックします。
 - b. [使用可能なメンバー]ボックスから[**Employees**]を選択し、右向きの矢印をクリックして、このグループを[メンバーの選択]ボックスに移動します。
 - c. [OK]をクリックします。

これで、人事アプリケーションが識別されました。

グループ メンバシップの属性マッピングの作成

Acme-Financial の一部の従業員が管理者であることを示すために、管理者グループ メンバシップにマップする属性マッピングを作成します。

注: このユース ケースの LDAP ユーザ ディレクトリの名前は、AcmeLDAP です。

グループ メンバシップの属性マッピングを作成する方法

1. [インフラストラクチャ]、[ディレクトリ]、[ユーザ ディレクトリ]、[ディレクトリの変更]をクリックします。

2. 変更するディレクトリを選択します。

AcmeLDAP を変更できるダイアログが表示されます。

3. 属性マッピングリストの[作成]をクリックします。

[属性マッピングの作成]ダイアログが表示されます。

4. [一般]グループ ボックスのすべてのフィールドに以下のように入力します。

名前

IsManager

説明

管理者グループを定義します。

5. [プロパティ]グループ ボックスの[グループ]ラジオ ボタンを選択し、以下を入力します。

定義

cn=managers、ou=Groups、o=acme-financial.com

[定義]フィールドの値は、ユーザ ディレクトリでグループが定義されている方法(ユース ケースの最初に示した方法)を示します。

6. [OK]をクリックします。

ユーザ ディレクトリ ダイアログに戻ります。

7. [サブミット]をクリックして、変更をサブミットします。

これで、グループ cn=managers、ou=Groups、o=acme-financial.com の IsManagers と呼ばれる属性が定義されました。

アプリケーション リソースの指定

保護するメイン アプリケーションのサブ領域を指定したら、アプリケーション ポリシーで保護するそのサブディレクトリ内の特定のリソースを指定できます。

このユース ケースでは、保護するリソースが 2 つあります。

- 福利厚生管理
- 勤務評価

メイン アプリケーションの特定のリソースまたは機能を指定する方法

1. [リソース]タブをクリックします。
2. [作成]をクリックします。
[リソース]ペインが開きます。
3. [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

Benefits Management

説明

従業員が各自の福利厚生を管理できます。

4. [属性]グループ ボックスの各フィールドに、値を入力します。このユース ケースでは以下を入力します。

リソース

managebenefits.jsp

- 手順 2 から 4 を繰り返します。ただし、以下の情報を入力します。

名前

勤務評価

説明

マネージャが従業員の評価報告書および給与査定を作成できます。

リソース

salaryincrease.jsp

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

これで、勤務評価アプリケーションに関連付けられたリソースが定義されました。

従業員とマネージャのロールの作成

保護を必要とするアプリケーションの固有コンポーネントを定義したら、ユーザーに割り当てることのできるロールを指定できます。ロールとは、特定のリソースへのアクセス権を持つユーザの集合です。ユーザのこれらの集合は、式で定義します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

ロールを作成する方法

- [ロール]タブをクリックします。
- [作成]をクリックします。
[ロールの作成]ペインが表示されます。
- [作成]ラジオ ボタンが選択されていることを確認し、[OK]をクリックします。
- [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

従業員

説明

Acme Financial Services のすべての従業員

5. [メンバシップ]グループ ボックスに、式を入力します。このユース ケースでは以下を入力します。

式

TRUE

式を作成するには、[式エディタ](#) (P. 264)を使用することができます。このエディタにアクセスするには、[編集]をクリックします。

6. [サブミット]をクリックします。
7. 手順 2 から 4 を繰り返して、以下のように「マネージャ」という 2 つ目のロールを作成します。

名前

マネージャ

説明

Acme Financial Services のマネージャ

式

BOOLEAN(IsManager)

IsManager は、LDAP ユーザ ディレクトリに対して定義された属性マッピングです。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

アプリケーションにデータを渡すレスポンスの使用

Acme Financial Services の従業員が人事アプリケーションを簡単に使えるように、各従業員の手当レコードの従業員 ID を渡すレスポンスを設定できます。

従業員 ID を渡すレスポンスを作成する方法

1. [アプリケーション]ダイアログの[レスポンス]タブをクリックします。
2. [レスポンスの作成]をクリックします。
[レスポンスの作成]ダイアログが表示されます。

3. フィールドには以下のように入力します。

名前

Employee ID

説明

従業員 ID を一覧表示します。

4. [レスポンス属性の作成]をクリックします。
[レスポンス属性の作成]ダイアログが表示されます。
5. フィールドには以下のように入力します。

属性

WebAgent-HTTP-Header-Variable

属性の種類

ユーザ属性

属性フィールド - 変数名

Personnel_Key

属性フィールド - 変数値

EmployeeID

注: レスポンス属性の詳細については「Web エージェント設定ガイド」を参照してください。

6. 他のすべてのフィールドはデフォルトのままにします。
7. メインの[レスポンス]タブに戻るまで[OK]をクリックします。

従業員 ID という名前のレスポンスが作成されました。従業員が自分の手当情報を表示するとき、このレスポンスから得られたデータが人事アプリケーションに返され、手当レコードにこの従業員の顧客 ID が表示されます。

ルールに基づいたポリシーの確立

リソースとルールを定義したら、これらのオブジェクトをアプリケーション セキュリティ ポリシーにグループ化することができます。

アプリケーション セキュリティ ポリシーを作成する方法

1. [ポリシー]タブをクリックします。
[ポリシー]ペインが開き、設定されたリソースとルールを示すテーブルが表示されます。このテーブルを見れば、どのルールがどのリソースへのアクセスを許可されているかをひとめで把握できます。
2. 以下の手順を実行します。
 - a. [Benefits Management]行で[Employees]ロールのチェック ボックスをオンにして、すべての従業員に各自の福利厚生を管理することを許可するポリシーを作成します。
 - b. [Performance Appraisals]行で[Managers]ロールのチェック ボックスをオンにして、マネージャのみに勤務評価にアクセスすることを許可するポリシーを作成します。
3. [サブミット]をクリックします。

ルールに基づいて、人事アプリケーション用の 2 つのセキュリティ ポリシーを作成しました。

注: リソースまたはルールを編集する必要がある場合は、[ポリシー]ペインではなく、各タブで変更を行う必要があります。

アプリケーションを説明するメタデータを含める

Acme-financial.com は、社内の人事アプリケーションに関する説明情報を必要としています。カスタム属性を使用して、アプリケーションを説明するメタデータを定義できます。

Acme-financial で必要としている情報は、アプリケーションの用途とアプリケーションが完了した日付です。

人事アプリケーションのメタデータを指定する方法

1. [カスタム属性]タブをクリックします。
[カスタム属性]ダイアログが表示されます。
2. [作成]をクリックします。
[名前]と[値]フィールドがあるテーブルが表示されます。
3. カスタム属性テーブルのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

App_Completed

値

November_22_2007

4. [作成]をクリックしてテーブルに別の行を追加し、さらに以下を入力します。

名前

Purpose

値

Human_Resource_Mgmt

5. [サブミット]をクリックします。

ユーザ マッピングと名前付き式を使用したアプリケーション セキュリティ ポリシー

このユース ケースでは、衣料品小売会社で、顧客がクレジット限度を超えたら Web 上でクレジットによる購入を行えないようにするルールを定義します。この会社のポリシーでは、顧客のクレジット限度を 1,000ドルとし、従業員のクレジット限度を 2,000ドルとしています。

属性マッピング、名前付き式(仮想ユーザ属性とユーザ クラス)、およびルールを使用して、会社のクレジット ポリシーを満たすように、アプリケーション セキュリティ ポリシーを作成することができます。

例:

- SiteMinder 環境には 2 つのユーザ ディレクトリが含まれています。
- ディレクトリ A には、従業員が格納されます。従業員は顧客になることもあります。そのため、ディレクトリ A では、以下に属する従業員は顧客でもあると識別されます。

`group:cn=Customers,ou=Groups,o=acme.com`

- ディレクトリ B は顧客のみ格納します。ディレクトリ B にユーザを格納することは、そのユーザが顧客であるという意味なので、ディレクトリ B に顧客を識別するユーザ属性はありません。

解決方法:

1. 属性マッピングを定義します。
2. 名前付き式を設定します。
3. 属性マッピングと式を使用して、ロールを設定します。
4. レスポンスを作成して、アプリケーションをさらにカスタマイズします。
5. アプリケーション セキュリティ ポリシーを作成します。

2 つのユーザ ディレクトリのマッピングの確立

小売会社には 2 つのディレクトリがあります。両方のユーザ ディレクトリで顧客を識別するユニバーサルなスキーマを作成するには、属性マッピングを使用します。属性マッピングは、管理 UI で作成します。

このユース ケースの属性マッピングを作成する方法

1. ディレクトリ A にはグループ メンバシップ属性を作成します。
 - 属性の名前を **IsCustomer** とします。
 - **IsCustomer** を **cn=Customers,ou=Groups,o=acme.com** として定義します。
2. ディレクトリ B には定数属性を作成します。
 - 属性の名前を **IsCustomer** とします。
 - **IsCustomer** を **"TRUE"** として定義します。

IsCustomer により、同じユーザ情報の共通のビューが生成されます。式で **IsCustomer** を参照して、ユーザが顧客かどうかを判別することができます。

属性マッピングを設定する方法の詳細な手順については、「[属性マッピングの定義 \(P. 277\)](#)」のセクションを参照してください。

クレジット限度をチェックするための名前付き式の定義

名前付き式を使用して、**SiteMinder** が各ユーザのクレジット限度と口座残高を計算できるようにします。式では、顧客がクレジット限度を超えているかどうかを判別することもできます。

このユース ケースの名前付き式を定義する方法

1. 仮想ユーザ属性を定義します。このユーザ属性は、顧客の場合は 1,000 ドルのクレジット限度を計算し、従業員の場合は 2,000 ドルのクレジット限度を計算します。
 - 属性の名前を **#CreditLimit** とします。
 - **#CreditLimit** を以下のように定義します。

```
IsCustomer?1000 2000
```

この計算には、**SiteMinder** でサポートされている式構文が含まれています。

2. アカウンティング データベースから口座残高を取得する仮想ユーザ属性を定義します。

- 属性の名前を **#Balance** とします。
- この属性を次のように定義します。

```
(MyLibrary.GetBalance(""))
```

この属性定義は衣類小売り業者によって定義されたアクティブな式です。

3. 顧客がそれぞれの信用限度を超えているかどうかを判断する、ユーザ クラス式を作成します。

- 属性に **@IsUnderCreditLimit** という名前を付けます。
- この属性を次のように定義します。

```
(#Balance > #CreditLimit)
```

仮想ユーザ属性およびユーザ クラス式の作成の詳細については、「[名前付き式の定義](#) (P. 258)」を参照してください。

オンライン ショッピング アプリケーションの保護

このユース ケースでは、ストアの Web ベースのショッピング アプリケーションの特定の条件に従ってアクセス権限を設定します。

Web ベースのショッピング アプリケーションを保護する方法

1. [ポリシー]-[アプリケーション]をクリックします。
2. [アプリケーションの作成]をクリックします。
[アプリケーションの作成]ペインが開きます。
3. [一般]タブをクリックします。
4. [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは、以下のデータを指定します。

名前

Online Catalog

説明

衣料品ストアの Web ベースのショッピング アプリケーションを識別します。

5. [詳細]グループ ボックスの各フィールドに、値を入力します。このユース ケースでは、以下のデータを指定します。

エージェントのタイプ

Web エージェント

エージェント

Web 小売エージェント

リソース フィルタ

/webcatalog

デフォルト リソース保護

保護

認証方式

基本

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

6. 保護されているリソースに関連付けられているユーザ ディレクトリを指定します。このディレクトリで **SiteMinder** はロール基準を満たすユーザを探します。
 - a. [追加/削除]をクリックします。
 - b. [使用可能なメンバー]ボックスから[IsCustomer]を選択し、右向きの矢印をクリックして、このグループを[メンバーの選択]ボックスに挿入します。

IsCustomers は、衣料品ストアに関連付けられている両方のディレクトリ内のユーザにマッピングされます。
 - c. [サブミット]をクリックします。

これで、オンライン カタログという新しいアプリケーションが作成されました。

保護を必要とするリソースの指定

このユース ケースでは、クレジット限度を超えたユーザがトランザクションを完了できないようにチェックアウトプロセスを保護します。そのため、作成したオンライン カタログ アプリケーションにリソースを追加する必要があります。

Web ベースのショッピング アプリケーションの特定のリソースを保護する方法

1. [ポリシー]-[アプリケーション]-[アプリケーション]-[アプリケーションの変更]をクリックします。
2. [検索]をクリックし、オンライン カタログ アプリケーションを選択します。[選択]をクリックします。
[アプリケーションの変更]ダイアログ ボックスが開きます。
3. [リソース]タブを選択します。
4. [リソース]グループ ボックスで[作成]をクリックします。
[リソースの作成]ダイアログ ボックスが表示されます。
5. [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

Checkout

説明

購入額を合計して代金を支払うことができます。

6. [属性]グループ ボックスの各フィールドに、値を入力します。このユース ケースでは以下を入力します。

リソース

total_charges.jsp

7. [アクション]グループ ボックスで **Web** エージェント アクションのラジオ ボタンを選択し、アクション **Get** および **Post** を選択します。
8. [OK]をクリックします。

Checkout というリソースが作成されました。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

顧客ロールの設定

リソースを設定したら、クレジット限度を超えていない限り顧客が Web ベースの購入を行うことを可能にするアプリケーション ロールを作成します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

このクレジット ベースのロールを作成する方法

1. [ロール]タブをクリックします。
2. [作成]をクリックします。
[ロールの作成]ダイアログ ボックスが表示されます。
3. [作成]ラジオ ボタンが選択されていることを確認し、[OK]をクリックします。
[ロールの作成]ダイアログ ボックスが開きます。
4. [一般]グループ ボックスのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

PurchasewithCredit

説明

顧客がクレジットを使用して購入代金を支払うことを示します。

5. [メンバシップの式]グループ ボックスに式を入力します。このユース ケースでは以下を入力します。

式

@IsUnderCreditLimit

ロール式は、#Balance と #CreditLimit の 2 つの仮想ユーザ属性式の積です。これらの式によって、ユーザがクレジット限度を超えているかどうかを計算します。

6. [OK]をクリックします。

PurchasewithCredit というロールが作成されました。このロールの値は、2 つの名前付き式の組み合わせです。

レスポンスによるアプリケーションのカスタマイズ

さらにパーソナライズされた操作性を顧客に提供するために、衣類小売り業者は信用限度を超えている顧客が信用限度の引き上げを申請できるレスポンスを設定できます。このレスポンスは、信用限度を超えている顧客を限度の引き上げを申請できる借入申込書にリダイレクトします。

レスポンスを作成する方法

1. [レスポンス]タブをクリックします。
2. [レスポンスの作成]をクリックします。
[レスポンスの作成]ダイアログが表示されます。
3. フィールドには以下のように入力します。

名前

CreditNotice

説明

信用限度を超えていることをユーザにアラートします。

4. [レスポンス属性の作成]をクリックします。
[レスポンス属性の作成]ダイアログが表示されます。
5. 以下のようにフィールドと設定を完了します。

属性

WebAgent-OnReject-Redirect

属性の種類

スタティック

属性フィールド - 変数値

http://catalog.retailcorp.com/credit_notice.jsp

注: レスポンス属性の詳細については「Web エージェント設定ガイド」を参照してください。

6. 他のすべてのフィールドはデフォルトのままにします。
7. [OK]をクリックします。

CreditNotice という名前のレスポンスが作成され、信用限度を超えている顧客に送信されます。

ショッピング アプリケーションに対するセキュリティ ポリシーの設定

リソース、ロール、およびレスポンスを定義したら、Web ベースのショッピング アプリケーションを保護するポリシーを設定します。

アプリケーション セキュリティ ポリシーを作成する方法

1. [ポリシー]タブをクリックします。
[ポリシー]ダイアログ ボックスが開き、Checkout リソースと PurchaseWithCredit ロールを示すテーブルが表示されます。
2. Checkout リソースに対して PurchaseWithCredit ロールを選択します。
この組み合わせにより、クレジット限度を超えていない限り、すべての顧客がストアのクレジットカードを使用して購入することを許可するポリシーが設定されます。さらに、ロールのチェック ボックスをオンにすると、[レスポンス]のグリッドにデータが設定されます。
3. Checkout リソースに対して CreditNotice レスポンスを選択します。

これで、支出限度を定義するルールに基づくオンライン カタログ アプリケーションのセキュリティポリシーが設定されました。さらに、レスポンスがポリシーに関連付けられているので、限度を超えた後も購入を続けている顧客にレスポンスが送信されます。

アプリケーションを説明するメタデータの指定

衣類小売り業者は、オンライン カタログ アプリケーションに関する説明情報を必要としています。カスタム属性を使用して、アプリケーションを説明するメタデータを指定できます。

衣類小売り業者は、このアプリケーションの管理者の電子メール アドレスを記載すると共に、アプリケーションがオンライン カタログ専用であることを説明したいと考えています。

オンライン カタログ アプリケーションのメタデータを指定する方法

1. [カスタム属性]タブをクリックします。
[カスタム属性]ダイアログが表示されます。
2. [作成]をクリックします。
[名前]と[値]フィールドがあるテーブルが表示されます。

3. カスタム属性テーブルのフィールドに値を入力します。このユース ケースでは以下を入力します。

名前

App_Function

値

online_retail

4. [作成]をクリックしてテーブルに別の行を追加し、さらに以下を入力します。

名前

Admin_email

値

jdcoe@retailcorp.com

5. [サブミット]をクリックします。

アプリケーション セキュリティ ポリシーの作成に関連するすべてのタスクを完了しました。

アプリケーションの詳細なポリシー コンポーネント

EPM に用意されている設定オプションを使用することで、以下のタイプのユーザは **SiteMinder** コンポーネントをデフォルト以外の設定に変更することができます。

- r12 より前のリリースの **SiteMinder** で使用されていたポリシー設計とインターフェースに精通していて、そのポリシーの設定を微調整する必要があるユーザ。
- デフォルトの設定よりもポリシーの詳細レベルを上げる必要があるユーザ。
- EPM で実装されたポリシーが、組織の要件またはその組織が準拠する必要がある規制の要件を満たしているかどうかを判別する必要がある監査者またはその他の担当者。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

詳細オプションを使用してポリシーを作成する方法

1. [ポリシー]-[アプリケーション]-[アプリケーションの作成]をクリックします。
[アプリケーションの作成]ペインが表示されます。
2. [一般]セクションと[コンポーネント]セクションに情報を入力し、[詳細]をクリックします。
[詳細]ペインが表示されます。
3. 以下のタスクのいずれかを実行します。
 - 新しいレلمを作成するには、[新規レلم]をクリックします。
 - 既存のレلمを変更するには、リストで、変更するレلمを見つけて、その名前の横にある編集矢印をクリックします。
4. レلمの詳細オプションを設定します。
5. 設定を終了したら、[OK]をクリックして変更を保存します。
[詳細]ペインが表示されます。ポリシーの詳細オプションが設定されています。
6. [閉じる]をクリックします。
[アプリケーションの作成]ペインが表示されます。
7. ポリシーのその他の部分の設定を続けます。

第 19 章：変数

このセクションには、以下のトピックが含まれています。

[eTelligent ルール](#) (P. 627)

[変数の概要](#) (P. 632)

[Web サービス変数](#) (P. 637)

[変数の作成](#) (P. 642)

eTelligent ルール

eTelligent ルールを使用することで、非常に詳細なアクセス制御基準を有効にする変数を定義することができます。これを、ポリシー式といいます。

ポリシー式は、ポリシー属性として実装されます。ポリシー式には、演算子とユーザ定義の変数が含まれます。これらの演算子と変数は、実行時、つまりユーザが実際に **Web** サイト上の保護されたリソースにアクセスする必要があるときに評価されます。

変数には、社内にあるローカル情報だけでなく、各種 **Web** サービスから提供されるリモート情報も格納できます。

eTelligent ルールで提供される変数は、管理 UI で使用することができます。変数オブジェクトを定義し、ポリシー式を使用してポリシー ロジックにそれらを組み込むことができます。また、変数は **SiteMinder** レスポンス オブジェクトに組み込むこともできます。

eTelligent ルールのコンポーネント要件

eTelligent ルールを使用するには、以下のコンポーネントが必要です。

- Web エージェント
- (オプション) Web エージェント オプション パック

Web エージェント オプション パックは、POST 変数を使用する予定の場合のみ必要です。

注: Web エージェントのインストールの詳細については、「*Web エージェント インストール ガイド*」を参照してください。Web エージェント オプション パックのインストールの詳細については、「*Web Agent Option Pack Guide*」を参照してください。

SiteMinder eTelligent ルールのメリット

- 複雑さが軽減され、カスタム コードを作成する必要がなくなります。
許可アクセスは、アプリケーション コードではなくグラフィカル ツールを使用して、ポリシー式で SiteMinder 管理者によって定義されます。バックエンド ビジネス アプリケーションのアクセス制御情報は SiteMinder ポリシー サーバで一元管理されるため、統合して調整する必要がありません。
- セキュリティ ポリシーでビジネス データを動的に使用できます。
リソースを保護するためのアクセス制御は、ローカル ユーザ情報と、発注金額などのユーザ入力情報に基づいて定義されます。
- 許可を決定するために、さまざまなタイプの情報を組み合わせることができます。
Web ブラウザのフォーム データや、ユーザ コンテキスト データ (ポリシー サーバにローカルに格納)、およびリモート データ (サービス機関を介して取得) を、ポリシー式で柔軟に組み合わせることができます。
- トランザクションの決定をオンラインで行うことができます。
保護されたリソースにアクセスするための許可が必要になるたびに、バックエンド ビジネス アプリケーションに戻る必要がありません。
- XML ベースのサードパーティ セキュリティ データを利用できます。
eTelligent ルールでは、信頼できるサービス機関との通信に標準の XML プロトコルを使用するため、Web サービス プロバイダの選択肢が広がります。

- ブール ロジックを使用できます。
SiteMinder セキュリティ管理者は、変数と論理演算子を組み合わせて使用して、ポリシー式を定義できます。
- 必要なポリシーの数を最小限に抑えます。
ロジックに基づいたポリシー式を使用することにより、必要なポリシーが少なくなり、ポリシー管理の負担が最小限に抑えられます。

eTelligent ルールの設定

eTelligent ルールを設定するために必要なタスクは、以下のとおりです。

- 変数を設定します。
- eTelligent ルールの変数を使用するポリシー式を設定します。
変数およびポリシー式は、管理 UI を使用して設定します。
- 以下の eTelligent ルールのプロパティファイルを変更します。
 - JVMOptions.txt
 - LoggerConfig.properties変更できるのは、LoggerConfig.properties ファイルのみです。

eTelligent ルールのプロパティ ファイル

eTelligent ルール用のプロパティファイルは次のとおりです。

- JVMOptions.txt
これは eTelligent ルールの必須ファイルです。このファイルのインストールロケーションは、*policy_server_home/config/* です。
- LoggerConfig.properties
このファイルは、eTelligent ルールのロギングの設定に必要です。このファイルは、次の場所にインストールされます。
policy_server_home/config/properties

JVMOptions.txt ファイル

JVMOptions.txt ファイルには、ポリシー サーバが Java 仮想マシンを作成するときに使用する設定が含まれています。Java 仮想マシンは、eTelligent ルールのサポートに使用されます。

必要なクラスが欠落しているというエラーが表示された場合には、**JVMOptions.txt** ファイルの **classpath** 命令文を変更する必要があります。**JVMOptions.txt** ファイルに含まれる設定の詳細については、**Java** のマニュアルを参照してください。

LoggerConfig.properties ファイルの変更

ポリシー サーバでは、**LoggerConfig.properties** ファイルを使用して、コマンドラインから **SiteMinder** サービスを起動するときに使用されるログ機能を指定することができます。ポリシー サーバ管理コンソールから **SiteMinder** サービスを起動した場合には、このファイルに含まれるプロパティは使用されません。通常、このファイルの設定はデバッグ時にのみ使用されます。

このファイルを修正して、デバッグ時により多くの出力を取得できるようにすることができます。

LoggerConfig.properties ファイルの例を、以下に示します。

```
// 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.
詳細については、Java のマニュアルの java.text.DateFormat を参照してください。
指定がない場合は、デフォルト ロケールのデフォルト フォーマットが使用されます。
DateFormatPattern=MMM d, yyyy h:mm:ss.S a
```

このファイルの設定は次のとおりです。

LoggingOn

ログ機能を有効または無効に設定します。ログ機能を有効にするには、このパラメータを **Y** に設定します。ログ機能を無効にするには、このパラメータを **N** に設定します。

LogLevel

このログに記録される詳細度を指定します。**LogLevel** には、次のいずれかを指定できます。

LOG_LEVEL_NONE

メッセージは記録されません。

LOG_LEVEL_ERROR

エラーメッセージのみが記録されます。

LOG_LEVEL_INFO

エラーメッセージと警告が記録されます。

LOG_LEVEL_TRACE

エラーメッセージと警告に加え、問題を追跡する場合に有用な全般的な処理情報も記録されます。

LogFileName

LogFileName を設定すると、このパラメータで指定されたファイルにすべてのログ出力が送られます。

AppendLog

起動時に既存のファイルにログ情報を追加するか、新しいファイルを作成するかを指定します。**LogFileName** パラメータで指定されたファイルに出力を追加するには、このパラメータを **Y** に設定します。起動時に新しいファイルを作成する場合は、このパラメータを **N** に設定します。

AlwaysWriteToSystemStreams

ログestreamの設定にかかわらず、**System.out** または **System.err** にメッセージを記録するには、このパラメータを **Y** に設定します。ログestreamが **System.out** または **System.err** に設定されている場合には、ログメッセージは重複して記録されます。これにより、**System.out/System.err** とファイルにメッセージを同時に記録できます。

DateFormatPattern

DateFormatPattern には、`java.text.DateFormat` コンストラクタへの有効な入力をどれでも指定できます。詳細については、Java のマニュアルの `java.text.DateFormat` を参照してください。

指定がないと、デフォルトロケールのデフォルトフォーマットが使用されます。

変数の概要

ポリシー サーバでは、変数とは、1 つの値に解決して、その値をリクエストの許可プロセスに組み込むことができるオブジェクトのことです。変数オブジェクトの値は、動的データの結果であり、実行時に評価されます。変数は、ポリシーやレスポンスの機能を拡張するための柔軟な手段として利用できます。

変数タイプ

使用できる変数のタイプは次のとおりです。

- [静的変数](#) (P. 632)
- [要求コンテキスト変数](#) (P. 633)
- [ユーザ コンテキスト変数](#) (P. 633)
- [フォーム ポスト変数](#) (P. 633)
- [Web サービス変数](#) (P. 634)

静的変数

スタティック変数は、文字列、ブールなど、特定のタイプの単純な名前/値ペアで構成されます。スタティック変数の主なメリットは、正しいプログラミング手法を実践できることです。スタティック変数を使用すれば、複数のポリシーで 1 つのデータを使用できるため、各ポリシーで定数の値を使用するたびに同じ値を繰り返し入力する必要がなくなります。

要求コンテキスト変数

SiteMinder が要求を処理するたびに、要求コンテキストが確立されます。このコンテキストは以下の項目を特定します。

アクション

GET または POST など、要求で指定されたアクションのタイプを示します。

リソース

/directory_name/ などの、要求されたリソースを示します。

サーバ

server.example.com などの、要求で指定されたサーバの完全名を示します。

コンテキスト要求の変数を使用すれば、これらのどの情報でも取得して、ポリシー式やレスポンスに含めることができます。このタイプの変数の主なメリットは、一切プログラミングロジックを使用せずに、きめ細かなコンテキスト要求情報を提供できることです。

ユーザ コンテキスト変数

ポリシー サーバがディレクトリ内のエントリに対してユーザを認証すると、ユーザコンテキストが作成されます。ユーザコンテキストは、ユーザディレクトリと、認証されたユーザに関連するディレクトリの内容に関する情報で構成されます。

ユーザコンテキスト変数には、ディレクトリ接続の属性、またはディレクトリの内容に基づく値を指定できます。このタイプの変数の主なメリットは、プログラミングロジックを一切使用せずに、特定のユーザコンテキストに基づいて柔軟にルールを定義できることです。

フォーム ポスト変数

通常、バックエンドアプリケーションに必要な情報の収集には、HTML フォームが使用されます。フォーム ポスト変数を使用すれば、HTML フォームや POSTed に入力されたあらゆる情報を取得できます。たとえば、アプリケーションに関連付けられたビジネスロジックで、アプリケーションへのログインに使用する HTML フォームに発注金額を指定する必要がある場合には、ユーザが入力した発注書の値を収集するフォーム POST 変数オブジェクトを作成します。この変数をポリシーに使用できます。

重要: フォーム ポスト変数は EJB やサーブレット エージェントではサポートされていません。したがって、EJB やサーブレットエージェントで実行されるポリシーには、フォーム **POST** 変数を使用しないでください。

このタイプの変数の主なメリットは、ポリシー サーバでポリシー式の一部に **POST** データを使用できることです。これにより、バックエンドのサーバ アプリケーションにセキュリティロジックを組み込む必要がなくなります。**HTTP POST** 変数を使用すれば、エージェントとポリシー サーバ間のネットワークの利用が効率化されます。エージェントは **HTTP** 変数の情報を **HTTP** ストリームから取得するだけで済み、ポリシー サーバはこの情報を許可の処理時に使用できます。

Web サービス変数

Web サービス変数は、ポリシーまたはレスポンスで使用する情報を **Web** サービスから取得する際に使用できます。このタイプの変数の主なメリットは、ポリシー サーバ管理者が **Web** サービスからリアルタイムで提供される動的な顧客情報に基づいてポリシーを定義できることです。

ポリシーでの変数の使用

変数を使用すれば、多様な動的データを取得してポリシー式に組み込むことで、ポリシーにビジネスロジックを含めることができます。管理 UI で変数オブジェクトを定義すると、[ポリシー]ダイアログ ボックスの[式]タブにある式でこれらの変数を使用することができます。複数の変数オブジェクトやブール演算子を使用して式を組み立てれば、非常に複雑なビジネスロジックをポリシーに組み込むことができます。

たとえば、あるアプリケーションへのアクセスを許可する条件として、ユーザのアカウントタイプの値と信用度の入力を求める式をポリシーに含めることができます。アカウントタイプが「gold」で、信用度が特定の値より高いユーザのみがソースへのアクセスを許可されるような式をポリシーに定義することができます。この例では、2 つの変数が必要です。これらの変数は、[ポリシー]ダイアログ ボックスの[式]タブにある式で組み合わせる必要があります。

レスポンスでの変数の使用

変数はレスポンスにも使用できます。管理 UI で変数オブジェクトを定義すると、これらの変数をレスポンスに使用することができます。レスポンスの値は、実行時にポリシー サーバが変数オブジェクトの値を解決した時点で作成されます。

ポリシー サーバが変数进行处理する方法

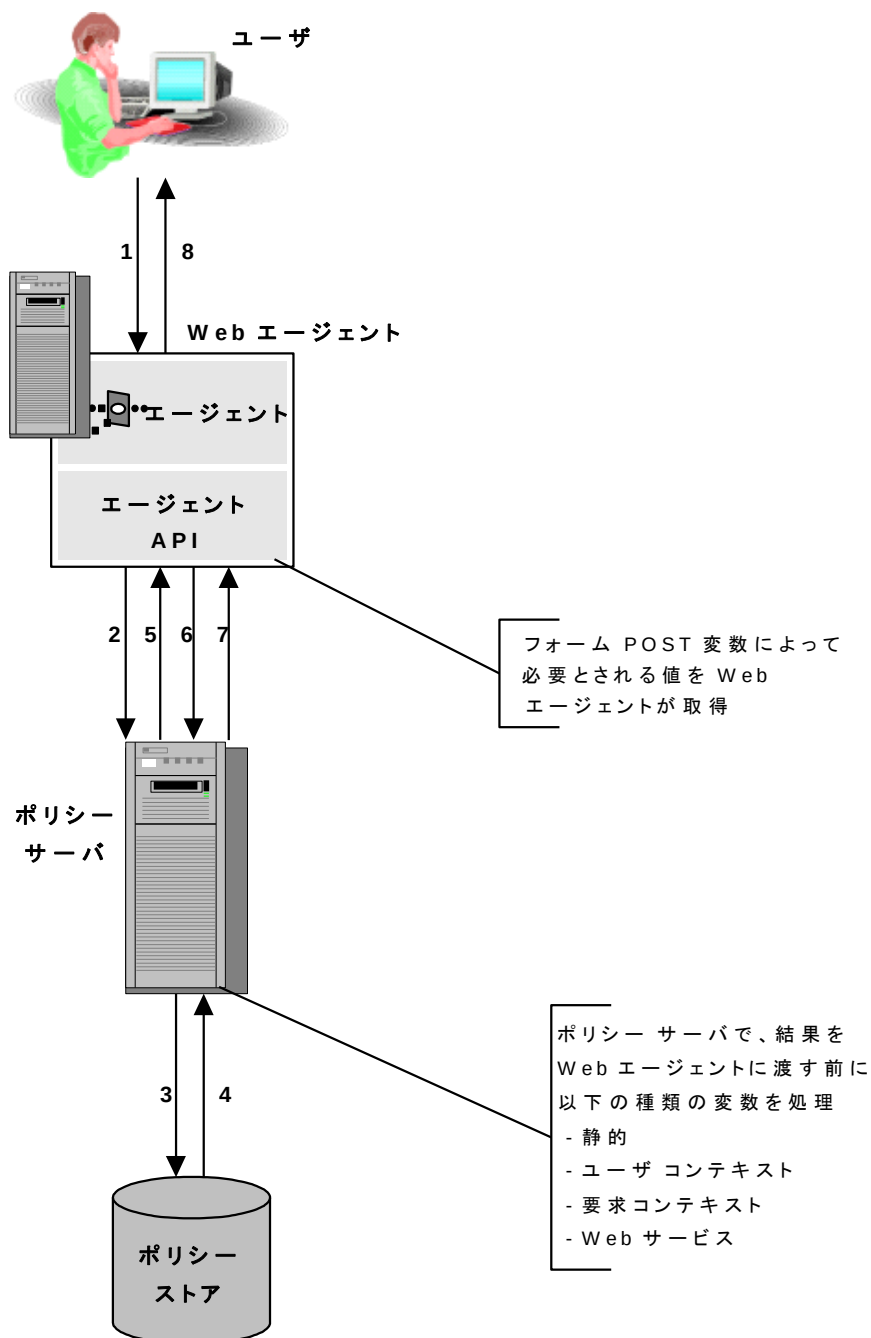
変数は、ポリシー サーバが要求の認証进行处理し、要求されたリソースに対してユーザのアクセスを許可するかどうかを判断するときに評価されます。変数処理の仕組みは、変数オブジェクトがレスポンスまたはポリシー式のどちらに含まれているかによって若干異なります。

ポリシー サーバがポリシー式に含まれている変数进行处理する方法

ポリシー評価において、ポリシー式に含まれている変数は、要求の認証時に未解決の変数リストに追加されます。ポリシー サーバが変数を解決すると、その変数は解決済みの変数リストに移動されます。ポリシー式に含まれるすべての変数の解決が終わると、ポリシー サーバはそのポリシー全体の内容に基づいて、アクセスを許可または拒否します。

以下の図に、要求されたリソースのポリシーに変数が含まれている場合に、Web エージェントとポリシー サーバによってユーザのリクエストの許可がどのように処理されるかを示します。この図には、Web サービスの変数は含まれていません。

注: 以下の図のプロセスでは、ユーザはすでに SiteMinder で認証済みであると見なします。認証されていないユーザの場合には、認証プロセスを経てから許可プロセスに進む必要があります。



1. ユーザは、SiteMinder Web エージェントによって保護されているリソースをサーバに要求します。
2. エージェントはリソースが保護されていることを確認し、ポリシー サーバが許可処理を開始します。
3. ポリシー サーバがポリシーストアから、要求されたリソースについてのポリシー情報を取得します。
4. ポリシー サーバは、要求されたリソースに関連付けられたポリシー式に含まれている未解決変数のリストを受け取ります。ポリシー サーバが、未解決変数リストに含まれているスタティック変数、ユーザ コンテキスト変数、およびコンテキスト要求の変数を評価します。
5. すべての変数と変数式の解決が終わると、ポリシー サーバは要求されたリソースへのユーザのアクセスを許可するかどうかを Web エージェントに伝えます。
6. 未解決変数リストにまだ未解決の変数が残っている場合、リストは **Not Resolved** インジケータと共に Web エージェント API レイヤーに渡されます。フォーム ポスト変数の値は Web エージェントによって解決され、解決済み変数リストのフォーム ポスト変数の値が含まれた新しい要求としてポリシー サーバに渡されます。
7. ポリシーにフォーム ポスト変数が含まれていると、ポリシー サーバは POST データから取得し新たに解決された値を使用してポリシーを処理します。
8. ユーザは要求したリソースへのアクセスを許可または拒否されます。

ポリシー サーバがレスポンスに含まれている変数を処理する方法

SiteMinder では、前述のセクションの説明どおりに、レスポンスに含まれている変数を処理します。フォーム ポスト変数はレスポンスには使用できないため、ポリシー サーバでレスポンスが起動する時点で変数はすべて処理されています。

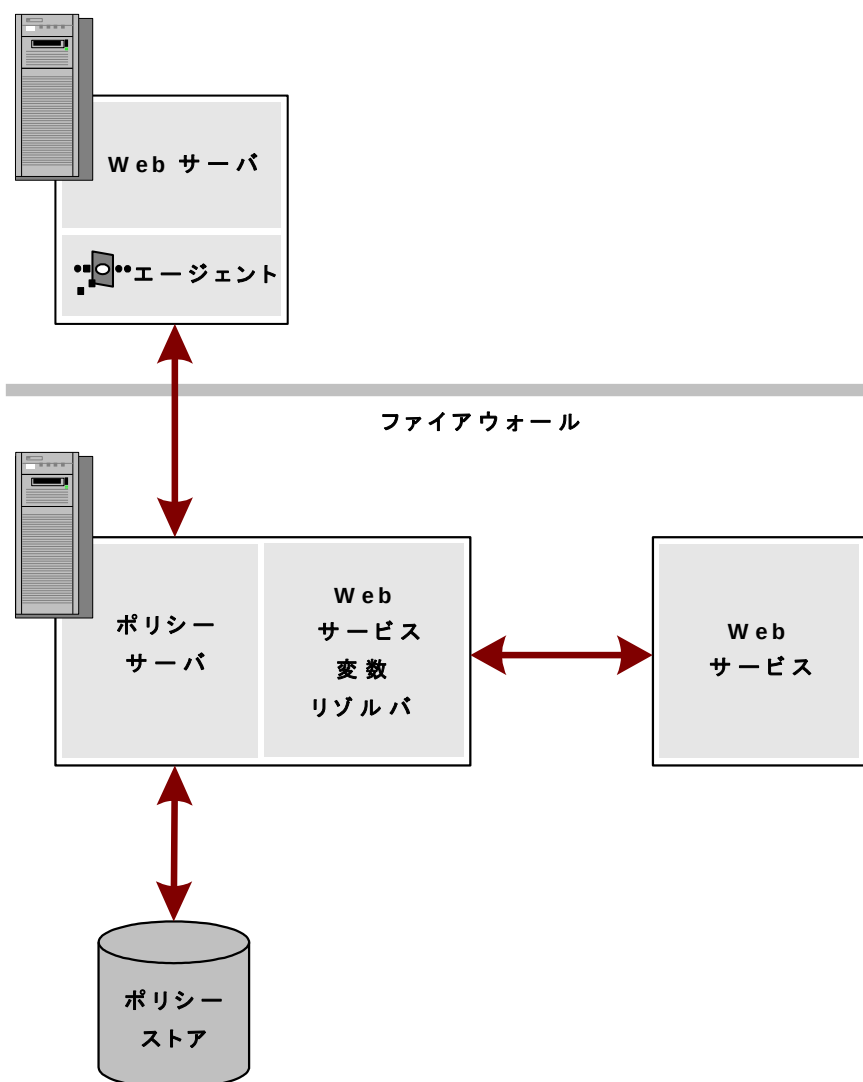
Web サービス変数

Web サービスの変数を使用すれば、Web サービスから取得した動的データをポリシー サーバのポリシーに組み込むことができます。Web サービスの変数は、Web サービスを呼び出すことで解決されます。ポリシー サーバは、Web サービスの変数の定義に従って SOAP 要求ドキュメントを送り、その応答として SOAP レスポンスドキュメントを受け取ります。ポリシー サーバは、SOAP レスポンスドキュメントから Web サービスの変数の値を取得します。

SOAP (Simple Object Access Protocol) は、軽量な XML ベースのプロトコルで、次の 3 つの部分で構成されます。

- メッセージの内容とメッセージの処理方法を記述するためのフレームワークを定義するエンベロープ
- アプリケーションで定義されたデータ型のインスタンスを表すための一連のエンコードルール
- リモートプロシージャコールとレスポンスを表すための表記規則

下図は、Web サービスに対して送られた Web サービスの変数が、SiteMinder 環境のイントラネット内、つまりファイアウォール内部のポリシー サーバと同じ側で解決される仕組みを示します。



このシナリオでは、Web サービスの変数に許可要求が関連付けられている場合、ポリシー サーバ側で Web サービスの変数リゾルバが呼び出されて変数を解決します。このリゾルバはポリシー サーバと同じ処理空間で実行します。

Web サービスの変数を定義する際に、ユーザは Web サービスに送る SOAP ドキュメント、認証情報、および SOAP ドキュメントの定義に応じてその他のパラメータを指定します。

リゾルバは指定された SOAP ドキュメントを Web サービスに送り、レスポンスから変数の値を取得して、その値をポリシー サーバに送ります。ポリシー サーバは値を受け取ると、許可要求を処理します。

ポリシー サーバと Web サービスの間にファイアウォールが存在しても、この 2 つの間の通信を許可するようにファイアウォールを設定することができます。ポリシー サーバは要求を発行し、レスポンスを読み込みます。したがって、ポリシー サーバから Web サービスへの要求送信を許可するようにファイアウォールを設定します。

ポリシー サーバと Web サービスの間に安全な SSL 接続を設定し、Web サービスのサーバ側証明書とポリシー サーバ側で設定された信頼できる CA のリストを使用して、Web サービスからポリシー サーバに着信レスポンスが送られるようにすることができます。

Web サービス変数のコンポーネント要件

Web サービス変数では、セッション サーバが必要です。

注: セッション サーバの設定の詳細については、「ポリシー サーバインストールガイド」を参照してください。既存のセッション サーバのアップグレードの詳細については、「*SiteMinder Upgrade Guide*」を参照してください。

Web サービス変数を解決する場合のセキュリティ要件

Web サービスの変数を処理する場合、ポリシー サーバと Web サービスの間に SSL 接続を設定する必要があります。また、Web サービスで認識されるように設定された WS-Security ヘッダーとユーザ名トークンを含めることもできます。WS-Security は、署名と暗号化を通じて、セキュリティトークンの伝播とメッセージの正当性/機密保護を提供する SOAP 拡張機能の標準セットです。

Web サービス変数を安全に処理するためには、次の条件を満たす必要があります。

- ポリシー サーバは既知の識別情報 (Web サービスアカウント) に代わって Web サービスに SOAP 要求を送信する必要があります。これは、SiteMinder がユーザディレクトリの属性にアクセスする際に使用するモデルと似ています。
- Web サービスのクレデンシャルを含む WS-Security ヘッダーに、base64 方式でエンコードされた SHA-1 ダイジェストを含めるように設定できます。
- Web サービスの変数は、SSL 接続とサーバ側の証明書を使用するように設定できます。つまり、ポリシー サーバに信頼できる CA のリストを設定する必要があります。

注: SSL 接続の場合は、Web サービス用のサーバ側証明書を設定する必要があります。ポリシー サーバには、信頼できる CA のリストを設定します。信頼できる CA を設定するには、「認証機関と Web サービス変数」で説明されている smkeydatabase ツールを使用します。

Web サービスの変数リゾルバの設定

ポリシー サーバで Web サービスの変数を解決するには、Web サービスに正しく接続するように Web サービスの変数リゾルバを設定する必要があります。Web サービスへの接続には、次の 2 つのパターンがあります。

- ポリシー サーバと Web サービスがファイアウォールの同じ側にある場合。ポリシー サーバと Web サービスの間にはファイアウォールは存在しません。
- ポリシー サーバと Web サービスの間にファイアウォールが存在するが、この 2 つの間で単方向の通信 (ポリシー サーバから Web サービスへの要求送信) が許可されるようにファイアウォールが設定されている場合。

Web サービスの変数機能を使用するためには、SmKeyTool コマンドラインユーティリティを使用して、信頼できる CA のリストをポリシー サーバに設定する必要があります。ロード バランシングまたはフェイルオーバー構成で複数のポリシー サーバを使用する場合には、各ポリシー サーバに同じ CA リストを設定する必要があります。

デフォルトの設定は、SiteMinder/Config/properties ディレクトリの WebServiceConfig.properties ファイルに格納されています。デフォルト設定はユーザが変更できます。

WebServiceConfig.properties のサンプル構成ファイル

```
# 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
```

認証機関と Web サービス変数

Web サービス変数を解決する際に SSL 接続を使用するには、ポリシー サーバが Web サービスへの接続を確立するときに使用できる、信頼できる認証機関 (CA) のリストを設定する必要があります。このためには、Web サービスに接続するポリシー サーバごとに **smkeydatabase** をセットアップする必要があります。

smkeydatabase は、WS-Security トークンによるメッセージの署名および検証に必要なキーと証明書を保管、管理、および取得することのできる、フラットファイル キーと証明書のデータベースです。また、サイトの公開キーを使用して暗号化されている対称 XML 暗号化キーを復号化する場合にも、このサービスを使用します。

SiteMinder **smkeytool** ユーティリティでは、新しい **smkeydatabase** を作成したり、既存の **smkeydatabase** を削除してから新しい **smkeydatabase** を作成したりすることができます。設定できるキー データベースは、1 つのポリシー サーバにつき 1 つのみです。

WS-Security ドキュメントのキー データベースに保管されるアイテム

以下のキーは、キー データベースに保管されます。

- WS-Security ドキュメントに署名して X509v3 トークンを生成するための、エンタープライズ秘密キーおよび証明書。
- WS-Security 文書に署名してトークンを生成するために使用する秘密キーに対応する公開キー証明書。これらの証明書は、WS-Security ドキュメントの検証に使用されます。

指定されたポリシー サーバが、**WS-Security** ドキュメントの署名/検証を行うことができます。署名および検証に使用されるキーと証明書は、ポリシー サーバが実行する内容に応じて、同じキー データベースに追加できます。

以下の表に、**WS-Security** ドキュメントの署名および検証の各要件を処理するために **smkeydatabase** に追加する必要があるオブジェクトを示します。

機能	WS-Security トークン タイプ	必要なデータベース オブジェクト
署名	すべて	Web サービス ホスト エンタープライズの秘密キーおよび証明書
X509 トークンの生成	X509v3	Web サービス ホスト エンタープライズの秘密キーおよび証明書
署名の検証	SAML アサーション、Sender Vouches	発行する Web サービス コンシューマ アプリケーションの証明書
	SAML アサーション、Holder-of-key	XML リクエスト対象と発行側の Web サービス コンシューマ アプリケーションの証明書。
	X.509v3、ユーザ名 (署名する場合)	信頼された発行者の証明書。

smkeytool という名前の **SiteMinder** ユーティリティを使用して **smkeydatabase** にアイテムを追加します。変更を行う前に、[smkeydatabase と smkeytool](#) (P. 649) について確認してください。

変数の作成

ポリシーまたはレスポンスで使えるようにする変数を作成します。変数は、ドメインオブジェクトです。変数は特定のポリシードメイン内に作成するか、**smobjimport** ツールを使用してドメインにインポートします。

ポリシードメインにオブジェクトをインポートする方法の詳細については、「ポリシー サーバ管理ガイド」を参照してください。

スタティック変数の作成

ポリシーまたはレスポンスで使用できるように、スタティック変数を作成します。

注: 解決された変数の値が 1 K を超えることはありません。

変数を作成する方法

1. 変数を追加するドメインを開きます。
2. [変数]タブをクリックします。
ドメインに関連付けられた変数のリストが表示されます。
3. [変数の作成]をクリックします。
[変数の作成]ペインが開きます。
4. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
変数の設定が表示されます。
5. [名前]フィールドに変数名を入力します。
6. [変数のタイプ]リストから[スタティック]を選択します。
スタティック変数の設定が表示されます。
注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。
7. [変数情報]グループ ボックスで、変数のデータ型および値を指定します。
8. [サブミット]をクリックします。

変数が、ドメインの[変数]タブ内に表示されます。これで、ポリシー式またはレスポンスの中で変数を使用できます。

コンテキスト要求変数の作成

ポリシーまたはレスポンスでできるように、コンテキスト要求変数を作成します。

注: 解決された変数の値が 1 K を超えることはありません。

変数を作成する方法

1. 変数を追加するドメインを開きます。
2. [変数]タブをクリックします。
ドメインに関連付けられた変数のリストが表示されます。
3. [変数の作成]をクリックします。
[変数の作成]ペインが開きます。
4. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
変数の設定が表示されます。

5. [名前]フィールドに変数名を入力します。

注: コンテキスト要求変数の名前は、先頭がパーセント文字(%)である必要があります。

例: %REQUEST_ACTION

6. [変数のタイプ]リストから[コンテキスト要求]を選択します。
コンテキスト要求の設定が表示されます。
7. [プロパティ]リストから変数値を選択します。
8. [OK]をクリックします。

変数が、ドメインの[変数]タブ内に表示されます。これで、ポリシー式またはレスポンスの中で変数を使用できます。

ユーザ コンテキスト変数の作成

ポリシーまたはレスポンスでできるように、ユーザ コンテキスト変数を作成します。

注: 解決された変数の値が 1 K を超えることはありません。

変数を作成する方法

1. 変数を追加するドメインを開きます。
2. [変数]タブをクリックします。
ドメインに関連付けられた変数のリストが表示されます。
3. [変数の作成]をクリックします。
[変数の作成]ペインが開きます。
4. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
変数の設定が表示されます。
5. [名前]フィールドに変数名を入力します。

注: ユーザ コンテキスト変数の名前は、先頭がパーセント文字(%)である必要があります。

例: %SM_USERPATH

6. [変数のタイプ]リストから[ユーザ コンテキスト]を選択します。
ユーザ コンテキストの設定が表示されます。
7. [プロパティ]リストから、変数の値を指定する、ユーザ コンテキストの部分を選択します。
戻り型の値が、[プロパティ]リストで選択した値に応じて、文字列またはブールのいずれかで表示されます。
8. (ユーザ プロパティおよびディレクトリ エントリの場合は必須)[プロパティ]フィールドに、変数値を指定するディレクトリまたはユーザ属性の名前を入力します。
9. (ユーザ プロパティおよびディレクトリ エントリの場合は必須)[バッファ]フィールドに、変数を格納するためのバッファのサイズ(バイト単位)を入力します。
10. (ディレクトリ エントリの場合は必須)[DN]フィールドに、ディレクトリ エントリの識別名を入力します。

11. [サブミット]をクリックします。
12. 変数が、ドメインの[変数]タブ内に表示されます。これで、ポリシー式またはレスポンスの中で変数を使用できます。

フォーム ポスト変数の作成

ポリシーでできるように、フォーム ポスト変数を作成します。

注: 解決された変数の値が 1 K を超えることはありません。

変数を作成する方法

1. 変数を追加するドメインを開きます。
2. [変数]タブをクリックします。
ドメインに関連付けられた変数のリストが表示されます。
3. [変数の作成]をクリックします。
[変数の作成]ペインが開きます。
4. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
変数の設定が表示されます。
5. [名前]フィールドに変数名を入力します。
6. [変数のタイプ]リストから[ポスト]を選択します。
フォーム ポストの設定が表示されます。
7. [フォーム フィールド名]フィールドに、フォームに含まれる POST 変数の名前を入力します。
8. [OK]をクリックします。
変数が、ドメインの[変数]タブ内に表示されます。変数がポリシー式で使えるようになりました。

Web サービス変数の作成

ポリシーまたはレスポンスでできるように、**Web** サービス変数を作成します。

注: 解決された変数の値が **1 K** を超えることはありません。

変数を作成する方法

1. 変数を追加するドメインを開きます。
2. [変数]タブをクリックします。
ドメインに関連付けられた変数のリストが表示されます。
3. [変数の作成]をクリックします。
[変数の作成]ペインが開きます。
4. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
変数の設定が表示されます。
5. [名前]フィールドに変数名を入力します。
6. [一般]グループ ボックスの[変数のタイプ]ドロップダウンリストで[Web サービス]を選択します。
Web サービスの設定が表示されます。
7. [定義]グループ ボックスの[戻り型]ドロップダウンリストでデータ型を選択します。
8. [リモート URL]グループ ボックスの[URL]フィールドに、Web サービスの URL を入力します。
9. [クエリを返す]グループ ボックスの[XPath]フィールドに XPath クエリを入力します。
注: ポリシー サーバは、このクエリを使用して、Web サービスから返された SOAP ドキュメントから Web サービス変数の値を取得します。
10. (オプション) [Web サービス認証情報]グループ ボックスの[認証情報が必要]チェック ボックスをオンにして、Web Service への接続時にポリシー サーバが使用するユーザ名およびパスワードを指定します。
11. (オプション) [SOAP ドキュメント]グループ ボックスの[変数...]をクリックして、既存の変数を SOAP メッセージに追加します。

12. (オプション) [HTTP ヘッダ] グループ ボックスの [追加] をクリックして、HTTP ヘッダを Web サービス変数に関連付けます。
13. [OK] をクリックします。
変数が、ドメインの [変数] タブに表示され、ポリシー式またはレスポンスで使用可能になります。

第 20 章: SiteMinder キー データベース管理

このセクションには、以下のトピックが含まれています。

[キー データベースの概要 \(P. 649\)](#)

[キー データベースのプロパティファイル \(P. 650\)](#)

[Smkeytool によるキー データベースの作成および管理 \(P. 653\)](#)

[smkeytool を使用したキー データベースの変更 \(P. 655\)](#)

キー データベースの概要

smkeydatabase という名前の SiteMinder キー データベースは、SiteMinder で署名、検証、暗号化、復号化のために使用されるキーおよび証明書のデータベースです。たとえば、OCSP リクエストとレスポンスを署名および検証するため、または WS-Security トークンでメッセージを署名および検証したりするためのキーと証明書を格納できます。

データベースは複数のファイルから構成されます。このデータベース内のキーと証明書は、smkeytool という名前の SiteMinder ツールを使用して管理および取得できます。

smkeydatabase には複数の秘密キーを格納できます。

smkeydatabase は SiteMinder ポリシー サーバと併せてインストールされます。ポリシー サーバでは、認定された FIPS (Federal Information Processing Standard) 140-2 準拠の暗号化ライブラリを使用しています。これにより SiteMinder 環境では、FIPS 準拠のアルゴリズムを使用して機密データを暗号化することができます。その結果、smkeydatabase 内のすべてのデータはこれらの FIPS 準拠アルゴリズムを使用して暗号化されます。

注: ポリシー サーバの旧バージョンから r12.0 SP3 にアップグレードする場合、データが正しく暗号化されるよう、smkeydatabase の移行手順について「SiteMinder Upgrade Guide」を参照してください。

Smkeydatabase のエイリアス

エイリアスを使用すると、smkeydatabase 内のすべての証明書または証明書と秘密キーのペアを容易に参照できるようになります。smkeydatabase 内のすべての証明書または証明書/秘密キーのペアには一意のエイリアスが必要です。

キー データベースのプロパティファイル

smkeydatabase プロパティファイル (smkeydatabase.properties) は、キー データベースにアクセスして管理するのに必要なコンフィグレーション パラメータを定義します。

smkeydatabase.properties ファイルは、以下の場所にインストールされています。

- `policy_server_home\config\properties` (Windows)
- `policy_server_home/config/properties` (UNIX)

このファイルは、以下のオプションを変更する場合のみ変更します。

- `NativeDBName` -- キー データベースの名前を指定します。
- `DBLocation` -- キー データベースが格納されているディレクトリを示します。
- `DBUpdateFrequencyMinutes` -- データベースがファイル システムから読み取られる頻度を指定します。

smkeydatabase.properties ファイルには以下の設定が含まれます。

- [DBLocation](#) (P. 651)
- [NativeDBName](#) (P. 651)
- [XMLDocumentOpsImplementation](#) (P. 651)
- [AffiliateIXMLSignatureImplementation](#) (P. 651)
- [IXMLSignatureImplementation](#) (P. 652)
- [EncryptedPassword](#) (P. 652)
- [IXMLEncryptDecryptImplementation](#) (P. 652)
- [DBUpdateFrequencyMinutes](#) (P. 652)

次に各設定の説明を示します。

DBLocation の設定

データベースがあるディレクトリへのパスを指定します。

データベースを手動で作成するときに **smkeytool** が使用する場所を入力します。

デフォルト: *policy_server_home/smkeydatabase*

NativeDBName の設定

データベースの名前を識別します。

データベースを作成するときに **smkeytool** で使用する名前を指定します。

デフォルト: **smkeydatabase**

XMLDocumentOpsImplementation の設定

XML 署名および検証を実行する **Java** クラスを指定します。

注: この値は変更しません。事前に設定されており、スタティックです。

デフォルト: **com.ca.smkeydatabase.api.XMLDocumentOpsImpl**

AffiliateIXMLSignatureImplementation の設定

署名と検証で低いレベルの暗号化を実行する **Java** クラスを指定します。

注: この値は変更しません。事前に設定されており、スタティックです。

デフォルト: **com.ca.smkeydatabase.api.XMLSignatureApacheImpl**

IXMLSignatureImplementation の設定

署名と検証で低いレベルの暗号化を実行する Transactionminder 向け Java クラスを指定します。

注: この値は変更しません。事前に設定されており、スタティックです。

デフォルト: `com.ca.smkeydatabase.api.XMLSignatureApacheImpl`

EncryptedPassword の設定

smkeydatabase パスワードを示します。

(データベースの作成時にポリシー ストア キーを使用して暗号化される) キー データベースの作成前に、このエントリにはダミー値が含まれます。

デフォルト: `encrypted_password_string`

IXMLEncryptDecryptImplementation の設定

アサーション、名前 ID および属性の暗号化と復号化を実行する Java クラスを示します。

注: この値は変更しません。事前に設定されており、スタティックです。

デフォルト: `com.ca.smkeydatabase.api.XMLEncryptDecryptApacheImpl`

DBUpdateFrequencyMinutes の設定

データベースがファイル システムから読み取られる頻度を示します。具体的には、メモリ内の smkeydatabase が期限切れになり再ロードされるまでの分数です。

この時間が経過するまで、データベースに行われた証明書とキーの追加、削除、変更といった操作はポリシー サーバに影響しません。値が 0 の場合、キー データベース キャッシュは完全に無効になります。値が -1 の場合、キャッシュはポリシー サーバが再起動されるまで永続します。

デフォルト: 60 分

Smkeytool によるキー データベースの作成および管理

smkeytool コマンドライン ユーティリティを使用すると、キー データベースにデータを入力し、そのデータベースを管理できます。このツールは、ポリシー サーバと共にインストールされます。

smkeytool を使用して、次の操作を実行できます。

- キー データベースの作成および削除
キー データベースは、ポリシー サーバごとに 1 つだけ作成できます。データベースの作成後に、キーと証明書を追加できます。
- 秘密キーと証明書の追加および削除
- 発行者証明書追加および削除
- キー データベースに格納されたすべての証明書のリスト表示

注: smkeytool は、smkeydatabase.properties ファイル内の値を使用します。smkeytool を実行する前に、このファイルが正しく設定されていることを確認してください。

smkeytool は、以下のディレクトリに配置されています。

- <policy_server_home>/bin (UNIX)
- <policy_server_home>\bin (Windows)

smkeytool ユーティリティは、以下の構文を使用してコマンドラインから実行します。

UNIX の場合

smkeytool.sh option [argument(s)]

Windows の場合

smkeytool.bat option [argument(s)]

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

以下の表は、オプションと引数の説明です。

オプション	引数	Function
-createDB または -cdb	<password>	キーと証明書を格納するための空のキー データベースを作成します。 指定したパスワードは、ポリシー ストア キーを使用して暗号化され、 smkeydatabase.properties ファイルに追加されます。
-deleteDB または -ddb	なし	smkeydatabase.properties ファイル内で指定されたキー データベースを削除します。
-addPrivKey または -apk	<private_key_filepath> <x.509_certificate_filepath> <password>	指定した秘密キーおよび対応する証明書ファイルをキー データベースに追加します。<password> は、ロードされる秘密キー ファイルの暗号化に使用されるパスワードです。キー データベースと関連付けられたパスワードではありません。
-deletePrivKey または -dpk	<x.509_certificate_filepath>	指定した証明書に基づいて、キー データベースから秘密キー エントリを削除します。
-addCert または -ac	<x.509_certificate_filepath>	キー データベースに証明書を追加します。
-deleteCert または -dc	<x.509_certificate_filepath>	指定した証明書に基づいて、キー データベースから証明書を削除します。
-listCerts または -lc	なし	キー データベースに格納されたすべての証明書の発行者名/件名 (DN) およびシリアル番号をリスト表示します。
-help または -h	なし	smkeytool の使用情報をリスト表示します。

注: smkeytool には、証明書の取り消し情報を管理するいくつかのコマンド オプションが備わっています。これらのオプションは SiteMinder Federation セキュリティ サービスの署名および暗号化機能でのみ使用できます。

smkeytool の使用例

- キー データベースの作成

UNIX の場合

```
smkeytool.sh -cdb password
```

Windows の場合

```
smkeytool.bat -cdb password
```

- 秘密キーと証明書の追加

UNIX の場合

```
smkeytool.sh -apk /opt/netegrity/webagent/certs/samplePrivKey.pkcs8  
/opt/netegrity/webagent/certs/sampleRobm.cer passphrase
```

Windows の場合

```
smkeytool.bat -apk "c:\program  
files\netegrity\webagent\certs\samplePrivKey.pkcs8" "C:\program  
files\netegrity\webagent\certs\sampleRobm.cer" passphrase
```

- 発行者証明書の追加

UNIX の場合

```
smkeytool.sh -ac /opt/netegrity/webagent/certs/sampleCARoot.cer
```

Windows の場合

```
smkeytool.bat -ac "c:\program  
files\netegrity\webagent\certs\sampleCARoot.cer"
```

smkeytool を使用したキー データベースの変更

Smkeytool は、ユーザが smkeydatabase にデータを読み込んだり管理できるようにする SiteMinder コマンドライン ユーティリティです。smkeytool ユーティリティは、ポリシー サーバ により以下の場所にインストールされます。

- *policy_server_home*/bin (UNIX)
- *policy_server_home*\bin (Windows)

smkeytool を使用して、次の操作を実行できます。

- キー データベースの作成および削除

キー データベースは、ポリシー サーバごとに 1 つだけ作成できます。データベースの作成後に、キーと証明書を追加できます。

- 秘密鍵の追加および削除
- パートナーの証明書の追加および削除
- キー データベースに格納されたすべての証明書のリスト表示
- CA のルート証明書のインポート
- クライアント証明書キーの追加

smkeydatabase に含まれていないコンシューマ機関でルートまたはチェーンの認証局 (CA) を使用している場合、smkeydatabase にこれを追加する必要があります。

たとえば、署名された VeriSign の CA サーバ側証明書は、Web エージェントオプション パックでインストールされたプロデューサ側 Web サーバを SSL 対応にするために使用します。SSL を介した基本認証にこの証明書を使用するには、コンシューマ側で smkeydatabase への VeriSign 証明書を追加します。証明書の追加は、コンシューマがサーバ側証明書を持つプロデューサと通信していることを確認するのに役立ちます。また、信頼された CA によって証明書が検証されたことを確認するのに役立ちます。

- smkeydatabase からのキー データのエクスポート
- 証明書失効リストの追加、表示、検証、および削除

証明書の変更に関する考慮事項

- 秘密キーまたは証明書を追加する場合、証明書ファイルから証明書メタデータを削除してから smkeydatabase にインポートします。「--BEGIN CERTIFICATE --」のマーカで始まり、「--END CERTIFICATE --」のマーカで終わるデータのみをインポートします。以下のマーカを必ず含めるようにしてください。
- キー データベースに新規証明書を追加するか、既存の証明書を更新する場合、変更を反映させるにはポリシー サーバを再起動する必要があります。ポリシー サーバを再起動しない場合は、ポリシー サーバとキー データベースが同期するまでに多少の時間がかかります。キー データベースの更新にかかる時間は、設定されたデータベース更新頻度によって変わります。データベース更新頻度は、smkeydatabase.properties ファイル内の DBUpdateFrequencyMinutes パラメータを使用して設定できます。

Smkeytool コマンドの構文およびオプション

Smkeytool は、smkeydatabase を管理するための多くのオプションを提供するコマンドライン ユーティリティです。

smkeytool ユーティリティは、以下の構文を使用してコマンドラインから実行します。

UNIX

```
smkeytool.sh -option [-引数]
```

Windows

```
smkeytool.bat -option [-引数]
```

コマンドラインからオプションなしで **smkeytool** を入力すると、すべてのコマンドライン オプションのリストを表示します。

smkeytool ユーティリティは、次のコマンド オプションおよび引数を使用します。

- [createDB](#) (P. 658)
- [addPrivateKey](#) (P. 659)
- [addCertOption](#) (P. 660)
- [addRevocationInfo](#) (P. 661)
- [changepassword](#) (P. 661)
- [deleteRevocationInfo](#) (P. 662)
- [deleteDB](#) (P. 662)
- [delete](#) (P. 662)
- [export](#) (P. 663)
- [findAlias](#) (P. 663)
- [importDefaultCACerts](#) (P. 664)
- [listCerts](#) (P. 664)
- [listRevocationInfo](#) (P. 664)

- [printCert](#) (P. 665)
- [renameAlias](#) (P. 665)
- [validateCert](#) (P. 665)
- [help](#) (P. 665)

次に各コマンド オプションの説明を示します。

createDB オプション

ストア キーおよび証明書への新しい **smkeydatabase** を作成します。デフォルトのディレクトリ名は **smkeydatabase** です。**smkeydatabase.properties** ファイルを変更して、**smkeydatabase** の場所を変更できます。

smkeydatabase 内の秘密キーはすべて、FIPS 準拠のアルゴリズムを使用して暗号化されます。

重要： データベースに複数のキーを格納するには、追加する最初のキーに別名 **defaultenterpriseprivatekey** を定義した後に、以降のキーを追加する必要があります。

-createDB の引数は以下のとおりです。

-password <password>

必須です。パスワードは、すべてのデータを暗号化形式でキー データベースに格納するために使用されます。6 ～ 32 文字の長さの値を使用できます。パスワードは、ポリシー ストア キーを使用して暗号化され、**smkeydatabase.properties** ファイルに追加されます。

-importDefaultCACerts

(オプション) データベースの作成中に、デフォルトの認証局 (CA) 証明書をインポートします。これらの証明書は **cacerts.keystore** ファイルからインポートされます。このファイルは、ポリシー サーバと共にインストールされ、すべてデフォルト CA 証明書を含んでいます。このオプションは、**-importDefaultCACerts** オプションの実行と同じです。

addPrivKey オプション

秘密キーと証明書のペアをキー データベースに追加します。データベース内には、複数の秘密キーおよび証明書を格納できますが、サポートされているのは RSA キーのみです。

注: 暗号化形式で **smkeydatabase** に格納されるのは秘密キーのみです。

証明書を作成する側のポリシー サーバは、単一の企業秘密キーを使用して、SAML メッセージに署名すると共に、証明書を使用する側から受信した暗号化 SAML メッセージを復号化します。通常、企業キーは、**smkeydatabase** で見つかった最初の秘密キーです。

-addPrivKey コマンドを使用すると、-keyfile オプションおよび -certfile オプションを組み合わせるか、または -keycertfile オプションを単独で使用するにより、キー データを指定できます。

-addPrivKey の引数は以下のとおりです。

-alias <alias>

必須です。データベース内の単一の証明書または証明書/秘密キー ペアに別名を割り当てます。別名は、一意の、英数文字のみを含む文字列である必要があります。

-certfile <cert_file>

この秘密キーに関連付けられた証明書の場所へのフル パスを指定します。PKCS1、PKCS5、および PKCS8 の各形式のキーに必須です。

-keyfile <private_key_file>

秘密キー ファイルの場所へのフル パスを指定します。PKCS1、PKCS5、および PKCS8 の各形式のキーに必須です。

-keycertfile <key_cert_file>

秘密キーおよび公開証明書データを含む PKCS12 ファイルの場所へのフル パスを指定します。PKCS12 形式のキーに必須です。

-password <password>

オプションです。キー/証明書ペアが最初に作成されている場合に、秘密キーの暗号化に使用されたパスワードを指定します。秘密キーを **smkeydatabase** に追加するときは、秘密キーが **smkeydatabase** に書き込まれる前に、それを復号化するためにこのパスワードを指定する必要があります。

注: このパスワードは **smkeydatabase** には格納されません。

秘密キーは、復号化されて **smkeydatabase** 内に配置された後に、**smkeydatabase** 自体のパスワードを使用して、再び暗号化されます。このパスワードは、**smkeydatabase** の作成時に指定されたものです。

addCert オプション

キー データベースに証明書を追加します。X.509 証明書形式の **V1**、**V2**、および **V3** バージョンがサポートされています。また、**DER** および **PEM** の各エンコード形式がサポートされています。**CA** 証明書を追加するときは、**Web** エージェントを再起動してください。

証明書を認証局として信頼することを示した場合は、この証明書が常に、**CA** 証明書として扱われます。

addCert の引数は以下のとおりです。

-alias <alias>

必須です。データベース内でこの秘密キーに関連付けられた証明書の別名。一意の、英数文字のみを含む文字列である必要があります。

-infile <cert_file>

必須です。新しく追加された証明書の場所へのフルパス。

- trustcacert

オプションです。追加されるユーザ プロバイダ証明書が **CA** 証明書であることを確認します。**Smkeytool** は、証明書にデジタル署名拡張子があり、証明書に同じ **IssuerDN** 値および **SubjectDN** 値があることを確認します。

-noprompt

(オプション) 証明書の追加を確認するメッセージは表示されません。

addRevocationInfo オプション

SAML 認証処理中に smkeydatabase がリストを見つけられるように、CRL の場所を指定します。smkeydatabase には、CRL の内容は格納されません。smkeydatabase は、ポリシー サーバの起動時、およびリフレッシュ間隔時間の経過後に、CRL の内容を読み取るだけです。

重要: smkeydatabase に CRL エントリを追加した場合は、ポリシー サーバを再起動する必要があります。

addRevocationInfo の引数は以下のとおりです。

-issueralias <issueralias>

必須です。CRL を発行する認証局の別名。

例: -issueralias verisignCA

-type (ldapcrl | filecrl)

必須です。リストが、証明書ファイルおよび LDAP CRL のいずれであるかを指定します。オプションは、ldapcrl または filecrl です。

-location <location>

必須です。CRL の場所を指定します。ファイルの場合は、ファイルへのフルパスを指定します。LDAP CRL の場合は、LDAP サーバ ノードへのフルパスを指定します。

ファイルの場所の例: -location c:¥crls¥siteminder_root_ca.crl

LDAP CRL の場所の例: -location "http://localhost:880/sn=siteminderroot,dc=crls,dc=com"

changePassword オプション

smkeydatabase のパスワードを変更できるようにします。パスワードを変更すると、古いパスワードの使用時に暗号化されたエントリがすべて、FIPS 準拠のアルゴリズムを使用して、新しいパスワードによって再暗号化されます。

changePassword の引数は以下のとおりです。

-password <password>

必須です。最初に smkeydatabase を作成するために使用した既存のパスワードを指定します。

-newpassword <new_password>

必須です。smkeydatabase の新しいパスワードを指定します。

deleteRevocationInfo オプション

データベースから CRL を削除します。

-deleteRevocationInfo の引数は以下のとおりです。

-issueralias <issuer_alias>

必須です。CRL を発行する認証局の名前。

-noprompt

(オプション) データベースからの CRL の削除を確認するメッセージは表示されません。

deleteDB オプション

smkeydatabase.properties ファイル内の設定データに基づいて、smkeydatabase を削除します。キー データベース内および別名データストア ファイル内のエントリがすべて削除されます。

-deleteDB の引数は以下のとおりです。

-noprompt

(オプション) データベースの削除を確認するメッセージは表示されません。

delete オプション

smkeydatabase から、既存の証明書を削除します。証明書に秘密キーが関連付けられている場合は、そのキーも削除されます。

-delete の引数は以下のとおりです。

-alias <alias>

必須です。削除する証明書の別名。

-noprompt

(オプション) データベースの削除を確認するメッセージは表示されません。

export オプション

smkeydatabase から、既存の証明書または秘密キーをファイルにエクスポートします。証明書データは、**PEM** エンコーディングを使用してエクスポートされます。秘密キー データは、**DER** エンコードによる **PKCS8** 形式でエクスポートされます。

-export の引数は以下のとおりです。

-alias <alias>

必須です。エクスポートする証明書またはキーを特定します。

-outfile <out_file>

必須です。エクスポートする証明書またはキーの出力ファイルへのフルパスを指定します。

-type (key|cert)

オプションです。証明書およびキーのどちらをエクスポートするかを示します。オプションを指定しない場合は、デフォルトで証明書が指定されます。

-password <password>

秘密キーをエクスポートする場合のみ必須です。ファイルへのエクスポート時に秘密キーを暗号化するために使用されるパスワードを指定します。公開キーを保持する証明書をエクスポートする場合、パスワードは不要です。これは、証明書はクリア テキストでエクスポートされるためです。

秘密キーはエクスポート時に、このパスワードを使用して、暗号化形式の出力ファイルにエクスポートされます。同じ秘密キーを **smkeydatabase** に再度追加するには、**-addPrivKey** コマンドを実行して、このパスワードを使用します。

findAlias オプション

すでに **smkeydatabase** 内にある証明書に関連付けられている別名を判定します。

-findAlias の引数は以下のとおりです。

-infile <cert_file>

必須です。検索する別名に関連付けられている証明書ファイルへのフルパス。

-password <password>

パスワードで保護されている **P12** ファイルが証明書ファイルとして指定されている場合にのみ必須のパスワード。

importDefaultCACerts オプション

ポリシー サーバと共にインストールされた `cacerts.keystore` ファイルから、デフォルトの信頼済み CA 証明書をすべて、`smkeydatabase` にインポートします。CA 証明書を使用して、作成側の Web サーバに関連付けられたサーバ証明書が確認されます。

listCerts オプション

キー データベースに格納されているすべての証明書のメタデータの一部をリストで表示します。

`-listCerts` の引数は以下のとおりです。

`-alias <alias>`

(オプション) 指定した別名に関連付けられた証明書およびキーのメタデータ情報をリストで表示します。このオプションは、ワイルドカード文字としてアスタリスク(*)をサポートしています。このワイルドカードは、別名の値の先頭、末尾、またはその両方に使用できます。コマンドシェルがワイルドカード文字を解釈しないように、アスタリスクは必ず、引用符で囲んでください。

listRevocationInfo オプション

`smkeydatabase` 内の現在の CRL のリストを表示します。`-listRevocationInfo` オプションは、CRL 名、タイプ(`file` または `ldap`)、およびデータベース内のすべての CRL の場所の印刷のみを実行します。

`-listRevocationInfo` の引数は以下のとおりです。

`-issueralias <issuer_alias>`

(オプション) CRL を発行する認証局の名前。このオプションは、ワイルドカード文字としてアスタリスク(*)をサポートしています。このワイルドカードは、別名の値の先頭、末尾、またはその両方に使用できます。コマンドシェルがワイルドカード文字を解釈しないように、アスタリスクは必ず、引用符で囲んでください。

printCert オプション

指定した証明書のメタデータの一部を表示します。このコマンドは特に、証明書のプロパティを表示するのが難しい UNIX システムで役立ちます。

-printCert の引数は以下のとおりです。

-infile <cert_file>

必須です。証明書ファイルの場所。

-password <password>

パスワードで保護されている P12 ファイルが証明書ファイルとして指定されている場合にのみ必須のパスワード。

renameAlias オプション

証明書に関連付けられている既存の別名を変更します。

-renameAlias の引数は以下のとおりです。

-alias <current_alias>

必須です。証明書に関連付けられている現在の別名。

-newalias <new_alias>

必須です。新しい別名。値は、英数文字のみを含む一意の文字列である必要があります。

validateCert オプション

(オプション) 証明書が失効しているかどうか示します。

-validateCert の引数は以下のとおりです。

-alias <alias>

必須です。データベース内でこの秘密キーに関連付けられた証明書の別名。一意の、英数文字のみを含む文字列である必要があります。

-infile <crl_file>

オプションです。smkeytool で、検証する証明書を検索する CRL ファイルを指定します。

help オプション

smkeytool ユーティリティの使用方法を表示します。

UNIX プラットフォームでの smkeytool の使用例

例: キー データベースの作成

この例は、smkeydatabase を作成するコマンドを示します。

```
smkeytool.sh -createDB -password siteminderdb
```

例: 秘密キーと証明書の追加

この例は、smkeydatabase に秘密キーと証明書を追加するコマンドを示します。この例では、以下に示すように証明書とキーが格納されているディレクトリから smkeytool を実行しているとします。

```
smkeytool.sh -addPrivkey -password keypswd -alias idplprivkey -keyfile privkey.pkcs8  
-certfile sample.crt
```

証明書とキーが格納されているディレクトリから smkeytool を実行していない場合は、以下に示すように、これらのアイテムが格納されているディレクトリのフルパスを指定する必要があります。

```
smkeytool.sh -addPrivkey -alias privkey1 -keyfile "export/ca/siteminder/certs/  
sampleprivkey.pkcs8" -certfile "export/ca/siteminder/certs/samplecert.crt"
```

例: 信頼された CA 証明書の追加

この例は、信頼された認証機関の証明書を追加する際のコマンドを示します。

重要: 信頼された証明書を追加する前に、認証機関から CA 証明書を取得してください。

信頼された CA 証明書を追加する方法

1. 以下を入力し、使用する証明書データベースに証明書がすでにあるかどうか確認します。
`smkeytool.sh -listCerts`
2. 以下を入力し、CA 証明書を追加します。
`smkeytool.sh -addCert -alias -splcacert -infile
/opt/netegrity/siteminder/certs/sampleCARoot.cer -trustcacert`
3. (任意) ポリシー サーバを再起動して、キー データベースへの変更を適用します。

ポリシー サーバを再起動しない場合は、ポリシー サーバとキー データベースが同期するまでに多少の時間がかかります。キー データベースの自動更新にかかる時間は、設定されたデータベース更新頻度によって変わります。データベース更新頻度は、`smkeydatabase.properties` ファイル内の `DBUpdateFrequencyMinutes` パラメータを使用して設定できます。

Windows プラットフォームでの smkeytool の使用例

例: キー データベースの作成

この例は、`smkeydatabase` を作成するコマンドを示します。

```
smkeytool.bat -createDB -password smdb
```

例: 秘密キーと証明書の追加

この例は、smkeydatabase に秘密キーと証明書を追加するコマンドを示します。この例では、以下に示すように証明書とキーが格納されているディレクトリから smkeytool を実行しているとします。

```
smkeytool.bat -addPrivkey -password keypswd -alias privkey1  
-keyfile sampleprivkey.pkcs8" -certfile samplecert.crt"
```

証明書とキーが格納されているディレクトリから smkeytool を実行していない場合は、以下に示すように、これらのアイテムが格納されているディレクトリのフルパスを指定する必要があります。

```
smkeytool.bat -addPrivkey -password keypswd -alias privkey1 -keyfile "c:\program  
files\ca\siteminder\certs\sampleprivkey.pkcs8"  
-certfile "c:\program files\ca\siteminder\certs\samplecert.crt"
```

例: 信頼された CA 証明書の追加

この例は、信頼された認証機関の証明書を追加する際のコマンドを示します。

重要: 信頼された証明書を追加する前に、認証機関から CA 証明書を取得してください。

信頼された CA 証明書を追加する方法

1. 以下を入力し、使用する証明書データベースに証明書がすでにあるかどうか確認します。

```
smkeytool.sh -listCerts
```
2. CA 証明書を追加するには、以下を入力します。

```
smkeytool.bat -addCert "c:\program files\ca\siteminder\certs\sampleCARoot.crt"  
-trustcacert
```
3. (任意) ポリシー サーバを再起動して、キー データベースへの変更を適用します。

ポリシー サーバを再起動しない場合は、ポリシー サーバが同期するまでに多少の時間がかかります。キー データベースの自動更新にかかる時間は、設定されたデータベース更新頻度によって変わります。データベース更新頻度は、smkeydatabase.properties ファイル内の DBUpdateFrequencyMinutes パラメータを使用して設定できます。

第 21 章: グローバル ポリシー、ルール、レスポンス

このセクションには、以下のトピックが含まれています。

[グローバル ポリシー \(P. 669\)](#)

[グローバル ポリシーを設定する方法 \(P. 674\)](#)

[グローバル ポリシーの有効と無効 \(P. 686\)](#)

[グローバル アクティブ ポリシーの設定 \(P. 686\)](#)

[グローバル ポリシーで許可される IP アドレス \(P. 687\)](#)

[グローバル ポリシーの時間制限の追加と削除 \(P. 690\)](#)

グローバル ポリシー

SiteMinder の標準ポリシーは、1 つのポリシードメインのコンテキストで作成されます。ただし、大規模な本番環境では何千ものドメインが存在することがあります。このような環境では、多数のドメインに共通する動作のタイプを定義しておくことが便利です。この動作はポリシーで表します。標準のポリシーを使用すると、同じ動作が必要なドメインごとに同じポリシーを作成する必要があります。グローバルポリシーを使用すれば、すべてのドメインで適用されるポリシー (および関連するルールとレスポンス) をシステムレベルオブジェクトとして設定することができます。

グローバルポリシーの説明では、次のような用語を使用します。

アクセス ルール

アクセスルールは、リソースへのアクセスを許可または拒否します。グローバルポリシーにはアクセスルールは含まれません。グローバルポリシーに追加できるのは、イベントルールだけです。

イベント ルール

イベントルールは、認証イベントまたは許可イベントが発生すると、起動します。すべてのドメインに渡って共通に実行される動作には、イベントルールを関連付けて、グローバルポリシーに含めることができます。

グローバル ポリシー

システム オブジェクトとして定義されるポリシー。

グローバル ルール

システム オブジェクトとして定義されるルール。

グローバル レスポンス

システム オブジェクトとして定義されるレスポンス。

ポリシー リンク

ポリシー定義に使用される論理的なエンティティ。ルールとレスポンスのペアで構成されます。1つのポリシーには、1つまたは複数のポリシーリンクを含めることができます。

グローバル ポリシー オブジェクトの特性

ここでは、標準の(非グローバル)ポリシー オブジェクトとの相違点を比較しながら、グローバル ポリシー オブジェクトの特性について概説します。

グローバル レスポンスと標準レスポンス

相違点

- システムレベルで定義される。グローバル レスポンスを定義できるのは、システムレベルの管理者のみです。
- 変数ベースの属性は使用できない。
- 任意のグローバル固有またはドメイン固有のポリシーに使用される。
- 特定のエージェントタイプに関連付けられている。
- レスポンスグループに追加できない。グローバルレスポンスグループというものはありません。

類似点

- アクティブな式を使用できる。
- 特定のポリシーに指定されていないと返されない。

グローバル ルールと標準ルール

相違点

- システムレベルで定義される。グローバル ルールを定義できるのは、システムレベルの管理者のみです。
- グローバル ルールのフィルタは、特定のレلمにバインドされない。グローバル ルールのフィルタは絶対フィルタで、正規表現を使用できる場合とできない場合があります。

- 特定のエージェントまたはエージェントグループにバインドされる。エージェントはルールを作成時に明示的に指定します。
- **SiteMinder** エージェントの場合にのみ使用可能です。**RADIUS** エージェントは、認証および許可イベントをサポートしていないため、グローバル ルールに関連付けることはできません。
- 認証イベントまたは許可イベントについてのみ定義される。
- グループ ポリシーでのみ使用される。
- ルール グループに追加できない。グローバル ルール グループというものはありません。
- グローバル ポリシー処理が有効になっている、どのドメインのリソースに対しても起動できる。

グローバル ポリシーと標準ポリシー

相違点

- システムレベルで定義される。グローバルポリシーを定義できるのは、システムレベルの管理者のみです。
- すべてのユーザにバインドされる。特定のユーザを指定してグローバル ポリシーに追加したり、除外したりすることはできません。
注: グローバル ポリシー処理は、個々のドメインで明示的に有効または無効にすることができます。
- 定義には、グローバル ルール、グローバル レスポンス、およびこれらのグローバル オブジェクトのグループのみが使用される。
- 変数ベースの属性や変数式は使用できない。
- 許可/拒否アクセスルールを含めることはできない。グローバル ポリシーに含めることができるのは、イベントルールだけです。
- 特定のリソース/レルムを指定してグローバル ポリシーに追加したり、グローバル ポリシーから除外したりすることはできない。グローバル ポリシーは、グローバル ポリシー処理が有効に設定されたドメイン上で、そのポリシーに定義されたルール フィルタの少なくとも 1 つに適合するすべてのリソースに対して適用されます。
- **Java Policy Management API** ではサポートされない。

類似点

- アクティブな式を使用できる。
- 特定のエージェントに関連付けられる。ただし、同じタイプのすべてのエージェントで構成されるグループを作成し、このグループにグローバル ルールをバインドすることは可能です。

グローバルポリシーの処理時には、起動したグローバルルールに定義されたレスポンスが他のレスポンスのリストに追加されます。グローバルルールは次の条件が満たされたときに起動します。

- アクセス対象のリソースがグローバルルールに定義された絶対リソースフィルタに適合する。
- 発生したイベントがグローバルルールに定義されている。
- 要求されたリソースが同じエージェント/エージェントグループによって保護されており、これらのエージェント/エージェントグループがルールに指定されている。
- アクセス対象のリソース/レルムが、グローバルポリシー処理が有効に設定されているドメインに属している。

詳細情報:

[ドメインのグローバル ポリシー処理の無効化](#) (P. 478)

SiteMinder グローバル ポリシーの概念

SiteMinder では、ポリシーベースのアクセス制御モデルを使用します。

SiteMinder ポリシーは、特定のリソースに対してユーザに与えられるアクセスのタイプと、ユーザがそのリソースにアクセスしたときに実行される処理を定義します。SiteMinder の標準ポリシーはそれぞれ、一連のユーザと一連のリソースを結び付けています。また、ユーザ、ルール、およびレスポンスをバインドして、リソースを保護するように設計されています。各ポリシーでは、そのポリシーが適用されるユーザまたはユーザのグループが指定されている必要があります。ユーザをポリシーに追加したり、ポリシーから除外したりできます。

さらに、標準のポリシーには少なくとも 1 つのルールまたはルールグループを指定する必要があります。ルールはポリシーの一部で、保護するリソースとルールを起動するアクションの種類を決定します。ルールでは、文字列ベースのリソースフィルタとアクションの組み合わせで、ポリシーに含まれる 1 つまたは複数のリソースが識別されます。これに対し、フィルタは、**レルム フィルタ**と**ルール フィルタ**で構成されます。標準 **SiteMinder** ポリシーのレルム、ルール、およびレスポンスの詳細については、以下を参照してください。

- [レルム内のグループ化リソース](#) (P. 487)
- [ルール グループ](#) (P. 517)
- [ルール](#) (P. 495)
- [レスポンスとレスポンス グループ](#) (P. 521)
- [ポリシー](#) (P. 549)

SiteMinder オブジェクトには、システム レベルとドメイン レベルの 2 つのタイプがあります。標準 (非グローバル) **SiteMinder** ポリシーでは、すべてのポリシー オブジェクトを特定のドメインのコンテキストで作成する必要があります。ただし、グローバル ポリシーは、システム レベルのポリシーであるため、**SiteMinder** 環境内のすべてのドメインにわたって適用できます。システム レベル権限を持つ管理者は、グローバルポリシーを定義できます。これには、グローバルルールとグローバルレスポンスも含まれます。これらのグローバルポリシーは、どのドメインのどのリソースにも適用できます。

グローバルオブジェクトは、標準のドメイン固有オブジェクトと似ています。グローバルポリシー定義でのグローバルオブジェクト役割は、ドメイン固有のポリシーオブジェクトとは異なりなります。主な違いはオブジェクトの作成方法と、オブジェクトを結合してポリシーを構成する仕組みです。ただし、グローバルドメインオブジェクトやグローバルレルムオブジェクトというものはありません。

グローバル ポリシー処理

ポリシーは、「[ポリシーの処理](#) (P. 588)」で説明されている方法で評価されます。加えて、以下の条件が満たされた場合には、グローバル ポリシーに追加されたグローバルルールが起動します。

- 要求されたリソースが、グローバル ポリシー処理が有効に設定されているドメインに属している。
- 要求されたリソースがグローバルルールに定義された絶対リソースフィルタに適合する。グローバルルールの場合、フィルタはレルムからではなく、ルールから取得されるのが重要なポイントです。

- グローバル ルールに定義されたイベントと同じイベントが発生する。
- 要求されたリソースが同じエージェント/エージェントグループによって保護されており、これらのエージェント/エージェントグループがルールに指定されている。

認証イベントまたは許可イベントが発生するたびに、起動したグローバルルールに定義されたレスポンスが他のレスポンスのリストに追加されます。

グローバル ポリシーを設定する方法

グローバル ポリシーは、グローバル ルール オブジェクトとグローバル レスポンス オブジェクトで構成され、レスポンス属性を含みます。以下は、グローバル ポリシーを作成する手順を示しています。

1. [認証イベントのグローバル ルールを作成](#) (P. 676)、または[許可イベントのグローバル ルールを作成](#) (P. 678)します
2. [グローバル レスポンスの設定](#) (P. 682)
3. [グローバル Web エージェントレスポンス属性の設定](#) (P. 683)
4. [グローバル ポリシーを設定します](#) (P. 684)

重要: グローバル ポリシーとドメイン固有ポリシーを設定して同じリソースに適用することができます。たとえば、アクセス制御用のドメイン固有ポリシーと、標準のレスポンス セットを指定するグローバル ポリシーを設定できます。ただし、グローバル ポリシーが機能するためには、ドメイン固有ポリシーに含まれているレムでイベント処理が許可されるように設定する必要があります。

グローバル ルール

グローバル ルールはグローバル ポリシーの一部で、グローバル ポリシーの処理をトリガするリソースおよびイベントを定義します。グローバル ルールは、ドメイン固有ルールとほぼ同じです。ただし、グローバル ルールは認証イベントまたは許可イベントに関連付ける必要があります。グローバル 許可/拒否アクセスルールはありません。

認証イベント用のグローバルルール

SiteMinder 認証イベントを含むグローバルルールを使用すると、ユーザがリソースにアクセスするための認証を得たときに発生するアクション (On-Auth イベント) を制御できます。

注: On-Auth イベントの結果はレルム単位で発生します。たとえば、OnAuthAccept ヘッダが設定されているレルム A からユーザがレルム B に移動しても、レルム B にはこのヘッダが設定されていないので、ヘッダを使用できません。ユーザがレルム A に戻ると、このヘッダは再度設定されます。

次のリストは、発生する可能性がある On-Auth イベントです。

On-Auth-Accept

認証に成功した場合に発生します。このイベントは、認証が成功した後、ユーザをリダイレクトするときに使用されることもあります。

On-Auth-Reject

On-Auth-Reject ルールを含むポリシーにバインドされたユーザの認証に失敗した場合に発生します。このイベントは、認証が失敗した後、ユーザをリダイレクトするときに使用されることもあります。

OnAuthAccept イベントと OnAuthReject イベントは、いずれも認証時 (ユーザがユーザ名とパスワードを入力したとき) および確認時 (ユーザの cookie からユーザ情報が読み込まれたとき) に起動します。ただし、認証時にのみ発生する特殊なアクションもあります。

レルムタイムアウトの上書き (EnforceRealmTimeouts が使用された場合を除く)。

EnforceRealmTimeouts オプションをサポートしている Web エージェントを使用していて、そのオプションをオンにしている場合を除き、ユーザのアイドルタイムアウトと最大タイムアウト時間は、そのユーザが最後に認証されたレルムの設定値がそのまま適用されます。これらの値は、ユーザが認証情報を再度入力するまで変わりません。

注: EnforceRealmTimeouts の詳細については、*SiteMinder 4.x Web and Affiliate Agent Quarterly Maintenance Release 4 Release Notes* のセクション 3.3 を参照してください。

リダイレクト。

リダイレクトが認証時のみに許可される理由はいくつかありますが、最大の理由は、OnAuth リダイレクトを確認時にも許可すると、リダイレクトが無限ループに入る可能性が非常に高いことです。

ユーザのパスワードへのアクセス。

パスワードは **SMSSESSION cookie** に保存されないため、ユーザが認証時に実際にパスワードを入力するまでわかりません。

On-Auth-Attempt

SiteMinder がユーザを認識できず、そのユーザを拒否したときに発生します(たとえば、未登録ユーザを、最初にユーザ登録サイトにリダイレクトできます)。

On-Auth-Challenge

カスタム チャレンジ/レスポンス認証方式がアクティブになったときに発生します(トークンコードなど)。

ユーザが認証または拒否されると、該当する **On-Auth** ルールに関連付けられたグローバルレスポンスが認証を要求してきたエージェントに返されます。

認証イベント用のグローバル ルールの作成

リソースにアクセスするユーザの認証時に発生する制御アクションの認証イベント用グローバル ルールを作成します。

グローバル ルールを作成する方法

1. [ポリシー]-[グローバル]をクリックします。
2. [グローバル ルール]-[グローバル ルールの作成]をクリックします。

[グローバル ルールの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. グローバル ルール名を入力します。
4. [レルム]グループ ボックスと[リソース]グループ ボックスで、エージェントとリソースの設定を指定します。

注: エージェントグループを指定し、さらに同じリソースに関連するドメイン固有ルールも設定すると、同じ処理が重複することになり、システムのパフォーマンスが低下する場合があります。ドメインに固有のルールには、グローバル ルールによって生成されたレスポンスを複製する可能性があることを考慮してください。このような場合は、ポリシー サーバが、重複するレスポンスを自動的に削除してから、要求元のエージェントに情報を渡すため、エージェントにはレスポンスが 1 つだけ返されます。

5. [アクション]グループ ボックスから認証イベントを選択します。

6. [アクション]リストから **On-Auth** イベントを 1 つ選択します。
7. [サブミット]をクリックします。

グローバル ルールが保存されます。

許可イベント用のグローバル ルール

SiteMinder 許可イベントを含むグローバル ルールを使用すると、SiteMinder が、要求したリソースに対する許可をユーザが得ているかどうかに基づいて、レスポンスを呼び出すことができます。リソースを保護するルールに **On-Access** イベントがある場合、許可イベントはユーザが認証されたあとに発生します。各自の権限に基づいてユーザにアクセス権が付与または拒否されたときには、その状況に応じたイベントが発生します。

次のリストは、発生する可能性がある **On-Access** イベントです。

On-Access-Accept

ユーザが許可された場合に発生します。このイベントは、リソースへのアクセスを許可されたユーザをリダイレクトすることができます。

On-Access-Reject

ユーザが許可されなかった場合に発生します。このイベントを使用して、リソースへのアクセスが許可されなかったユーザをリダイレクトすることができます。

ユーザが許可または拒否されると、該当する **On-Access** ルールに関連するレスポンスが許可を要求してきたエージェントに返されます。

OnAccessReject ルールのポリシーに関する考慮事項

OnAccessReject イベントを含むグローバルなルールを作成する場合は、ポリシー サーバがグローバルなポリシーを処理する方法と、**OnAccessReject** ルールによって生じる特別な状況を検討してください。

GET/POST ルールと同じポリシーに **OnAccessReject** ルールを含めると、**OnAccessReject** ルールは起動しません。ユーザの認証時に、SiteMinder はユーザの ID を解決します。したがって、**OnAccessReject** ルールと GET/POST ルールが同じポリシー内にあると、リソースへのアクセスが許可されるべきユーザと、**OnAccessReject** イベントにリダイレクトされるべきユーザが同一になります。ユーザはアクセスを許可されるので、拒否イベントが適用されることはないからです。

この矛盾を解消するには、**OnAccessReject** ルール用の別のポリシーを作成し、このルールを適用するユーザを指定します。このポリシーには他のイベントルールを含めることもできます。

たとえば、LDAP ユーザディレクトリで、**User1** にはリソースへのアクセスを許可し、グループ、**ou=People**、**o=company.com** のそれ以外のユーザはすべて **OnAccessReject** ページにリダイレクトするように設定するとします。この場合には、次の 2 つのポリシーを作成する必要があります。

Policy1

このポリシーには、**User1** のアクセスを許可する **GET/POST** ルールを含めます。

Policy2

OnAccessReject ルールとリダイレクトレスポンスを含めて、グループ **ou=People**、**o=company.com** を指定します。

User1 は許可されているため、**User1** がリソースにアクセスしても **OnAccessReject** ルールは起動しません。ただし、グループ、**ou=People**、**o=company.com** に含まれる他のすべてのユーザについては、リソースにアクセスしようとすると、**OnAccessReject** ルールが起動します。これらのユーザにはリソースへのアクセスが許可されていないからです。

許可イベント用グローバル ルールの作成

リソースにアクセスするユーザの認証時に発生する制御アクションの認証イベント用グローバルルールを作成します。

グローバル ルールを作成する方法

1. [ポリシー]-[グローバル]をクリックします。
2. [グローバル ルール]-[グローバル ルールの作成]をクリックします。

[グローバル ルールの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. グローバル ルール名を入力します。
4. [レルム]グループ ボックスと[リソース]グループ ボックスで、エージェントとリソースの設定を指定します。

注: エージェントグループを指定し、さらに同じリソースに関連するドメイン固有ルールも設定すると、同じ処理が重複することになり、システムのパフォーマンスが低下する場合があります。ドメインに固有のルールには、グローバル ルールによって生成されたレスポンスを複製する可能性があることを考慮してください。このような場合は、ポリシー サーバが、重複するレスポンスを自動的に削除してから、要求元のエージェントに情報を渡すため、エージェントにはレスポンスが 1 つだけ返されます。

5. [アクション]グループ ボックスから認証イベントを選択します。
6. [アクション]リストから On-Auth イベントを 1 つ選択します。
7. [サブミット]をクリックします。

グローバル ルールが保存されます。

グローバル ルールの有効と無効

グローバル ルールを有効にすることで、ユーザが指定されたリソースにアクセスして認証、または許可イベントをトリガすると、**SiteMinder** が必ずルールを起動するようになります。グローバル ルールを無効にすると、ユーザが指定されたリソースにアクセスして認証、または許可イベントをトリガしても、**SiteMinder** がルールを起動しません。

グローバル ルールを有効または無効にする方法

1. グローバル ルールを開きます。
2. ルールを有効にするには[有効]チェック ボックスを選択し、無効にするにはクリアします。
3. [サブミット]をクリックします。

ルールが保存されます。

グローバル ルールへの時間制限の追加

グローバル ポリシーに時間制限を追加することで、グローバル ポリシーが特定の時間にのみ実行されるようになります。時間制限で指定していない時間帯にユーザがリソースにアクセスしても、ポリシーは起動しません。

時間制限をルールに追加する方法

1. グローバル ポリシーを開きます。
2. [時間制限]グループ ボックスの「設定」をクリックします。

[時間制限]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. 開始日および有効期限を指定します。
4. 毎時間制限テーブルで時間制限を指定します。

注: 各チェック ボックスは 1 時間に相当します。チェック ボックスで選択した時間帯は、ルールが起動し、指定されているリソースに適用されます。チェック ボックスで選択していない時間帯は、ルールが起動しないため、指定されているリソースに適用されません。

5. [OK]をクリックします。
6. 時間制限が保存されます。

アクティブ グローバル ルールの設定

外部のビジネス ロジックに基づいて動的に許可を行うアクティブ ルールを設定します。このときポリシー サーバでは、ユーザ自ら作成する共有ライブラリの関数を呼び出します。この共有ライブラリは許可 API で指定されたインターフェースに適合する必要があります。この API はソフトウェア開発キットで使用できます。

注: 共有ライブラリの詳細については、「*Programming Guide for C*」を参照してください。

アクティブ ルールを設定する方法

1. [アクティブ ルール]グループ ボックスのフィールドに、ライブラリ名、関数名、関数パラメータを指定します。

アクティブ ルール文字列が[アクティブ ルール]フィールドに表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [サブミット]をクリックします。

アクティブ ルールが保存されます。

グローバル ルールの削除

グローバル ルールを削除すると、そのグローバル ルールを含むすべてのグローバル ポリシーからルールが自動的に削除されます。ただし、グローバルポリシーはシステムから削除されません。ルールを削除した後もグローバル ポリシーが機能するかどうかを確認します。

グローバル ポリシーには、グローバル ルールが最低でも 1 つは必要です。

注: ポリシー サーバオブジェクトの変更および削除の詳細については、「[Manage Policy Server Objects \(P. 61\)](#)」を参照してください。

グローバル レスポンス オブジェクト

グローバル レスポンスはグローバル ポリシーの一部で、ユーザがグローバル ルールで指定された認証イベントまたは認可イベントをトリガした後に返す属性を定義します。

注: グローバル レスポンスはドメイン ポリシーでも使用できます。グローバルレスポンスが返されるためには、ドメイン固有のポリシーまたはグローバルポリシーに追加する必要があります。ポリシー内で、グローバルレスポンスはドメイン固有レスポンスと同じように処理されます。

グローバル レスポンスの設定

認証または許可イベントが発生した後に返す属性を定義するために、関連するグローバル ルールでグローバル レスポンスを設定します。

グローバル レスポンスを設定する方法

1. [ポリシー]-[グローバル]をクリックします。
2. [グローバル レスポンス]、[グローバル レスポンスの作成]をクリックします。
[グローバル レスポンスの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. グローバル レスポンスの名前を入力します。
4. SiteMinder エージェント タイプ リストから SiteMinder エージェント タイプを選択します。
5. [サブミット]をクリックします。

グローバル レスポンスが保存されます。これで、レスポンスにレスポンス属性を追加できます。各エージェント タイプにレスポンス属性を追加する方法については、次のセクションを参照してください。

グローバル レスポンスのレスポンス属性

SiteMinder では、グローバル レスポンスごとに 1 つ以上のレスポンス属性が含まれています。レスポンス属性によって、ポリシー サーバから SiteMinder エージェントに渡される情報が識別されます。SiteMinder では、エージェント タイプごとに異なるレスポンス属性を設定できます。

グローバル レスポンス属性のタイプ

SiteMinder は、さまざまなタイプのレスポンス属性に対応しています。レスポンス属性のタイプによって、ポリシー サーバがレスポンス属性の該当する値を見つける場所が決まります。グローバル レスポンスに設定できるレスポンス属性のタイプは、ドメイン固有のレスポンスに設定可能なレスポンス属性のタイプと同じです。

グローバル Web エージェント レスポンス属性の設定

ポリシー サーバが SiteMinder エージェントに渡す各情報を格納するレスポンス属性を設定できます。Web エージェントレスポンス属性は、HTTP ヘッダ変数、cookie 変数、他のリソースへのリダイレクト、テキスト、タイムアウト値をサポートしています。Web エージェントレスポンス属性タイプの詳細については、「Web エージェント設定ガイド」を参照してください。

注: CA SOA セキュリティ マネージャを購入している場合は、「CA SOA Security Manager Policy Configuration Guide」に WebAgent-SAML-Session-Ticket-Variable レスポンス属性タイプに関する説明があります。

レスポンス属性を作成する方法

1. [レスポンス]ペインの[属性リスト]グループ ボックスで、[レスポンス属性の作成]をクリックします。

[レスポンス属性の作成]ペインが表示されます。

2. ドロップダウンリストからレスポンス属性を選択します。

注: レスポンス属性の詳細については、「Web エージェント設定ガイド」を参照してください。

3. [属性の種類]グループ ボックスで属性タイプを選択します。

[属性フィールド]グループ ボックスの各フィールドが、指定した属性タイプと一致するように更新されます。

4. [属性フィールド]グループ ボックスの各フィールドに入力します。

注: レスポンスで利用できる、自動的に生成された SiteMinder ユーザ属性の一覧については、「[SiteMinder が生成するユーザ属性](#) (P. 538)」を参照してください。

5. (省略可) [詳細]グループ ボックスの[スクリプト]フィールドで属性を編集します。

注: [詳細]グループ ボックスで属性を編集すると、[属性のセットアップ]グループ ボックスが閉じます。

6. [属性キャッシング]グループ ボックスで、[キャッシュ値]または[値の再計算間隔]を指定します。

7. [サブミット]をクリックします。

レスポンス属性の作成タスクが、処理のためにサブミットされて、[レスポンス]ペインの[属性リスト]にレスポンス属性が追加されます。

グローバル ポリシー オブジェクトを設定する方法

グローバル ポリシーの設定では、以下の手順を完了する必要があります。

1. [グローバル ポリシーの作成](#) (P. 684)
2. [グローバル ポリシーへのグローバル ルールの追加](#) (P. 684)
3. (オプション) [グローバル ルールとレスポンスの関連付け](#) (P. 685)

グローバル ポリシーの作成

グローバル ポリシーを作成して、ユーザがリソースと対話する方法を定義します。

グローバル ポリシーを作成する方法

1. [ポリシー]-[グローバル]をクリックします。
2. [グローバル ポリシー]-[グローバル ポリシーの作成]をクリックします。
[グローバル ポリシーの作成]ペインが開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. グローバル ポリシー名を入力します。
4. グローバル ルールおよびグローバル レスポンスを追加します。

グローバル ポリシーへのグローバル ルールの追加

グローバルルールは、グローバルポリシーに含まれる特定のリソースを示します。グローバルポリシーには、少なくとも 1 つのグローバルルールを追加する必要があります。

グローバル ポリシーにグローバル ルールを追加する方法

1. [ルール]タブをクリックします。
[ルール]グループ ボックスが表示されます。
2. [ルールの追加]をクリックします。
[使用可能なルール]ペインが表示され、使用可能なグローバル ルールをリストします。
注: 必要とするグローバル ルールが表示されない場合は、[新規ルール]をクリックします。このようにして作成したルールは、グローバル ポリシーに追加されます。
3. 追加するグローバル ルールを選択し[OK]をクリックします。
[ルール]グループ ボックスに、選択したルールとルール グループがリストされます。
4. (オプション)レスポンス、またはレスポンス グループとルールを関連付けます。

グローバル ルールとレスポンスの関連付け

グローバル レスポンスは、ルール起動時に発生するアクションを示します。ルールが起動すると、関連するレスポンスも起動します。

グローバル ルールとレスポンスを関連付ける方法

1. レスポンスを関連付けるグローバル ルールの[レスポンスの追加]をクリックします。
[使用可能なレスポンス]ペインが開き、ここに、使用可能なレスポンス、レスポンス グループ、およびグローバル レスポンスのリストが表示されます。
2. レスポンス、レスポンス グループ、またはグローバル レスポンスを選択して、[OK]をクリックします。
[ルール]グループ ボックス内でレスポンスが開かれ、それぞれのルールに関連付けられます。
注: 必要なレスポンスが存在しない場合は、新規作成をクリックしてレスポンスを作成します。

グローバル ポリシーの有効と無効

管理 UI では、グローバル ポリシーの有効と無効を切り替えられます。ポリシーを作成すると、デフォルトではポリシーは有効に設定されます。ポリシーが有効な場合、グローバルルールに指定されたリソースにユーザがアクセスしようとする、グローバルポリシーに含まれるグローバルルールが起動します。

グローバル ポリシーを無効にした場合、ポリシーに含まれるルールは起動しません。

ポリシーを有効または無効にする方法

1. ポリシーを開きます。
2. [有効]チェック ボックスをオンまたはオフにします。

このチェック ボックスをオンにすると、ポリシーは有効になります。このチェック ボックスをオンにすると、ポリシーは無効になります。無効なポリシーは起動しません。

3. [サブミット]をクリックします。

ポリシーが保存されます。

グローバル アクティブ ポリシーの設定

アクティブなポリシーを使用して、外部のビジネスロジックに基づく動的な許可を行います。ポリシー サーバがユーザ作成の共有ライブラリの関数を呼び出すと、アクティブなポリシーが許可の決定に追加されます。

この共有ライブラリは、許可 API (ソフトウェア開発キットの別売) で指定されているインターフェースに準拠している必要があります

注: 詳細については、「API Reference Guide for C」を参照してください。

グローバルポリシーにアクティブなポリシーを設定する手順は、ドメイン固有ポリシーにアクティブなポリシーを設定する手順と同じです。

アクティブなポリシーを設定する方法

1. グローバル ポリシーを開きます。
2. [詳細]グループ ボックスで、[アクティブなポリシーの編集]チェック ボックスをオンにします。
アクティブなポリシーの設定が表示されます。
3. [ライブラリ名]フィールドに、共有ライブラリの名前を入力します。
4. 共有ライブラリに、アクティブなポリシーを実装する関数の名前を入力します。
5. [サブミット]をクリックします。
ポリシーが保存されます。

グローバル ポリシーで許可される IP アドレス

以下の情報を指定してポリシーのリソースにアクセスするユーザを絞り込み、該当する場合にグローバル ポリシーを起動します。

- IP アドレス
- ホスト名
- サブネット マスク
- IP アドレスの範囲

たとえば、ポリシーにルールを追加してリソースへのアクセスを許可した後で、IP アドレスの範囲を指定した場合、指定した範囲の IP アドレスからログインしたユーザだけが、保護されたリソースにアクセスできます。

グローバル ポリシーに対する単一 IP アドレスの指定

単一 IP アドレスを指定すると、ユーザが、指定した IP アドレスを使用してポリシーのリソースにアクセスした場合にのみ、グローバル ポリシーが起動します。

単一の IP アドレスを指定する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。

IP アドレスの設定が表示されます。

3. [単一ホスト]ラジオ ボタンを選択します。
単一ホストに固有の設定が表示されます。
4. IP アドレスを入力し、[OK]をクリックします。

[IP アドレス]グループ ボックスに、その IP アドレスが表示されます。

注: IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。

5. [サブミット]をクリックします。
ポリシーが保存されます。

グローバル ポリシーのホスト名の追加

ホスト名を指定し、指定したホストからポリシーのリソースにアクセスしたユーザにのみグローバル ポリシーが起動するようにします。

ホスト名を指定する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。

IP アドレスの設定が表示されます。

3. [ホスト名]ラジオ ボタンを選択します。
ホスト名に固有の設定が表示されます。

4. ホスト名を入力し、[OK]をクリックします。
ホスト名が[IP アドレス]グループ ボックスに表示されます。
5. [サブミット]をクリックします。
ポリシーが保存されます。

グローバル ポリシーのサブネット マスクの追加

サブネット マスクを指定し、指定したサブネット マスクからポリシーのリソースにアクセスしたユーザにのみグローバル ポリシーが起動するようにします。

サブネット マスクを追加する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。
IP アドレスの設定が表示されます。
3. [サブネット マスク]ラジオ ボタンを選択します。
サブネット マスクに固有の設定が表示されます。
4. [IP アドレス]フィールドに IP アドレスを入力します。
注: IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。
5. [サブネット マスク]フィールドに、サブネット マスクを入力します。
6. [OK]をクリックします。
サブネット マスクが[IP アドレス]グループ ボックスに表示されます。
7. [サブミット]をクリックします。
ポリシーが保存されます。

グローバル ポリシーの IP アドレスの範囲の追加

IP アドレスの範囲を指定し、アドレスの範囲に該当する IP アドレスを使ってポリシーのリソースにアクセスしたユーザにのみポリシーが起動するようにします。

IP アドレスの範囲を追加する方法

1. ポリシーを開きます。
2. [IP アドレス]グループ ボックスで[追加]をクリックします。
IP アドレスの設定が表示されます。
3. [範囲]ラジオボタンを選択します。
IP アドレスの範囲に固有の設定が表示されます。
4. [最小値]フィールドに、開始 IP アドレスを入力します。
注: IP アドレスはわからないが、アドレスのドメイン名がわかっている場合は、DNS 検索を選択します。完全修飾ホスト名を入力して、IP アドレスを検索します。
5. [最大値]フィールドに、終了 IP アドレスを入力します。
6. [OK]をクリックします。
[IP アドレス]グループ ボックスに、IP アドレスの範囲が表示されます。
7. [サブミット]をクリックします。
ポリシーが保存されます。

グローバル ポリシーの時間制限の追加と削除

グローバルポリシーには、時間制限を追加できます。時間制限を追加すると、指定した時間の範囲内でグローバルポリシーが起動します。時間制限で指定していない時間帯にユーザがリソースにアクセスしても、グローバル ポリシーは起動しません。

注: 時間制限は、ポリシー サーバがインストールされたサーバのシステム クロックに基づいています。

ポリシーに時間制限を追加する方法

1. ポリシーを開きます。
2. [時間]グループ ボックスで[設定]をクリックします。

[時間制限]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. 開始日および有効期限を指定します。
4. 毎時間制限テーブルで時間制限を指定します。

注: 各チェック ボックスは 1 時間に相当します。チェック ボックスで選択した時間帯は、ルールが起動し、指定されているリソースに適用されます。チェック ボックスで選択していない時間帯は、ルールが起動しないため、指定されているリソースに適用されません。

5. [OK]をクリックします。
時間制限が保存されます。

第 22 章：インパーソネーション(偽装)

このセクションには、以下のトピックが含まれています。

[インパーソネーションの概要](#) (P. 693)

[インパーソネーション処理](#) (P. 694)

[インパーソネーションでのセキュリティに関する考慮事項](#) (P. 696)

[インパーソネーション用のポリシー サーバ オブジェクト](#) (P. 699)

[インパーソネーションのサンプル実装](#) (P. 712)

インパーソネーションの概要

インパーソネーション機能を使用すると、権限を持つユーザが、そのセッションを終了することなく、別のユーザのロールで動作することができます。この機能は、SM_SERVERSESSIONSPEC に使用します。

- カスタマーサービススタッフ (Customer service representatives: CSR) が顧客のアカウントを使用してアクセスの問題を調査する場合。
- ヘルプデスクスタッフが社員のアカウントを使用してアクセスの問題を調査する場合。
- 休暇中または外出中の同僚のアカウントを使用する場合。
- その他、あるユーザが別のユーザの識別情報を一時的に借用する必要がある場合。

インパーソネーション機能を使用すると、上記のような状況で、安全に作業を行うことができます。すべての場合において、パスワードは公開されず、ユーザが別のユーザとして動作することはできません。

インパーソネーションの説明では、次のような用語を使用します。

インパーソネーション セッション

別のユーザの識別情報を利用することを目的として作成されるユーザ セッション。

偽装対象

その識別情報が、権限のあるユーザによって利用されるユーザ。

偽装者

他のユーザの識別情報を利用できる、権限のあるユーザ。

インパーソネーション認証方式

権限を持つユーザが、自身の識別情報を維持したまま、別のユーザになりすますことが可能な認証方式。

セッション

ユーザ セッションとも呼ばれます。認証からログアウトまでの時間を表す用語。

セッション仕様

情報セッション チケットまたはセッション スペックとも呼ばれます。ユーザと現在のセッションの特性を説明する情報を表す用語で、この情報は、ポリシー サーバ上に独自のフォーマットで格納されます。

SMSESSION

ポリシー サーバのセッション仕様を含む **Web** エージェント **Cookie** の名前。

注: セッションの詳細については、「ポリシー サーバ管理ガイド」を参照してください。

インパーソネーション処理

別のユーザを偽装するプロセスは、次のとおりです。

- 権限を持つユーザ(偽装者)が、**SiteMinder** で保護されているリソースにアクセスして、**SiteMinder** セッションを確立します。このリソースに該当するのは、管理 UI、または、**SiteMinder** ポリシーで保護されている他のアプリケーションまたはリソースです。
- 偽装者は現在確立されている識別情報を使用してフォームにアクセスします。これにより、偽装者は偽装されるユーザ(偽装対象)を指定できます。偽装対象の認証情報が偽装者によって提示されることはありません。このフォームは、インパーソネーションセッションを有効にする特殊なロジックが組み込まれた **.fcc** ファイルです。

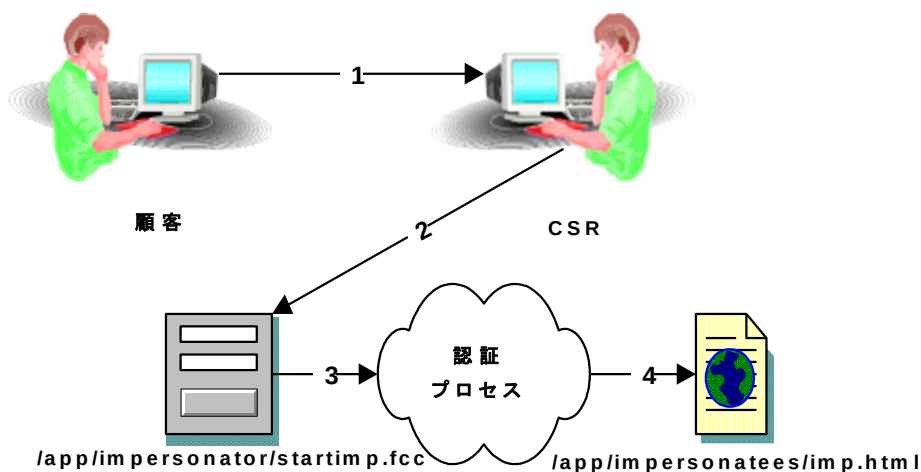
- 偽装者は偽装対象に関する情報を送信します。

ポリシー サーバは、偽装者が一連のポリシーを使用してインパーソネーションを実行できるかどうかを判断します。

- 偽装対象のインパーソネーションを実行するための十分な権限が偽装者にあるかどうかを判断します。
- 偽装者の確立されたセッションを使用して、偽装者が偽装対象としてアクセスすることを許可します。
- 偽装者のアクティビティをすべて監査します。

偽装者は、インパーソネーション認証方式で保護されたレルム内のリソースを指すターゲットが含まれる .fcc ファイルに直接アクセスします。これにより、インパーソネーションセッションが開始されます。

次の図は、カスタマーサービススタッフ (CSR) が .fcc ファイルに直接アクセスして、インパーソネーションセッションを開始する場合の流れを示します。



1. CSR は顧客から問題が発生したという連絡を受けます。CSR は、顧客の問題を解決する最善の方法は、インパーソネーションであると判断します。

2. CSR はインパーソネーション開始ページにアクセスして、インパーソネーション処理を開始します。この例の場合の開始ページの名前は「/app/impersonators/startimp.fcc」です。フォーム証明書コレクタ (Forms Credential Collector: FCC) によってフォームが表示されます。CSR は偽装対象の顧客の名前を入力します。偽装対象を明確化するためにその他の属性を入力する必要がある場合もあります。この図には掲載されていませんが、.fcc ファイルには、認証方式で使用する CSR のセッション仕様を収集する命令文が含まれています。FCC のターゲットが、インパーソネーション認証方式で保護されている SiteMinder レルムにあるため、インパーソネーション認証方式はポリシー サーバ上で起動されます。
3. ポリシー サーバは FCC によって収集された情報を受け取り、この情報をもとに、設定されたユーザディレクトリのいずれかに偽装対象の顧客が存在するかどうかを判断します。顧客が見つかり、ポリシー サーバは CSR がその顧客を偽装できるかどうか、顧客が偽装対象となるかどうかを判断します。2 つの条件が適合すると、ポリシー サーバは偽装者を認証し、CSR はインパーソネーションセッションが存続する間、顧客を装うことができるようになります。
4. 最後に、ポリシー サーバは CSR (顧客を偽装) を FCC のターゲットに送ります。

インパーソネーションでのセキュリティに関する考慮事項

顧客を偽装している間の偽装者のセッション仕様は、他の顧客のセッション仕様とほとんど同じであるため、SiteMinder からは区別できません。主な違いは、偽装者の識別名を示すフィールドと、偽装者が最初に認証されたユーザ ディレクトリを示すフィールドが追加されることです。これにより、偽装を利用したリソースへのアクセスはすべて、追加のチェックを経るよう強制されます。また、ポリシー サーバでインパーソネーションアクティビティを記録して監査することも可能になります。

インパーソネーションセッション仕様は、インパーソネーションの連鎖を防止する目的にも利用されます。ポリシー サーバが偽装者の DN フィールドとユーザディレクトリのフィールドが使用中であると判断すると、以降のインパーソネーションは許可されず、ログインしようとしても拒否されます。これにより、偽装者がインパーソネーションを重ねて利用して、本来ならアクセスが許可されないリソースにアクセスするのを防止することができます。

認証方式保護レベルの有効性

ユーザを偽装する場合、偽装者が最初に認証された保護レベルはチェックされません。通常は、より高い保護レベルが設定されている認証方式で保護されたリソースにアクセスしようとする、新しい認証情報の入力が必要です。ただし、偽装者はもともと権限を持つユーザであるため、インパーソネーション セッションで改めて認証情報の入力が必要はありません。保護レベルは、レルム内のリソースにアクセスするために使用される認証情報の強度を示すものです。インパーソネーションの場合、偽装対象ユーザに固有の認証情報がないため、保護レベルは考慮されません。

注: インパーソネーションセッションが終了すると、再び、通常のように保護レベルの確認が行われるようになります。

セッションアイドルタイムアウト

インパーソネーション セッションの実行中に、偽装者の本来のセッションがアイドル状態になり、タイムアウトが発生することがあります。タイムアウトが起こるかどうかは、偽装者が最初に認証されたレルムのアイドルタイムアウト値によって決まります。インパーソネーション セッションは、偽装者の本来のセッションに設定されているアイドル タイムアウト時間を超えて続行されると、本来のセッションと同時に終了します。この問題を回避するには、偽装者が通常認証を受けるレルムのセッションアイドルタイムアウト時間を長く設定してください。

インパーソネーションの制限

インパーソネーションを実行するには、インパーソネーションレルム内で偽装者と偽装対象をポリシーに関連付ける必要があります。偽装者のポリシーには **ImpersonateStart** イベントに対して起動するルールを含め、偽装対象のポリシーは **ImpersoanteStartUser** イベントに対して起動するように設定する必要があります。これらのオブジェクトを設定する手順は、「インパーソネーション用のポリシー サーバ オブジェクト」で説明されています。

インパーソネーションレلمおよびイベント

インパーソネーションは、インパーソネーション プロセスを開始するように特定されたレلمで開始されます。偽装者がインパーソネーションレلم内のリソースを要求すると、インパーソネーション認証方式によって認証情報の入力が必要です。認証情報の入力にはフォームを使用します。この場合、フォームには通常のユーザ名とパスワードではなく、インパーソネーション対象ユーザのユーザ名を入力する必要があります。フォームの .fcc ファイルには、パスワードをインパーソネーターの現在のセッション仕様に設定するロジックが含まれています。

ポリシー サーバはインパーソネーション認証方式を使用して、インパーソネーターの現在のセッションが有効であることを確認します。その後で、インパーソネーションレلمに関連付けられたポリシードメインにインパーソネーション対象ユーザが存在するかどうかを確認します。ユーザが見つかり、認証が行われます。

ディレクトリで偽装対象ユーザの存在を確認した後、ポリシー サーバは偽装者にインパーソネーションを実行する権限があるかどうか、偽装対象ユーザがインパーソネーション可能であるかどうかを確認する必要があります。これらの権限の設定と実行では **SiteMinder** ポリシーとルールのほか、2 つのインパーソネーション イベントが使用されます。

インパーソネーション イベントは、認証イベントや許可イベントと似ていますが、受動的に起動される点が異なります。ポリシー では、偽装者と偽装対象ユーザを偽装者イベント ルールに明示的にバインドする必要があります。ポリシー が存在しないか、偽装者および偽装対象ユーザの偽装者イベント ルールが含まれていないと、インパーソネーションは許可されません。

インパーソネーションは、**SiteMinder** ポリシーとルール、およびインパーソネーション イベントを使用してレلمで許可または拒否することができます。レلمは、すべてのインパーソネーションを許可するか、指定された偽装者と偽装対象のみにインパーソネーションを制限するように設定できます。

インパーソネーション用のポリシー サーバ オブジェクト

企業内でインパーソネーションを実行するには、いくつかのポリシー サーバ オブジェクトを設定する必要があります。これらのオブジェクトを組み合わせることで、ユーザが別のユーザを偽装するために必要な認証およびポリシーの権限付与が可能になります。インパーソネーションを実行するには、以下のオブジェクトが必要になります。

インフラストラクチャ オブジェクト

エージェント

インパーソネーションには、**Web** エージェントと、それに関連付けられたポリシー サーバ エージェント オブジェクトが必要です。

注: インパーソネーションを実行するには、1 つ以上の **SiteMinder Web** エージェントをインストールしている必要があります。**Web** エージェントのインストールについては、「**Web** エージェント インストール ガイド」で説明しています。

認証方式

インパーソネーションには、インパーソネーション テンプレートに基づいた認証方式オブジェクトが必要です。

ユーザディレクトリ

インパーソネーションには、偽装者と偽証対象者を含むユーザ ストアを指す、1 つ以上のユーザ ディレクトリ オブジェクトが必要です。2 人のユーザ (偽装者と偽証対象) は、属性値またはグループのメンバーで区別できなければなりません。

ドメイン

インパーソネーションには、上記のユーザ ディレクトリ オブジェクトを含むポリシー ドメイン オブジェクトが必要です。

ドメインオブジェクト:

レルム

2 つ以上のレルム オブジェクトを設定する必要があります。1 つのレルムには偽証対象ユーザがアクセス可能なリソースを含めます。もう 1 つのレルムには、インパーソネーションセッションを初期化するために必要なリソースとルールを含めます。このレルムがインパーソネーションレルムになります。

ルール

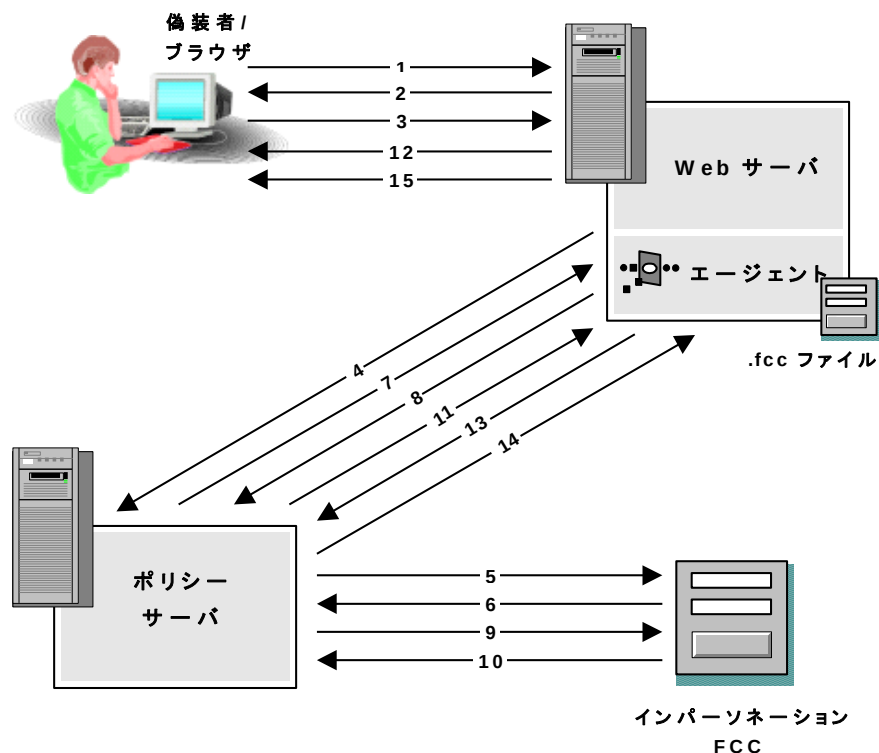
インパーソネーションには、所定のアクセス制御ルールが必要です。また、インパーソネーションセッションを開始するには、**ImpersonateStart** イベントが設定されたルールを偽装者用に作成する必要があります。**ImpersonateStartUser** イベントが設定されたルールがないと、偽装対象のインパーソネーションは許可されません。

ポリシー

インパーソネーションには、一連のリソースを保護するために必要なポリシーに加えて、追加のポリシーが必要です。このポリシーは、インパーソネーションレム内のリソースへのアクセスを許可し、十分な権限を持つユーザを偽装者として認定して、偽装対象となる一連のユーザを限定します。

インパーソネーションセッションの開始方法

以下の図は、権限を持つユーザがインパーソネーション セッションを開始するための処理フローの詳細を示します。



1. 既に認証された偽装者がインパーソネーション .fcc ファイルにアクセスして、別のユーザのインパーソネーションを開始します。
2. .fcc ファイルは保護されていないか、CSR にアクセス許可が与えられています。これについては図示していません。Web エージェントにより、手順 で要求されたフォーム(.fcc)が表示されます。FCC には非表示のフォームフィールドがあり、ターゲットがインパーソネーション認証方式で保護されているレルム内にあることを示すように設定されています。
3. 偽装者が偽装対象ユーザの名前を入力します。
4. Web エージェントが .fcc ファイルのターゲットフィールドをもとにレルムを特定します。
5. ポリシー サーバがインパーソネーション認証方式で必要とされる認証情報を判断します。
6. 必要な認証情報が返されます。
7. ポリシー サーバは .fcc ターゲットを保護しているレルムを通知します。
8. Web エージェントは、偽装者のセッション仕様をパスワードとして使用して、偽装対象ユーザを認証しようとします。
9. インパーソネーション認証方式の機能が呼び出され、認証情報(偽装対象ユーザの名前と偽装者のセッション仕様)を持つユーザを認証します。偽装者のセッション仕様の有効性が検証されます。ポリシー サーバによってユーザが明確化され、インパーソネーションセッションの認証が行われます。
10. ポリシー サーバによって、イベントポリシーが ImpersonateStart イベントに関連付けられていることが確認されます。偽装者に適用されるポリシーがある場合には、そのポリシーに ImpersonateStartUser イベントが関連付けられているかどうかを確認するチェックも実行されます。どちらのポリシーのルールにもレスポンスを関連付けることができます。

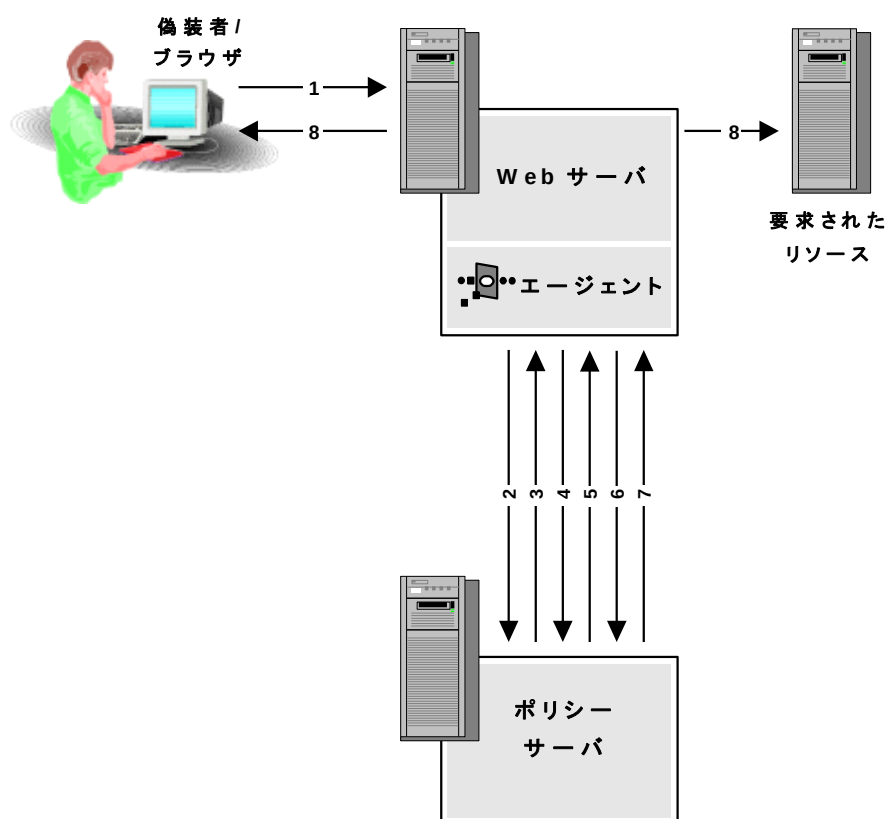
11. インパーソネーション認証方式を使用して偽装者の認証が完了すると、偽装者は任意のアプリケーションでユーザを偽装することができるようになります。

注: この段階では、偽装者がリソースにアクセスしていないため、インパーソネーションは実行されていません。

セッション仕様が Web エージェントに返されます。セッション仕様には、偽装対象ユーザの DN と偽装者の DN、偽装対象ユーザおよび偽装者の本来のディレクトリが含まれています。

Web エージェントが既存の **SMSESSION** を **SMSAVEDSESSION** に移動し、FCC の **@pushsession** ディレクティブに従って、新しいセッション仕様と同じ **SMSESSION Cookie** を新たに設定します。

12. 偽装対象ユーザが .fcc ターゲットリソースへのアクセスを許可されているかどうかを Web エージェントが判断します。
13. 偽装対象ユーザが .fcc ターゲットリソースへのアクセスを許可されているかどうかをポリシー サーバが判断します。
14. 偽装者が .fcc インパーソネーションファイルで指定されたリソースにアクセスします。たとえば、偽装者の DN と偽装対象の DN を示す .jsp ページにアクセスします。



1. 偽装者が(別のユーザを装って)新しいアプリケーションに移動します。
2. Web エージェントはポリシー サーバに確認して、リソースが保護されているかどうかを判断します。
3. 偽装者が要求されたリソースにアクセスします。
4. リソースは保護されており、この Cookie ドメインの偽装者の Web ブラウザには **SMSESSION Cookie** が存在するため、Web エージェントはポリシー サーバを呼び出してセッション仕様を検証します。
5. 現在の認証ロジックに従ってセッション仕様が確認されます。ただし、このロジックで確認できるのは、セッション仕様の内容に従ってインパーソネーションが発生しているということだけであるため、偽装者とユーザについては別途確認が行われます。偽装者は、**ImpersonateStart** イベントのルールが含まれる現在のレルム内のイベントポリシーに関連付けられていなければなりません。また、ユーザは、偽装対象となり得るユーザに適用される **ImpersonateStartUser** イベントの同様のポリシーに関連付けられていなければなりません。

6. **Web** エージェントは、ユーザがリソースへのアクセスを許可されているかどうかをポリシー サーバに確認します。
7. ポリシー サーバは **Web** エージェントに許可レスポンスを返します。許可レスポンスでは、アクセスが許可または拒否されていることを示すだけでなく、アクセスが拒否されている場合にはその理由（このリソースに対してルールが起動していない、そのユーザについてはアクセスが明示的に禁止されている、許可レベルが低い、セッションがタイムアウトしているなど）を示すことができます。これらのアクセス拒否の理由はすべて、ユーザ体験を反映しています。ただし、インパーソネーションセッションでは保護レベルは考慮されません。
8. 要求されたリソースが返されます。

startimp.fcc を使用するエージェントの設定

インパーソネーション セッションを呼び出すには、サンプルの **startimp.fcc** などの **.fcc** ファイルをインパーソネーション認証方式で指定する必要があり、**Web** エージェントで処理する必要があります。デフォルトでは、**Web** エージェントは **.fcc** ファイル拡張子を持つファイルを保護しません。

Web エージェントでインパーソネーション セッションを初期化できるようにするには、以下のいずれかの措置を取る必要があります。

- **Web** エージェントを、**.fcc** 拡張子が無視されないように設定する。
- インパーソネーションを呼び出すために使用する **.fcc** ファイルに、別の（無視されない）ファイル拡張子を使用する（たとえば、**.ifcc**）。

インパーソネーション認証方式の設定

インパーソネーションではインパーソネーション認証方式を使用する必要があります。この方式は、偽装者がインパーソネーションプロセスを開始するための手段として使用されます。インパーソネーションセッションを開始するには、偽装者は **.fcc** ファイルに直接アクセスします。このファイルは、インパーソネーション認証方式で保護されたリソースです。

.fcc ファイルによるインパーソネーションの有効化

インパーソネーションでは、適切な認証方式と .fcc ファイルを使用した認証プロセスが非常に重要になります。インパーソネーション認証方式の実行を可能にする .fcc ファイルは、SiteMinder 対応の Web サーバにインストールされるファイルです。このファイルには、拡張子 .fcc が付いています。この拡張子が付いたファイルには、Web エージェントによる特殊な処理を適用できます。インパーソネーション認証方式は、HTML フォーム認証方式によく似ています。

インパーソネーションでは、インパーソネーション認証方式用の .fcc ファイルに直接アクセスして認証方式を呼び出すことができます。正しい認証方式は、.fcc ファイルのターゲット パラメータから呼び出されます。この値は、@target ディレクティブを使用するか、ターゲット (非表示のフォーム ポスト変数) を適切な Web ページに設定して、.fcc ファイルにハードコーディングできます。

注: ターゲットリソースは、インパーソネーション認証方式で保護されているレルム内になければなりません。

インパーソネーションでの .fcc ファイルの基本要件

@smpushsession=true 命令文にもインパーソネーション プロセスを開始する FCC を含める必要があります。この命令文は、セッション cookie に偽装対象ユーザのセッション仕様を保持するために、現在のセッション cookie の内容を別の cookie に保存するように Web エージェントに指示します。

認証が行われるためには、インパーソネーション認証方式を処理する Web エージェントに対して適切な認証情報を提示する必要があります。ユーザ名には、偽装対象ユーザの名前を指定します。パスワードは、偽装者のセッション仕様に設定します。必要に応じて、その他の属性を追加します。これには、フォームが送信された時点で cookie またはヘッダーの内容を命令文に代入する FCC の機能を利用します。FCC はこの機能を使用して、@password ディレクティブをユーザセッション仕様に設定し、必要に応じて他のデータを追加します。.fcc ファイルの詳細については、「[HTML フォーム認証方式](#) (P. 324)」を参照してください。

インパーソネーション プロセスを終了するには、インパーソネーション認証方式で保護されたレルムに別の .fcc ファイルを追加します。この .fcc ファイルでは、インパーソネーションプロセスを開始する際に使用されたレルム内のリソースを指すように @target 命令文を設定します。また、認証プロセスを強制的に終了するように、@smredirect 命令文を同じリソースに設定します。最後に、@smpopsession=true 命令文を使用して元のセッション cookie を復元します。

インパーソネーションに使用される FCC 命令文

インパーソネーション用の .fcc ファイルを作成するときには、ファイルに次のような命令文を使用します。

@logout

SiteMinder からユーザをログアウトし、SMSESSION cookie を削除します。

@smheaders

HTTP 要求ヘッダを FCC ネームスペースに追加します。インパーソネーションの場合、このディレクティブはセッション仕様ヘッダ **SMSEVERSESSIONSPEC** (または **SM_SERVERSESSIONSPEC**、**「SMSEVERSESSIONSPEC および LegacyVariables に関する注意事項」**を参照) の内容を FCC ネームスペースに渡します。これにより、セッション仕様をパスワードとして使用できるようになります。

@smpushsession

ユーザが、別のユーザを「偽装」した後に、元のセッションに戻ることができるようにします。この命令文は「真」に設定する必要があります。

@smpopsession

@smpushsession を使用した後、元のセッションに戻ります。この命令文は「真」に設定する必要があります。

@smredirect

要求を指定されたターゲットにリダイレクトします。

@target

URL を処理した後のリダイレクト先を FCC に指示します。

@password

パスワードの内容をポリシー サーバに渡すかどうかを指定します。

@smaltcreds

カスタム認証方式で、4KB を超えるサイズのクレデンシャルの送信が可能になります。これは、**@password** ディレクティブを使用する場合と同じ方法で使用できます。クレデンシャルが、**@smaltcreds** を使用して FCC に送信されると、その値がログイン中に、バイト バッファとしてポリシー サーバに送信されます。これは、4K バイトに制限されているパスワード フィールドが使用されないようにするためです。**@smaltcreds** ディレクティブは、既存の作成済み認証方式では使用できませんが、カスタム認証には使用できます。カスタム認証方式の開発者は、ログイン中にエージェント API 経由で渡されるユーザ クレデンシャル構造の `[IpszCertBinary]` フィールドで **@smaltcreds** クレデンシャルを検索するように、認証方式ライブラリのコードを作成する必要があります。

@username

ポリシー サーバに渡すユーザ名を指定します。

% および \$ 置換機能

「%」機能と「\$\$」機能は、Perl 言語のスカラー変数と同じように、データ代入に使用されます。「%NAME%」は、POST 処理で "NAME" を "NAME" に関連付けられたデータに置換する際に使用します。「\$\$NAME\$\$」は GET 処理で "NAME" を "NAME" に関連付けられたデータに置換する際に使用します。

FCC を使用したセッション仕様の取得

Web エージェントは、要求ごとに一連のヘッダにセッション仕様を追加します。セッション仕様ヘッダは、いくつかの FCC 命令文を使用してパスワードとして送信できます。**SMSERVERSESSIONSPEC** ヘッダを FCC のネームスペースに含めるには **@smheaders** ディレクティブを使用し、**SMSERVERSESSIONSPEC** ヘッダの内容をパスワードとして設定するには **@password** ディレクティブを使用します。

SM_SERVERSESSIONSPEC/SMSERVERSESSION ヘッダーは、Web エージェント構成ファイルで **DisableSessionVars** がデフォルト値である「偽」に設定されている場合にのみ使用できます。

注: **LegacyVariables** Web エージェントパラメータを「はい」に設定する場合は、ヘッダが **SM_SERVERSESSIONSPEC** である必要があります。**LegacyVariables** Web エージェントパラメータを「いいえ」に設定する場合は、ヘッダが **SMSERVERSESSIONSPEC** である必要があります。IIS 6.0 Web サーバ上の Web エージェントでは、**LegacyVariables** パラメータがサポートされていません。IIS 6.0 上の Web エージェントの正しいヘッダは常に **SMSERVERSESSIONSPEC** です。

インパーソネーション用の .fcc ファイルの例

次の .fcc ファイルが偽装者によって直接呼び出されるか、インパーソネーション認証方式によって呼び出されると、インパーソネーションプロセスが開始されます。

```
@username=%USER%
@smheaders=%SMSERVERSESSIONSPEC%
@password=%SMSERVERSESSIONSPEC%
@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>
<tr>
<td colspan=2><input type=hidden name=target value="server.example.com/app/
impersonatee/successimp.htm ">
</td></tr>
<tr><td><input type=submit></td></tr>
</table></form></body>
</html>
```

偽装者によって次の .fcc ファイルが呼び出されると、元のセッションに戻ります。

```
@smpopsession=true
@target=/impersonators/end.htm
@smredirect=/impersonators/end.htm
```

FCCCompatMode の効果

従来の Web エージェントは、デフォルトで `fcccompatmode` を「はい」に設定しますが、フレームワーク Web エージェントは `fcccompatmode` を「いいえ」に設定します。モードが「はい」に設定されると、Web エージェントはインパーソネーションリクエストを処理できます。ただし、モードが「いいえ」に設定されている場合は、Web エージェントがインパーソネーションリクエストを処理するために、以下の 2 つの要件を満たす必要があります。

- Web エージェント設定パラメータ `EncryptAgentName` が `False` に設定されている必要があります。
- Web エージェント設定パラメータ `AgentName` が、インパーソネーションを開始する fcc ファイル内、または、インパーソネーションセッションの開始時にエージェントに送信されるクエリ データ内でコーディングされている必要があります。

インパーソネーション イベントの設定

インパーソネーション イベントはルール イベントです。権限を持つユーザが別のユーザを偽装するには、このイベントを設定してポリシーに含める必要があります。

インパーソネーションを可能にするには、次の各インパーソネーションイベントを少なくとも 1 つ設定する必要があります。

ImpersonateStart

適切なポリシーに追加すると、このイベントを含むルールによって、インパーソネーション セッションの開始が可能になります。

ImpersonateStartUser

適切なポリシーに追加することで、このイベントを含むルールによって、特定のユーザのインパーソネーションが可能になります。

インパーソネーション用のポリシーの設定

インパーソネーションが正常に機能するためには、いくつかのポリシーを設定する必要があります。

インパーソネーションを開始するリソースについては、次のようなポリシーを設定します。

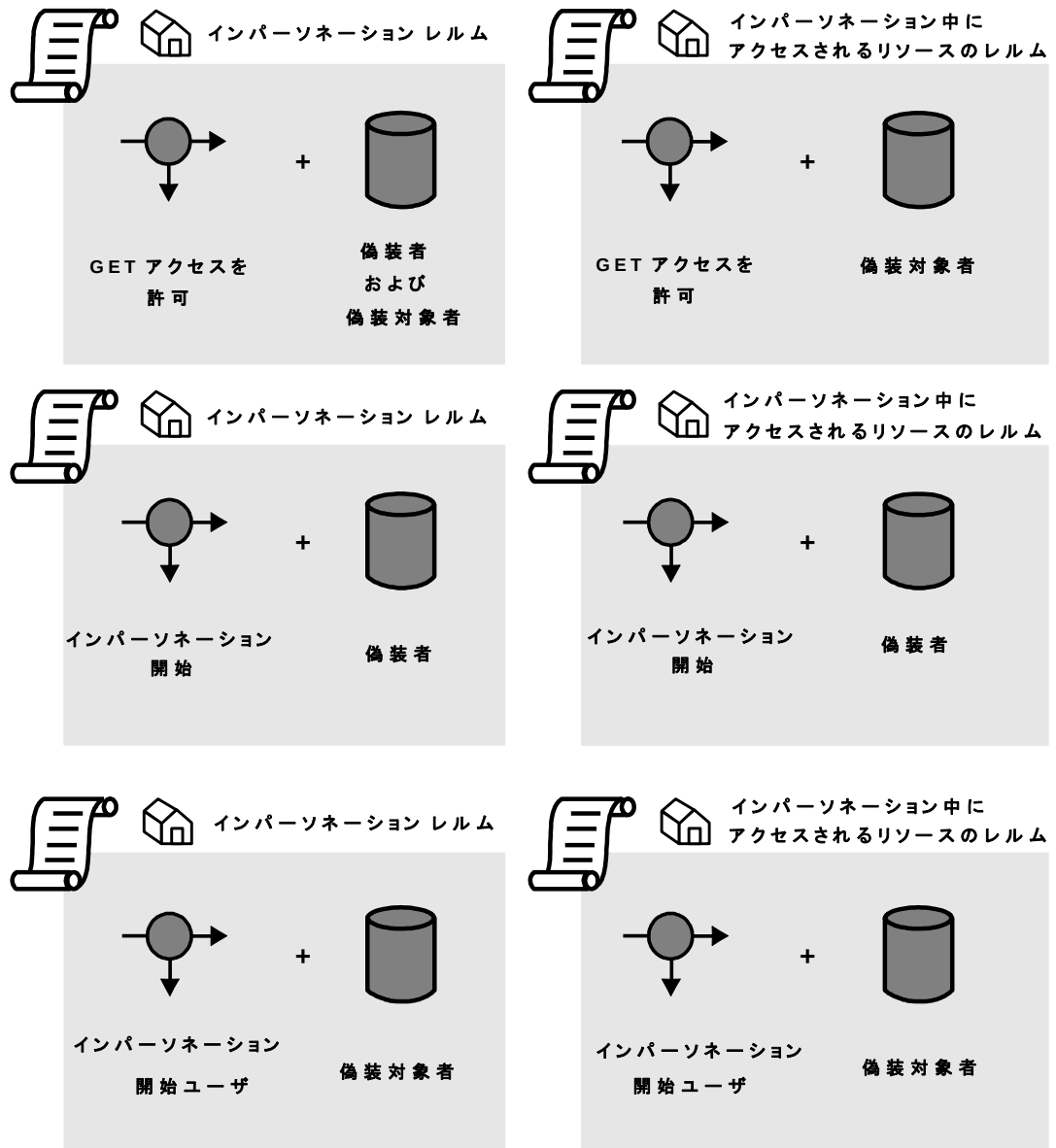
- アクセス制御ポリシー。偽装者と偽装対象の両方について設定します。たとえば、偽装者と偽装対象の両方について、GET アクセスルールが含まれるポリシーを設定します。
- ImpersonationStart ルールと偽装者を含むインパーソネーションポリシー。
- ImpersonationStartUser ルールと偽装対象を含むインパーソネーションポリシー。

偽装者がアクセスするリソースについては、次のポリシーを設定します。

- インパーソネーション対象ユーザのアクセス制御ポリシー。たとえば、インパーソネーション対象ユーザの **GET** アクセス ルールを含むポリシーです。
- **ImpersonationStart** ルールと偽装者を含むインパーソネーションポリシー。
- **ImpersonationStartUser** ルールと偽装対象を含むインパーソネーションポリシー。

注: インパーソネーターがアクセスする各リソースには、ポリシーを設定する必要があります。

以下の図は、リソースにアクセスするインパーソネーターに必要な最低限のポリシーを示します。



複数 cookie ドメインのサポート

1 つのサイトが複数の Cookie ドメインで構成されている場合は、異なる Cookie ドメイン内にあるリソースの間を移動する間に、偽装者の識別情報が混乱する可能性があります。この問題を回避するには、現在の cookie ドメインを除く、すべての cookie ドメインの SMSESSION cookie をクリアする必要があります。これには、インパーソネーションの開始と終了に使用されるフォームを変更します。これらのフォームに、現在の cookie ドメインを除く、すべての cookie ドメインから SMSESSION cookie を削除するコードをサーバ側で呼び出す HTML またはスクリプトを追加します。

たとえば、Web エージェントが、`yourcompany.com`、`subsidiaryA.com` および `subsidiaryB.com` でインストールされたとします。インパーソネーションを実行する Web エージェントは `yourcompany.com` ドメインにあった可能性があります。この場合にインパーソネーションが正常に機能するためには、インパーソネーションを開始および終了する .fcc ファイルが、`subsidiaryA.com` と `subsidiaryB.com` でサーバ側の機能（おそらく JSP ページ）を呼び出して、これらの cookie ドメインから SMSESSION cookie を削除する必要があります。

インパーソネーションのサンプル実装

ここでは、簡単なインパーソネーションの実装例を説明します。インパーソネーションを実行するために必要な最低限のポリシー サーバ オブジェクトは次のとおりです。

インフラストラクチャ オブジェクト

エージェント

インパーソネーションには、Web エージェントと、それに関連付けられたポリシー サーバ エージェント オブジェクトが必要です。

注： インパーソネーションを実行するには、1 つ以上の SiteMinder Web エージェントをインストールする必要があります。Web エージェントのインストールについては、「Web エージェント インストール ガイド」で説明しています。

認証方式

インパーソネーション認証方式テンプレートに基づくインパーソネーション認証方式が必要です。ここで定義する例では、認証方式の名前は「Impersonation Auth」です。

ユーザ ディレクトリ

インパーソネーションには、偽装者と偽証対象者を含むユーザ ストアを指す、1 つ以上のユーザ ディレクトリ オブジェクトが必要です。2 人のユーザ (偽装者と偽装対象) は、属性値またはグループのメンバーで区別できなければなりません。

ドメイン

ポリシー ドメインが必要です。ここで定義する例では、ポリシー ドメインの名前は「Impersonation Domain」です。

ドメイン オブジェクト:

レルム

ここで定義する例では、「Impersonation」および「App1」という 2 つのレルムが必要になります。「Impersonation」レルムでは、「Impersonation Auth」認証方式を使用する必要があります。「App1」レルムでは、任意の認証方式を使用できます。

ルール

ここで説明する例では、すべてのリソースに「Get」アクションへのアクセスを許可するルールを、「Impersonation」レルムの下に設定する必要があります。つまり、ルールの [リソース] フィールドにアスタリスク(*)を入力します。また、インパーソネーション イベントのルールも設定する必要があります。この場合、偽装者が該当するポリシーに含まれている場合にインパーソネーションを許可するルールと、偽装対象が別の該当するポリシーに含まれている場合にその対象が偽装されることを許可するルールが必要です。

ルール

「Impersonation」レルムに設定したものと同様のルールを、「App1」レルムの下にも作成する必要があります。

ポリシー

ここで説明する例の場合は、6 つのポリシーが必要です。「Impersonation」レルムの各ルールと「App1」レルムの各ルールにそれぞれ 1 つずつポリシーを定義します。

サンプル インパーソネーション実装の評価

必要なポリシー サーバオブジェクトをすべて作成したら、管理者は次の手順を実行してインパーソネーションセッションを開始します。

1. 偽装者となる管理者が、**SiteMinder** で保護されたネットワークにログインします。
2. 管理者はクレデンシャルを入力し、**SiteMinder** から認証されて許可を受けます。
3. 管理者が **Web** ブラウザで **imp.fcc** ファイルにアクセスします。
4. 偽装対象となるユーザのユーザ ID の入力を求められます。場合によっては、偽装対象ユーザに関する追加情報の入力を求められることもあります。
5. 管理者が情報を送信します。
6. ポリシー サーバはインパーソネーション用に定義されたポリシーを使用して、次のことを判断します。
 - 管理者に偽装者としての役割を果たす権限があるかどうか
 - 偽装対象ユーザが偽装されることを許可されるかどうか
7. 2 つの条件が満たされれば、偽装者が偽装対象ユーザを装うことが許可されます。

注: インパーソネーション セッションは、カスタム アプリケーション(.jsp ページ、サーブレットなど)を使用せずに、ポリシー サーバの監査ログ機能で追跡管理できます。ただし、インパーソネーションセッションを監視および追跡管理するためのカスタム **Web** アプリケーションを作成しておくとう便利です。

インパーソネーションのサンプルフォーム

インパーソネーションにはいくつかの **Web** ベースのフォームが必要になります。これらのフォームは、それぞれの環境で使用するインタフェースに合わせてカスタマイズする必要があります。ここでは、インパーソネーションに必要なフォームの基本要件と、フォームの内容に関する推奨事項を説明します。

インパーソネーションプロセスを開始するためのフォーム

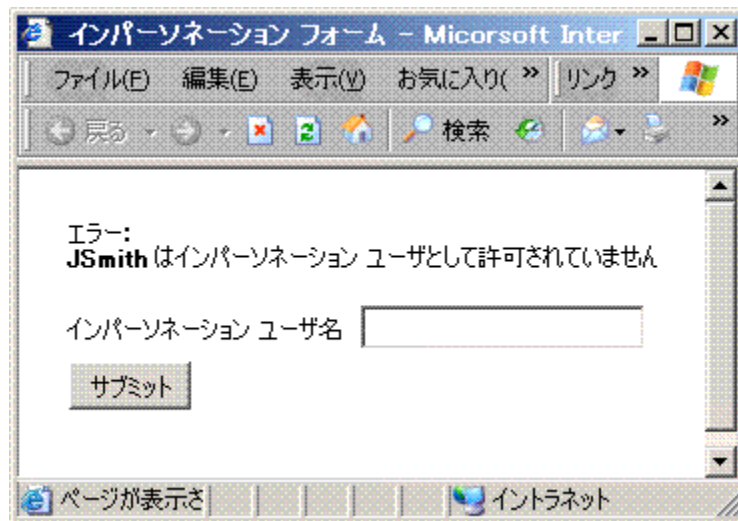
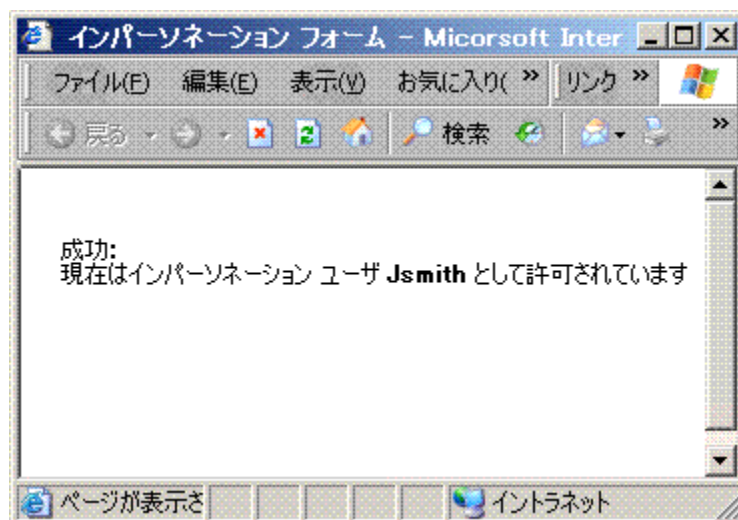
フォームには、インパーソネーション処理を呼び出す **.fcc** ファイルで表示する必要がある基本要素が含まれています。

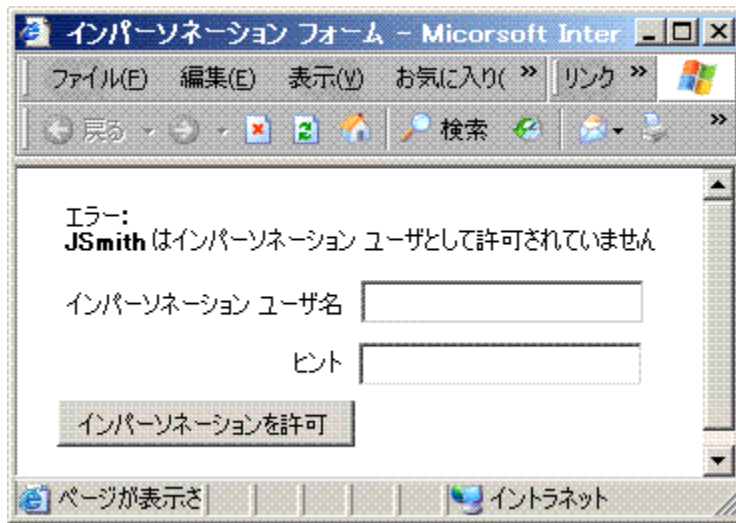
ヒントのフィールドは、偽装対象ユーザを特定するための追加の属性が必要になるようにインパーソネーションを設定できることを示しています。このヒントは、インパーソネーション認証方式で追加属性のリストとして設定できます。属性には、社員番号など、ユーザ ディレクトリでユーザを一意に定義するその他のユーザ属性などの情報を含めることができます。

インパーソネーション結果フォーム

インパーソネーションを試みた場合は、その成功と失敗の両方の結果を表した形式で、そのことを伝える必要があります。

注: 以下の形式は、各形式の基本的な要件を示しています。インパーソネーション形式はすべて、カスタマイズして、.fcc ファイルから参照されるようにする必要があります。

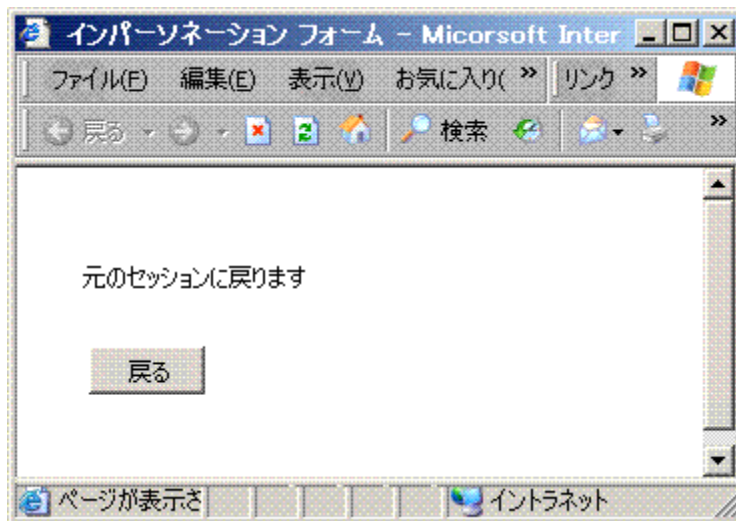




インパーソネーションログオフフォーム

インパーソネーションセッションを終了するためのフォームも設定する必要があります。

注: 以下の形式は、その基本的な要件を示しています。インパーソネーション形式は、カスタマイズして、.fcc ファイルから参照されるようにする必要があります。



第 23 章: パスワード ポリシー

このセクションには、以下のトピックが含まれています。

- [パスワード サービスの概要 \(P. 717\)](#)
- [パスワード ポリシーの作成 \(P. 724\)](#)
- [パスワードの有効期限の設定 \(P. 725\)](#)
- [パスワード構成の設定 \(P. 726\)](#)
- [パスワード正規表現 \(P. 726\)](#)
- [パスワードの制限の設定 \(P. 729\)](#)
- [高度なパスワード オプションの設定 \(P. 730\)](#)
- [ユーザが開始するパスワード変更 \(P. 731\)](#)
- [パスワード サービスでリダイレクトされる際のログイン ID の削除 \(P. 735\)](#)
- [CGI ベースのパスワード サービスの HTML フォームのカスタマイズ \(P. 736\)](#)
- [サブリットベース パスワード サービスの JSP 形式のカスタマイズ \(P. 742\)](#)
- [追加属性情報を必要とする認証方式 \(P. 748\)](#)
- [パスワード ポリシーのトラブルシューティング \(P. 750\)](#)

パスワード サービスの概要

パスワード サービスを利用すると、LDAP ユーザディレクトリまたはリレーショナルデータベースでユーザ パスワードの管理が可能になり、保護されているリソースに対するセキュリティをさらに強化できます。ユーザ パスワードを管理するには、パスワード ポリシーを作成してルールと制限を定義し、パスワードの有効期限、構成、使用方法を決定します。

パスワード サービスは、パスワードを管理だけではなく、以下のような場合にユーザがパスワードを選択できるようにします。

- 指定したタイミング(パスワードの期限が切れたときなど)
- ユーザがパスワード変更の必要性を感じたとき
- 新規ユーザが登録サービスを使用して登録を行ったとき

パスワード ポリシーを使用すると、特別な管理作業を行わなくても、ユーザは有効なパスワードを確実に選択できます。

パスワード サービスが有効な場合、保護されているリソースにユーザがアクセスしようすると、**SiteMinder** はパスワード ポリシーを呼び出します。次に、パスワード サービスはユーザのクレデンシャルを評価します。パスワード ポリシーに定義された条件に基づいてユーザのパスワードが期限切れになった場合は、不正なアクセスからリソースを保護するためにユーザのアカウントを無効にできるほか、パスワードの変更をユーザに強制できます。ユーザのアカウントを無効にした場合は、管理者が再度有効にする必要があります。

パスワード ポリシーは、ユーザディレクトリかデータベース全体に、またはネームスペースといわれるディレクトリもしくはデータベースのサブセットに関連付けることができます。同じユーザディレクトリまたはネームスペースに複数のパスワード ポリシーを設定できます。この場合、指定した優先順位に従ってポリシーが適用されます。

パスワード サービスには、パスワード サービスを実行するための 3 つのメカニズムがあります。

CGI-based パスワードサービス

(非推奨) パスワード サービス CGI とカスタマイズ可能な HTML フォーム

FCC-based パスワードサービス

(デフォルト) パスワード サービス FCC とカスタマイズ可能な HTML フォーム

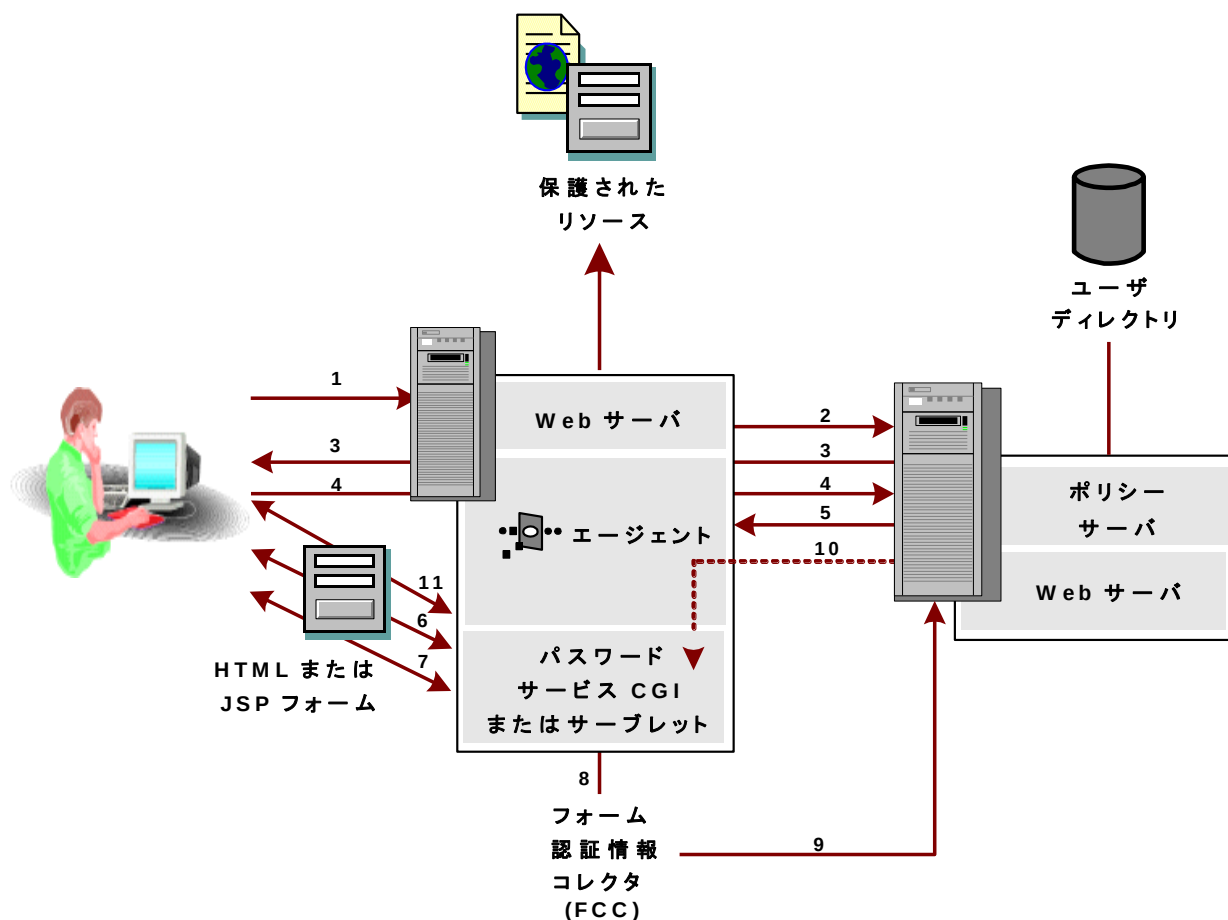
注: FCC ベースのパスワード サービスの詳細については、「Web エージェント設定ガイド」を参照してください。

サーブレットベースのパスワード サービス

パスワード サービス サーブレットとカスタマイズ可能な JSP フォーム

パスワード サービスの機能

次の図で、Web リソースを保護するように設定された SiteMinder 環境でパスワード サービスがどのように機能するかを説明します。ここでは、ユーザのパスワードが失効し、変更する必要がある場合を例として使用します。



ユーザが保護されたリソースにアクセスしようとした場合

1. ユーザは、Web サーバにリクエストを送信して、Web ページにアクセスしようとします。
2. SiteMinder Web エージェントはリクエストをインターセプトし、リクエストされたリソースが保護されているかどうかをポリシー サーバに確認します。
3. リソースが保護されている場合、ポリシー サーバはブラウザにユーザ証明を要求します。
4. ユーザは、ポリシー サーバに証明を送信します。

5. ポリシー サーバはユーザを認証します。

ユーザが認証されたら、ポリシー サーバは、パスワードポリシーに関連付けられているディレクトリまたはデータベースにユーザ情報が保存されているかどうかを確認します。ユーザ情報が保存されていれば、ポリシー サーバはパスワード ポリシー条件に基づいて、ユーザのパスワードが有効かどうかを確認します。

6. パスワードの有効期限が切れている場合、ポリシー サーバはエージェントにパスワードサービスサーブレットを指すリダイレクト URL を送信します。

このリダイレクト URL を使用して、エージェントはブラウザをリダイレクトし、パスワードサービスサーブレットを要求します。

7. CGI またはサーブレットは、ユーザへの表示に使用する適切な HTML/JSP フォームをテンプレートセットの中から決定し、それをブラウザに表示します。

この HTML フォームを使ってメッセージが表示され、ユーザがリダイレクトされた理由が説明されます。このメッセージで、ユーザは新旧のパスワードを入力し、さらに確認用に新しいパスワードをもう一度入力するように要求されます。

8. フォームへのユーザ入力が完了すると、パスワードサービス CGI またはサーブレットは、エージェントから受け取った情報、および暗号化されたユーザ入力情報を FCC (フォーム証明書コレクタ) に渡します。

9. FCC がパスワードサービス CGI またはサーブレットから受け取った情報をポリシー サーバに渡します。

ポリシー サーバは、新しいパスワードをパスワードポリシーと突き合わせてチェックし、パスワードの有効性を確認後にパスワードを変更します。

10. ポリシー サーバは、要求をエージェントにもう一度送信し、ブラウザをパスワードサービス CGI/サーブレットにリダイレクトします。

11. パスワードサービスサーブレットはメッセージを表示して、パスワードが正常に変更されたことをユーザに通知します。

ユーザがメッセージを確認すると、最初に要求したページにリダイレクトされます。

Apache Web サーバの前提条件

環境で実行されているオペレーティングシステムによっては、CGI ベースのパスワード サービスが確実に実装されるように、手動で Apache httpd.conf 設定ファイルを編集する必要があります。

Solaris 用の httpd.conf 設定ファイルを変更するには、ServerRoot 行の後に PassEnv 行を追加します。

例:

```
ServerRoot "/export/home/smuser/apache/"
PassEnv LD_LIBRARY_PATH "/export/home/smuser/netegrity/webagent/bin"
```

Windows 用の httpd.conf 設定ファイルを変更するには、以下の行を追加します。

```
# To use CGI scripts:
AddHandler cgi-script .exe
AddHandler smformsauth-handler .fcc
AddHandler smcookieprovider-handler .ccc
```

パスワード サービスの注意事項

パスワードサービスをセットアップする場合、次の点を考慮することが重要です。

- 複数のポリシー サーバでのパスワード サービスの時差による影響
パスワード サービスと複数のポリシー サーバを使用している場合は、ユーザ アカウントが突然無効になったり、有効期限が切れる前にパスワードの変更を強制されたりすることがないように、ポリシー サーバシステムの時刻設定を合わせてください。

- ディレクトリ サーバのパスワード ポリシーの使用

Netscape LDAP ディレクトリ サーバなどのディレクトリ サーバを使用すると、簡単なパスワード ポリシーを作成できます。このパスワード ポリシーは、SiteMinder ポリシーの前に評価されます。SiteMinder ポリシーの制限よりもディレクトリ サーバ ポリシーの制限の方が厳しい場合、ディレクトリ サーバは、SiteMinder に通知することなく承諾、または拒否します。SiteMinder は、ユーザがログインしようとしても認識せず、パスワード管理機能を一切提供しません。

重要： パスワードポリシーを使用するには、ディレクトリ サーバのすべてのポリシーを使用不可にするか、それらのポリシーに対する制限をパスワード ポリシーの制限よりも緩和します。

CGI またはサーブレットの場合のカスタム パスワード サービス ディレクトリの注意事項

デフォルト以外の Web サーバ上でカスタム パスワード サービス ディレクトリをセットアップするには、次の作業を行う必要があります。

1. デフォルト以外の Web サーバからアクセス可能な場所に `smpwservicescgi.exe` をコピーします。

CGI はリダイレクト URL から `agent_path` を取得して、デフォルトの `/siteminderagent/pw` パスを置き換えます。

2. 2 つの仮想ディレクトリ、`pw` および `pwcgi` を作成します。

後者は CGI ディレクトリです。以下に例を示します。

- `<custom directory>/pwcgi`
- `<custom directory>/pw`

ユーザがパスワード サービス CGI またはサーブレットにリダイレクトされると、パスワード サービス CGI またはサーブレットは、ポリシー サーバから取得した情報を基にパスワードが無効である理由を判断し、フォームを表示して情報を提供するか、ユーザに追加の認証情報を要求します。

パスワードサービス CGI またはサーブレットが保護されていないことを確認してください。SiteMinder がサーブレットの上位にあるディレクトリを保護している場合は、次の設定でレルムを作成してください。

- CGI を使用している場合のリソースフィルタ:
siteminderagent/pwcgi/smpwservicescgi.exe
- サーブレットを使用している場合のリソース フィルタ:
siteminderagent/pwservlet/PSWDChangeServlet
- デフォルトリソース保護: 保護なし

このレルムに対するポリシーは作成しないでください。

注: SSL 接続を使用していないエージェントを介してリソースにアクセスしているユーザがパスワードを変更する場合は、ユーザの新しいパスワード情報が、安全な接続を介して受信されます。パスワードを安全に変更するには、[リダイレクト URL] フィールドを使用して、SSL 接続を介してユーザをリダイレクトするパスワード ポリシーを設定します。

デフォルトの CGI リダイレクト URL の変更

デフォルトで、SiteMinder Web エージェントのインストールには以下が含まれています。

`/web_agent_install_dir/pw/smpwservicescgi.exe`

認証サーバでユーザを正しく明確化できない場合、ユーザはこのディレクトリにリダイレクトされます。

リダイレクト先をカスタマイズするには、NETE_PWSERVICES_REDIRECT 環境変数をリダイレクト先の相対 URL に設定し、Web エージェントに関連付けられている Web サーバを再起動します。

パスワード ポリシーの作成

保護されたリソースにセキュリティレイヤを追加するには、パスワード ポリシーを作成します。

パスワード ポリシーを作成する方法

1. [ポリシー]、[パスワード]をクリックします。
2. [パスワード ポリシー]、[パスワード ポリシーの作成]をクリックします。

[パスワード ポリシーの作成]ペインが表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

3. ポリシー名を入力します。
4. [ディレクトリ]リストから、ポリシーを適用するユーザ ディレクトリを選択します。
5. ポリシーをディレクトリ全体に適用するのか、一部に適用するのか指定します。

注:ディレクトリの一部にのみポリシーを適用する場合は、[検索]をクリックしてポリシーを適用するディレクトリの一部を指定します。

6. [リダイレクト URL]フィールドでリダイレクト URL を指定します。
 - パスワード サービス CGI を使用するには、以下を指定します。
`http://<server.company.org>/siteminderagent/pwcgi/smpwservicescgi.exe`
注: カスタムのパスワード サービス ディレクトリを設定している場合は、デフォルトのパスワード サービス CGI パスと異なることがあります。パスワード サービス CGI は廃止予定です。
 - パスワード サービス FCC を使用するには、以下を指定します。
`http://<server.company.org>/siteminderagent/forms/smpwservices.fcc`
注: パスワード サービス FCC はデフォルトのリダイレクト URL です。パスワード サービス FCC の詳細については、「Web エージェント設定ガイド」を参照してください。
 - パスワード サービス サーブレットを使用するには、以下を指定します。
`http://<server.company.org>/siteminderagent/pwservlet/PSWDChangeServlet`
7. 有効期限、構成、式、制限、詳細設定などを使用し、必要なパスワード ロジックを反映したポリシーを設定します。

パスワードの有効期限の設定

イベントを定義してパスワードに有効期限を設定します。定義したイベントがトリガされると、ポリシー サーバはそのユーザ アカウントを無効にし、必要に応じて新しい Web ページにユーザをリダイレクトします。このようなイベントには、何度もログインに失敗する、アカウントが未使用、などがあります。

注: 有効期限の設定はオプションです。有効期限の設定を有効にしない場合は、該当するフィールドに何も指定しないでください。

パスワードの有効期限を設定する方法

1. [有効期限]タブをクリックします。

パスワードの有効期限の設定が表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [有効期限]グループ ボックスで、[正常ログインの追跡]、[失敗したログインを追跡する]、[ログイン追跡に失敗したときに認証を実施する]チェックボックスを選択し、ユーザ ログインの追跡設定を指定します。

注: アカウントの未使用に基づいてアカウントを無効にする場合は、[正常ログインの追跡]チェックボックスを選択する必要があります。ログインの失敗に基づいてアカウントを無効にする場合は、[失敗したログインを追跡する]チェックボックスを選択する必要があります。

3. [パスワードが変更されない場合は失効]グループ ボックスで、パスワードを変更する必要がある頻度を設定します。
4. [不正なパスワード]グループ ボックスで、不正なパスワードが使われた回数の上限を指定する必要があります。
5. [パスワードが使用されなかったため失効]グループ ボックスで、パスワードを未使用のままにできる日数を指定します。

注: 未使用のパスワードを期限切れにする必要がない場合は、パフォーマンス上の理由からこのオプションは設定しないことをお勧めします。

6. [サブミット]をクリックしてパスワード ポリシーを保存するか、別のタブをクリックしてパスワード ポリシーの作業を続けます。

パスワード構成の設定

新しく作成されるパスワードの文字構成をコントロールするには、パスワード構成ルールを設定します。

注: 構成ルールはオプションです。構成ルールを有効にしない場合は、該当するフィールドに何も指定しないでください。

パスワード構成の制限を設定する方法

1. [構成]タブをクリックします。

パスワード構成の設定が表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [最小文字数]と[最大文字数]フィールドには、それぞれパスワードの最小文字数と最大文字数を入力します。
3. [最大]フィールドに、パスワードで連続して表示できる文字の最大数を入力します。
4. [最小文字数]グループ ボックスのそれぞれに、許容できる文字タイプと最小要件を指定します。

注: Netscape 4.1 ディレクトリ サーバとパスワード サービスを使用している場合は、印刷不可能文字の最小数を指定しないでください。Netscape 4.1 Directory Server では、印刷不可能文字をサポートしていません。

5. [サブミット]をクリックしてパスワード ポリシーを保存するか、別のタブをクリックしてパスワード ポリシーの作業を続けます。

パスワード正規表現

パスワードの正規表現照合では、テキスト パターンを指定して文字列照合を行い、パスワードがそれに一致するかどうかでそのパスワードの有効性を判断できます。

たとえば、パスワードの最初の文字は数字であり、この文字は最後の文字ではないことを条件とする場合、正規表現を設定してこの要件を適用すれば、すべてのパスワードがチェックされます。

正規表現の構文

以下のテーブルは、パスワードを照合する正規表現を作成する際に使用できる文字について説明します。この構文は、レلمを指定する場合のリソース照合をサポートする正規表現構文で構成されています。

量指定子(+、*、?)はデフォルトで **Greedy** です。つまり、全体的な照合を中断することなく、文字列内で可能な限り多くの要素と一致します。控えめ (**Non-greedy**) な照合を行うには、量指定子に「?」を追加します。控えめな量指定子を使用すると、照合時に文字列のできるだけ少ない要素と一致します。

正規表現構文は以下のようになります。

文字	結果
¥	メタキャラクタ(「*」など)を引用する場合に使用
¥¥	1 つの「¥」文字と一致
(A)	副次式のグループ化 (パターンの評価順序に影響)
[abc]	単純な文字クラス (文字 abc... の中の任意の 1 文字に一致)
[a-zA-Z]	範囲指定のある文字クラス (a から z、A から Z の範囲内にある 1 文字に一致)
[^abc]	文字 abc... 以外の任意の 1 文字に一致
.	改行以外の任意の 1 文字と一致
^	行の先頭に一致
\$	行の末尾に一致
A*	A が 0 回以上繰り返すものに一致 (greedy)
A+	A が 1 回以上繰り返すものに一致 (greedy)
A?	A が 1 回または 0 回現れるものに一致 (greedy)
A*?	A が 0 回以上繰り返すものに一致 (reluctant)
A+?	A が 1 回以上繰り返すものに一致 (reluctant)
A??	A が 0 回または 1 回現れるものに一致 (reluctant)
AB	A の後に B が続くものに一致

文字	結果
A B	A または B のいずれかが現れるものに一致
¥1	1 番最初のかっこで囲まれた副次式への後方参照
¥n	n 番目のかっこで囲まれた副次式への後方参照

制限: 各正規表現に含むことができる副次式は、その正規表現自体を含めて 10 個までです。副次式の数、正規表現に含まれる左側かっこの数に 1 を加えたものと同等です。

正規表現照合の設定

文字列マッチングに使用するテキストパターンを指定する正規表現を設定します。パスワードは、正規表現に一致する、または一致しないのを条件として有効にします。各正規表現エントリは、記述タグと表現定義で構成される名前/値のペアです。

パスワードに正規表現照合を使用するかどうかはオプションです。正規表現を使用する場合は、一致する必要がある、または一致してはいけないエントリを表現に指定するだけです。正規表現照合を行う必要がない場合は、正規表現エントリを作成しないでください。

パスワードに正規表現を設定する方法

1. [パスワード ポリシー] ダイアログで、[正規表現] タブを選択します。
[正規表現] グループ ボックスに、空のテーブルが表示されます。
2. [追加] をクリックし、表現を追加します。
[パスワード正規表現] ダイアログが表示されます。
3. 次のいずれかのラジオ ボタンをオンにします。
 - 一致する
このオプションを選択する場合は、パスワードが一致する必要がある正規表現を定義します。
 - 一致しない
このオプションを選択する場合は、各正規表現にパスワードが一致してはいけないエントリを追加します。

4. フィールドに値を入力します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

5. [OK]をクリックします。

正規表現がテーブルに追加されます。[一致しない]を選択している場合は、[一致なし]列にチェックボックスが表示されます。

パスワードの制限の設定

パスワードの使用について制限を設けるには、パスワードの制限を設定します。たとえば、以下の制限があります。

- パスワードの再使用を許可する経過期間
- 以前使用したパスワードと異なっていなければならない程度

また、セキュリティリスクがあるか、ユーザの個人情報が含まれると判断した用語を、ユーザに指定させないようにすることもできます。

注:制限はオプションです。制限を有効にしない場合は、該当するフィールドに何も指定しないでください。

パスワードの制限を設定する方法

1. [制限]タブをクリックします。

パスワードの制限の設定が表示されます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. [再使用]グループ ボックスに、以前と同じパスワードの使用が許可される経過期間、またはそのパスワードからの変更回数を指定します。

注:両方の条件を指定すると、パスワードを再使用できるようになるには両方の条件を満たす必要があります。

例:パスワード ポリシーで、以前のパスワードを再使用できるようになる条件として 365 日が経過していること、パスワードを 12 回指定していることを設定します。1 年経過した時点で、6 つのパスワードしか指定しなかった場合、ユーザはさらに 6 つのパスワードを指定しないと、最初のパスワードを再使用できません。

3. [必要な変更]グループ ボックスに、新しいパスワードが以前のパスワードと異なっていなければならない程度を指定します。
4. [プロファイル属性]グループ ボックスに、パスワード ポリシーで、ユーザ プロファイルに保存されている個人情報と比較する連続する文字の数を指定します。
5. [ディクショナリ]グループ ボックスに、禁止されているパスワードのユーザ定義ディクショナリへのパスとそのディクショナリ内の値と比較する文字列の長さを指定します。
6. [適用]をクリックして変更内容を保存します。または、[OK]をクリックして変更内容を保存し、管理 UI に戻ります。

高度なパスワード オプションの設定

高度なパスワード ポリシー オプションを設定し、サブミットされたパスワードを検証して保管する前にあらかじめ処理するよう指定します。高度なパスワード ポリシーでは、ポリシーに優先度を割り当てられるため、同じユーザ ディレクトリやネームスペースに複数のパスワード ポリシーが適用される場合でも評価が予測できるようになります。

注: 前処理オプションの設定は任意です。ユーザディレクトリまたはネームスペースに割り当てられるパスワード ポリシーごとに、一意のパスワード ポリシー評価の優先順位を指定する必要があります。

高度なパスワード オプションを設定する方法

1. [詳細設定]タブをクリックします。

高度なパスワード ポリシーの設定が表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

2. サブミットされたパスワードを評価と保管の前に処理する際のオプションを、[パスワードの前処理]グループ ボックスで指定します。

注: 同じユーザ ディレクトリまたはネームスペースに適用される各パスワード ポリシーには、それぞれ同じ前処理オプションを指定する必要があります。

3. (オプション) 同じユーザ ディレクトリやネームスペースに複数のパスワード ポリシーが適用されている場合は、[パスワード ポリシーの優先順位]グループ ボックスで優先順位を各パスワード ポリシーに指定します。

注: 評価の優先順位の範囲は、0-999 です。999 が最も高くなります。

ユーザが開始するパスワード変更

ユーザによるパスワード変更を使用すると、管理者の介入なしにエンドユーザがパスワードを変更することができます。ユーザは、[パスワードの変更要求]フォームにアクセスするリンクをクリックして、パスワードの変更を選択できます。

[パスワードの変更]リンクの追加

ユーザによるパスワード変更を有効にするには、ポリシー サーバ管理者が HTML ページに [パスワードの変更] のリンクを追加する必要があります。たとえば、管理者がこのリンクをログインページに追加すると、ユーザはログイン時にパスワードを変更できるようになります。

パスワード サービス CGI へのパスワードの変更リンクの追加

[パスワードの変更要求] フォームへのリンクは、次のとおりです。

```
<a href="/siteminderagent/pwcgi/smpwservicescgi.exe?SMAUTHREASON=34&TARGET=<URL_of_Protected_Web_Page_After_Password_Change>">Change Password</font></a>
```

ここで、<URL_of_Protected_Web_Page_After_Password_Change> は、保護された Web ページの URL で、パスワードの変更後にユーザに表示されます。たとえば、<http://server.myorg.org/AfterPasswordChange.html> などです。

注: このリンクを保護されていないページに追加すると、すべてのユーザが自分のパスワードを変更することができます。パスワードを変更した後にユーザに表示される URL は保護されている必要があります。保護されていない場合、パスワードの変更は失敗します。

また、SMAGENTNAME は FCCCompatMode が NO に設定されている場合、指定する必要があります。

たとえば、FCCCompatMode を「NO」に設定している場合、以下のようにリンクに SMAGENTNAME を指定します。

パスワードの変更リンクは以下のようになります。

```
"<a href="/siteminderagent/pwcgi/smpwservicescgi.exe?SMAUTHREASON=34&SMAGENTNAME=$$SMAGENTNAME$$&TARGET=$TARGET$">Change Password</font></a>"
```

特定のユーザに CGI でのパスワードの変更を可能にする

特定のユーザに対してのみ、パスワードの変更を有効にするには、次の手順を実行します。

特定のユーザがパスワードを変更できるようにする方法

1. PWLogin.template ファイルに対する権限を変更します。

- a. 次のディレクトリに移動します。

`web_agent_installation_dir/pw`

ここで、`web_agent_installation_dir` は Web エージェントがインストールされている場所です。

- b. [ファイル]、[プロパティ]の順に選択し、読み取り専用の属性をオフにします。

2. PWLogin.template ファイルのテキストを、次の手順で編集します。

- a. テキストエディタを使用して PWLogin.template を開きます。

- b. テンプレートの該当する場所に以下の行を追加します。

```
<a href="/siteminderagent/pwcgi/smpwservicescgi.exe?  
SMAUTHREASON=34&TARGET=$$TARGET$$">Change Password</a>
```

- c. ファイルを保存します。

3. 管理 UI にアクセスします。

4. 認証方式を、次の設定で作成します。

- サーバ名: パスワードサービス CGI のあるサーバの名前です。以下に例を示します。

`myserver.mycompany.org`

- ターゲット: `/siteminderagent/pwcgi/smpwservicescgi.exe`

5. 必要に応じて、新しいポリシードメインを作成します。

このポリシードメインには、自分のパスワード変更を許可するユーザが登録されたユーザディレクトリを含める必要があります。

新しいポリシードメインを作成しない場合、既存のポリシードメインを選択します。

6. 保護するディレクトリを指定するレルムを作成します。[認証方式]リストボックスで、手順 4 で作成した認証方式を選択します。

7. 保護するリソースを指定するレルムの下に、ルールを作成します。

注: 保護するディレクトリ内のすべてのリソース(*)を指定するルールを作成するのであれば、ローカライズしたパスワード サービスに合わせて別にルールを作成する必要はありません。

8. 作成したルールとパスワードの変更を可能にするユーザ/グループを結び付けるポリシーを作成します。

パスワード サービス サブレットへのパスワードの変更リンクの追加

[パスワードの変更要求]フォームへのリンクは、次のとおりです。

```
<a href="/siteminderagent/pwservlet/PSWDChangeServlet?
SMAUTHREASON= 34&TARGET=<URL_of_Web_Page_After_Password_Change>">
Change Password</font></a>
```

ここで、*URL_of_Web_Page_After_Password_Change* はパスワードの変更後にユーザに表示される Web ページの URL です。たとえば、<http://server.myorg.org/AfterPasswordChange.html> などです。

このリンクを保護されていないページに追加すると、すべてのユーザが自分のパスワードを変更することができます。

特定ユーザにサブレットでのパスワードを変更を可能にする

特定のユーザに対してのみ、パスワードの変更を有効にするには、次の手順を実行します。

特定のユーザがパスワードを変更できるようにする方法

1. PWLogin.jsp ファイルに対する権限を変更します。

- a. 次のディレクトリに移動します。

```
web_agent_installation_dir/pwservlet/default
```

ここで、*web_agent_installation_dir* は Web エージェントがインストールされている場所です。

- b. [ファイル]、[プロパティ]の順に選択し、読み取り専用の属性をオフにします。

2. PWLogin.jsp ファイルのテキストを、次の手順で編集します。
 - a. テキストエディタを使用して PWLogin.jsp を開きます。
 - b. 以下を追加します。

```
<p align="center"><font size="1">  
<a href="/siteminderagent/pwservlet/  
PSWDChangeServlet?SMAUTHREASON=34&TARGET  
=%TARGET%>&SMAGENTNAME=<%=agentname%>">Change Password</font></a>
```
 - c. ファイルを保存します。
3. 管理 UI にアクセスします。
4. 認証方式を、次の設定で作成します。
 - サーバ名: パスワード サービス サブレットのあるサーバの名前です。
以下に例を示します。

myserver.mycompany.org
 - ターゲット: /siteminderagent/pwservlet/PSWDChangeServlet
5. 必要に応じて、新しいポリシードメインを作成します。

このポリシードメインには、自分のパスワード変更を許可するユーザが登録されたユーザディレクトリを含める必要があります。

新しいポリシードメインを作成しない場合、既存のポリシードメインを選択します。
6. 保護するディレクトリを指定するレルムを作成します。[認証方式]リストボックスで、手順 4 で作成した認証方式を選択します。
7. 保護するリソースを指定するレルムの下に、ルールを作成します。

保護するディレクトリ内のすべてのリソース (*) を指定するルールを作成する場合、ローカライズされたパスワード サービスに対して個別のルールを作成する必要はありません。
8. 作成したルールと、パスワードの変更が許可されたユーザ/グループをバインドするポリシーを作成します。

ユーザによるパスワードの変更

ユーザが自分のパスワードを変更するには、次の処理を実行する必要があります。

1. [パスワードの変更]をクリックします。
管理 UI に[パスワードの変更リクエスト]フォームが表示されます。
2. 必要な情報を入力し、[パスワードの変更]ボタンをクリックします。
管理 UI に、別の[パスワード変更情報]ページが表示されます。これは、ユーザのパスワードが変更されたことを意味します。

パスワード サービスでリダイレクトされる際のログイン ID の削除

パスワード サービスの処理中、ユーザのリクエストは何度かリダイレクトされます。要求がリダイレクトされると、リクエスト URL にユーザが入力したログイン ID (通常はユーザ名) がデフォルトで追加されます。このデフォルト動作を変更して、ログイン ID (ユーザ名) がリダイレクト URL に追加されないようにするには、次のいずれかの手順を実行します。

パスワード サービスでリダイレクトされる際にログイン ID を削除する方法 (Windows の場合)

1. 次のレジストリキーを追加します。
`HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Authentication\DisallowUsernameInURL`
2. DWORD の値は次のいずれかに設定します。
 - 0 - リクエスト URL に UID を追加するデフォルト動作を適用します。
 - 1 - デフォルト動作を変更して、リクエスト URL に UID が追加されないようにします。

パスワード サービスでリダイレクトされる際にログイン ID を削除する方法 (UNIX の場合)

1. 次のディレクトリに移動します。
`<policy-server-install-dir>/registry/`
2. テキストエディタで、次のファイルを開きます。
`sm.registry`

3. 次のレジストリキーを追加します。
HKEY_LOCAL_MACHINE¥SOFTWARE¥Netegrity¥SiteMinder¥CurrentVersion¥Authentication¥DisallowUsernameInURL
4. DWORD の値は次のいずれかに設定します。
 - 0 - リクエスト URL に UID を追加するデフォルト動作を適用します。
 - 1 - デフォルト動作を変更して、リクエスト URL に UID が追加されないようにします。

注: レジストリ設定を変更して、パスワード サービスのリダイレクトでログイン ID が追加されないようにする場合、パスワード サービスは自動的に PWnn というプレフィクスが付いたテンプレートを使用します。これらのテンプレートについては、「[CGI ベースのパスワード サービス HTML フォーム テンプレート](#) (P. 736)」で説明されています。

CGI ベースのパスワード サービスの HTML フォームのカスタマイズ

パスワード サービス CGI では HTML フォームが使用されます。このフォームを使用して、ユーザはパスワードを変更することができます。これらのフォームをカスタマイズして、商標を作成したり、フォームに表示される情報を変更したりすることができます。

CGI ベースのパスワードサービス HTML フォームテンプレート

パスワードサービス CGI では、HTML フォームを使用してユーザからパスワード情報を収集し、ポリシー サーバに送ります。ポリシー サーバは、送られた情報をパスワードポリシーと突き合わせてチェックし、別のフォームを使用してメッセージをパスワードサービスに送り返します。

SiteMinder Web エージェントのインストールには、カスタマイズできるフォームテンプレートが用意されています。フォームのサンプルは、次の場所にインストールされます。

- Windows の場合: <installdir>\SiteMinder Web Agent\pw
- UNIX: <installdir>/siteminder/webagent/pw

次のような テンプレートがあります。

テンプレート	説明
PWAccountInactive.template	アカウントがアクティブでないために無効になったことを、ユーザに通知します。
PWnnAccountInactive.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWAccountInactive.template と同じです。
PWChange.template	保護されたリソースにアクセスする前にパスワードを変更するように、ユーザに要求します。ユーザは、古いパスワード、新しいパスワードを入力し、新しいパスワードの確認入力をします。
PWnnChange.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWChange.template と同じです。
PWChangeAccept.template	パスワードが正常に変更されたことを、ユーザに通知します。
PWChangeErr.template	内部的な問題(サーバの過負荷など)が発生した場合、パスワードが変更されなかったことをユーザに通知します。サーバが過負荷状態の場合は、パスワードが変更できないことがあります。
PWnnChangeErr.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWChangeErr.template と同じです。
PWExcessLogin.template	ログイン試行回数の限度を超えたことを、ユーザに通知します。ユーザは、指定された時間待機してからログインし直すか、管理者に問い合わせるパスワードを再使用できるようにする必要があります。
PWnnExcessLogin.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWExcessLogin.template と同じです。
PWChangeMismatch.template	選択したパスワードが無効であることをユーザに通知して、新しいパスワードを選択するように要求します。

テンプレート	説明
PWnnChangeMismatch.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWChangeMismatch.template と同じです。
PWChangeOption.template	パスワードの有効期限が迫っている場合、ユーザに今パスワードを変更させることも、後で変更するように通知することもできます。ユーザは、古いパスワード、新しいパスワードを入力し、新しいパスワードの確認入力を行います。
PWnnChangeOption.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWChangeOption.template と同じです。
PWExpired.template	パスワードの有効期限切れをユーザに通知します。
PWnnExpired.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWExpired.template と同じです。
PWAccountDisabled.template	アカウントが無効であるため、管理者に問い合わせるようにユーザに通知します。
PWnnAccountDisabled.template	ログイン ID がパスワードサービスリダイレクト URL に追加されない点を除き、PWAccountDisabled.template と同じです。
PWSelfChangeLogin.template	ユーザがログインページからパスワードを変更できるようにします。

CGI ベースのパスワード サービス HTML テンプレートの変更

他の HTML ドキュメントをカスタマイズできるのと同様に、パスワードサービス HTML テンプレートもレイアウトや書式をカスタマイズできます。

ただし、パスワード サービス テンプレートをカスタマイズする際には、以下を変更しないように注意してください。

- テンプレートファイルの先頭にある JavaScript 関数
- `<form>` HTML タグ。パスワードサービス CGI では `<form>` 出力を使用して、情報を適切なスクリプトに渡して処理します。
- `$text$` 表記を使用するテキスト。パスワードサービス CGI では `$` 記号で示された変数に格納された情報を使用します。

これらの要素を変更すると、パスワードサービスが正常に動作しなくなる可能性があります。

CGI ベースのパスワード サービスのプロパティ ファイル

パスワード サービスのプロパティファイルによって、ユーザからの情報を収集したり、ステータスメッセージを表示したりする際に使用される HTML ページに表示されるテキストが決まります。該当するテキストは次のとおりです。

- フィールド名
- ダイアログのタイトル
- ステータス メッセージとユーザへの指示
- リンクとボタンのテキスト

HTML ページを生成するため、パスワードサービス CGI はパスワードサービス テンプレートを使用します。テンプレート内にパラメータがあると、CGI はプロパティファイルで定義されている必要な識別子を検索します。プロパティファイルでは、識別子と生成される HTML ページに追加される文字列がペアになっています。

以下に、パスワード サービスのプロパティファイルの内容例を示します。

`szPWNewPWChange` = 新規パスワード

等号 (=) の右側の文字列を変更して、生成される HTML ページに表示されるラベルを変更することができます。ただし、左側の識別子を変更しないでください。

注: パスワード サービスのプロパティファイルでは、すべての文字が **UTF-8** エンコード方式である必要があります。UTF-8 エンコード方式以外の文字 (Windows のウムラウトやチルダ、または 8 ビット ASCII 文字コードなど) は正しく処理されません。

パスワード サービスによって、次の言語のプロパティファイルがインストールされます。

- 英語 (US-EN): PasswordServicesUS-EN.properties
- フランス語 (FR-FR) : PasswordServicesFR-FR.properties
- 日本語 (JP-JP): PasswordServicesJP-JP.properties

これらのファイルは、以下のディレクトリにあります。

- Windows: `web_agent_installation\pw`
- UNIX: `web_agent_installation/pw`

ここで、`<web_agent_installation>` は、SiteMinder Web エージェントがインストールされている場所です。

CGI ベースのパスワード サービスのローカライズ

パスワードサービスのテンプレートファイルでは、複数の言語を使用できるように **Unicode** がサポートされています。デフォルトのエンコーディングは **UTF-8** であり、これによってユーザは **UTF-8** をサポートする言語でパスワードを入力できるようになります。

フランス語または日本語用にパスワード サービス テンプレートをローカライズするには、「[CGI ベースのパスワード サービスのプロパティファイル \(P. 739\)](#)」の説明に従って、フランス語および日本語のサンプル プロパティファイルを変更してください。

ローカライズされたパスワード サービスを使用するようにサイトを設定する方法については、「[サブレットベースのパスワード サービス JSP フォームの概要 \(P. 742\)](#)」を参照してください。

ローカライズしたパスワード サービス プロパティ ファイルの作成と使用

5.x QMR 2 以前の Web エージェントでは、パスワードサービス変更フォームに適切な言語のエンコード方式が設定されるように、SMLOCALE 変数の値が渡されていました。このためには、変数 SMLOCALE=<country>-<language> を以下のような保護されたリソースのターゲット URL に追加する必要があります。

`http://server.mysite.org/dir1/targetpage.html?SMLOCALE=FR-FR`

5.x QMR 2 以降では、言語のエンコード方式が Web ブラウザから読み込まれるようになったため、このような手動による作業は不要になりました。パスワードサービスの CGI は、Web ブラウザから ACCEPT_LANGUAGE 変数を読み込み、パスワード変更フォームをどの言語で表示するかを決定します。

SiteMinder Web エージェントに付属している変更フォームでは、言語ごとに異なるプロパティファイルを使用します。プロパティファイルは、HTML ページ内で表示されるテキストの特定の要素を定義します。

以下のプロパティファイルは SiteMinder Web エージェントのインストール時に自動的にインストールされます。

- 英語: PasswordServicesUS-EN.properties
- フランス語: PasswordServicesFR-FR.properties
- 日本語: PasswordServicesJP-JP.properties

ACCEPT_LANGUAGE にプロパティファイルのない言語が指定された場合には、英語のプロパティファイルが使用されます。

他の言語のプロパティファイルを作成および使用する方法

1. SiteMinder に付属しているプロパティファイルの 1 つを目的の言語に翻訳します。

たとえば、PasswordServicesUS-EN.properties ファイルをコピーし、等号(=)より右にあるすべての文字を翻訳します。

2. パスワードサービスの規則に従ってファイルに名前を付けます。命名規則は次のとおりです。

`PasswordServices<country>-<language>.properties`

注: RFC 3066 では、命名規則の一部として使用する言語タグが定義されています (<http://www.ietf.org/rfc/rfc3066.txt>)。

注: ブラウザが正しい言語に設定されていることを確認してください。

1 つのブラウザで複数の言語を設定した場合、パスワード サービスは、そのリストの中にある最初の言語だけを使用します。パスワードサービスの CGI が、その言語に対応するプロパティ ファイルを見つけることができない場合、英語のプロパティファイルが使用されます。ブラウザに対して指定された他の言語は無視されます。

サブレットベース パスワード サービスの JSP 形式のカスタマイズ

パスワード サービス サブレットでは *JavaServer Page (JSP)* フォームが使用されます。このフォームを使用して、ユーザはパスワードを変更することができます。これらのフォームをカスタマイズして、商標を作成したり、フォームに表示される情報を変更したりすることができます。

サブレットベースのパスワードサービス JSP フォームの概要

パスワード サービス サブレットでは、JSP フォームを使用してユーザからパスワード情報を収集し、ポリシー サーバに送ります。ポリシー サーバは、送られた情報をパスワード ポリシーと突き合わせてチェックし、別のフォームを使用してメッセージをパスワード サービス サブレットに送り返します。

SiteMinder Web エージェントのインストールには、カスタマイズできるフォームテンプレートが用意されています。JSP フォームのサンプルは、次の場所にインストールされます。

- Windows の場合: <Web-Agent-Install-Dir>\jpw
- UNIX の場合: <Web-Agent-Install-Dir>/jpw

ここで、<Web-Agent-Install-Dir> は、SiteMinder Web エージェントがインストールされているディレクトリです。

次のような テンプレートがあります。

テンプレート	説明
PWAccountInactive.jsp	アカウントがアクティブでないために無効になったことを、ユーザに通知します。

テンプレート	説明
PWChange.jsp	保護されたリソースにアクセスする前にパスワードを変更するように、ユーザに要求します。ユーザは、古いパスワード、新しいパスワードを入力し、新しいパスワードの確認入力を行います。
PWChangeAccept.jsp	パスワードが正常に変更されたことを、ユーザに通知します。
PWChangeErr.jsp	内部的な問題(サーバの過負荷など)が発生した場合、パスワードが変更されなかったことをユーザに通知します。サーバが過負荷状態の場合は、パスワードが変更できないことがあります。
PWExcessLogin.jsp	ログイン試行回数の限度を超えたことを、ユーザに通知します。ユーザは、指定された時間待機してからログインし直すか、管理者に問い合わせるパスワードを再使用できるようにする必要があります。
PWChangeMismatch.jsp	選択したパスワードが無効であることをユーザに通知して、新しいパスワードを選択するように要求します。
PWChangeOption.jsp	パスワードの有効期限が迫っている場合、ユーザに今パスワードを変更させることも、後で変更するように通知することもできます。ユーザは、古いパスワード、新しいパスワードを入力し、新しいパスワードの確認入力を行います。
PWExpired.jsp	パスワードの有効期限切れをユーザに通知します。
PWAccountDisabled.jsp	アカウントが無効であるため、管理者に問い合わせるようにユーザに通知します。
PWSelfChangeLogin.jsp	ユーザがログインページからパスワードを変更できるようにします。
PWLogin.jsp	ユーザが認証情報を入力してログインできるようにします。この .jsp ファイルは、HTML フォーム ベースの認証方式のリダイレクトパスがサーブレット(つまり、/siteminderagent/pwservlet/PSWDChangeServlet)を指す場合に、ユーザに表示されるログイン用のフォームを生成します。

テンプレート	説明
UserMsg.jsp	UserMsg.jsp は、拒否の理由に応じてメッセージを返します。
PWExpiredLogin.jsp	ユーザのセッションが失効した場合、この .jsp ファイルはユーザが新しいセッションを確立できるようにログイン画面を表示します。
PSWDChangeError.jsp	この .jsp ファイルは、内部的な問題が発生したためにパスワード変更要求の処理が失敗し、パスワードが変更されなかったことをユーザに通知します。
PWAuthLevelLogin.jsp	セッションが確立しているユーザが、ログイン時に要求したリソースよりも高い保護レベルが設定されているリソースにアクセスしようとする、この .jsp ファイルはそのユーザが再度認証を受けるためのフォームを表示します。ユーザはこのフォームで認証を受けた後、保護レベルが高いリソースにアクセスすることができます。
Resource.jsp	UTF-8 として内容の種別を指定し、すべてのパスワードサービスフォームのローカライズされた文字列を含むリソースバンドルを示します。

サブレットベースのパスワード サービス JSP フォームの変更

他の HTML ドキュメントをカスタマイズできるのと同様に、パスワードサービス JSP フォームもレイアウトや書式をカスタマイズできます。

以下を変更しないように注意してください。

- テンプレートファイルの先頭にある JavaScript 関数
- `<form>` HTML タグ

ポリシー サーバは、`<form>` 出力を使用して情報をパスワード サービス サブレットに渡し、処理します。

- `<%text%>` 表記で囲まれたパラメータ

これらの `<%text%>` パラメータは、パスワード サービス サーブレットで使われます。既存のパラメータは変更できませんが、必要に応じて `<%text%>` を追加したり、パスワードサービス固有の `<%text%>` パラメータを使用することができます。

JSP テンプレートファイルごとに使用される `<%text%>` パラメータは異なります。

サーブレットベースのパスワード サービスのローカライズ

サーブレットベースのパスワードサービスでは、1 セットのテンプレート .jsp ファイルを使用します。このファイルには、外部化されたプロパティファイルから文字列/メッセージが実行時に入力されます。 .jsp ファイルで使用される言語は、ユーザのブラウザのロケール設定で決まります。

ブラウザが複数のロケールを使用するように設定されている場合は、指定されたロケールが順に比較され、パスワード サービス フォームでサポートされる言語と一致した最初のロケールによって、ユーザに表示されるフォームに使用される言語が決まります。

注: ユーザのブラウザで言語設定が指定されていない場合、デフォルトのプロパティファイルである `PasswordServices.properties` が使用されます。

フォームに表示される文字列の変更

.jsp ファイルでは、文字列とメッセージが、次のようなフォーマットのタグによって表されます。

```
<%=getI18NParameter (resourcebundle, TitleName)%>
```

たとえば、タイトル「SiteMinder パスワード サービス」は、`PWChange.jsp` では次のタグで表されます。

```
<%=getI18Nparameter (resourcebundle, "PWChange.Title")%>
```

ここで、「`resourcebundle`」は、プロパティファイルから読み込まれた名前/値のペア文字列が含まれるオブジェクトで、「`PWChange.Title`」は、プロパティファイルから値が取得されるキーです。

.jsp ファイルのタグは、プロパティファイルに定義されている対応する識別子の値に置き換わります。

以下に、パスワードサービスのプロパティファイルの内容のフォーマット例を示します。

```
PWChange.Title = SiteMinder Password Services
```

等号の右側のテキスト文字列を修正して、ユーザに表示されるフォームに現れるラベルに変更できます。

新しい文字列の追加

フォームに新しい文字列またはメッセージを追加するには、.jsp ファイルに新しいタグを追加して、関連するプロパティファイルに対応するエントリを追加する必要があります。たとえば、**NewTag** という新しいタグを、**PWScreen.jsp** という .jsp ファイルに追加するには、以下を実行する必要があります。

1. 以下のように、プロパティファイルに新しいプロパティを追加します。

```
PWScreen.NewTag=NewValue
```

2. .jsp ファイルに新しいタグを追加します。

```
<%=getI18NParameter (resourcebundle, "PWScreen.NewTag")%>
```

.jsp ファイルのタグは実行時に **NewValue** という文字列に置き換わります。

プロパティファイル

SiteMinder には、デフォルトで次のプロパティファイルが含まれています。

PasswordServices.properties

デフォルトのプロパティファイルであるこのファイルには、英語 (**EN**) の鍵/値のペアが含まれています。

PasswordServicesJA.properties

このファイルには、日本語 (**JA**) のキー/値のペアが含まれています。

プロパティファイルの文字はすべて **UTF-8** 方式でエンコードされています。自分でプロパティファイルを作成したり、既存のファイルにプロパティファイルを追加したりする場合には、**UTF-8** 方式でエンコードする必要があります。

プロパティ名には、次のいずれかの規則に従って名前を付ける必要があります。

- PasswordServices<LANGUAGE>-<COUNTRY>.properties

または

- PasswordServices<LANGUAGE>.properties

ここで、「<LANGUAGE>」は、言語を表す 2 文字のコード(英語は EN、フランス語は FR、日本語は JA など)で、「<COUNTRY>」は、国を表す 2 文字のコード(米国は US、日本は JP など)です。たとえば、カナダのフランス語用のプロパティファイル名は次のようになります。`named PasswordServicesFR-CA.properties`

新しい言語のサポート

プロパティファイルを使用して、サポートする言語を追加できます。

新しい言語をサポートする方法

1. UTF-8 方式でエンコードされたプロパティファイルを作成します。
2. このファイルに、必要な形式の名前を付けます。
3. .jsp ファイルと共に、このファイルを <Web-Agent-Install-Dir>\jpw ディレクトリに配置します。

英語以外の言語のデフォルト言語としての設定

ユーザのブラウザで言語設定が指定されていない場合は、デフォルトのプロパティファイルである `PasswordServices.properties` が使用されます。

英語以外の言語をデフォルト言語として設定する方法

1. 次のファイルの名前を別の名前に変更します。
`web_agent_install_dir¥jpw¥PasswordServices.properties`
2. 新しいデフォルトのプロパティファイルを作成して、次のような名前を付けます。
`PasswordServices.properties`
3. 新しい `PasswordServices.properties` ファイルを次のディレクトリに保存します。
`<Web-Agent-Install-Dir>\jpw\`

追加属性情報を必要とする認証方式

追加属性を必要とする認証方式で動作するようにパスワード サービスを設定するには、ユーザがポリシー サーバにデータをサブミットする前に、以下をパスワード サービスの HTML フォーム `.template` ファイルとサンプルの `.jsp` フォーム ファイルに追加する必要があります。

```
document.PWChange.OLDPASSWORD.value="PASSWORD="+escape( document.PWChange.OLDPASSWORD.value)
+"&<attribute>="+escape(document.PWChange.<attribute>.value)
```

パスワード サービスの `.template` ファイルおよびサンプルの `.jsp` ファイルは以下の場所にインストールされます。

- `web_agent_installation/pw`
- `web_agent_installation/jpw/default`
- `web_agent_installation/pwservlet/default`

たとえば、以下にサンプルの `PWChange.jsp` ファイルを変更する方法を示します (太字でハイライトされているテキストが追加する新しい行です)。

```
<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
}
}
```


以下にサンプルの **PWChange.template** ファイルを変更する別の例を示します (太字でハイライトされているテキストが追加する新しい行です)。

```
<!-- 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
}
}
```

パスワード サービス サブレットには、複数の言語をサポートする **Unicode** が用意されています。デフォルトのエンコーディングは **UTF-8** であり、これによってユーザは **UTF-8** をサポートする言語でパスワードを入力できるようになります。

パスワード サービス サブレットは、ブラウザが設定する **HTTP** プロトコル標準の **ACCEPT-LANGUAGE** リクエスト ヘッダを基に、ユーザの言語設定を判断します。サブレットが **ACCEPT-LANGUAGE** の値 (**EN-US**、**FR** などの標準言語エンコーディングを示す値) を取得できる場合は、以下の対応する名前のディレクトリでローカライズされた適切な **JSP** を探します。

- NT: **web_agent_installation_dir\jpw\<encoding>**
- UNIX: **web_agent_installation_dir/jpw/<encoding>**

ここで、**web_agent_installation_dir** はルート インストール ディレクトリ、**<encoding>** は標準の言語エンコーディングです。

パスワード サービスはインストール時、サンプルの米国英語 **JSP** ファイルのセットも **jpw** ディレクトリにインストールします。パスワード サービスは、**ACCEPT-LANGUAGE** 値に対応する **JSP** ディレクトリを見つけられない場合、または **ACCEPT-LANGUAGE** 値を取得できない場合に、このフォルダの内容を使用します。

別の言語に合わせてサブレットベースのパスワード サービスをローカライズするには、翻訳された **JSP** 形式の完全なセットを含み、適切な名前を付けたディレクトリを新しく作成する必要があります。

ローカライズした JSP フォームが格納されているディレクトリ名の例

言語	フォルダの場所
英語 (米国)	EN
英語 (英国)	EN-GB
フランス語	FR
ドイツ語	DE
日本語	JP

別の言語をデフォルトとしてローカライズする場合は、インストールされているデフォルト ディレクトリの名前を **EN** に変更し、新しいデフォルト言語に翻訳した JSP フォームの完全なセットを含むデフォルト ディレクトリを新しく作成します。

パスワード ポリシーのトラブルシューティング

以下のセクションでは、パスワード ポリシーの実装時に発生する可能性のある問題の解決策について説明します。

新しいユーザ パスワードが拒否される

症状:

ユーザ指定のパスワードが常に拒否されます。

解決方法:

パスワード ポリシーの制限が厳しすぎたり、不適切な設定を行っている可能性があります。最小限必要な内容と、パスワードの長さの設定を確認します。

ユーザ アカウントが誤って無効にされる

症状:

ログインの試行回数が、許可されたログイン失敗回数を超えていないにもかかわらず、ユーザ アカウントが無効になっています。

解決方法:

パスワード設定が正しいかどうかを確認します。誤ったパスワードが特定の回数、続けて入力された後にアカウントを無効にするための設定値が小さすぎる可能性があります。

設定値が小さすぎると、異なるユーザ ディレクトリに存在する 2 名以上のユーザが同じユーザ名である場合に、問題が発生します。ポリシー サーバがユーザに対して許可を与えるときは、まず、すべてのユーザ名から、ログイン ユーザ名と対応するものを見つけ、次にそのパスワードが一致するかを確認します。パスワードが一致しないユーザ名を検出すると、ポリシー サーバはエラーを記録します。このエラーの発生回数が、不正なパスワードの設定の[パスワードが無効になるまでの入力失敗回数]フィールドに指定された回数を超えると、そのアカウントは無効になります。

ユーザ アカウントが途中で無効になる

症状:

マルチ ポリシー サーバ環境でユーザ アカウントが、設定した時間より早く無効になります。

解決方法

ポリシー サーバ間の時刻設定に差がないかどうか確認してください。

パスワードの変更が強制される

症状:

マルチ ポリシー サーバ環境でユーザ アカウントが、設定した時間より早く、パスワードを変更するよう求められます。

解決方法:

ポリシー サーバ間の時刻設定に差がないかどうか確認してください。

LDAP ユーザが無効にならない

症状:

パスワード ポリシーで LDAP ユーザを無効にすることができません。

解決方法:

以下の点をチェックしてください。

- ポリシーを結び付けようとしている LDAP ディレクトリが書き込み可能である。LDAP ディレクトリの各ユーザのレコードにパスワードが保存可能である必要があります。
- ユーザプロファイル属性の[パスワード属性]、[パスワードデータ]、および[無効フラグ]に対して、ユーザディレクトリ設定が有効になっている。パスワード属性、パスワード データおよび無効なフラグ ユーザ プロファイル属性の設定については、「[ディレクトリ属性の指定 \(P. 183\)](#)」で説明されています。

Active Directory ユーザがパスワードを変更できない

症状:

Active Directory ユーザ ディレクトリ内に格納されたユーザが、パスワードを変更できません。

解決方法:

以下の点をチェックしてください。

- ポリシーがバインドされている Active Directory ユーザ ディレクトリが、安全な(SSL)接続を使用するように設定されている。
- ポリシーがバインドされている Active Directory ユーザ ディレクトリが、unicodePWD のパスワード属性を使用するよう設定されている。

パスワードが誤っていることを示すメッセージが表示されない

症状:

ユーザが、無効な現在のパスワードを含むパスワード変更リクエストをサブミットしたときに、現在のパスワードが誤っていることを示すメッセージを含む[パスワード変更情報]画面が表示されません。さらに、ポリシー サーバによってユーザが以下へリダイレクトされます。

- メッセージを表示しないログイン画面 (ユーザ ディレクトリで設定されたポリシーに **On-Auth-Reject-Redirect** レスポンスがバインドされていない場合)
- ユーザ ディレクトリで設定されたポリシーにバインドされた **On-Auth-Reject-Redirect** レスポンスに関連付けられた URL

解決方法:

HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\PolicyServer にある DisallowForceLogin レジストリ キーを有効にします。

DisallowForceLogin

変更リクエストに入力された現在のパスワードが無効な場合は、現在のパスワードを再入力するための[パスワード変更情報]画面にユーザをリダイレクトします。

KeyType: REG_DWORD

値: 0(無効)または 1(有効)

デフォルト: 0(無効)

注: 0 および 1 以外の値を指定した場合は、キーが無効になり、ポリシー サーバによって、現在のパスワードが無効であることを示すメッセージが表示されずに、ユーザがログイン画面にリダイレクトされます。

第 24 章: SiteMinder テストツール

このセクションには、以下のトピックが含まれています。

[テスト ツールの概要](#) (P. 755)

[テスト環境エージェントの設定](#) (P. 756)

[機能テストの実行](#) (P. 762)

[リグレーション テストの実行](#) (P. 767)

[ストレス テスト](#) (P. 768)

[証明書ベースの認証テスト](#) (P. 771)

テスト ツールの概要

SiteMinder テストツールは、エージェントとポリシー サーバの間の対話をシミュレートするユーティリティです。このツールは、ポリシー サーバの機能をテストします。テストツールは、エージェントとしての役割を果たし、実際のエージェントと同じようにポリシー サーバに対して要求を送信します。これにより、SiteMinder の設定を実際に展開する前にテストできます。

SiteMinder テストツールは、Windows システムでのみ使用可能です。このツールを使用すると、Windows ベースの Web エージェントおよび RADIUS エージェントを無制限にエミュレートできます。SiteMinder テストツールは、すべてのプラットフォーム上の任意のポリシー サーバとの接続を作成できますが、Solaris 上にある 4.x より後の Web エージェントは、SiteMinder テストツールでテストできません。SiteMinder 5.x 以降の Web エージェント用のポリシーをテストするには、以下のいずれかを実行します。

- Web エージェントの登録およびテストには、Perl スクリプト インターフェースを使用してください。
- SiteMinder Web エージェントをインストールして設定し、それを手動でテストします。

SiteMinder テストツールは、次の 3 種類のテストを実行します。

機能

ポリシーをテストして、正しく設定されていることを確認します。

リグレーション

ポリシー ストアの移行や新機能の実装などの変更が、SiteMinder に影響を及ぼすかどうかをテストします。

ストレス

ポリシー サーバが多数のリクエストを受信した場合のパフォーマンスをテストします。

注: スクリプト インターフェースを使用して、ポリシーをテストすることもできます。「Programming Guide for Perl」を参照してください。

テスト環境エージェントの設定

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント]グループ ボックスで設定できます。

テスト ツールがシミュレートするエージェントは、管理 UI で設定する必要があります。

エージェント情報を設定するには、次のオプションを指定してください。

エージェントのタイプ

以下のいずれかのエージェントタイプを指定します。

バージョン 4

SiteMinder 4.x エージェントをシミュレートします。

バージョン 5

SiteMinder 5.x エージェントをシミュレートします。

注: テストツールで SiteMinder 5.x Web エージェントのシミュレーションを行うには、テストツールを実行するシステムで **smreghost.exe** アプリケーションを実行する必要があります。**smreghost.exe** ファイルは Web エージェントに含まれています。詳細については、「Web エージェント インストール ガイド」を参照してください。このファイルは、<Policy Server install dir>/siteminder/bin ディレクトリにも保存されています。

RADIUS

RADIUS デバイスをシミュレートします。

エージェント名

エージェントの名前を入力します。この名前が 管理 UI に表示されます。このフィールドは、バージョン 4 エージェントとバージョン 5 エージェントの必須フィールドです。

秘密キー

エージェントの共有秘密キーを入力します。エージェントの作成時に入力した共有秘密キーと同じキーを入力してください。[秘密キー]フィールドは、バージョン 4 エージェントと RADIUS エージェントの必須フィールドです。

(オプション) サーバ

エージェントがインストールされているサーバの完全な名前を入力します。たとえば、<http://www.myorg.org> のポリシー サーバをテストする場合には、このフィールドに「www.myorg.org」と入力します。このフィールドは、バージョン 4 エージェントのシミュレーションを行う場合に使用します。

SmHost.conf へのパス

シミュレートするバージョン 5 エージェントの設定を含む SmHost.conf ファイルへのパスを入力します。[参照]ボタンで SmHost.conf ファイルを参照できます。

ポリシー サーバの識別

テストツールには、[STMNDR エージェント]グループ ボックスで指定されたエージェントとの対話をシミュレートする際に使用するポリシー サーバについての情報が必要です。必要な情報は、選択したエージェントのタイプによって若干異なります。

バージョン 4 エージェントと RADIUS エージェント シミュレーションで使用するポリシー サーバのセットアップ

バージョン 4 エージェントおよび RADIUS エージェントのシミュレーションには、テストで使用するポリシー サーバの IP アドレスとポートを指定する必要があります。複数のポリシー サーバ環境をシミュレートする場合には、ポリシー サーバの役割 (プライマリまたはセカンダリ) を指定できます。

バージョン 4 エージェントと RADIUS エージェントのシミュレーションで使用するポリシー サーバをセットアップする方法

1. 必要に応じて、次のポリシー サーバ オプションを指定します。

ポリシー サーバ

プライマリ ポリシー サーバまたはセカンダリ ポリシー サーバのどちらかを指定するかを示します。

IP アドレス

ポリシー サーバの IP アドレスを指定します。デフォルトでは、このフィールドにはローカルシステムの IP アドレスが入力されます。

許可ポート、認証ポート、アカウントिंग ポート

許可、認証、アカウントिंगの要求の受信に使用する TCP ポートを指定します。これらのフィールドには、ポリシー サーバのデフォルトのポート番号が入力されます。

タイムアウト

テストツールがポリシー サーバからの応答を待つ時間を秒単位で表示します。

2. 次のいずれかの動作モードを選択してください。

フェイルオーバー

フェイルオーバーを有効にします。フェイルオーバーの実行時に、テストツールは最初のポリシー サーバに要求を転送します。最初のポリシー サーバに障害が発生すると、テストツールは要求を次のポリシー サーバにリダイレクトします。

ラウンドロビン

ラウンドロビンモードのロードバランシングを有効にします。ラウンドロビンモードのロードバランシングでは、プライマリポリシー サーバとセカンダリポリシー サーバの間で要求処理が分散されます。接続ごとに、テストツールは 2 つのポリシー サーバに交互に要求を振り分けます。

3. [接続]をクリックして、テストツールがポリシー サーバに接続できることを確認してください。

テストツールがポリシー サーバに接続すると、IsProtected と DoManagement の各ストップライトが緑色に変わります。

注: ポリシー サーバ接続をテストする前に、エージェントを指定する必要があります。

バージョン 5 エージェントの場合に必要なポリシー サーバ情報

バージョン 5 エージェントのシミュレーションには、テストで使用するポリシー サーバの IP アドレスとポートを指定するか、ポリシーストアに格納されたホスト設定オブジェクトのポリシー サーバ接続情報を使用します。

デフォルトでは、テストツールが **SmHost.conf** ファイルを使用してポリシー サーバへの最初の接続を確立するときにポリシーストアからポリシー サーバ情報が取得されます。ポリシー サーバ情報を手動で指定するには、[オーバーライド] チェックボックスをオンにして、「[バージョン 4 エージェントおよび RADIUS エージェントのシミュレーション用ポリシー サーバのセットアップ](#) (P. 757)」にある説明に従って、ポリシー サーバ情報を入力します。

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント]グループ ボックスで設定できます。

テストツールを選択します。

次の表に示すテスト モードのいずれかを使用して、テストの実行方法と結果の表示方法を特定できます。選択したテストモードによっては、スクリプト情報を指定することもできます。

モード	説明
対話式	データを入力し、テストを実行して、その結果を直ちに[Server Response]セクションに表示できます。
レコード	[Interactive]動作と、テスト結果をプレーンテキスト スクリプトファイルに書き出すスクリプト生成機能を併用できます。このファイルは後で入力ファイルとして使用して、再生モードでテストを再実行できます。 複数のテストを実行して、結果を同じスクリプトファイルに記録できます。テストツールは、テスト結果をファイルの最後に追加します。このスクリプトファイルを再帰テストやストレステストに利用できます。 必要に応じて、出力スクリプトファイルにコメントを追加できます。これには、テストを実行する前にコメントを[Comment]フィールドに入力します。 記録を停止するには、新しいモードを指定します。 注: [レコード]を選択した場合は、テスト結果を保存するテキスト ファイルのパスとファイル名を[出力スクリプト]フィールドに入力する必要があります。

モード	説明
Basic Playback	[Record] モードで作成されたスクリプトファイルを使用して、再帰テスト用の連続テストを自動化します。[Basic Playback] を選択した場合、[Input Script] フィールドと [Output Script] フィールドのみが選択可能になります。
Advanced Playback	マルチスレッド化されたストレステストを実行します。[Advanced Playback] を選択した場合、[Control Script] フィールドのみが選択可能になります。

リソース情報の指定

テストを実行する対象リソースを指定できます。リソースを指定すると、ブラウザで URL を入力するユーザをシミュレートできます。

リソース情報を指定するには、次のオプションに値を指定します。

リソース

レルムに設定された、**SiteMinder** で保護されているリソースの相対パスを入力します。このパスは、**Web** サーバのパブリッシング ディレクトリに対する相対パスです。たとえば、「/protected/」と入力します。

アクション

テスト対象のルールで指定されたエージェント アクション、認証イベント、または許可イベントを入力します。

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント] グループ ボックスで設定できます。

ユーザ クレデンシャルの指定

テストツールでは、ポリシーでユーザを認証または許可できるかどうかをテストするためにユーザの認証情報が必要になります。

ユーザ クレデンシャルを指定するには、次のフィールドに入力します。

ユーザ名

リソースへのアクセスに使用するユーザ名を入力します。

パスワード

[ユーザ名] に入力したユーザのパスワードを入力します。

CHAP パスワード

RADIUS CHAP 認証方式を使用している場合は、このチェック ボックスをオンにします。

証明書ファイル

保護されているリソースで、ユーザの認証に証明書が必要な場合は、テストツールが証明書認証をシミュレートできるように、証明書ファイルを指定する必要があります。

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント]グループ ボックスで設定できます。

エンコーディング仕様の設定

[エンコーディング仕様]フィールドを使用すると、言語エンコーディング パラメータを指定できます。テストツールはこのパラメータを使用して、Web エージェントと同じ方法でヘッダーをエンコーディングします。エンコーディングされたレスポンス属性データは [属性]フィールドに表示されます。

言語エンコーディングの詳細については、「Web エージェント設定ガイド」を参照してください。

エンコーディング仕様を設定するには、エンコーディング仕様の値を、以下のように入力します。

encoding_spec, wrapping_spec

各項目の説明:

- encoding_spec は、UTF-8、Shift-JIS、EUC-J、または ISO-2022 JP のいずれかのエンコーディングタイプを表すテキスト文字列です。
- wrapping_spec は、ラッピング仕様ですが、RFC-2047 にする必要があります。

注: このフィールドを空欄のままにすると、デフォルトで、ラッピングを使用しない UTF-8 エンコーディングが使用されます。

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント]グループ ボックスで設定できます。

設定の保存とロード

設定を保存しておいてロードすれば、エージェントやリソース、ユーザ情報などのユーザ入力情報を再度入力する手間を省くことができます。

[設定を保存]ボタンで、現在の設定をプレーンテキストの .ini ファイルに保存することができます。[設定を保存]をクリックすると、ファイル名の入力が求められます。

保存した設定を取得するには、[設定をロード]をクリックし、ファイルを選択して、[開く]をクリックします。ファイルに定義された値がユーザ設定フィールドに自動的に入力されます。

テスト時にテストツールがシミュレートするエージェントは、[SiteMinder エージェント]グループ ボックスで設定できます。

機能テストの実行

SiteMinder テストツールを使用すると、シミュレートされた実際の環境で、ポリシーの機能をテストできます。機能テストを実行するには、次の条件を満たしている必要があります。

- ポリシー サーバが設定され、実行されている
- 管理 UI で SiteMinder エージェントが設定されている

注: テスト ツールで SiteMinder エージェント v5.x をシミュレートする場合は、そのエージェントで、4.x のサポートが有効になっている必要があります。

- ポリシードメインに任意のユーザ ディレクトリのタイプが設定されていること
- ルールとユーザが関連付けられたポリシーが設定されていること

SiteMinder では、次のような機能テストを実行できます。

IsProtected

指定したリソースがポリシーで保護されているかどうかを示します。

IsAuthenticated

ポリシー サーバが、一連のユーザ クレデンシャルを、ユーザ ディレクトリと照合して認証できるかどうかを示します。

ユーザ認証情報を認証する際、ポリシー サーバは認証情報とユーザディレクトリのエントリを照合します。認証情報がエントリに一致すると、ポリシー サーバはセッションチケットを作成して、ユーザを認証します。

「実際の」 SiteMinder 稼動環境では、ユーザが追加のリクエストを送信すると、SiteMinder は、ユーザ クレデンシャルをディレクトリに照合して再確認するのではなく、セッション チケットが有効かどうかを確認します。デフォルトでは、ユーザがセッションチケットを持っているかどうかに関係なく、テストツールは IsAuthenticated テストが実行されるたびにユーザの認証を行います。

ユーザのセッション チケットの有効性を確認するようにテストツールを設定するには、IsAuthenticated テストを実行する前に、テストツールのコメントフィールドに「検証 (Validate)」と入力します。ただし、セッション チケットの有効性を確認する前に、SiteMinder がユーザの認証を行う必要があります。

注:「検証」は、対話モードで複数のテストを実行する場合 ([Repeat count] フィールドを使用)、および [Playback] モードで指定できます。

IsAuthorized

ポリシー サーバがポリシーに基づいてユーザを認証できるかどうかを示します。

これらのテストは、上記の順序で実行する必要があります。たとえば、IsAuthenticated テストを実行するには、その前に IsProtected テストを実行する必要があります。この順序は、SiteMinder がユーザのアクセス権限に判断する際の手順を反映しています。

機能テストを実行するときには、テストツールを使って次のようなタスクを実行できます。

DoAccounting

最も新しいサーバトランザクションを記録します。

DoManagement

エージェント キャッシュをクリアするキャッシュ クリア コマンドなどのエージェント コマンドを要求します。テストツールがポリシー サーバから最新の情報を受け取るようにするには、**DoManagement** を実行します。

機能テストを実行する方法

1. テスト環境の設定

注: スクリプト インターフェースを使用してポリシーをテストすることもできます。「**Programming Guide for Perl**」を参照してください。

2. (オプション) [Command] グループ ボックスの [Repeat count] フィールドにテストの実行回数を指定します。

3. [Command] グループボックスで、次のいずれからのテストを選択して実行します。

- IsProtected
- IsAuthenticated
- IsAuthorized

4. IsAuthenticated テストを実行するときに、ユーザ クレデンシャルをユーザ ディレクトリと照合して認証するのではなく、ユーザのセッション チケットの有効性を確認するようにテスト ツールを設定する場合は、[コメント] フィールドに「検証」と入力します。

注: ユーザのセッション チケットの有効性を確認するには、その前にユーザが認証されている必要があります。ユーザが認証されると、そのユーザのセッション チケットが **SiteMinder** によって作成されます。

機能テストの結果

以下の表は、各機能テストの結果を示します。

isProtected の状況	結果
成功	テストツールの[Message]フィールドに Protected と表示されます。これは、テストツールがポリシー サーバに正常に接続し、リソースがポリシーによって保護されていることを示します。 また、ポリシー サーバから返された値が次のフィールドに入力されます。 レルム名 リソースを含んでいるレルムの名前 レルム OID レルム オブジェクト識別子 クレデンシャル リソースを保護するために使用される認証方式 リダイレクト 認証方式で使用するリダイレクト文字列(指定されている場合)。証明書および HTML フォームベースの認証方式ではすべてこの文字列が返されます。通常、この文字列は、エージェントにフォームを表示する場所を指定します。
失敗	テストツールの[Message]フィールドに Error または Not Protected と表示されます。「Error」と表示された場合は、テストツールがポリシー サーバに接続できなかったことを意味します。「Not Protected」と表示された場合は、指定したリソースがポリシーによって保護されていないことを示します。 テストが失敗した場合には、次のことを実行してください。 ポリシーが正しく設定されていることを確認します。 認証サーバログでデバッグ情報を確認します。

isAuthenticated の状況	結果
成功	[Message]フィールドに「Authenticated」と表示され、ポリシー サーバから返された値が次のフィールドに入力されます。 セッション ID SiteMinder によって割り当てられた一意のセッション ID。ポリシー サーバは、この ID を使用して、セッション情報を格納する Cookie を識別します。 属性 ポリシー サーバがレスポンスで返す属性。以下に例を示します。 <pre> ID 長さ ASCII 形式 16 4 > id 215 len 005 'LDAP:' - '4 </pre> レスポンスには、ユーザが認証されたユーザディレクトリの名前が表示されます。 注: ユーザが入力した情報を削除することなく、[属性]フィールドに表示されたレスポンスを消去するには、[リセット]ボタンをクリックします。 Reason テストの結果に関連付けられた理由コード。このフィールドは、SiteMinder SDK を使用する開発者に情報を提供するために使用されます。理由コードは SmApi.h に一覧されます。
失敗	テストツールの [Message]フィールドに「Not Authenticated」と表示されます。 テストが失敗した場合には、次のことを実行してください。 ユーザ認証情報が有効であることを確認します。 認証サーバログでデバッグ情報を確認します。

IsAuthorized の状況	結果
成功	[Message]フィールドに「Authorized」と表示され、SiteMinder によって割り当てられたセッション ID が [Session ID]フィールドに表示されます。この ID は、セッション情報が格納されている cookie を識別します。
失敗	テストツールの [Message]フィールドに「Not Authorized」と表示されます。 テストが失敗した場合には、次のことを実行してください。 ポリシーが正しく設定されていることを確認します。 許可サーバログでデバッグ情報を確認します。

平均経過時間の計算

テストを実行すると、[Command]グループボックスの[Elapsed Time]フィールドにテストの所要時間が表示されます。システムのパフォーマンスにはばらつきがあるため、複数回実行したテストの経過時間の平均値を算出すると、より正確な結果を得ることができます。

平均経過時間を計算する方法

1. [Repeat Count]フィールドで、テストの実行回数を指定します。
テストが指定された回数だけ実行され、経過時間の合計が表示されます。
2. 表示された経過時間をテストの実行回数で割ると、平均経過時間を割り出すことができます。

リグレッション テストの実行

リグレッション テストでは、ポリシー ストアのアップグレードや新機能の実装などの、SiteMinder に対して行われた変更がポリシーに影響するかどうかをテストできます。リグレッション テストを実行するには、現在の環境で 1 度テストを実行し、目的の変更を行ってから、再度テストを実行します。2 つのテストの結果を比較すると、SiteMinder が変更の影響を受けるかどうかを判断できます。

リグレッション テストを実行する方法

1. 機能テストは[Record]モードで実行します。必要に応じて、[Comment]フィールドにコメントを入力します。入力したコメントは出力ファイルに表示されます。
2. テストが完了したら、[Mode]グループボックスの[Basic Playback]を選択します。
3. [Script Information]グループボックスで、[Input Script]フィールドにテキストファイルの名前を入力します。
このファイル名は、[Record]モードで作成した[Output Script]ファイルと同じ名前であればなりません。
4. [Output Script]フィールドに、出力ファイルの名前を指定します。
5. [Command]グループボックスで [Run Script]をクリックします。
入力スクリプトが実行され、出力スクリプトファイルが作成されます。
6. 入力スクリプトファイルと出力スクリプトファイルを比較します。

ストレス テスト

テスト ツールを使用して、ポリシー サーバが同時に複数のリクエストを受け取ったときの SiteMinder のパフォーマンスをテストできます。ストレス テストを使用して、ポリシー サーバと複数のエージェントの対話や、1 つのエージェントとポリシー サーバとの複数スレッドでの対話をシミュレートできます。

ストレス テストは、[Advanced Playback]モードで実行します。テスト ツールは、どのテストを何回実行するかを指示をスレッド制御ファイルから受け取ります。スレッド制御ファイルに指定された指示が実行されると、テスト結果が出力ファイルに書き込まれます。

スレッド制御ファイル

[Advanced Playback]モードでテスト ツールが実行するテストの回数とスレッドの数は、スレッド制御ファイルによって決まります。このファイルには、テスト ツール自体のスクリプト言語で書かれた複数の命令行が含まれています。また、# 記号で始まるコメントも含まれています。

基本命令は、「<スクリプトファイル名>,<スクリプト実行回数>,<スレッドの数>」というフォーマットで表記されます。

以下に例を示します。

```
# c:\temp\test_data.txt, 8, 6
```

この命令行は次のような内容を指示しています。

- 入力スクリプトは c:\temp\test_data.txt
- テストツールはこのスクリプトを 8 回実行します。
- このスクリプトは 6 つのスレッドによって同時実行されます。

テストツールは、テスト結果をファイルに書き込みます。出力ファイル名は、入力ファイル名に _out# が追加された名前になります。# はスレッド数を示し、1 ずつ増えていきます。たとえば、上記のテストの出力ファイルは、c:\temp\test_data.txt_out1 から c:\temp\test_data.txt_out6. になります。

テストツールのスクリプト言語には、以下の表に示す、スクリプトファイルの出力を制御するためのコマンドが含まれています。以下のサンプル スレッド制御ファイルの図を参照してください。

コマンド	説明
<code>.report</code>	テスト結果をまとめた最終レポートを出力ファイルとして生成します。このレポートには、各サーバ要求のステータスは含まれません。これがデフォルトです。
<code>.output</code>	各サーバ要求のステータスを含めた総合的なテスト結果を最終レポートとして出力ファイルに生成します。
<code>.viewstats</code>	最終的な統計情報をテキストエディタに表示します。
<code>.verbose</code>	各サーバ要求の詳細を含む出力ファイルを生成します。
<code>.brief</code>	各サーバ要求の簡単な結果を含む出力ファイルを生成します。このオプションは、 <code>.output</code> コマンドを使用した場合にのみ有効になります。
<code>.sleep</code>	指定した時間 (ミリ秒単位) テストツールを一時停止できます。これにより、断続的なサーバ要求のシミュレーションを実行できます。
<code>.connect <setting_file></code>	設定ファイルの情報を使用してテストツールを初期化し、1 つのエージェントとのマルチスレッドテストをセットアップします。デフォルトのマルチスレッドテストは、スレッドごとに複数のエージェントと 1 つのエージェントの動作をシミュレーションするように構成されます。このオプションを使用すれば、1 つのエージェントと複数のスレッドのシミュレーションテストを実行するようにセットアップすることができます。
<code>.disconnect</code>	テストツールの初期化を解除して、1 つのエージェントとのマルチスレッドテストの終了を指示します。

```
.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
```

レポートの表示

ストレステストを実行すると、結果をまとめたレポートが生成されます。このレポートには、以下の情報が含まれています。

- テストの開始時刻と終了時刻
- 合計経過時間
- 要求処理にかかった最短時間、最長時間および平均時間
- 要求総数
- スループット
- テストの実行回数とその結果

このレポートは、スレッド制御ファイルがあるディレクトリに保存されます。レポート名は、スレッド制御ファイル名に `_stats` が追加された名前になります。たとえば、スレッド制御ファイルの名前が `thread.txt` の場合、レポート名は `thread.txt_stats` になります。

```
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

証明書ベースの認証テスト

テストツールを使用して、ユーザ認証とユーザ許可をシミュレートすることができます。証明書ベースの認証の場合には、追加の設定が必要になります。

発行者 DN および証明書のその他の属性のフォーマットは、Web サーバのベンダーによって異なる場合があります。たとえば、同一の証明書でも IIS Web サーバ上にあるものと iPlanet Web サーバ上にあるものでは、発行者 DN は異なります。証明書ベースの認証方式をテストするには、使用している Web サーバに合わせて適切に証明書マッピングを設定する必要があります。

カスタム マッピングを必要とする証明書属性

一般的な証明書属性の中には、Web サーバのベンダーの間で表現が異なるものがあります。テストツールでエラーになりやすい属性は次のとおりです。

電子メール アドレス

ベンダーによって E または Email と表記されます。

米国の州

ベンダーによって S または ST と表記されます。

ユーザ ID 番号

ベンダーによって UID または UserID と表記されます。

テスト用のカスタム属性マッピング

テストツールは、Web ブラウザおよび Web サーバ経由で正常に動作していても、証明書認証方式に使用した場合は、正しく動作しません。認証ログを確認すると、テストツールで期待される発行者 DN の属性と実際に DN に表示された属性が異なることがわかります。たとえば、テストツールでは州名に S が期待されるのに対して、実際の IssuerDN には ST が使用されている場合があります。この状況は、「[カスタム マッピングが必要な証明書属性 \(P. 771\)](#)」にあるリストの他の属性に似ています。

この問題は、発行者 DN やその他の属性のフォーマットが、Web サーバのベンダーが使用している（およびテスト ツールで想定されている）フォーマットと異なる場合に発生します。たとえば、同一の証明書でも IIS Web サーバ上にあるものと Netscape Web サーバ上にあるものでは、発行者 DN は異なります。

この問題を解決するには、ポリシー サーバが異なる Web サーバの IssuerDn を受け入れられるように、管理者が発行者 DN のマッピングを作成する必要があります。

発行者 DN マッピング

発行者 DN は Web サーバのベンダーによって表記方法が異なる場合があります。たとえば、ある Web サーバでは発行者 DN は次のように表記されます。

CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, S=Western Cape, C=ZA

一方、発行者 DN を次のように表記する Web サーバもあります。

CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, ST=Western Cape, C=ZA

2 つの表記方法をサポートするには、管理者が 2 つの発行者 DN のマッピングを作成する必要があります。

注: さまざまな Web サーバおよび SiteMinder テスト ツールがサポートされるように、管理者がマッピングを作成することをお勧めします。

カスタム証明書マッピングの作成

ポリシー サーバの証明書マッピング機能を使用して、証明書のカスタム マッピングを作成できます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

証明書マッピング内にカスタム属性を作成して使用する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [証明書マッピング]-[証明書マッピングの作成]をクリックします。
[証明書マッピングの作成]ペインが開きます。

3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

証明書マッピングの設定が開きます。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [発行者 DN]フィールドに、完全な発行者 DN を入力します。
5. [マッピング]グループ ボックスで[カスタム]ラジオ ボタンを選択します。
[マッピング式]フィールドが開きます。
6. カスタム マッピング式を入力します。

この表記法は、証明書マッピングに指定できる 2 種類の異なる属性を指定するときにも使用できます。

- 電子メール:

%{E/Email}

- ST:

%{S/ST}

- ユーザ ID:

%{UID/UserID}

注: カスタム マッピング式の詳細な説明は、「[カスタム マッピングが必要な証明書属性 \(P. 771\)](#)」にあります。

7. [サブミット]をクリックします。

カスタム マッピングが保存されます。これで、ポリシー サーバは、発行者 DN での電子メール属性の表記方法が異なる Web サーバおよびテスト ツールからのリクエストを、問題なく処理できるようになります。この処理は、「[カスタム マッピングが必要な証明書属性 \(P. 771\)](#)」に記載された、他のあらゆる属性に使用できます。

付録 A: SSL 認証方式のトラブルシューティング

このセクションには、以下のトピックが含まれています。

[概要](#) (P. 775)

[SSL 構成](#) (P. 776)

[SSL のトラブルシューティング](#) (P. 781)

概要

高度な SSL 認証方式を設定するには、Web サーバが SSL を使用するよう適切に設定されている必要があります。SSL 接続を介した認証方式を設定するときに発生する問題の多くは、実際には SSL 設定の問題である可能性があります。したがって、SSL を介した認証方式のトラブルシューティングの最初の手順は、SSL が適切に設定されていて、動作していることを確認することです。この確認は、SiteMinder Web エージェントと対話することなく実行されるため、これらのコンポーネントを個別に分析できます。

SSL 接続機能の決定

SSL を介した認証方式のトラブルシューティングの最初の手順は、SSL が適切に設定されて動作していることを確認することです。この確認は、SiteMinder Web エージェントと対話することなく実行されるため、これらのコンポーネントを個別に分析できます。

SSL 接続を確立できるかどうかを判断する方法

1. SSL を介した認証方式を使用するレلمを保護している SiteMinder Web エージェントを無効にします。

注: Web エージェントの無効化の詳細については、「Web エージェント設定ガイド」を参照してください。

2. ブラウザを使用して、次の URL の 1 つを閲覧します (証明書付きのブラウザを使用)。

- `https://web_server_name:port` (Netscape Web Server)
- `https://web_server_name:port/<SSL Virtual Directory>` (IIS Web Server)
- `https://web_server_name:port` (Apache Web Server)

SSL 接続が証明書を要求するよう設定されている場合は、証明書を選択するように促されます。

SSL 接続を確立できない場合は、「[SSL 設定](#) (P. 776)」を参照して、SSL の設定の詳細を確認してください。SSL 接続は確立できたが、**SiteMinder** を設定できなかった場合は、「[SSL のトラブルシューティング](#) (P. 781)」を参照してください。

SSL 構成

SiteMinder を使用する前に、SSL を設定し、正しく動作させる必要があります。SSL 接続を行うためには、受信する証明書の認証局を信頼する必要があります。たとえば、ブラウザが **VeriSign** で署名された証明書を提示する場合、**VeriSign** 認証局が **Web** サーバにインストールされ信頼されていなければなりません。提示されたクライアント証明書の信頼だけでなく、サーバ自体がクライアントに提示する証明書を持っている必要もあります。クライアントは、証明書を発行した認証機関を信頼する必要があります。これによって、相互認証が可能となります。これらの証明書をインストール後、**Web** サーバを SSL を使用するように、また必要に応じて証明書を要求するよう設定できます。

SSL の設定情報の詳細については、**Web** サーバソフトウェアに付属のマニュアルを参照してください。このセクションでは、**Web** サーバおよび **Web** ブラウザを設定して SSL 接続を正しく確立する手順を説明します。正しく SSL を設定しても接続上の問題が発生する場合は、このセクションの末尾にある一般的な問題を参照してください。

Netscape Web サーバ証明書のインストール

まだ **Web** サーバでキーを生成していない場合は、キーを生成する必要があります。キーの生成は、コマンドライン ユーティリティを使用して実行します。詳細については、**Netscape** のマニュアルを参照してください。

Netscape Web サーバ証明書をインストールする方法

1. キーを作成し、証明書をリクエストします。キーの作成は、**Netscape Server Administration** で **[Keys & Certificates]**、**[Request Certificate]** の順に選択して行います。
2. 証明書リクエストを作成してから、認証機関に移動してリクエストをロードします。認証局がサーバ証明書を作成します。
3. 作成されたサーバ証明書を **Web** サーバにロードします。**Web** サーバへのロードは、**[Keys & Certificates]** の **[Install Certificate]** で実行します。すべての情報を入力してから、**[Certificate For]** セクションの下に **[This Server]** がオンになっていることを確認します。
4. 証明書がロードされたら、その所在を確認して **[Manage Certificates]** で信頼されていることを確認します。**ServerCert** 証明書を検索します。証明書を選択し、信頼されていることを確認します。
5. **[OK]** をクリックして **Web** サーバを再起動します。

SSL を使用する Netscape Web サーバの設定

Netscape Web サーバ証明書をインストールしてから、証明書を要求して SSL を使用するように Netscape Web サーバを設定する必要があります。

SSL Web サーバ証明書を要求する方法

1. **Netscape Server Administration** で、**[Admin Preferences]** をクリックします。
2. **[Encryption On/Off]** をクリックして、暗号化が有効になっていることを確認します。
3. 証明書認証方式、もしくは基本認証と組み合わせた証明書認証方式を使用している場合は、証明書を要求する必要があります。証明書の要求は、**[Encryption Preferences]** 設定の **[Require Certificates]** オンに設定されている状態で実行します。証明書をインストールしたブラウザから、**https://servername:port** にアクセスできることを確認します。

注: **[Required Certificates for the Certificate]** または **[Basic Authentication Scheme]** はオンにしないでください。

Netscape のクライアント証明書に対する Web サーバの信頼

認証局が既に Web サーバにインストールされている場合は、次のセクションに進んでください。インストールされていない場合は、SSL Web サーバ上に認証局の証明書をインストールします。

Web サーバ上で Netscape のクライアント証明書に対する信頼を有効にする方法

1. 認証局の証明書を取得し、画面に表示するか、ファイルに保存します。
2. Netscape Server Administration で、[Keys & Certificates]をクリックします。
3. [Install Certificate]を選択します。
4. [Certificate For]フィールドで、[Server Security Chain]を選択します。
5. [Certificate Name]フィールドに説明を入力します。
6. 認証局の証明書をファイルに保存した場合は、[Message is in this file]フィールドにファイル名を入力します。ファイルに保存していない場合は、[Message text (with headers)]ラジオボタンをオンにして、[Message text (with headers)]フィールドに証明書を貼り付けます。
7. [OK]をクリックして Web サーバを再起動します。

Netscape 認証局の信頼の確立

認証局が Web サーバ内にインストールされている場合は、それらの間の信頼を確立できます。

Netscape 認証局の信頼を確立する方法

1. Netscape Server Administration で、[Keys & Certificates]をクリックします。
2. [Manage Certificates]を選択します。
3. 認証局を選択します。システムが証明書の詳細情報をダイアログボックスに表示します。
4. [Trust]を選択します。
5. [OK]をクリックして Web サーバを再起動します。

Windows のクライアント証明書に対する Web サーバの信頼

適切な認証局の証明書をインストールして、クライアント証明書を信頼する必要があります。

SSL Web Server は、認証局ごとに証明書が必要です。主な認証局は既にインストールされています。証明書スナップインを使用して、Windows オペレーティングシステムの証明書を設定できます。詳細については、Windows のマニュアルを参照してください。

SSL を使用する IIS Web サーバの設定

安全なポートが Web サーバ上で実行可能であることを確認します。通常、これはポート 443 です。ポートを確認するには、管理コンソールで Web サーバを右クリックします。[Web サイト]タブに SSL ポートが表示されます。ポート番号が設定されていることを確認します。

高度認証方式では Web サーバに仮想ディレクトリを作成します。これらの仮想ディレクトリは、特定の認証方式に要求されたときに SSL と証明書を自動的に要求するように設定されます。しかし、テストを目的とした仮想ディレクトリを作成したいこともあります。[ディレクトリ セキュリティ]タブの[セキュリティで保護された通信]を使用すると、テスト目的の仮想ディレクトリに証明書を要求するよう設定できます。

`https://servername:port/virtual directory` - ブラウザで証明書が要求されることを確認してください。

IIS Web サーバ証明書のインストール

まだ Web サーバでキーを生成していない場合は、キーを生成する必要があります。これは、管理コンソールのキー マネージャで行います。キーマネージャにアクセスするには、次の手順に従います。

注: この手順は、IIS 3 と IIS 4 で若干異なります。

IIS Web サーバ証明書をインストールする方法

1. 管理コンソールで、Web サーバを右クリックして[プロパティ]をクリックします。
2. [ディレクトリ セキュリティ]タブをクリックします。
3. [セキュリティで保護された通信]パネルで[キーマネージャ]をクリックします。

4. キーの下の[新しいキーの作成]をクリックし、ウィザードに従って処理を進めます。

キーを作成すると、前述した手順で作成したファイルを使用して証明書を要求できます。証明書認証局 (Certificate Authority) に移動して、サーバの証明書を要求します。証明書を受け取るには、手順 1 で作成した証明書リクエスト情報を貼り付ける必要があります。証明書を受け取ったら、管理コンソールの[ディレクトリ セキュリティ]タブに戻り、[キー マネージャ]をクリックして、次の手順で説明するキーの証明書をインストールします。

5. キーの名前を右クリックし、[キー証明書のインストール]をクリックします。
6. Web サーバを再起動します。

Apache のクライアント証明書に対する Web サーバの信頼

使用している Web サーバに認証局が既にインストールされている場合は、次のセクションに進んでください。インストールされていない場合は、SSL Web サーバ上に認証局の証明書を、次の手順に従ってインストールします。

Web サーバ上で Apache のクライアント証明書に対する信頼を有効にする方法

1. 次の Apache のコンポーネントをインストールして設定します。
 - Apache
 - OpenSSL
 - Mod_SSL
 - Mod_so- このモジュールは Apache の一部ですが、Web サーバの設定中は使用可能でなければなりません。
 - RSAREf-2.0

Web サーバのインストールの詳細については、「ポリシー サーバ インストール ガイド」を参照してください。
2. CA 証明書を `apache/conf/ssl.crt` ディレクトリに x509 b64 形式でコピーします。
3. `apache/conf/ssl.crt` ディレクトリで `make` を実行します。
4. Web サーバを再起動します。

Apache Web Server 証明書のインストール

Apache Web サーバに証明書をインストールする手順は、個々の設定によって異なります。これらのコンポーネントを設定する方法の詳細については、Mod_SSL および OpenSSL のマニュアルを参照してください。

SSL のトラブルシューティング

以下のセクションでは、SSL 認証方式で処理する際に生じる最も一般的な問題について詳しく説明します。

証明書用のプロンプトはありませんでした

証明書の入力を要求されなかった場合は、SSL が適切に設定されていることを確認してください。Web エージェントがインストールされている場合は、Web エージェントを無効にします。まず最初に行うことは、簡単な SSL 接続の確認です。

SSL 接続を確立できるかどうかを判断する方法

1. SSL を介した認証方式を使用するレلمを保護している SiteMinder Web エージェントを無効にします。

注: Web エージェントの無効化の詳細については、「Web エージェント設定ガイド」を参照してください。

2. ブラウザを使用して、次の URL の 1 つを閲覧します (証明書付きのブラウザを使用)。

- `https://web_server_name:port` (Netscape Web Server)
- `https://web_server_name:port/<SSL Virtual Directory>` (IIS Web Server)
- `https://web_server_name:port` (Apache Web Server)

SSL 接続が証明書を要求するよう設定されている場合は、証明書を選択するように促されます。

前の手順に従っても、証明書のプロンプトが表示されない

証明書のプロンプトを受け取っていない場合、手立てとして 5 つの手順が考えられます。

- [すべての Netscape ブラウザが、毎回証明書を要求するように設定されていることを確認する](#) (P. 782)
- [すべての Web サーバが SSL を使用し、証明書を要求するように設定されていることを確認する](#) (P. 782)
- [SiteMinder 仮想ディレクトリごとに、次の設定を確認する](#) (P. 783)
- [Web サーバの証明書の有効期限を確認する](#) (P. 783)
- [ブラウザ証明書の有効期限を確認する](#) (P. 784)

すべての Netscape ブラウザが、毎回証明書を要求するように設定されていることを確認する

Netscape ブラウザでは、同一の証明書は自動的にパスするように設定できます。この設定は、ユーザに対して証明書の選択を要求せずに、証明書を使用する SSL 接続を確立します。

すべての Netscape ブラウザが、毎回証明書を要求するように設定されていることを確認する方法

1. Netscape ブラウザで、ツール バーの[Security]をクリックします。
2. [Navigator]をクリックします。
3. [Certificates to Identify You to a Web Site]セクションのドロップダウン ボックスで、[Ask Every Time]が設定されていることを確認します。

すべての Web サーバが SSL を使用し、証明書を要求するように設定されていることを確認する

Netscape Web Server の場合

1. Netscape Server Administration で、[Admin Preferences]をクリックします。
2. [Encryption On/Off]をクリックして暗号化が有効になっていることを確認し、[OK]をクリックします。
3. [Encryption Preferences]をクリックして[Required Certificates]がオンになっていることを確認します。
4. Web サーバを再起動します。

IIS Web サーバの場合

仮想ディレクトリ `SMGetCredCert`、`SMGetCredCertOptional`、`SMGetCredNoCert` が作成されて正しく設定されていることを確認します。

- `SMGetCredCert` - [クライアント証明書を要求する]がオンになっている
- `SMGetCredCertOptional` - [クライアント証明書を受諾する]がオンになっている
- `SMGetCredNoCert` - [クライアント証明書を無視する]がオンになっている

注: SiteMinder SSL 認証セットアップの一部として、SiteMinder は認証方式が要求する SSL 接続のタイプに基づいて SSL 仮想ディレクトリを設定します。

SiteMinder 仮想ディレクトリごとに、次の設定を確認する

SiteMinder 仮想ディレクトリごとに、次の設定を確認する方法

1. 管理コンソールで、仮想ディレクトリを右クリックして[プロパティ]をクリックします。
2. [ディレクトリ セキュリティ]タブをクリックします。
3. [セキュリティで保護された通信]をクリックします。

Apache Web サーバの場合

`httpd.conf` ファイルで、`SSLVerifyClient` が次のように設定されていることを確認します。

- SSL を介した基本の場合: `SSLVerifyClient none`
- 証明書または基本の場合: `SSLVerifyClient optional`
- 証明書/証明書および基本の場合: `SSLVerifyClient require`

注: 証明書が必須、またはオプションとされている Apache Web サーバでは、`httpd.conf` ファイルの "`SSL Verify Depth 10`" 行をコメントアウトする必要があります。

Web サーバの証明書の有効期限の確認

Netscape サーバ

1. Netscape Server Administration で、[Keys & Certificates]をクリックします。
2. [Manage Certificates]をクリックします。

3. [ServerCert]をクリックします。
4. サーバ証明書が信頼されていて、有効期限が切れていないことを確認します。サーバ証明書が存在しない、または有効期限が切れている場合は、[「Netscape Web サーバ証明書のインストール \(P. 776\)」](#)の手順に従って新しい証明書をリクエストする必要があります。

IIS サーバ

1. 管理コンソールで、**Web** サーバを右クリックして[プロパティ]をクリックします。
2. [ディレクトリ セキュリティ]タブをクリックします。
3. [セキュリティで保護された通信]パネルで[キーマネージャ]をクリックします。
4. プロパティを表示するキーを選択して、キーの有効期限が切れていないことを確認します。
5. 変更を行う必要がある場合は、**Web** サーバを再起動します。

Apache サーバ

Apache Web サーバ証明書の有効期限が切れると、サーバの起動時に証明書の有効期限が切れていることを示すエラー メッセージが表示されます。

ブラウザ証明書の有効性の確認

Netscape ブラウザ

1. Netscape ブラウザで、ツール バーの[Security]をクリックします。
2. [Security Info]パネルで、[Certificates]の下の[Yours]をクリックします。
3. [These are your certificates]の下のリストボックスに、証明書の一覧が表示されます。表示されていない場合は、証明書をインストールする必要があります。
4. 証明書の一覧が表示されている場合は、[Verify]をクリックして証明書が有効であることを確認します。

Internet Explorer ブラウザ

1. IE ブラウザのメニューで、[ツール]-[インターネット オプション]を選択します。
2. [コンテンツ]タブで、[証明書]、[個人]を選択します。
3. 証明書の一覧が表示されることと、その証明書が有効であることを確認します。証明書の一覧が表示されていないか有効でない場合は、新しい証明書をインストールします。

証明書のプロンプトの表示後に認証失敗を受け取る

Apache Web サーバ

- SSL Web Server に指定した証明書の認証局が含まれていることを確認します。
- 指定した証明書の認証局を SSL Web Server が信頼していることを確認します。
- SSL Verify Depth 10 がコメントアウトされていることも確認します。

Netscape Web Server

証明書の認証局が一覧表示されていて、証明書の信頼の有効期限が切れていないことを確認します。証明書が表示されていないか有効でない場合は、新しい CA 証明書をインストールします。

IIS Web Server

証明書が一覧表示されていることと、その証明書が有効であることを確認します。証明書の一覧が表示されていないか有効でない場合は、新しい証明書をインストールします。インストール先ディレクトリへのアクセスが可能であれば、証明書が正しくインストールされていることを確認できます。

ポリシー サーバと Web エージェントの設定の確認

ユーザの特定の Web サーバに基づいて、前のトピックの手順を完了した後に、ポリシー サーバと Web エージェントの設定を確認します。

ポリシー サーバと Web エージェントの設定が正しいことを確認する方法

1. ポリシー サーバが正しく作成されていることを確認します。
2. Web エージェントが、正しいポリシー サーバの情報を持っていることを確認します。
3. Web エージェントが有効であることを確認します。
4. Web エージェントとポリシー サーバを再起動します。

SiteMinder ポリシーによってアクセスが許可されるはずですが、SSL 認証失敗のメッセージを受信しました

これは、呼び出されたポリシーは存在していますが、ユーザのアクセスは不正に拒否されている状態です。この問題は、いくつかの設定エラーによって発生します。一般的なエラーは次のとおりです。

- SSL サーバがクライアント証明書を要求するように設定されていません。その結果、クライアントが証明書を渡さないため、SiteMinder は認証処理を実行できません。Web エージェントのログイン オプションを有効にすると、この状況を確認できます。ログには、ユーザが不明であることが示されているはずです。この問題を解決するには、SSL Web サーバの証明書要求をオンにします。
- ポリシーが正しく作成されませんでした。ポリシーのユーザを確認して、選択内容が正しいことを確認してください。
- Apache Web Server の場合は、SSL Verify Depth が正しく設定され、コメントアウトされていることを確認します。

「見つかりません」エラー メッセージの表示

このメッセージは、通常、認証方式のパラメータ設定が正しくない場合に発生します。リダイレクトが正しく設定されていないため、Web サーバが SSL Web エージェントのコンポーネントを見つけることができません。

証明書認証方式または基本認証方式を実行しているが、基本クレデンシャルを入力できません。

Netscape Web サーバでは、*証明書または基本*の認証方式の場合に、Web サーバで暗号化が有効である必要がありますが、証明書は必要ありません。Netscape Server Administration の [Encryption Preferences] セクションで、[Require Certificate] 設定が [No] になっていることを確認します。

付録 B: LanMan ユーザ ディレクトリ

このセクションには、以下のトピックが含まれています。

[LanMan ユーザ ディレクトリの概要 \(P. 789\)](#)

[LanMan ディレクトリ接続の前提条件 \(P. 790\)](#)

[LanMan ディレクトリ接続の設定 \(P. 790\)](#)

[Windows ユーザ ディレクトリのフェイルオーバー \(P. 792\)](#)

[LanMan ユーザディレクトリの検索条件 \(P. 792\)](#)

LanMan ユーザ ディレクトリの概要

Windows 環境では、ポリシー サーバがディレクトリ サービス内のリソースの列挙と管理を行う際、**Microsoft Active Directory Service Interface (ADSI)** 層を使用します。この層は、分散コンピューティング環境にあるさまざまなネットワークプロバイダのディレクトリ サービスの機能を抽出します。ただし、現在の **ADSI** のバージョンには、ポリシー サーバのパフォーマンスを低下させる可能性がある制約があります。

ADSI を使用している場合、各 **Windows** ディレクトリリクエストは常に、まず **PDC** (プライマリドメインコントローラ) を経由する必要があります。これにより、**PDC** が処理しなければならないネットワークトラフィックはさらに増えます。このジレンマに対するカスタムソリューションは、ポリシー サーバが **Windows** ディレクトリリクエストを **BDC** (バックアップドメインコントローラ) に渡して **PDC** を経由しない方法です。ポリシー サーバは、**LanMan** ディレクトリ接続を使用してこの種のカスタムソリューションを処理します。

LanMan ユーザディレクトリ接続オプションを使用すると、**Windows** レジストリ内の各ユーザディレクトリの検索に使用される、**BDC** のフェイルオーバーリストを指定できます。**LanMan** ディレクトリ接続を使用すると、ポリシー サーバは、**Windows** ディレクトリリクエストを **PDC** に渡す代わりに、レジストリリスト内の最初のアクティブ **BDC** に送信します。

LanMan ディレクトリ接続の前提条件

ポリシー サーバが LanMan ディレクトリ接続を使用して Windows ディレクトリ内のユーザデータにアクセスできるようにするには、次の条件を満たす必要があります。

- ファイル SmDsLanman.dll が、ポリシー サーバのインストール ディレクトリ内にある必要があります。デフォルトの場所は以下のとおりです。

installation_directory\netegrity\siteminder\bin

- LanMan ディレクトリへの接続を設定する必要があります。

LanMan ディレクトリ接続の設定

LanMan ユーザ ディレクトリを設定できます。以下に、ポリシー サーバへのユーザ ディレクトリ接続を作成する手順を示します。

1. [LanMan ディレクトリ接続用のレジストリ キーの設定](#) (P. 790)
2. [LanMan ユーザ ディレクトリ接続の設定](#) (P. 791)

LanMan ディレクトリ接続用のレジストリ キーの設定

LanMan ディレクトリ接続の設定では、まず適切なレジストリ キーを設定します。

LanMan ディレクトリ接続用のレジストリ キーを設定する方法

1. Windows の[スタート]メニューから[ファイル名を指定して実行]を選択します。
[ファイル名を指定して実行]ダイアログが表示されます。
2. **regedit** と入力し、[OK]をクリックします。
レジストリ エディタが表示されます。
3. 次のレジストリキーを変更します。
 - HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\Siteminder\CurrentVersion\Ds にある NameSpaces キーは、次の文字列値に設定されています。
"LDAP:;ODBC:;OCI:;WinNT:;Custom:;AD:"
 - NameSpaces レジストリ キーの値データに、文字列 Lanman を追加します。

4. 次のレジストリキーを作成します。

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion  
\Ds\Lanman_DC
```

5. Lanman_DC キーの下に NT ドメイン名のレジストリキーを作成します。

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion  
\Ds\Lanman_DC\<NT_domain_name>
```

以下に例を示します。

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion  
\Ds\Lanman_DC\MyDomain
```

6. 新しく作成された NT ドメインキーの下に **DWORD** 型のレジストリ値 **NumUserDir** を作成します。値データには、この NT ドメインにあるユーザディレクトリセットの実数(最大 16)を入力します。
7. BDC の各フェールオーバー リストに、**UserDir0**、**UserDir1**、...**UserDirN** といった文字列レジストリ値を 0 から順番に作成します。
8. 各フェールオーバー リストの文字列を、コンマで区切りながら入力します。**SmDsLanman** はリストを読み取り、各フェールオーバーリストの最初のアクティブ BDC を探して **Windows NT** のユーザとグループを検索します。
9. 他の NT ドメインについて、手順 5 ～ 7 を繰り返します。
10. ポリシー サーバ サービスを再起動します。

注: ポリシー サーバの開始と停止の詳細については、*ポリシー サーバ管理ガイド*を参照してください。

LanMan ユーザ ディレクトリ接続の設定

ポリシー サーバが **LanMan Directory** ユーザ ストアと通信するように、ユーザ ディレクトリ接続を設定することができます。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

LanMan ユーザ ディレクトリ接続を設定する方法

1. [インフラストラクチャ]-[ディレクトリ]をクリックします。
2. [ユーザ ディレクトリ]、[ユーザ ディレクトリの作成]をクリックします。
[ユーザ ディレクトリの作成]ペインが表示されます。

3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。

[ユーザ ディレクトリの作成: <名前>] ペインが表示されます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [一般]グループ ボックスのフィールドにユーザ ディレクトリの名前および説明を入力します。
5. [ネームスペース]リストから[LanMan]を選択します。
[LanMan 設定]が開きます。
6. [ドメイン コントローラ キー]フィールドでレジストリ キーを設定した NT ドメインの名前を入力します。
7. [サブミット]をクリックします。

ユーザ ディレクトリの作成タスクが処理のためにサブミットされます。

Windows ユーザ ディレクトリのフェイルオーバー

LanMan ユーザディレクトリ接続用に作成する、レジストリキーのリストによって、フェイルオーバーの順序が決定されます。

LanMan ユーザディレクトリの検索条件

LanMan ディレクトリ接続は Windows ユーザディレクトリ接続の 1 タイプです。LanMan ディレクトリ接続は通常の Windows 接続と同じように機能しますが、実際のドメインコントローラがリクエストを処理する点が異なります。このことは、ユーザディレクトリの検索を実行する手順には影響しません。

付録 C: CA SSO/WAC の統合

このセクションには、以下のトピックが含まれています。

[概要 \(P. 793\)](#)

[統合された SiteMinder および CA SSO のアーキテクチャ例 \(P. 794\)](#)

[SiteMinder と CA SSO の統合の前提条件 \(P. 801\)](#)

[SiteMinder から CA SSO へのシングル サインオンの設定 \(P. 802\)](#)

[CA SSO クライアントから SiteMinder へのシングル サインオンの設定 \(P. 805\)](#)

[CA SSO から SiteMinder へのシングル サインオンの設定 \(P. 806\)](#)

[smetssockie Web エージェント アクティブ レスポンス属性の設定 \(P. 808\)](#)

[smauthetssso カスタム認証方式の設定 \(P. 810\)](#)

概要

SiteMinder は、SiteMinder から CA SSO 環境へのシングル サインオンを提供しています。ユーザは、SiteMinder または CA SSO 環境にログインして SiteMinder によって認証された後は、両方の環境で認証されたことになります。認証されたユーザは、許可されている限り、どちらの環境でも、保護されているリソースにアクセスできます。クレデンシャルを再入力する必要はありません。各環境でのユーザの許可は、有効なポリシーに基づいて決定されます。

保護されたリソースへのユーザのアクセスが許可されている場合、SiteMinder および CA SSO は、それら自身のセッション ストアにそれぞれ、ユーザのクレデンシャルを維持しています。さらに、それら独自のセッション クレデンシャルを持っており、それを互いに読み取ることはできないため、ユーザ クレデンシャルは別々に維持されます。これらのクレデンシャルは別のストア内に保管されているため、シングル サインオンを有効にするには、SiteMinder ポリシー サーバおよび CA SSO ポリシー サーバが、同じ Cookie ドメインの一部であり、同じユーザまたは認証ストアを共有する必要があります。

このシングル サインオン設定では、SiteMinder ポリシー サーバおよび CA SSO ポリシー サーバを、同じマシンと別のマシンのいずれにも配置できます。SiteMinder では、Web エージェント、セキュア プロキシ サーバのいずれかまたは両方を含むことができます。使用している SiteMinder 環境に基づいて、Web エージェントまたはセキュア プロキシ サーバを使用してください。CA SSO は、eTrust Web Access Control (WAC) Web エージェントを使用しており、SiteMinder によるシングル サインオンを有効にするために現在の環境を変更する必要はありません。

注: SiteMinder および CA SSO について十分に理解し、作業に慣れた後に、これらの製品間のシングル サインオンを設定してください。サポートされている SiteMinder、SiteMinder セキュア プロキシ サーバ、CA SSO、および各バージョンの eTrust WAC のリストについては、[テクニカル サポート サイト](#)の 6.0 SiteMinder およびエージェントのプラットフォーム マトリックスを参照してください。

統合された SiteMinder および CA SSO のアーキテクチャ例

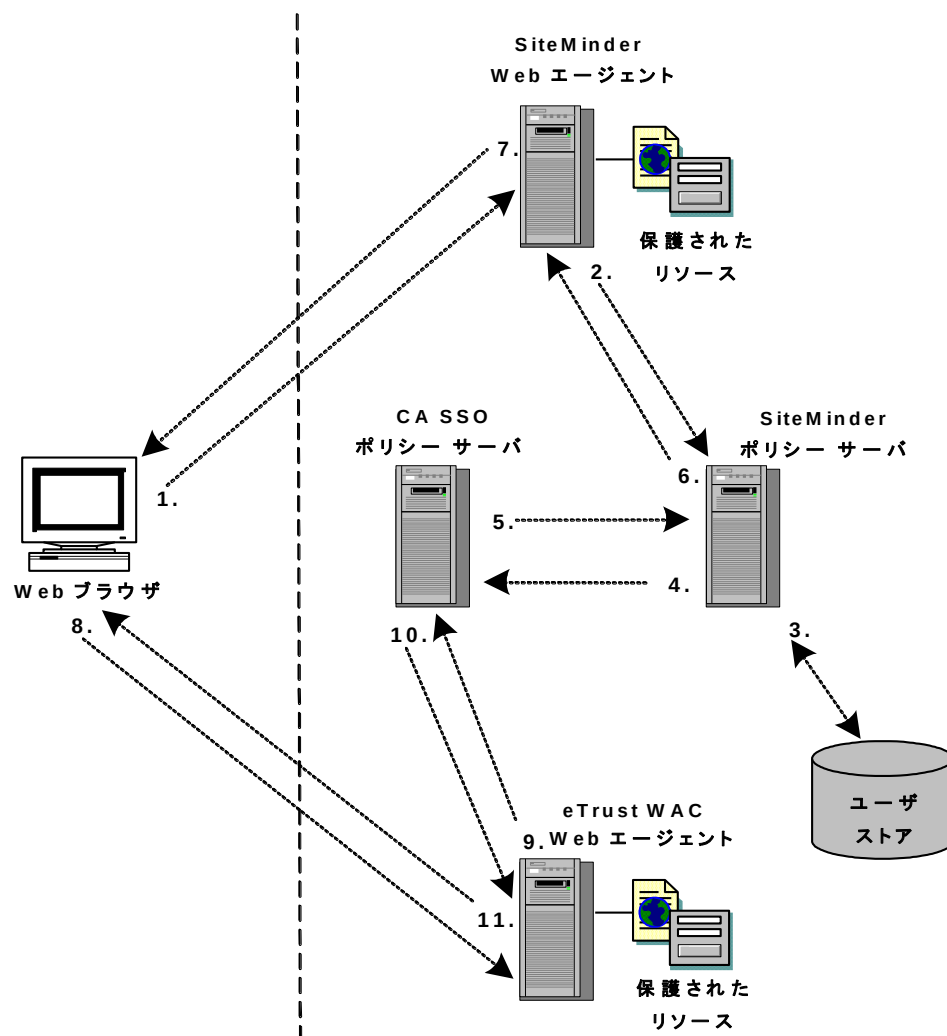
以下は、SiteMinder と CA SSO 環境の間のシングル サインオンの 3 つの例です。

1. ユーザは、Web ブラウザを使用する SiteMinder で認証され、その後に、CA SSO で保護されたリソースにアクセスします(「[例 1: ユーザが、CA SSO の前に、SiteMinder で保護されたリソースにアクセス \(P. 795\)](#)」を参照)。
2. ユーザは、デスクトップ CA SSO クライアント経由で CA SSO で認証され、その後に、Web ブラウザを使用して、SiteMinder で保護されたリソースにアクセスします(「[例 2: 認証された CA SSO クライアントユーザが SiteMinder リソースにアクセス \(P. 797\)](#)」を参照)。
3. ユーザは、Web ブラウザを使用して CA SSO で認証され、その後に、SiteMinder で保護されたリソースにアクセスします(「[例 3: ユーザが、SiteMinder の前に、CA WAC で保護されたリソースにアクセス \(P. 799\)](#)」を参照)。

注: これらの例では、SiteMinder ポリシー サーバおよび CA SSO ポリシー サーバでの認証による、保護されているリソースへのアクセス手順は、わかりやすくするために省略されています。

CA SSO 前の SiteMinder で保護されたリソースへのユーザ アクセス

以下の例は、WAC で保護されたリソースの前に、SiteMinder で保護されたリソースにアクセスするユーザを示します。

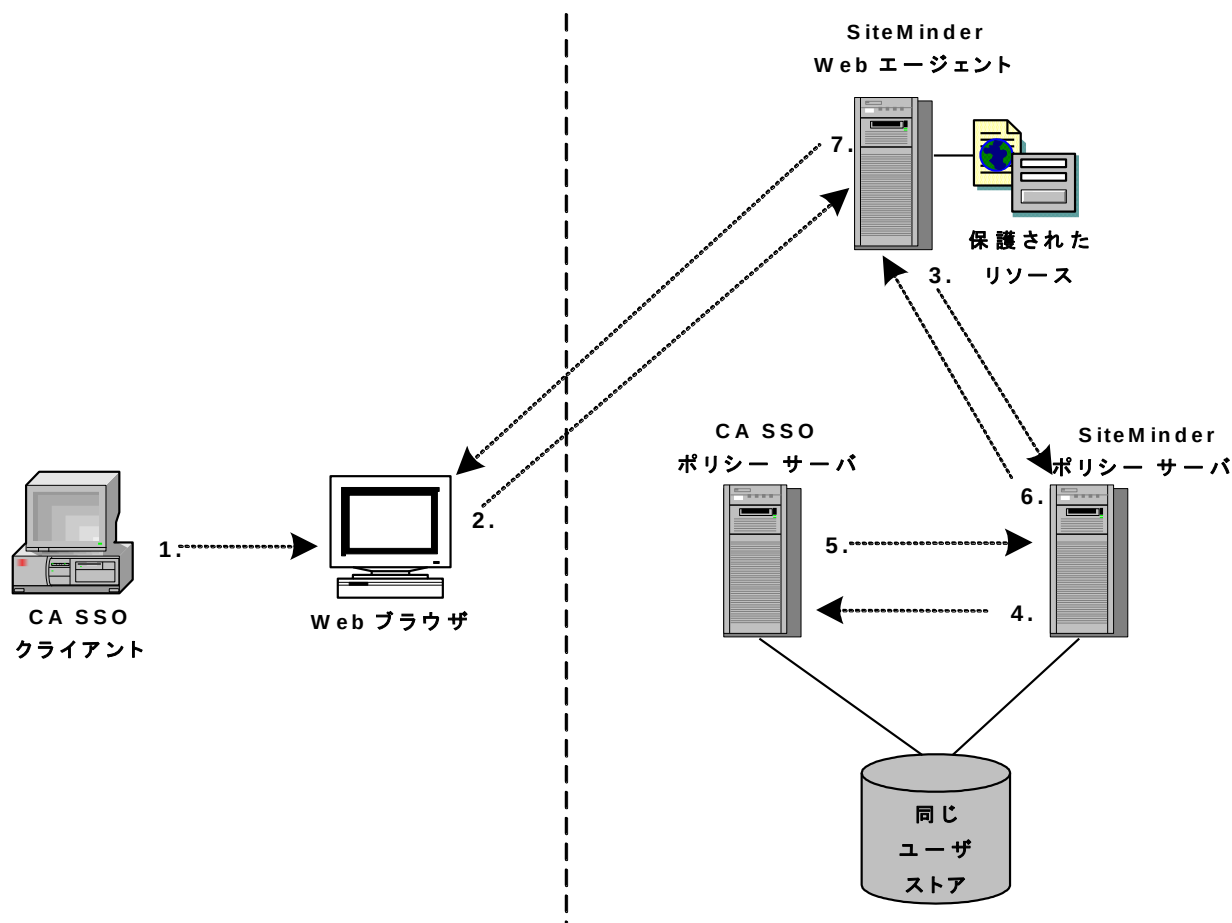


1. ユーザが SiteMinder で保護されたリソースにアクセスしようとする、SiteMinder Web エージェント/セキュア プロキシ サーバはそのリクエストをインターセプトします。ユーザはエージェント/SPS に認証クレデンシャルを渡します。
2. Web エージェント/セキュア プロキシ サーバはクレデンシャルを SiteMinder ポリシー サーバに転送し、検証します。
3. SiteMinder ポリシー サーバは、ユーザのクレデンシャルがユーザストアで有効であることを確認します。

4. 認証に成功すると、SiteMinder ポリシー サーバは CA SSO ポリシー サーバに、SiteMinder ユーザの CA SSO cookie を発行して返すように要求します。
5. CA SSO ポリシー サーバはユーザを検証し、そのユーザの CA SSO Web 認証クレデンシャルを SiteMinder ポリシー サーバに転送します。
6. SiteMinder ポリシー サーバは、SiteMinder Web エージェント/セキュア プロキシ サーバに CA SSO Web 認証クレデンシャルを転送します。
7. SiteMinder Web エージェント/セキュア プロキシ サーバは、ユーザのブラウザに CA SSO Web 認証と SiteMinder cookie を設定し、ユーザにリソースを返します。
8. ユーザが CA SSO リソースにアクセスしようとする、eTrust WAC Web エージェントはそのリクエストをインターセプトします。
9. eTrust WAC Web エージェントは、ユーザの CA SSO Web 認証 cookie クレデンシャルを CA SSO ポリシー サーバで検証します。
10. CA SSO ポリシー サーバは、eTrust WAC Web エージェントにユーザのクレデンシャルが有効であることを通知します。
11. eTrust WAC Web エージェントにより、ユーザは CA SSO で保護されたリソースにアクセスできます。

認証済み CA SSO クライアント ユーザの SiteMinder リソースへのアクセス

以下の例は、SiteMinder で保護されているリソースにアクセスする、認証済み CA SSO クライアント ユーザを示します。



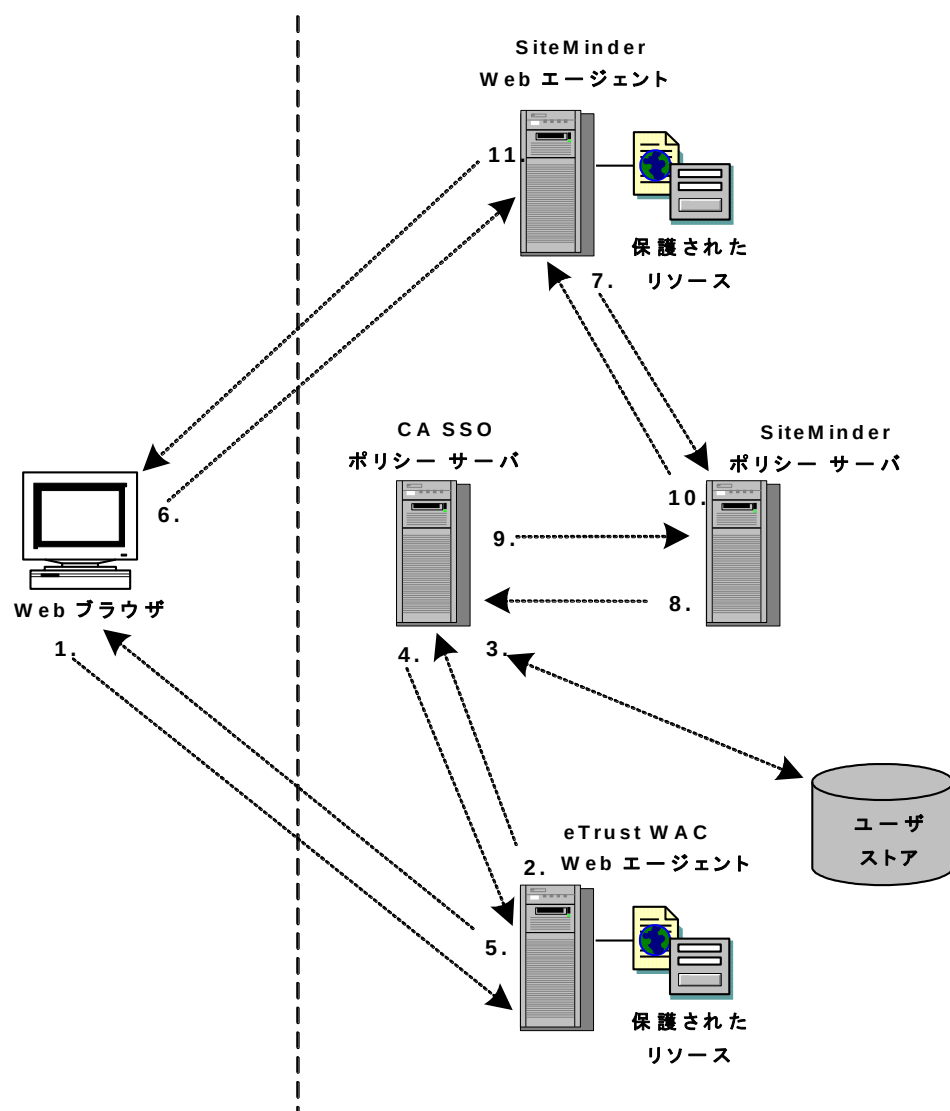
1. 認証済み CA SSO クライアント ユーザが Web ブラウザを起動します。この時、CA SSO クライアントはブラウザに CA SSO Web 認証 cookie を設定します。
2. ユーザが SiteMinder で保護されているリソースに Web ブラウザでアクセスしようとする、そのリクエストは SiteMinder Web エージェント/セキュアプロキシサーバでインターセプトされます。
3. SiteMinder Web エージェント/セキュアプロキシサーバは、CA SSO Web 認証 cookie を SiteMinder ポリシーサーバに転送します。
4. SiteMinder ポリシーサーバは CA SSO ポリシーサーバに CA SSO Web 認証 cookie を転送します。

5. CA SSO ポリシー サーバは CA SSO Web 認証 cookie を検証し、SiteMinder ポリシー サーバにユーザ名を返します。
6. SiteMinder ポリシー サーバは、返されたユーザ名を SiteMinder ユーザストアで検証し、対応する SiteMinder cookie を発行して SiteMinder Web エージェント/セキュア プロキシ サーバに返します。
7. SiteMinderWeb エージェント/セキュア プロキシ サーバは、リクエストされたリソースをユーザに返します。このユーザはこの時点で、SiteMinder と CA SSO 環境に必要な認証 cookie クレデンシャルを持っています。

SiteMinder にアクセスする前の eTrust WAC で保護されたリソースへのユーザ アクセス

以下の例は、SiteMinder の前に WAC で保護されたリソースにアクセスするユーザを示します。

注: この例は、環境で IIS6 WAC エージェントが使われているものとします。IIS6 WAC エージェントは、以下の例に当てはまるただ 1 つのプラットフォームです。



1. ユーザが **CA SSO-protected** リソースにアクセスしようとする、**eTrust WAC Web** エージェントはそのリクエストをインターセプトします。ユーザはエージェントに認証クレデンシャルを渡します。
2. **Web** エージェントはクレデンシャルを **CA SSO** ポリシー サーバに転送し、検証します。
3. **CA SSO** ポリシー サーバは、ユーザのクレデンシャルがユーザストアで有効であることを確認します。
4. **CA SSO** ポリシー サーバは、ユーザの **eTrust SSO Web** クレデンシャルを **eTrust WAC Web** エージェントに転送します。
5. **eTrust WAC Web** エージェントはユーザの **CA SSO Web** 認証 cookie を **Web** ブラウザに設定します。
6. ユーザが **SiteMinder** で保護されたリソースにアクセスしようとする、**SiteMinder Web** エージェント/セキュア プロキシ サーバはそのリクエストをインターセプトします。
7. **SiteMinder Web** エージェント/セキュア プロキシ サーバは、ユーザの **CA SSO Web** 認証クレデンシャルを **SiteMinder** ポリシー サーバに転送します。
8. **SiteMinder** ポリシー サーバは、ユーザの **CA SSO Web** 認証クレデンシャルを **eTrust SSO** ポリシー サーバに転送します。
9. **CA SSO** ポリシー サーバはユーザの **CA SSO Web** 認証クレデンシャルを検証し、ユーザ名を **SiteMinder** ポリシー サーバに転送します。
10. **SiteMinder** ポリシー サーバは、返されたユーザ名を **SiteMinder** ユーザストアで検証し、対応する **SiteMinder cookie** を発行して **SiteMinder Web** エージェント/セキュア プロキシ サーバに返します。
11. **SiteMinder Web** エージェント/セキュア プロキシ サーバは、ユーザのブラウザに **SiteMinder cookie** を設定し、リクエストされたリソースにユーザがアクセスできるようにします。

SiteMinder と CA SSO の統合の前提条件

SiteMinder と CA SSO の間でシングル サインオン統合を設定する前に、次の手順を実行します。

1. CA SSO をインストールして設定します。

注: CA SSO ポリシー サーバのインストール時には、以下の情報を収集します。

- SSO 管理者の名前およびパスワード。SiteMinder ポリシー サーバでは、`smauthetsso` 認証方式を使用した CA SSO ポリシー サーバへの認証時に、管理者の名前およびパスワードを使用します。
- SSO チケット暗号化キー。SiteMinder ポリシー サーバの `smetssocookie` アクティブ レスポンスではこの値が必要になります。

2. SiteMinder 環境および CA SSO 環境が、同じ FIPS モード (AES 暗号化) で実行されていることを確認します。

重要: 両方の環境が同じ FIPS モードで実行されていない場合は、統合に失敗します。

3. 以下の点について考慮してください。

- 統合が FIPS のみのモードで実行される場合、SiteMinder ポリシー サーバは **r12.0 SP1 CR3** 以降で実行されている必要があります。必要な場合は、CA SSO ポリシー サーバと通信する SiteMinder ポリシー サーバをアップグレードしてください。
- ポリシー サーバがインストールされているオペレーティング システムで `smauthetsso` 認証方式がサポートされていることを確認します。統合するには、`smauthetsso` 認証方式が設定されている必要があります。詳細については、**r12.0 SP3 SiteMinder プラットフォーム サポート マトリックス** を参照してください。

SiteMinder から CA SSO へのシングル サインオンの設定

SiteMinder は、SiteMinder から CA SSO 環境へのシングル サインオンを提供しています。

SiteMinder Web エージェント、またはセキュア プロキシ サーバを使用した SiteMinder から CA SSO へのシングル サインオンを有効にする方法

Web エージェントまたはセキュア プロキシ サーバと一緒にインストールされている SiteMinder SSO プラグインを有効にします。

r12.0 SP3 IIS 6.0 または Apache 2.0 Web エージェントの場合

- WebAgent.conf ファイルの以下の行からコメント(#) 文字を削除します。
#LoadPlugin=<Path to eTSSOPlugin.dll or libetssoplugin.so>
- IIS 6.0/Apache 2.0 の WebAgent.conf にアクセスするには、「Web エージェント設定ガイド」の「WebAgent.conf ファイルのロケーション」(すべての Web エージェント)」を参照してください。
- プラグインを有効/無効にする方法については、「Web エージェント設定ガイド」で、IIS 6.0 および Apache 2.0 用の Web エージェントプラグインのロードについて参照してください。

注: WebAgent.conf ファイルを変更してから Web サーバを再起動します。これで新しい設定が有効になります。

6.0 セキュア プロキシ サーバの場合

- <SPS_install_dir>\proxy-engine\conf\defaultagent\WebAgent.conf にある WebAgent.conf ファイルの以下の行からコメント(#) 文字を削除します。
#LoadPlugin=<Path to eTSSOPlugin.dll or libetssoplugin.so>

注: WebAgent.conf ファイルを変更してからセキュア プロキシ サーバを再起動します。これで新しい設定が有効になります。

WAC Web エージェントを使用したシングル サインオンを有効にする方法

1. 以下のパラメータを設定し、WAC Web エージェントの `webagent.ini` ファイルにドメインを設定します。

`DomainCookie=<domain>`

ここで `<domain>` は、CA SSO と SiteMinder Web エージェントで同じドメインになります (たとえば、`test.com`)。

このファイルは WAC Web エージェント マシン上の以下の場所にインストールされます。

`C:\Program Files\CA\WebAccessControl\WebAgent\webagent.ini`

2. 以下の Web サーバと認証方法の設定が `webagent.ini` ファイルにあることを確認します。
 - 「Authentication methods」と「The default authentication method」パラメータは、SSO として設定します。
 - `WebServerName`、`PrimaryWebServerName`、`AgentName`、`NTLMPath` および `Secure` は、CA SSO Web Access Control がインストールされているマシンをポイントする必要があります。
 - `ServerName` 属性は、CA SSO ポリシー サーバがインストールされているマシンの IP アドレスをポイントする必要があります。

注: WAC Web エージェントの設定の詳細については、WAC のマニュアルを参照してください。

CA SSO ポリシー マネージャ検証手順

1. SiteMinder と CA SSO ポリシー サーバが同じユーザ ストア、または認証ストアを使用することを確認します。
2. 以下があることを確認します。
 - SSO 管理者の名前およびパスワード。smauthetsso 認証方式による CA SSO ポリシー サーバでの認証時に、SiteMinder ポリシー サーバは、管理者の名前とパスワードを使用します。
 - SSO チケット暗号化キー。これは、SiteMinder ポリシー サーバの smetssocookie アクティブ レスポンスに必要です。

注: ポリシー サーバの設定の詳細については、CA SSO のマニュアルを参照してください。

SiteMinder ポリシー サーバの設定手順

1. 管理 UI を使用して、Web エージェント、エージェント設定オブジェクト、ホスト設定オブジェクトを作成します。詳細については、「ポリシー サーバ インストール ガイド」および「Web エージェント インストール ガイド」を参照してください。

2. 同じユーザまたは認証ストアを使用するように SiteMinder と CA SSO ポリシー サーバを設定します。

SiteMinder ユーザ ストアの設定手順については、このガイドのユーザ ディレクトリに関する章を参照してください。

CA SSO 認証ストアについては、CA SSO のマニュアルを参照してください。

3. smetssocookie (証明書) カスタム アクティブ レスポンスを設定します。
4. SiteMinder Web エージェントでリソースを保護するために、管理 UI を使用してドメイン、レルム、ルールを作成します。

注: ルールの作成時、それらに smetssocookie カスタム アクティブ レスポンスを追加します。

検証手順の概要

1. SiteMinder Web エージェント、および WAC Web エージェントによって保護されているリソースにアクセスするためのクレデンシャルをユーザを設定します。
2. SiteMinder ポリシー サーバと 管理 UI をホストしている Web サーバを再起動します。
3. SiteMinder Web エージェントによって保護されているリソースにアクセスし、この Web エージェントに適切なユーザ クレデンシャルを渡します。
4. このリソースにアクセスできるようになってから、同じブラウザ セッションで WAC Web エージェントで保護されているリソースをリクエストします。
クレデンシャルを要求されることなく、このリソースにアクセスできます。

詳細情報:

[smauthetsso カスタム認証方式の設定 \(P. 810\)](#)

[smetssocookie Web エージェント アクティブ レスポンス属性の設定 \(P. 808\)](#)

[ドメイン \(P. 471\)](#)

[レルム \(P. 481\)](#)

[ルール \(P. 495\)](#)

[Web エージェント アクションのルールの設定 \(P. 504\)](#)

CA SSO クライアントから SiteMinder へのシングル サインオンの設定

SiteMinder には、CA SSO クライアントから SiteMinder へのシングル サインオン機能があります。

CA SSO クライアントから SiteMinder へのシングル サインオンを有効にする方法

SiteMinder ポリシー サーバの設定手順

1. 管理 UI を使用した `smauthetsso` カスタム認証方式の設定
2. SiteMinder Web エージェントでリソースを保護するために、管理 UI を使用してドメイン、レルム、ルールを作成します。
3. リソースを保護するために `smauthetsso` カスタム認証方式を設定します。
4. CA SSO クライアントで保護されているブラウザにすでにアクセスできるユーザに、保護されているリソースへのアクセスを付与するポリシーを作成します。

CA SSO クライアント検証手順

CA SSO クライアントの `SsoClnt.ini` ファイルで以下を設定します。

注: `SsoClnt.ini` ファイルは、CA SSO クライアントマシン上の `C:\Program Files\CA\CA SSO\Client` にインストールされます。CA SSO クライアントの設定に関する詳細については、CA SSO マニュアルを参照してください。

`DomainNameServer=<eSSO_WA_FQDN> <SM_WA_FQDN>`

`eSSO_WA_FQDN`

(オプション) WAC Web エージェントに、完全修飾ドメイン名を指定します。

`SM_WA_FQDN`

SiteMinder Web エージェントに完全修飾名を指定します。

検証手順の概要

1. CA SSO クライアント、SiteMinder ポリシー サーバ、管理 UI をホストしている Web サーバを再起動します。
2. SSO クライアントから保護されているブラウザにアクセスし、SiteMinder ポリシー サーバで保護されているリソースの URL を入力します。

SiteMinder による再認証なしでリソースにアクセスできるはずです。

CA SSO から SiteMinder へのシングル サインオンの設定

SiteMinder には CA SSO から SiteMinder へのシングル サインオン機能があります。

CA SSO から SiteMinder へのシングル サインオンを有効にする方法

SiteMinder ポリシー サーバの設定手順

1. 管理 UI を使用した smauthetsso カスタム認証方式の設定
2. SiteMinder Web エージェントでリソースを保護するために、管理 UI を使用してドメイン、レルム、ルールを作成します。

詳細については、「ドメイン、レルムのリソースのグループ化、ルール」を参照してください。

3. リソースを保護するために smauthetsso カスタム認証方式を設定します。

WAC Web エージェント検証手順

1. DomainCookie=<domain> を設定し、WAC Web エージェントの webagent.ini ファイルにドメインを設定します。

注: ドメインに指定する値は CA SSO および SiteMinder Web エージェントで同じである必要があります。ファイルは WAC Web エージェント マシンの C:\Program Files\CA\WebAccessControl\WebAgent\webagent.ini にインストールされます。

2. 以下の Web サーバと認証方法の設定が webagent.ini ファイルにあることを確認します。
 - 「Authentication methods」と「The default authentication method」パラメータは、SSO として設定します。
 - WebServerName、PrimaryWebServerName、AgentName、NTLMPATH および Secure は、CA SSO Web Access Control がインストールされているマシンをポイントする必要があります。
 - ServerName 属性は、CA SSO ポリシー サーバがインストールされているマシンの IP アドレスをポイントする必要があります。
 - WAC Web エージェントの設定の詳細については、CA SSO のマニュアルを参照してください。

注: WAC Web エージェントの設定の詳細については、WAC のマニュアルを参照してください。

SiteMinder Web エージェントまたはセキュア プロキシ サーバの設定手順は次のとおりです。

1. Web エージェントまたはセキュア プロキシ サーバと一緒にインストールされる SSO プラグインを有効にすると、WebAgent.conf ファイルの以下の行からコメント文字 (#) を削除して、SSO クライアント cookie を認証できます。

#LoadPlugin=path_to_eTSSOPlugin.dll | path_to_libetssoplugin.so

注: WebAgent.conf ファイルは以下の場所にあります。

r12.0 SP3 IIS 6.0 または Apache 2.0 Web エージェント

「Web エージェント設定ガイド」を参照してください。

6.0 セキュア プロキシ サーバ

SPS_install_dir\proxy-engine\conf\defaultagent\

SPS_install_dir

セキュア プロキシ サーバのインストール ディレクトリ

2. ポリシー サーバを再起動します。

検証手順の概要

1. WAC Web エージェント、SiteMinder ポリシー サーバ、および 管理 UI をホストする Web サーバを再起動します。
2. WAC Web エージェントによって保護されているリソースにアクセスし、有効なクレデンシャルを渡します。
3. 同じブラウザで、SiteMinder Web エージェントによって保護されたリソースにアクセスします。

SiteMinder による再認証なしでリソースにアクセスできるはずです。

詳細情報:

[smauthetsso カスタム認証方式の設定](#) (P. 810)

smetssocookie Web エージェント アクティブ レスpons属性の設定

smetssocookie Web エージェント アクティブ レスpons属性は、SSO cookie を生成して Web ブラウザに送信します。SSO cookie では、SiteMinder で認証されたユーザが WAC または CA SSO で保護されたコンテンツに再認証を行うことなくアクセスできます。

smetssocookie Web エージェント アクティブ レスpons属性を設定する方法

1. [ポリシー]、[ドメイン]をクリックします。
2. [レスpons]、[レスponsの作成]をクリックします。
[レスponsの作成]ペインが表示されます。
3. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[レスponsの作成:ドメインの選択]ペインが表示されます。
4. 目的のドメインの左にあるラジオ ボタンをオンにし、[次へ]をクリックします。
[レスponsの作成:レスponsの定義]ペインが表示されます。
5. レスponsの名前と説明を定義します。
6. SiteMinder ラジオ ボタンがオンになっており、Web エージェントが[エージェントタイプ]ドロップダウンリストに表示されていることを確認します。
7. [レスpons属性の作成]をクリックします。
[レスpons属性の作成]ペインが表示されます。
8. [オブジェクトの新規作成]が選択されていることを確認し、[OK]をクリックします。
[レスpons属性の作成:<名前>]ペインが表示されます。
9. [属性]ドロップダウンリストをクリックし、WebAgent-HTTP-Cookie-Variable を選択します。
10. [属性の種類]グループ ボックスにある[アクティブ レスpons]ラジオ ボタンをオンにします。
[属性フィールド]グループ ボックスにその他のフィールドが表示されます。
11. [Cookie 名]フィールドに、**SSOTK** と入力します。
12. [ライブラリ名]フィールドに、**smetssocookie** と入力します。

13. [関数名]フィールドに、**GenEtssCookie** と入力します。

注: 関数名は大文字/小文字を区別します。

14. [パラメータ]フィールドで、トークンの順番を以下のように定義します。

<CA_PS_Host_Name>;<SSO_Auth_Host>;<SSO_AuthMethod>;<EncryptionKey>

CA_PS_Host_Name

CA SSO ポリシー サーバのホスト名を指定します。

SSO_Auth_Host

CA ポリシー マネージャで SSO 認証ホスト名を指定します。このホスト名は、Web Access Control Resources、Configuration Resources、Authentication Host の順に選択して指定できます。

必須値: SSO_Authhost

SSO_AuthMethod

SSO 認証方法を定義します。

必須値: SSO

EncryptionKey

CA ポリシー マネージャで、SSO 認証ホスト名で使用するチケット暗号化キーを定義します。

cookie スクリプトが[スクリプト]フィールドに表示されます。

注: 読みやすくするために、トークンの前後にスペースを入力してもかまいません。

15. [サブミット]をクリックします。

[レスポンス属性の作成]タスクが処理のためにサブミットされ、[レスポンスの作成: レスポンスの定義]ペインが再度表示されます。

16. [完了]をクリックします。

レスポンスの作成タスクが処理のためにサブミットされます。タスクが完了して作成されたレスポンスは、OnAuthAccept ルールに追加できます。

smauthetsso カスタム認証方式の設定

CA SSO SiteMinder (smauthetsso) 認証方式では、SiteMinder ポリシー サーバで CA SSO 認証クレデンシアルを検証できるため、CA SSO/WAC 環境ですでに認証済みのユーザは SiteMinder に対して再度認証を受ける必要がありません。このカスタム認証方式では、ログイン クレデンシアルとして受け取った CA SSO cookie を CA SSO ポリシー サーバで検証してユーザ名を抽出し、そのユーザ名が SiteMinder ユーザ ストアにあることを確認します。この認証方式は、cookie、cookieorbasic または cookieorforms モードで設定できます。

CA SSO ポリシー サーバの 1 つが何らかの理由で失敗した場合に別の CA SSO ポリシー サーバにフェイルオーバーするよう設定が可能です。フェイルオーバーを設定するには、[認証方式] ダイアログ ボックス上の[方式のセットアップ] グループ ボックスのパラメータ フィールドで、CA SSO ポリシー サーバのカンマ区切りリストを指定します。

注: 以下の手順では、新しいオブジェクトを作成しているとします。また、既存のオブジェクトのプロパティをコピーしてオブジェクトを作成することもできます。詳細については、「ポリシー サーバ オブジェクトの複製」を参照してください。

認証方式を設定する方法

1. [インフラストラクチャ]-[認証]をクリックします。
2. [認証方式]-[認証方式の作成]をクリックします。
[認証方式の作成] ペインが開きます。
3. [OK]をクリックします。

認証方式の設定が開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

4. [認証タイプのスタイル] リストから[カスタム テンプレート]を選択します。
認証方式固有のフィールドとコントロールが開きます。

注: それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ]をクリックします。

5. [ライブラリ] フィールドに **smauthetsso** を入力します。
6. [秘密キー] と [秘密キーの確認入力] フィールドに、CA SSO ポリシー サーバ管理者のパスワードを入力して確認します。

7. [パラメータ]フィールドで、トークンの順番を以下のフォーマットで定義します。

Mode [*;* <*Target*>] *;* *AdminID* *;* *CAPS_Host* *;* *FIPS_Mode* *;* *Identity_File*

注: トークンはセミコロンで区切ります。読みやすくするために、各トークンの前後にスペースを入力してもかまいません。

例: `cookie ; SMPS_sso ; myserver.myco.com ; 0 ; /certificates/def_root.pem`

例: `cookieorforms ; /siteminderagent/forms/login.fcc ; SMPS_sso ; myserver.myco.com ; 1 ; /certificates/def_root.pem`

モード

認証方式で受け取るクレデンシャルのタイプを指定します。選択肢は、`cookie`、`cookieorbasic`、`cookieorforms` です。`cookie` は CA SSO cookie し受け取れないように指定します。`cookieorbasic` は、CA SSO cookie が渡されない場合にログイン名とパスワードの判断で基本認証方式を使用することを指定します。`cookieorforms` は、CA SSO cookie が渡されない場合にログイン名とパスワードの判断でフォーム認証方式を使用することを指定します。

ターゲット

HTML フォーム認証方式で使用する .fcc ファイルのパス名を指定します。

注: この値は `cookieorforms` モードの場合のみ必要です。

AdminID

CA SSO ポリシー サーバで使用する CA SSO ポリシー サーバ管理者のユーザ名を指定します。`SiteMinder` は、CA SSO ポリシー サーバへの認証で CA SSO cookie の検証をリクエストする際、この管理者のユーザ名とパスワードを使用します。

CAPS_Host

CA SSO ポリシー サーバがあるホストの名前を指定します。

FIPS_Mode

ポリシー サーバを動作させる FIPS モードを指定します。ゼロ (0) は非 FIPS モードを指定します。1 (1) は FIPS モードを指定します。

Identity_File

CA SSO ID ファイルへのパスを指定します。ポリシー サーバは、このファイルを使用して CA SSO ポリシー サーバと通信します。

8. [サブミット]をクリックします。

認証方式は保存され、場合によってはレルムに割り当てられます。

詳細情報:

[HTML フォーム認証方式](#) (P. 324)

付録 D: Enterprise Log Manager 統合

Enterprise Log Manager (ELM) は、SiteMinder と ELM の統合を設定する方法について詳しく説明された SiteMinder コネクタ ガイドを提供します。どのガイドを使用するかは、SiteMinder が監査情報をテキストファイル (smaccess.log) または ODBC データベース内のどちらに保存するように設定されているかによって決まります。

ELM コネクタ ガイドを特定する方法

1. [CA Enterprise Log Manager Integration Matrix](#) にアクセスします。
2. Identity Management をクリックします。Identity Management は [Product Integrations] の下にあります。

SiteMinder コネクタ ガイドは ELM で使用されるログ センサのタイプに基づきます。

3. 以下のいずれかを実行します。
 - SiteMinder が監査情報をテキスト ファイル内に格納する場合、ファイル ログ センサ用のコネクタ ガイドを使用します。
 - SiteMinder が監査情報を ODBC データベース内に格納する場合、ODBC ログ センサ用のコネクタ ガイドを使用します。

これらの各ガイドは、必要なコネクタを作成する際に ELM 管理 UI から参照できます。コネクタの作成時にこれらのガイドにアクセスするには、[ヘルプ] をクリックしてください。

付録 E: RADIUS サーバとしてのポリシー サーバの使用方法

このセクションには、以下のトピックが含まれています。

[RADIUS サーバとしてのポリシー サーバの使用](#) (P. 815)

[RADIUS クライアント/サーバ アーキテクチャ](#) (P. 816)

[ポリシー サーバを使用した RADIUS 認証の動作](#) (P. 816)

[RADIUS 環境のポリシー](#) (P. 818)

[RADIUS ポリシー ドメインのレスポンス](#) (P. 823)

[RADIUS 環境への SiteMinder の展開](#) (P. 832)

[RADIUS デバイスの保護に関するガイドライン](#) (P. 833)

[同機種 RADIUS 環境でユーザを認証する方法](#) (P. 834)

[1 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザの認証](#) (P. 836)

[2 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法](#) (P. 841)

[RADIUS エージェントグループの概要](#) (P. 846)

[RADIUS エージェントグループのセットアップ](#) (P. 847)

[RADIUS レスポンスのグループ化](#) (P. 848)

[RADIUS のトラブルシューティングとテスト](#) (P. 849)

RADIUS サーバとしてのポリシー サーバの使用

RADIUS (リモート認証ダイヤルイン ユーザ サービス) は、NAS (ネットワーク アクセス サーバ) デバイスと RADIUS 認証サーバの間で、セッション認証と設定情報を交換するためのプロトコルです。ポリシー サーバは、RADIUS 認証サーバとして使用できます。

RADIUS プロトコルは、次の機能を提供する NAS デバイスで頻繁に使用されます。

- インターネット サービス プロバイダ (ISP) のプロキシ サービス
- ファイアウォール
- 企業のダイヤルアップサービス

RADIUS クライアント/サーバ アーキテクチャ

RADIUS は、NAS デバイスが提供する通信テクノロジーと、認証サーバが提供するセキュリティテクノロジーを切り離すことによって、セキュリティを簡素化します。RADIUS のセキュリティ機能は、分散クライアント/サーバアーキテクチャを使用して、ネットワークへのリモートアクセスとネットワークサービスを保護します。ポリシー サーバは、RADIUS サーバです。RADIUS クライアントは、NAS デバイスです。

NAS デバイスは、次のいずれかを実行します。

- ダイアルイン プロトコル (SLIP、PPP など) をサポートし、RADIUS 認証サーバを使用してユーザを認証し、ユーザをネットワークにルーティングします。
- ネットワークへのファイアウォール経由の直接接続をサポートし、RADIUS 認証サーバを使用してユーザを認証し、ネットワークへのアクセス権を付与します。

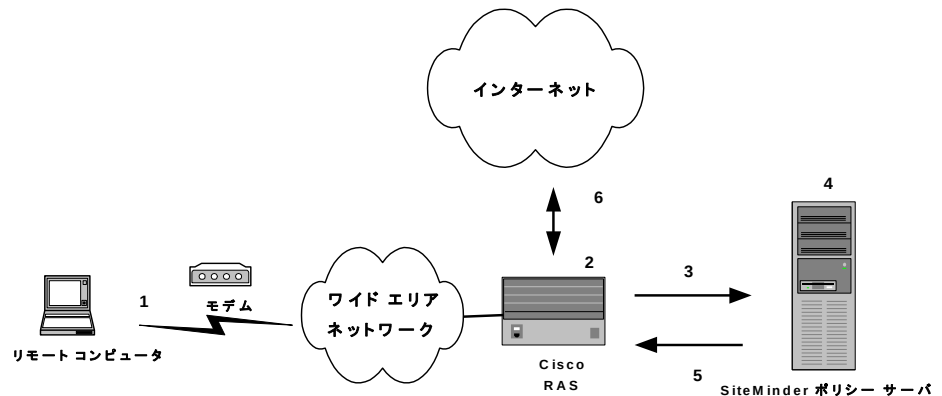
ポリシー サーバは、この章の説明どおりに設定することで、RADIUS 認証サーバとして使用できます。ポリシー サーバは RADIUS サーバとして、RADIUS 認証方式と定義済みユーザディレクトリを使用して、RADIUS ユーザを認証します。

注: RADIUS アカウンティングを使用するには、RADIUS アカウンティング サーバを別途設定する必要があります。ポリシー サーバは、アカウンティング サーバに ACK レスポンスを送信することによって、NAS デバイスの要求を満たします。ただし、アカウンティング情報はファイルに記録できます。

ポリシー サーバを使用した RADIUS 認証の動作

ポリシー サーバは、NAS デバイスとの一連の通信によってユーザを認証します。SiteMinder がユーザを認証すると、NAS は、そのユーザに該当するネットワークサービスへのアクセス権を付与します。

次の図は、この認証プロセスを示しています。



1. モデムからダイヤルインするユーザは、インターネットへの接続を可能にする Cisco RAS (NAS デバイス) への接続を開こうとします。
2. RAS は、ユーザを認証するために RADIUS ユーザプロファイルを使用する必要があるかどうかを判別します。
3. RAS は、ポリシー サーバにユーザ接続リクエストを送信します。
4. ポリシー サーバは、次のいずれかの方法でユーザ名とパスワードを取得します。
 - パスワード認証プロトコル (PAP) を使用して認証します。
PAP は、ホストに、双方向ハンドシェイクでホストを識別する簡単な方法を提供する PPP 認証プロトコルです。認証は、リンクの初期確立時にのみ実行され、暗号化を使用しません。
 - チャレンジ ハンドシェイク式認証プロトコル (CHAP) を使用して認証します。
CHAP も、安全な PPP 認証プロトコルです。CHAP は、3 方向ハンドシェイクと暗号化を使用してホストの識別情報を定期的に確認する方法を提供します。認証は、リンクの初期確立時に実行されます。RAS は、接続が実行された後に、いつでも認証プロセスを繰り返すことができます。
 - Security Dynamics ACE/Server または Secure Computing SafeWord サーバを使用して認証します。

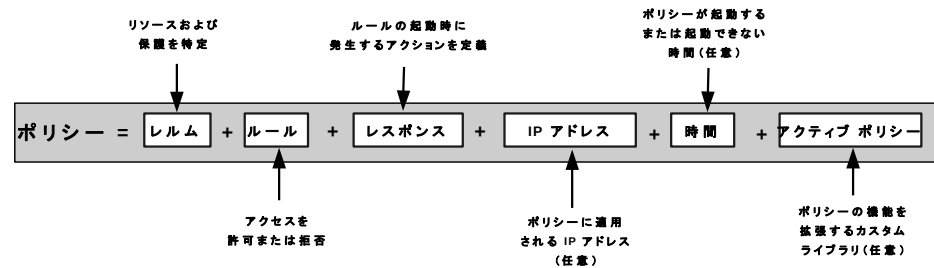
5. ポリシー サーバは、認証レスポンスを RAS に送信します。
 6. 次のいずれかが実行されます。
 - 認証が失敗すると、RAS は接続を拒否します。
 - 認証が成功すると、RAS は、ユーザの認証に対する SiteMinder のレスポンスから属性のリストを受け取ります。属性は、ユーザのネットワークセッションを設定するユーザ プロファイルとして使用されます。
- RAS は、ポリシー サーバに、セッションが開始されたこと、およびいつセッションが終了するかを通知します。

RADIUS 環境のポリシー

SiteMinder の RADIUS ポリシーは、RADIUS エージェントによって実施され、次の要素とバインドして作成されます。

- 認証ルール
- レスポンス
- ユーザまたはユーザグループ
- 必要に応じて、IP アドレス、時刻、およびアクティブなポリシー

次の図に、ポリシーの基本構造を示します。

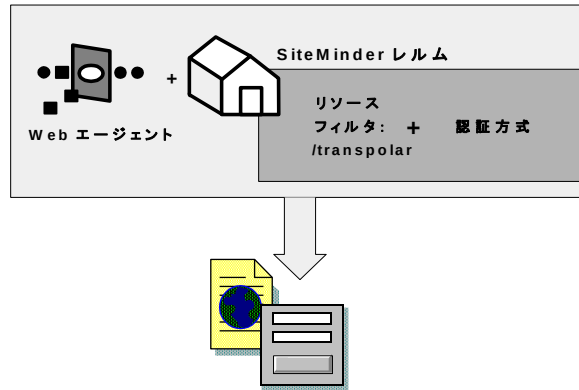


RADIUS ポリシーは、SiteMinder エージェントが使用するポリシーに含まれる要素と同じ要素で構成されていますが、RADIUS エージェントは、これらのコンポーネントを異なる方法で解釈します。ルール、レلم、およびレスポンスは、次の表に示すように、異なる機能を実行します。

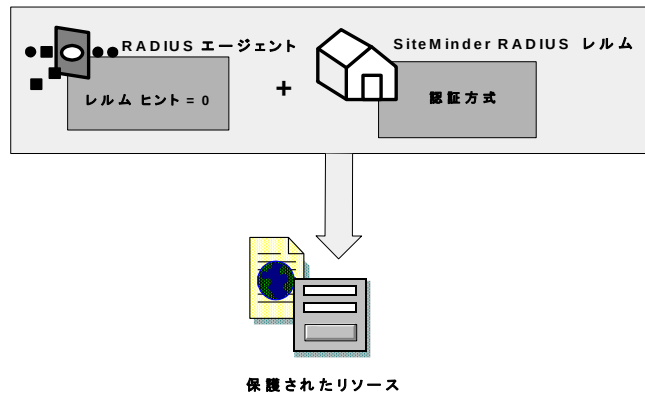
ポリシーのコンポーネント	RADIUS ポリシーでの機能	SiteMinder エージェント ポリシーでの機能
レلم	■ エージェントの識別 ■ 認証方式の識別 ■ セッションタイムアウトの定義	■ リソースフィルタ (SiteMinder エージェントが管理するドメイン内のディレクトリ) の定義 ■ エージェントの識別 ■ 認証方式の識別 ■ リソースのステータス (保護されているかいないか) の定義 ■ 処理するイベント (認証または許可) の識別 ■ セッションタイムアウトの定義
ルール	■ 認証のみ ■ アクセスの許可または拒否 ■ 時間制限またはアクティブルール制限の定義	■ リソースフィルタの定義 ■ アクション (Web エージェントアクション、許可イベント、または認証イベント) の定義 ■ アクセスの許可または拒否 ■ 許可および認証 ■ 時間制限またはアクティブルール制限の定義
レスポンス	■ 認証イベントに対して返す値の定義	■ 許可イベントに対して返す値の定義 ■ 認証イベントに対して返す値の定義 ■ 許可拒否イベントに対して返す値の定義 ■ 認証拒否イベントに対して返す値の定義

RADIUS リソースと非 RADIUS リソース

RADIUS ポリシーの要素は、RADIUS 環境でのリソースの識別方法に基づいて、部分的に異なる方法で扱われます。SiteMinder エージェント環境では、特定のリソースは、レルムの定義に含まれるリソースフィルタを使用して識別されます。リソースフィルタは、リソースのディレクトリ位置を識別します。次の図に示すように、レルムの定義も、Web エージェントと認証方式を識別します。



次の図に示すように、RADIUS 環境では、保護されているリソースが、別の場所に保存されます。フィルタを使用してリソースを識別するレルムの代わりに、RADIUS エージェントは、レルム ヒントを使用してリソースを識別します。レルム ヒントは、ユーザを認証するドメインをポリシー サーバが確立することを可能にする属性です。レルムヒントは、エージェントが保護する特定のレルムを識別するか、エージェントが NAS デバイス全体を保護しなければならないことを示します。



レルム ヒントの使用

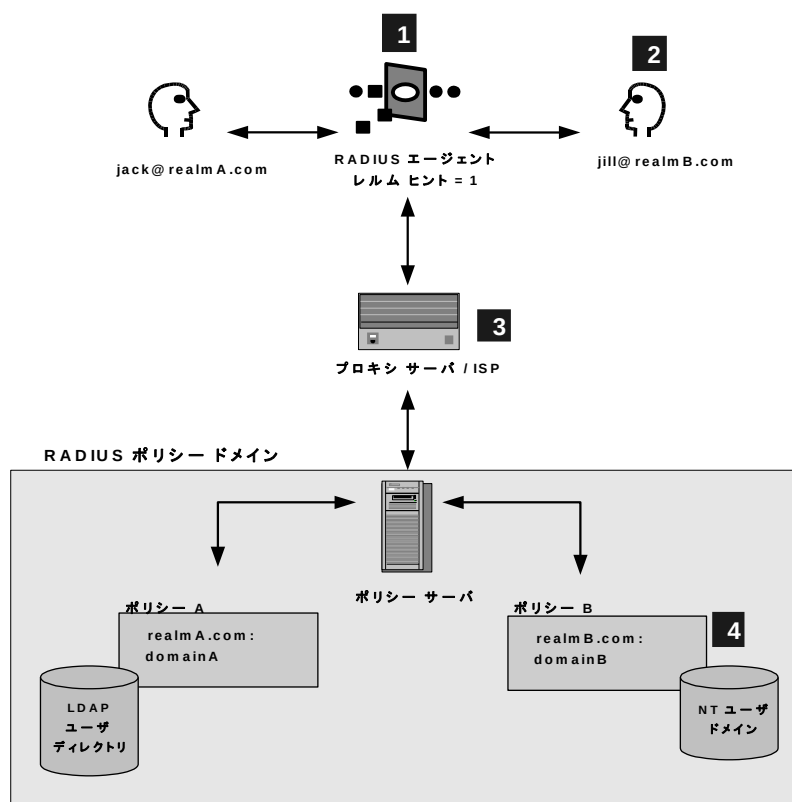
RADIUS エージェントは、異なるドメイン (domainA や domainB など) のユーザを認証する必要がある NAS デバイスをどのような方法で保護するのでしょうか。レルム ヒントは、ユーザを認証する適正なドメインを SiteMinder で判断できるようにする RADIUS 属性です。RADIUS エージェントに、次のいずれかのレルム ヒント値を提供する必要があります。

- 0 -- (デフォルト) ポリシー ドメインにレルムが 1 つしかなく、ヒントが不要であることを示します。レルムは、NAS デバイスに直接結び付けられます。
- 1 -- (RADIUS ユーザ名属性) SiteMinder は、この属性のユーザ名からレルム名を解析し、後で説明するように、関連付けられたドメインを探します。
- ドメインの実際の名前を含む属性。この属性は、すべての NAS デバイスに使用できるわけではありません。詳細については、使用している NAS デバイス製品のマニュアルを参照してください。

レルム ヒントが 1 に設定されている場合、レルム名はユーザ名属性から解析されます。user_name-realm のセパレータは、「@」または「/」である必要があります。

- セパレータが「@」の場合は、「@」の後の要素がレルム名です。たとえば、jack@realmA.com の場合、レルムは realmA.com です
- セパレータが「/」の場合は、「/」の前の要素がレルム名です。たとえば、x5/jack の場合、レルムは x5 です

次の図と説明は、プロキシサーバがユーザを認証する適正な SiteMinder ドメインを判断する方法を示しています。



1. 1 つの RADIUS エージェントが、両方の SiteMinder ドメインを保護します。
RADIUS エージェントに設定されているレルムヒント値は、1 です。
2. Jill が ISP のプロキシサーバにアクセスしようとする、RADIUS エージェントはそのリクエストをインターセプトし、Jill のユーザ名属性 jill@realmB.com をポリシーサーバに転送します。
3. ポリシーサーバが、ユーザ名属性から user_name と realm_name を解析します。

例: jill@realmB.com。ここで、jill は user_name、realmB.com は realm_name です。

ポリシーサーバが、realm_name に関連付けられているドメインを識別します。realmB.com に関連付けられているドメインは、domainB です。

4. ポリシーサーバが、user_name を適切なディレクトリで認証します。
user_name jill は、ポリシー B: realmB.com:domainB 用に定義された NT ユーザドメインで認証されます。

RADIUS ポリシードメインのレスポンス

ユーザを認証する場合、SiteMinder レスポンスを使用して、RADIUS 属性を NAS デバイスに返すことができます。ユーザが認証されると、属性は、セッションの特徴を設定し、認証されたユーザのユーザ プロファイルを定義します。ユーザ プロファイルは、NAS デバイスによって使用されます。たとえば、レスポンス内の属性を使用して、RADIUS ユーザ セッションの時間制限を定義できます。

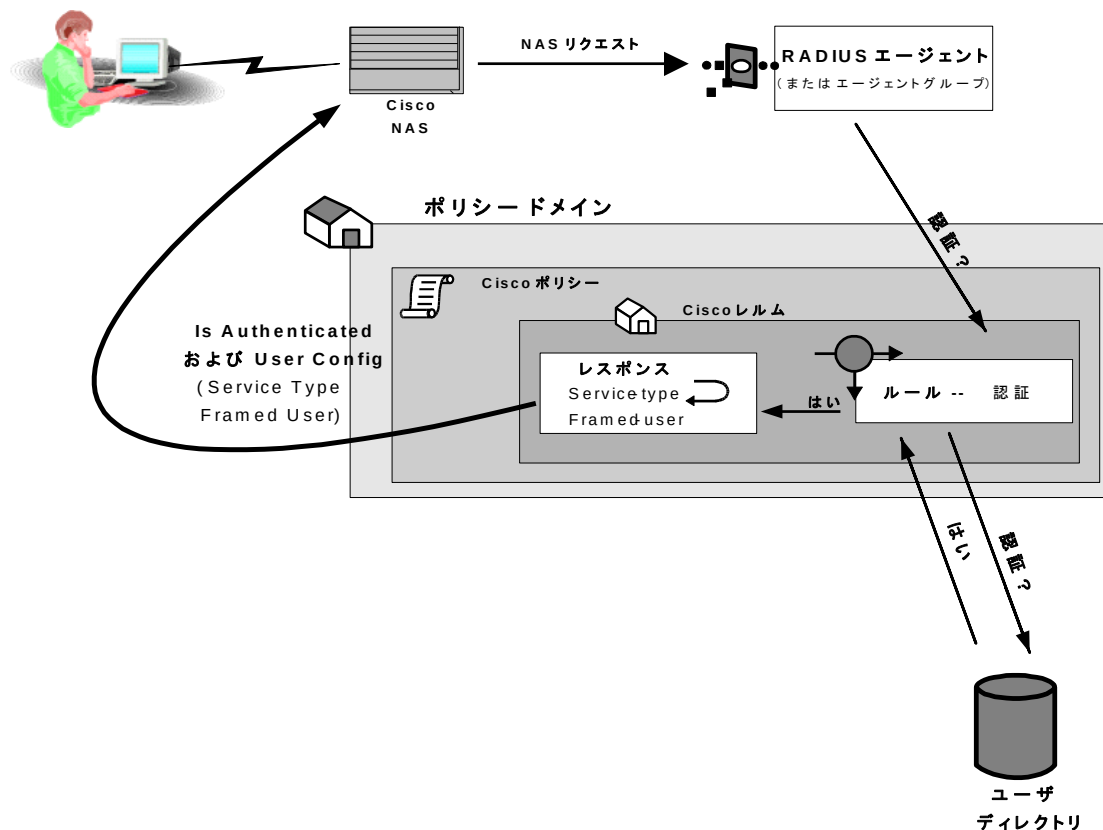
権限をユーザに割り当てるユーザ プロファイル情報を、レスポンスを使用して NAS デバイスに提供することができます。たとえば、同じリソースへのアクセスがあるユーザには制限なしで、また別のユーザには制限ありで許可することができます。本来 RADIUS は認証メカニズムにすぎませんが、レスポンスをこのように使用することで、ユーザに許可を与えることができるようになります。

注: NAS が認証だけを指定する場合、デフォルトで、SiteMinder は RADIUS 属性を返しません。NAS が認証のみを指定するときに RADIUS 属性を返すには、「[常に RADIUS 属性を返す SiteMinder の設定 \(P. 826\)](#)」にある手順に従ってください。

レスポンスの機能

RADIUS レスポンスは、認証ルールとペアになっています。ルールによってユーザが認証されると、RADIUS レスポンスが発行されます。ルールによってユーザが認証されないと、レスポンスは発行されません。

レスポンスが発行されると、ポリシー サーバは、レスポンスに含まれる属性を NAS デバイスに送信します。次の図に示すように、この情報はユーザのセッションのカスタマイズに使用されます。



属性タイプ

レスポンスでは、次の属性を使用できます。

- ユーザ属性
- DN 属性
- アクティブ レスポンス属性
- RADIUS 属性

ユーザ属性

この属性は、LDAP、Windows NT、または ODBC ユーザ ディレクトリのユーザに関する情報を返します。ユーザ属性は、ユーザディレクトリから取得され、RADIUS デバイスの動作を変更するために使用できます。

DN 属性

この属性は、ユーザに関連付けられている LDAP ディレクトリ オブジェクトに関するプロフィール情報を返します。たとえば、DN 属性は、ユーザのグループや組織単位 (OU) などの LDAP オブジェクトに関する情報を返す場合があります。

アクティブ レスポンス属性

この属性は、SiteMinder 許可 API を使用して開発されたカスタム ライブラリからの値を返します。SiteMinder がカスタム ライブラリの関数を呼び出すと、アクティブ レスポンスが生成されます。

RADIUS 属性

この属性は、以下のエージェントタイプ属性によって定義される値を返します。

RADIUS

RFC (Request for Comment) 2138 の RADIUS プロトコル仕様で定義されている、一般 RADIUS 属性。この属性の識別子は、1 ～ 25 と 27 ～ 63 です。この属性のいくつかは、同じレスポンスで複数使用される場合があります。

どの RADIUS エージェントタイプも、一般 RADIUS 属性を含むレスポンスを返すことができます。

RADIUS 拡張

NAS デバイスのディレクトリ ファイルに定義された属性。この属性は、一般 RADIUS 属性では定義されない、使用している NAS デバイス固有の値を定義します。この属性の固有識別子は、一般 RADIUS 属性のために予約されている範囲外のもの (64 以降) です。たとえば、Lucent は、*Ascend-Disconnect-Cause* と呼ばれる拡張 RADIUS 属性 (識別子 195) を提供します。

拡張 RADIUS 属性のベンダータイプに適合するエージェントタイプだけが、その属性を使用できます。たとえば、Shiva エージェントタイプは、Shiva 用の拡張 RADIUS 属性を使用できますが、Cisco エージェントタイプは、この拡張属性をレスポンスで使用できません。レスポンスで使用される拡張属性は、RADIUS クライアントの辞書ファイルで定義される属性と適合する必要があります。

デフォルトで、SiteMinder は定義済み RADIUS 拡張属性を、これらの属性を使用する Ascend (Lucent) などのいくつかのエージェントタイプ用に提供します。また、必要に応じて、任意の RADIUS エージェントタイプ用に、追加の RADIUS 拡張属性を定義できます。

ベンダー固有

NAS デバイスの辞書ファイルで定義された属性。識別子として 26 を使用します。ベンダー固有属性を使用すると、一般 RADIUS 属性によって提供されない値の属性を定義できます。一部のベンダーは、RADIUS 拡張属性の代わりに、または RADIUS 拡張属性に追加して、ベンダー固有属性を使用します。たとえば、Cisco は、RADIUS 拡張属性を使用しません。ただし、この NAS デバイスは、いくつかのベンダー固有属性 (Cisco-AVpair、Account-Info など) をサポートしています。

ベンダー固有属性を使用して、情報を他のプロトコルに渡すことができます。たとえば、Cisco-AV Pair 属性用のベンダー固有属性を定義して、TACACS+ 情報を TACACS+ サーバに渡すことができます。

ベンダー固有属性は、RADIUS クライアントのベンダータイプに適合するレスポンスでのみ定義できます。

デフォルトで、SiteMinder は、定義済みベンダー固有属性を、これらの属性を使用する Network Associates の Sniffer エージェントタイプなどのいくつかのエージェントタイプ用に提供します。また、必要に応じて、任意の RADIUS エージェントタイプ用に、追加のベンダー固有拡張属性を定義できます。

注: RADIUS 属性の詳細については、*RADIUS プロトコル仕様 RFC (Request for Comment) 2138* を参照してください。

常に RADIUS 属性を返す SiteMinder の設定

一部の NAS デバイスでは、NAS が認証だけを指定している場合でも、RADIUS が Access-Accept のレスポンスを返すと想定しています。NAS が認証だけを指定する場合、デフォルトでは、SiteMinder は RADIUS 属性を返しません。

NAS デバイスに常に RADIUS 属性を返すには、次のパラメータを使用して新しいレジストリ値を作成します。

- 値タイプ--DWORD
- 値名--HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\Authentication\AlwaysReturnRadiusAttrs
- 値データ--0 より大きい数値

注: インストール プログラムは、AlwaysReturnRadiusAttrs のレジストリエントリを作成しません。エントリが作成されて設定されるまで、SiteMinder は、デフォルト値の 0 を使用します。

AlwaysReturnRadiusAttrs を 0 よりも大きい値に設定すると、認証サーバのデバッグ ログに次のメッセージが表示されます。

```
Radius Attributes will be returned regardless of  
RA_SERVICE_TYPE_AUTHENTICATE_ONLY
```

エージェント タイプの属性の作成

レスポンスで属性を使用するには、レスポンスを返すエージェントタイプが属性を使用できるようにする必要があります。属性をエージェントタイプで定義することによって、エージェントタイプが属性を使用できるようになります。多数のエージェントタイプが、ベンダー固有属性および RADIUS 拡張属性で定義済みですが、必要に応じて、拡張 RADIUS 属性、一般 RADIUS 属性、およびベンダー固有属性をエージェントタイプに追加できます。

属性の複数のインスタンスの定義

同じエージェントタイプでベンダー固有属性の複数のインスタンスを定義できます。ベンダー固有属性の複数のインスタンスを定義すると、その属性のインスタンスごとに、NAS デバイスに異なる値を送信できます。たとえば、Cisco エージェントの場合、次のベンダー固有属性を、すべて同じ識別子 (26) を使用して定義できます。

- Cisco-AVpair
- Account-Info
- Command-Code

レスポンス内で属性を使用できる回数を定義する設定は、管理 UI の[エージェントタイプ属性の変更]ペインにあります。

属性を複数回使用できるように設定するには、[アクセスの受け付け]の値を[ゼロまたは複数]に設定する必要があります。

定義する属性のタイプは、レスポンスを返すエージェントのベンダータイプと適合している必要があります。たとえば、ベンダー固有の **Cisco** 属性を返すことができるのは、**Cisco** エージェントだけです。

エージェントによってレスポンスが返される場合、レスポンスのパケット構造は、レスポンスを送信する **RADIUS** エージェントのタイプを反映します。たとえば、**Cisco** エージェントによって返されるレスポンスのパケット構造には、ベンダー ID と文字列の長さが含まれています。

エージェントタイプに属性を定義する方法

1. 管理 UI にログインします。
2. [インフラストラクチャ]-[エージェントタイプ]-[エージェントタイプの変更]をクリックします。
[エージェントタイプの変更]ペインが開きます。
3. 検索条件を指定して[検索]をクリックします。
検索条件に一致するエージェントタイプのリストが表示されます。
4. エージェントタイプを選択して[選択]をクリックします。
[エージェントタイプの変更: 名前]ペインが開きます。
5. [エージェントタイプ属性]グループ ボックスの[エージェントタイプ属性の作成]をクリックします。
[エージェントタイプ属性の作成]ペインが開きます。
6. オブジェクトの新規作成が選択されていることを確認し、[OK]をクリックします。
[エージェントタイプ属性の作成: 名前]ペインが開きます。
7. [一般]グループ ボックスの各フィールドに、エージェントタイプの名前および説明を入力します。
8. [RADIUS タイプ]リストで[RADIUS]、[RADIUS 拡張]、または[ベンダー固有]を選択します。
9. [データ型]リストで、属性に含まれるデータの型を選択します。

10. [識別子]フィールドで、以下の属性識別子の 1 つを入力します。

■ 一般 RADIUS

属性識別子は、RADIUS プロトコル仕様の中で定義されています。一般 RADIUS 属性の識別子を上書きすることはできますが、通常は、事前に定義された一般 RADIUS 属性の定義を保持する必要があります。この定義が、RADIUS 仕様 (RFC 2138) に適合しているためです。

例: Callback-Id 変数の属性を作成するには、[識別子]フィールドに「20」と入力します。

■ RADIUS 拡張

属性識別子は、ベンダードキュメントの中で定義されています。

例: Ascend-Callback 変数の属性を作成するには、[識別子]フィールドに「246」と入力します。

■ ベンダー固有

属性識別子は 26 です。

例: Cisco エージェントの属性を作成して、このエージェントが TACACS+ を使用できるようにするには、[識別子]フィールドに「26」と入力します。

注: 属性識別子の詳細については、使用している RADIUS ベンダーのマニュアルを参照してください。

11. [RADIUS の動作]グループ ボックスの各フィールドに対する RADIUS コードを選択します。RADIUS コードは次のとおりです。

拒否

レスポンスで属性を使用することはできません。

ゼロまたは 1

同じレスポンス内で、属性のインスタンスを返すことができないか、または 1 つ返すことができます。この値を選択して、レスポンスで属性を使用した場合は、レスポンスで属性を使用した後に、その属性が[属性]リストから削除されます。

ゼロまたは複数

同じレスポンス内で、属性のインスタンスを返すことができないか、または複数返すことができます。

1 のみ

レスポンスで、属性のインスタンスを 1 つだけ返す必要があります。この値を選択して、レスポンスで属性を使用した場合は、レスポンスで属性を使用した後に、その属性が[属性]リストから削除されます。

[RADIUS]グループ ボックスにあるフィールドは次のとおりです。

アクセスのリクエスト

ユーザが特定の NAS へのアクセスを許可されているかどうかを判定するための情報を提供します。[アクセスのリクエスト]のパケットは、そのユーザのためにリクエストされている任意の特別なサービスに関する情報も提供します。

アクセスの受け付け

ユーザへのサービスの提供を開始するために必要な特定の設定情報を提供します。

注: レスポンスで属性を使用するには、[アクセスの受け付け]の値を[ゼロまたは 1]、[ゼロまたは複数]、または[1 のみ]に設定する必要があります。

アクセスの拒否

受信した属性の値のいずれかを受け付けることができない場合に、情報を送信します。このコードは、応答メッセージとしてよく使用されます。

アクセスのチャレンジ

NAS デバイスがチャレンジ/レスポンスに対して設定されている場合に、情報を送信します。

アカウントिंग リクエスト

提供されているサービスのタイプ、およびサービスの提供先であるユーザの説明です。

アカウントング レスポンス

アカウントングのリクエストが正常に記録された場合に、情報を送信します。RADIUS のアカウントング レスポンスには、属性が含まれている必要はありません。

12. データ型が数値である場合は、[値]グループ ボックスの[作成]をクリックします。
13. 属性の記号名を[記号名]フィールドに入力し、属性の実際の数値を[数値]フィールドに入力して、[OK]をクリックします。
[エージェント タイプ属性の変更]ペインが再び開き、属性名前/値ペアが[値]グループ ボックスに追加されます。
注: 複数の属性名/値ペアを作成するには、手順 12 および 13 を繰り返します。記号名を値にマッピングすると、名前を憶えるだけで済みます。
14. [サブミット]をクリックします。
[エージェント タイプの変更]ペインが再び開き、エージェント タイプ属性が[エージェント タイプ属性]グループ ボックスに追加されます。
15. [サブミット]をクリックします。
[エージェント タイプの変更]タスクが、処理のためにサブミットされます。
注: タスクが完了し、このエージェント タイプのためのレスポンスを作成するときに、エージェント タイプに追加したエージェント タイプ属性を属性リストから選択できます。

既存属性の変更

作成した属性や RADIUS エージェント用に定義済みの属性を変更できます。たとえば、Ascend エージェントタイプ用に定義済みの Ascend-PPP-Address 属性を変更できます。

注: 既存の属性を変更した場合、その属性をすでに使用しているレスポンス内の属性は、動的には更新されません。レスポンスで属性が使用されている場合は、更新した属性を使用してレスポンスを作成し直してください。

すべての RADIUS エージェントタイプは、RFC 2138 で定義されているように、一般 RADIUS 属性を使用するように事前に設定されています。これらの属性は、各 RADIUS エージェントタイプによって使用されるように提供されています。

重要: 一般属性を上書きしたり、一般 RADIUS エージェントで新しい属性を定義したりすると、変更内容が、すべての RADIUS エージェントに適用されます。たとえば、一般 RADIUS エージェントの Filter ID 属性を変更した場合は、他のすべての RADIUS エージェントタイプ (Cisco、Shiva、Livingston、Ascend、Checkpoint など) も変更されます。

エージェントタイプ属性を変更する方法

1. 管理 UI にログインします。
2. [インフラストラクチャ]タブからエージェントを選択します。
3. [エージェントタイプの変更]をクリックします。
4. [検索]をクリックします。
5. エージェントタイプを選択して[選択]をクリックします。
[エージェントタイプの変更]ペインが開きます。
6. 属性の左側にある[編集]ボタンをクリックして、エージェントタイプの値を変更します。
7. [サブミット]をクリックして変更内容を保存します。

RADIUS 環境への SiteMinder の展開

SiteMinder は、さまざまな異なる RADIUS 環境で認証サービスを提供するように設定できます。

- 単一の NAS デバイス (Cisco RAS など) と単一のユーザディレクトリで構成される同機種環境。この環境については、「[同機種 RADIUS 環境でのユーザの認証方法](#)」で説明されています。
- 複数の NAS デバイス (Checkpoint ファイアウォールと Cisco RAS など) と単一のユーザディレクトリで構成される異機種環境。この環境については「[1 つのユーザディレクトリを持つ異機種 RADIUS 環境でのユーザの認証 \(P. 836\)](#)」で説明されています。
- 複数の NAS デバイス (Checkpoint ファイアウォールと Cisco RAS など) と複数のユーザディレクトリで構成される異機種環境。この環境については、「[2 つのユーザディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法 \(P. 841\)](#)」で説明されています。

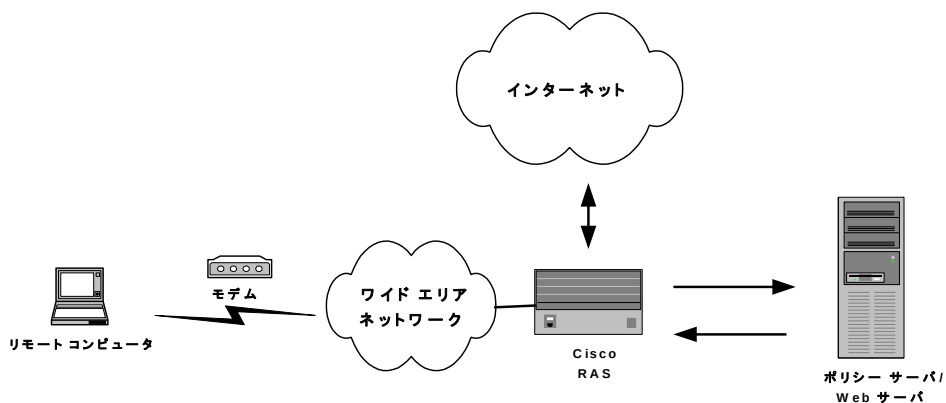
RADIUS デバイスの保護に関するガイドライン

SiteMinder を RADIUS 環境に展開する際は、以下のガイドラインに注意してください。

- 同じポリシードメイン内のレルム名は、同じでなければなりません。
- 1 つのポリシー内では、1 つのタイプの RAS デバイスだけを保護できます。各ベンダーは、個別の辞書ファイルを使用するため、1 つのポリシーで複数の RADIUS デバイスを保護することはできません。1 つのポリシー内のレスポンスは、返される属性を同じ方法で解釈する必要があります。さまざまな RAS デバイスが存在する異機種環境では、RADIUS デバイスのタイプごとに個別のポリシーを定義してください。
- 1 つのポリシードメイン内で複数のユーザディレクトリを定義できます。複数のユーザディレクトリを定義する場合は、検索順序を指定してください。
- 異なる NAS ベンダー用の RADIUS エージェントを単一の一般 RADIUS エージェントグループに組み合わせて、同じエージェントグループを、各タイプの RADIUS エージェント用の個別のポリシーで使用できます。たとえば、エージェントグループに Shiva エージェントと Cisco エージェントが含まれている場合、Shiva ポリシーと Cisco ポリシーを作成します。各ポリシーには同じルールとレルムが追加されるため、時間の節約になります。ただし、同じルールの各インスタンスに関連付けられるレスポンスは異なります。Cisco ポリシーは Cisco レスポンスを一般ルールに関連付け、Shiva ポリシーは Shiva レスポンスを一般ルールに関連付けます。

同機種 RADIUS 環境でユーザを認証する方法

同機種 RADIUS 環境は、最も簡単に保護できる環境です。1 つのポリシーのみを使用して、RADIUS デバイスを保護できます。このタイプの環境には、次の図に示すように、1 つの RADIUS デバイス (Cisco RAS など) と 1 つのユーザ ディレクトリのみが含まれています。



同機種 RADIUS 環境で SiteMinder をセットアップする方法

1. 次の手順で、システムを設定します。
 - a. 「[RADIUS エージェントの設定](#) (P. 157)」にある説明に従って、RADIUS エージェントを定義します。
 - b. 「ユーザ ディレクトリのセットアップ」にある説明に従って、RADIUS ユーザの認証に使用するユーザ ディレクトリをセットアップします。
 - c. 必要に応じて、管理者を定義し、認証方式を変更することもできます。

2. 次の手順で、ポリシードメインを設定します。
 - a. 「認証方式の作成」にある説明に従って、RADIUS 認証方式 (CHAP または PAP) を作成します。
 - b. 「[RADIUS エージェントで保護されたレルムの設定](#) (P. 489)」にある説明に従って、RADIUS エージェントおよび RADIUS 認証方式を識別するレルムを定義します。
 - c. 「[認証のルールの設定](#) (P. 505)」にある説明に従って、認証されたユーザを、RADIUS エージェントで保護されたレルムにアクセス可能にするためのルールを定義します。
 - d. 「[レスポンスの設定](#) (P. 528)」および「[RADIUS エージェントレスポンス属性](#) (P. 527)」にある説明に従って、NAS デバイスにユーザ プロファイルを提供するレスポンスを定義します。このレスポンスは、レスポンス属性を使用してセッションの特性も設定します。
 - e. 「[ポリシーの設定](#) (P. 553)」にある説明に従って、ルールとレスポンスにユーザ ディレクトリをバインドするポリシーを作成します。

ユーザ ディレクトリのセットアップ

使用している Windows NT または UNIX プラットフォームでサポートされている任意のユーザディレクトリを使用して RADIUS ユーザを認証できます。

ユーザディレクトリ情報にユーザの権限に関する情報が含まれている場合は、ユーザ属性を使用してレスポンスを作成できます。ユーザ属性が RADIUS デバイスに返されると、属性は、ユーザセッションを設定するために使用されます。

次のディレクトリを使用できます。

- ODBC 対応データベース
- NTドメイン
- Netscape または NDS LDAP

ポリシードメインのセットアップ

ポリシードメインは、RADIUS ユーザの名前、ドメインを変更できる管理者の名前、および RADIUS エージェントが保護しているレルムを含む 1 つまたは複数のユーザディレクトリを識別する必要があります。

認証方式の作成

次の認証方式を使用できます。

- パスワード認証プロトコル (PAP)

PAP は、ホストに、双方向ハンドシェイクでホストを識別する簡単な方法を提供する PPP 認証プロトコルです。認証は、リンクの初期確立時にのみ実行され、暗号化を使用しません。

- チャレンジハンドシェイク式認証プロトコル (CHAP)

CHAP も、安全な PPP 認証プロトコルです。CHAP は、3 方向ハンドシェイクと暗号化を使用してホストの識別情報を定期的に確認する方法を提供します。認証は、リンクの初期確立時に実行されます。RAS は、接続が実行された後に、いつでも認証プロセスを繰り返すことができます。

- Security Dynamics ACE/Server または Secure Computing SafeWord サーバ

注: 認証方式の作成の詳細については、このガイドの「認証方式」の章を参照してください。

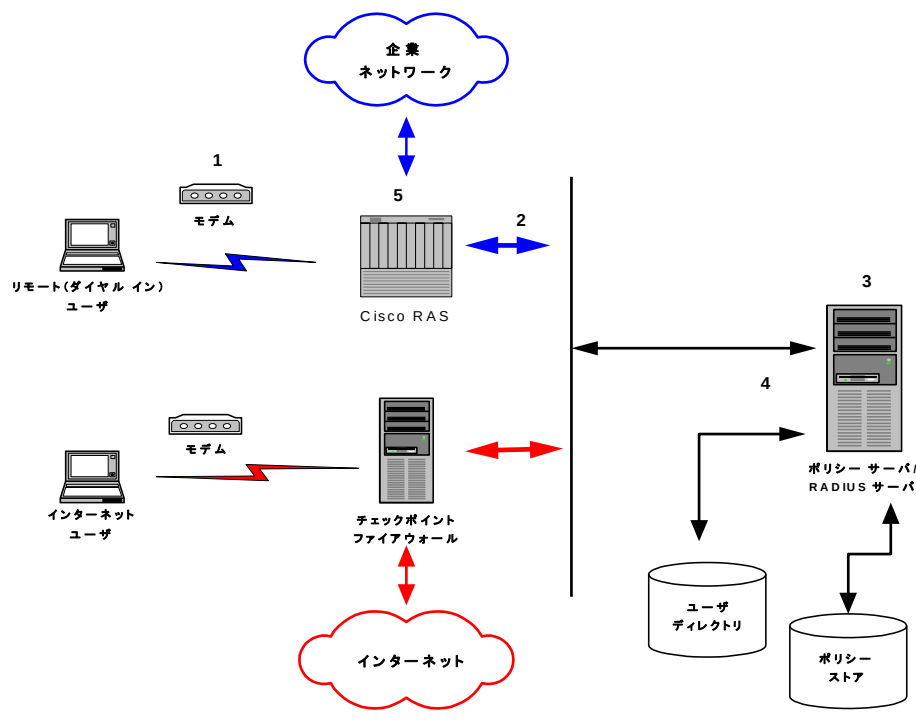
1 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザの認証

複数の NAS デバイスで管理される複数のレルムを含む RADIUS 環境では、ポリシー サーバの展開が強力、かつ複雑になります。この場合、ポリシー サーバを、同時に複数の RADIUS クライアントに対応する RADIUS 認証サーバとして使用できます。

異機種構成を使用する利点は、各 RADIUS クライアントに対して同じ RADIUS 認証サーバ(つまり、ポリシー サーバ)を使用することによって、作業時間を節約できることです。

1つのディレクトリを持つ異機種環境でユーザが認証される仕組み

以下の図は、異機種構成の例です。



この図のネットワークポロジでは、ポリシーサーバが、2つのNASデバイス（Cisco RASとCheckpointファイアウォール）のユーザを認証します。ポリシーサーバは、1つのユーザディレクトリを使用してユーザを認証します。

各NASデバイスは、独自のRADIUSエージェントを備えており、これらのエージェントはレルムヒントを使用して設定されています。ポリシーサーバは、ユーザ認証のリクエストを受信すると、RADIUSエージェントのレルムヒントを使用して、認証されたユーザがアクセスできるリソース（ドメイン）を判断します。

使用されているユーザディレクトリが1つの場合の認証プロセスは、以下のとおりです。

1. リモートユーザがモデムからダイヤルインすると、Cisco RASは、ユーザを認証するためにRADIUSユーザプロファイルを使用する必要があるかどうかを判別します。
2. RASは、ポリシーサーバにユーザ接続リクエストを送信します。

3. ポリシー サーバは、RAS 用に定義されたポリシーを有効化し、Cisco RAS に関連付けられた RADIUS エージェントは、次の作業を実行します。
 - a. レルム ヒントによるユーザのドメインの判断
 - b. エージェント用に設定された認証方式を使用して、ユーザの名前とパスワードを取得します。
 4. ポリシー サーバは、ユーザ情報を、ユーザディレクトリとポリシーストアに基づいて評価します。
 5. ポリシー サーバは、認証レスポンスを Cisco RAS に送信します。RAS は、次のいずれかの作業を実行します。
 - 認証が失敗すると、RAS は接続を拒否します。
 - 認証が成功すると、RAS は、RADIUS サーバのデータベースにあるユーザプロファイルから属性のリストを受け取り、ユーザのネットワークアクセスを確立します。
- RAS は、ポリシー サーバに、セッションが開始されたこと、およびいつセッションが終了するかを通知します。

インターネットユーザが Checkpoint ファイアウォール経由でインターネット サービスプロバイダにダイヤルアップしようとする場合も、同様の認証プロセスが発生します。Checkpoint ファイアウォール用に定義された RADIUS エージェントは、レルムヒントを使用して、インターネットユーザがアクセスしているドメインを判別します。ユーザが認証されると、ポリシー サーバは、ファイアウォールにセッションを確立するための適切な属性を渡します。

両方の NAS デバイス用のユーザ情報は、同じユーザディレクトリに保存されます。ポリシー サーバは、認証リクエストを受信するたびに、同じデータディレクトリを使用してユーザを認証します。

システムとポリシードメインの設定

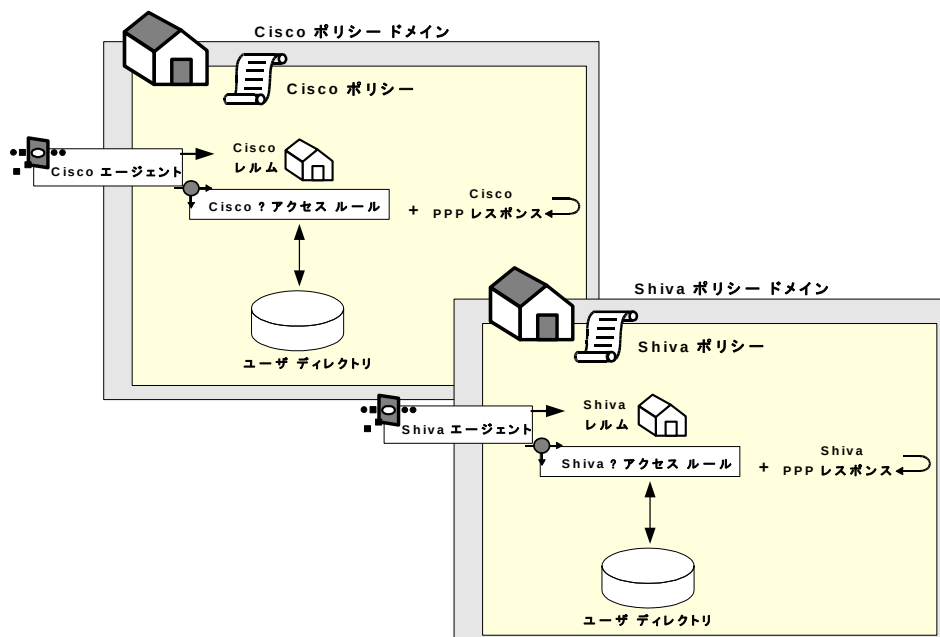
このシステム設定は、同機種環境と異なり、2 つのエージェントを作成する必要があります。

ポリシードメイン内には、Cisco エージェントおよび Checkpoint エージェント用のルールとレスポンスを含む 1 つのポリシーがあります。

前述の 1 つのディレクトリを持つ異機種環境で SiteMinder をセットアップするには、以下の手順を実行する必要があります。

1. 次の手順で、システムを設定します。
 - a. 「[1 つのディレクトリを持つ異機種環境のエージェントの定義 \(P. 840\)](#)」の説明にあるように、2 つの RADIUS エージェントを定義します。
 - b. 「[ユーザ ディレクトリの設定 \(P. 840\)](#)」の説明にあるように、RADIUS ユーザの認証の対象となるユーザ ディレクトリをセットアップします。
 - c. 「ポリシードメインの作成」の説明にあるように、1 つのポリシードメインを作成します。
 - d. 「認証方式の作成」の説明にあるように、認証方式を作成します。
2. 次の手順で、ポリシードメインを設定します。
 - a. 2 つのレルム (Cisco RAS 用のレルムと Checkpoint ファイアウォール用のレルム) を定義します。各レルムは、RADIUS エージェントと RADIUS 認証方式をバインドします。
 - b. 認証されたユーザが適切なレルムにアクセスすることを許可する 2 つのルールを定義します。各ルールは、レルムとアクセス許可/拒否イベントをバインドします。
 - c. ユーザプロフィールを NAS デバイスに提供し、レスポンス属性を使用してセッションの特徴を設定する 2 つのレスポンスを定義します。各 NAS デバイスが異なる辞書ファイルを使用するため、各デバイス用に個別のレスポンスを定義する必要があります。
 - d. Cisco ルールを Cisco レスポンスにバインドし、Checkpoint ルールを Checkpoint レスポンスにバインドする 1 つのポリシーを作成します。このポリシーはまた、ポリシードメインのコンポーネント (ルールとレスポンスのグループ) を RADIUS ユーザ ディレクトリにバインドします。

次の図に、このポリシードメインの構造を示します。



1つのディレクトリを持つ異機種環境のエージェントの定義

この環境では、次のような2つの RADIUS エージェントを設定する必要があります。

- 1つのエージェントは、Cisco RAS に関連付けられます。
- 1つのエージェントは、Checkpoint ファイアウォールに関連付けられます。
- どちらの RADIUS エージェントも、レルムヒントとして1を使用する必要があります。これにより、各エージェントが、保護するドメインを正しく識別できます。

ユーザディレクトリの設定

ポリシーサーバは、同じユーザディレクトリを使用して両方の NAS デバイスのユーザを認証できます。

ポリシードメインの作成

ポリシードメインは、RADIUS ユーザの名前、ドメインを変更できる管理者の名前、および RADIUS エージェントが保護しているレルムを含むユーザディレクトリを識別する必要があります。RADIUS 環境が使用するユーザディレクトリが 1 つの場合、必要なポリシードメインは 1 つのみです。

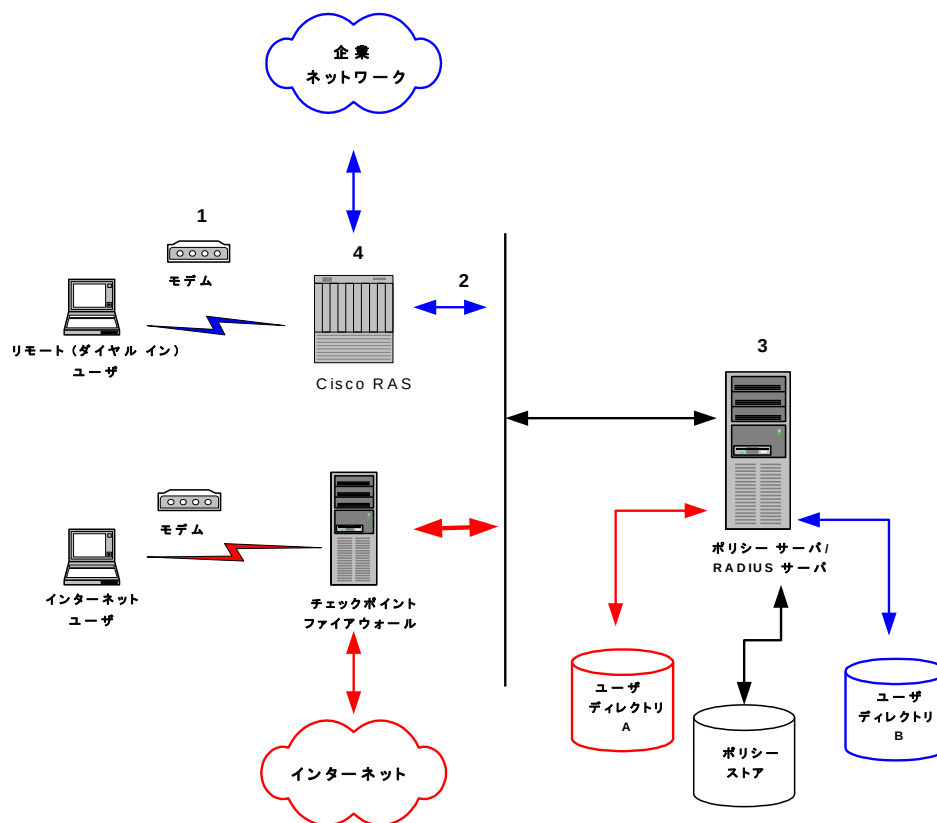
2つのユーザディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法

複数の NAS デバイスのユーザ情報が、それぞれ個別のユーザディレクトリに保存されている場合、複数の NAS デバイスのユーザを認証するようにポリシーサーバを設定することもできます。NAS デバイスは、ベンダータイプが異なってもかまいません。

この構成には、次のようないくつかの利点があります。

- 1つのポリシードメインで 2つのユーザディレクトリを使用することにより、各ディレクトリの管理を異なる担当者に委託できます。
- 複数の RADIUS クライアントのユーザを認証するようにポリシーサーバを設定することにより、作業時間を節約できます。各 RADIUS クライアント用に個別の認証サーバをインストールして設定する必要はありません。
- 既存のユーザディレクトリを使用するため、効率的です。ユーザ情報を単一のディレクトリに結合する必要はありません。

次の図では、2つのユーザ ディレクトリを使用する異機種構成の例について説明します。



前のセクションで説明したトポロジとは異なり、このポリシー サーバは、2つのユーザ ディレクトリを使用してユーザを認証します。Cisco RAS ユーザのユーザ情報はユーザ ディレクトリ A に格納されます。Checkpoint ファイアウォールのユーザ情報はユーザ ディレクトリ B に格納されます。ポリシー サーバは、これらのディレクトリの両方を使用してユーザを認証できます。

設定を2つのポリシー ドメインに分割することによって、レルムヒントが不要になります。各 RADIUS エージェントはそれぞれ別々のポリシー ドメインに存在し、1つのレルムにのみバインドされます。

使用されているユーザディレクトリが2つの場合の認証プロセスは、以下のとおりです。

1. リモートユーザがモデムからダイヤルインすると、Cisco RAS は、ユーザを認証するために RADIUS ユーザプロファイルを使用する必要があるかどうかを判別します。
2. RAS は、ポリシー サーバにユーザ接続リクエストを送信します。
3. ポリシー サーバは、RAS 用に定義されたポリシーを有効化し、RADIUS エージェントは、そのエージェント用に設定された認証方式を使用して、ユーザの名前とパスワードを取得します。
4. ポリシー サーバは、ユーザ情報を、ポリシーのドメインに関連付けられているユーザディレクトリとポリシーストアに基づいて評価します。
5. ポリシー サーバは、認証レスポンスを Cisco RAS に送信します。RAS は、次のいずれかの作業を実行します。
 - 認証が失敗すると、RAS は接続を拒否します。
 - 認証が成功すると、RAS は、RADIUS サーバのデータベースにあるユーザプロファイルから属性のリストを受け取り、ユーザのネットワークアクセスを確立します。

RAS は、ポリシー サーバに、セッションが開始されたこと、およびいつセッションが終了するかを通知します。

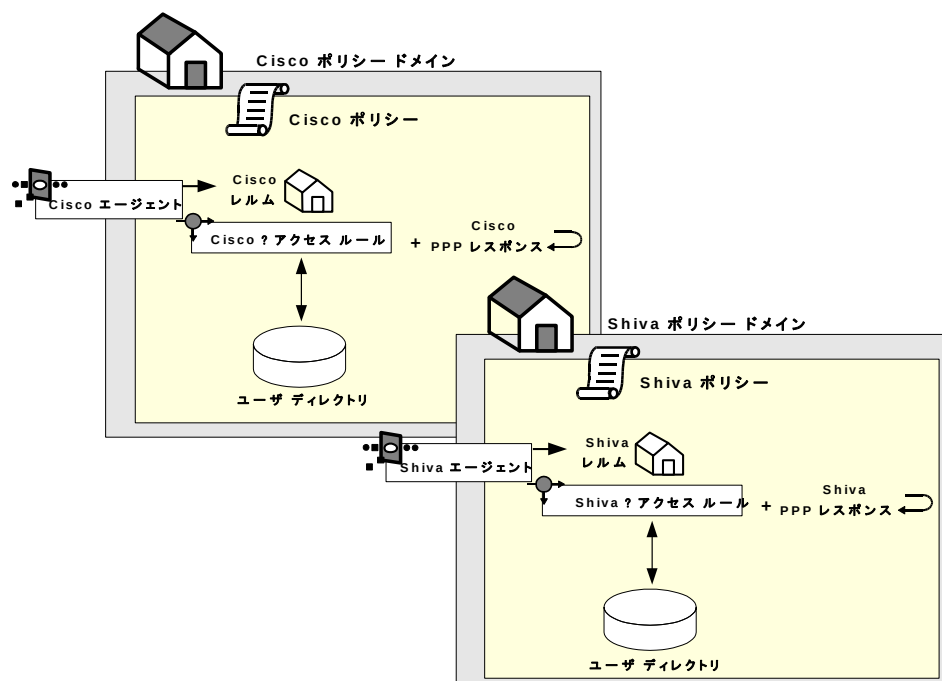
インターネットユーザが Checkpoint ファイアウォール経由でインターネットサービスプロバイダにダイヤルアップしようとする場合も、同様の認証プロセスが発生します。ただし、ポリシーサーバは、インターネットユーザの認証情報を異なるユーザディレクトリに基づいて評価します。

システムとポリシードメインを設定する方法

前述の 2 つのユーザディレクトリを持つ異機種環境を設定するには、以下の手順を実行する必要があります。

1. 次の手順で、システムを設定します。
 - a. 「[2つのディレクトリを持つ異機種環境のエージェントの定義 \(P. 845\)](#)」の説明にあるように、2 つの RADIUS エージェントを定義します。
 - b. 「[ユーザディレクトリのセットアップ \(P. 845\)](#)」の説明にあるように、ユーザディレクトリをセットアップします。
 - c. 「[2つのポリシードメインの作成 \(P. 846\)](#)」の説明にあるように、2 つのポリシードメインを作成します。
2. 次の手順で、ポリシードメインを設定します。
 - a. 1 つのレルムを定義します。レルムは、RADIUS エージェントと RADIUS 認証方式をバインドします。
 - b. 認証されたユーザがレルムにアクセスすることを許可するルールを定義します。各ルールは、レルムとアクセス許可/拒否イベントをバインドします。
 - c. ユーザプロフィールを NAS デバイスに提供したり、必要に応じて、レスポンス属性を使用してセッションの特徴を設定したりするレスポンスを定義します。
 - d. ルールをレスポンスにバインドするポリシーを作成します。このポリシーはまた、ルールとレスポンスを RADIUS ユーザディレクトリにバインドします。

次の図に、これら2つのポリシードメインの構造を示します。



2つのディレクトリを持つ異機種環境のエージェントの定義

この環境では、次のような2つのRADIUSエージェントを設定する必要があります。

- 1つのエージェントは、Cisco RASに関連付けられます。
- 1つのエージェントは、Checkpoint ファイアウォールに関連付けられます。
- どちらのRADIUSエージェントにも、レルムヒントは不要です。

ユーザディレクトリのセットアップ

RADIUS ユーザ情報を含む各ユーザディレクトリを、ポリシーサーバで設定する必要があります。各ディレクトリは別々のポリシードメインにバインドされるため、各ポリシードメインに異なる管理者を定義できます。

2 つのポリシードメインの作成

Cisco RAS 用に 1 つのポリシードメインを作成し、Checkpoint ファイアウォール用に 1 つのポリシードメインを作成する必要があります。ポリシードメインを定義する際に、各ドメインを適切なユーザ ディレクトリに関連付けます。

RADIUS エージェント グループの概要

RADIUS エージェントグループを作成すると、複数の RADIUS エージェントを同時に管理できるようになり、各 RADIUS エージェント用に個別のレلمを作成して設定する必要がなくなります。1 つのレلمを使用する場合、すべての RADIUS エージェント用に同じセッションタイムアウトと同じ認証方式を同時に定義できるため、作業時間を節約できます。

RADIUS エージェントのグループは、異なるタイプの NAS デバイスのエージェントを含むことができます。たとえば、1 つのエージェントグループに、Shiva LAN Rover RAS、Checkpoint ファイアウォール、および Cisco RAS のエージェントを含めることができます。これらの RADIUS エージェントをすべて含むエージェントグループは、認証方式とセッションタイムアウト要件を定義する 1 つのレلمに関連付けられます。

エージェント グループは、頻繁には変更されないスタティック環境に最適です。エージェントグループを使用すると、多数の NAS デバイスのユーザを認証する STMNDR をすばやく設定できます。使用している環境が静的ではなく、新しい NAS デバイスが追加または削除されることによって頻繁に変更される場合は、エージェントグループの使用に適さないことがあります。このような場合は、通常、エージェントをグループ化しない方が、RADIUS エージェントを簡単に追加または削除できます。エージェントが分離されており、グループ化されていない場合、通常、より迅速に、個別のエージェントを検出し、ポリシーを変更または削除できます。

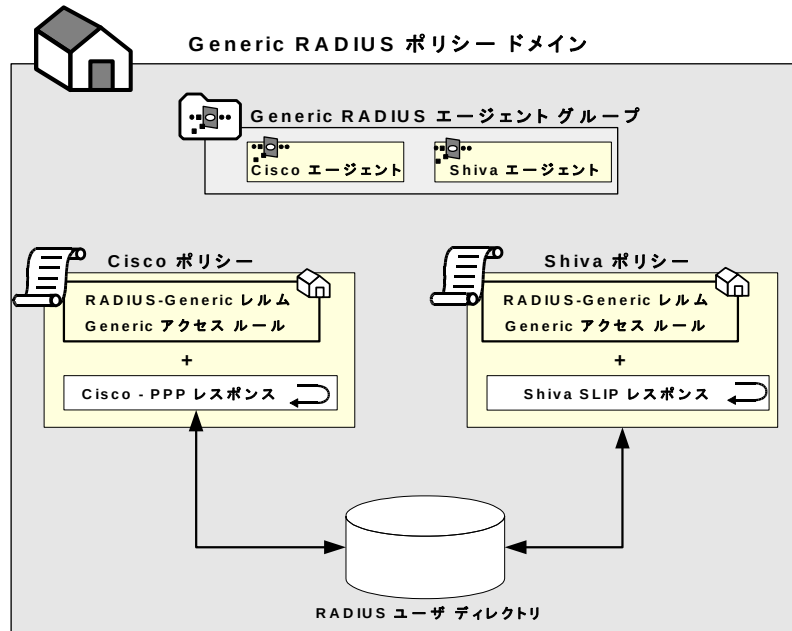
RADIUS エージェントグループのセットアップ

RADIUS エージェントグループを使用する場合、一般に、各タイプの RADIUS エージェント用に個別のポリシーを設定します。各タイプの RADIUS エージェント用に個別のポリシーを使用すると、ポリシードメインの共通要素 (レルム、ルールなど) を各ポリシーで共有できます。これらの共通要素を共有することによって、作業時間を節約できます。

ルールやレルムとは異なり、各ポリシーのレスポンスは共有されません。各ポリシーには、ポリシー内の RADIUS エージェントのデバイスタイプに対応する独自のレスポンスがあります。レスポンス内の属性は、NAS デバイスの辞書ファイルによって提供される属性と適合します。たとえば、Cisco RAS 用のレスポンスは、Cisco RAS が Cisco 辞書ファイルを使用して解読できる属性を提供する必要があります。

注: RADIUS エージェントグループに表示されるすべての NAS デバイスは、同じユーザディレクトリを共有する必要があります。これらのデバイスは、同じユーザディレクトリを共有しないと、同一のポリシードメインに存在できないため、同じ一般レルムまたは一般ルールを共有できません。

以下に示した例は、Cisco RAS 用のエージェントと Shiva RAS 用のエージェントの両方を含む 1 つの RADIUS エージェントグループです。このエージェントグループは、Cisco ポリシーと Shiva ポリシーの両方によって共有されます。これらのポリシーは、アクセスを許可する同じ一般ルールと、同じ一般レلمを共有し、エージェントグループを同じ認証方式にバインドします。ただし、各ポリシーのレスポンスは、それぞれに固有のものです。



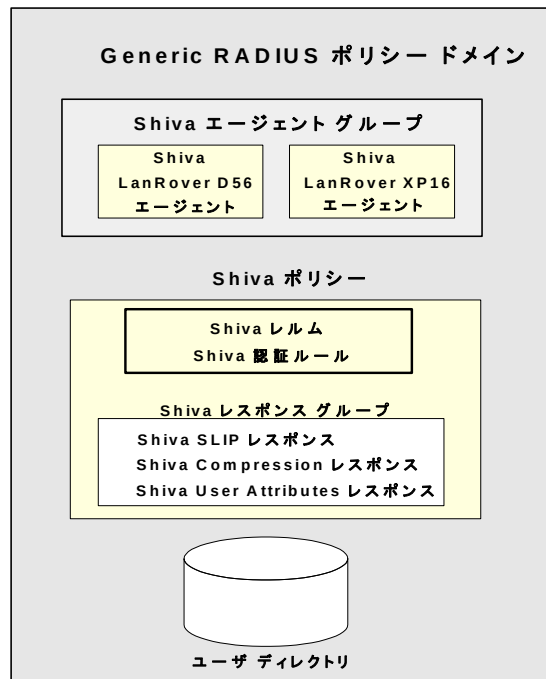
RADIUS エージェントグループをセットアップする手順は、「[エージェントグループの設定](#) (P. 160)」にあります。

RADIUS レスポンスのグループ化

RADIUS レスポンスグループは、同じタイプのエージェント (Generic RADIUS など) 用に定義されたレスポンスの集まりです。ルールが起動すると、レスポンスグループ内の、そのルールとペアになっているすべての RADIUS レスポンスが発行されます。

同じ辞書ファイルを使用するには、レスポンスが同じエージェントタイプである必要があります。たとえば、2 つの Cisco レスポンス、または 2 つの一般 RADIUS レスポンスをそれぞれ同じグループに結合することができます。しかし、一般 RADIUS レスポンスと Cisco レスポンスを同じグループに結合することはできません。

RADIUS レスポンスグループを使用する利点は、少ない数のポリシーを使用してポリシードメインを設定できることです。各 RADIUS エージェント用の個別のレルムと個別のポリシーを作成する代わりに、同じタイプの RADIUS エージェントをグループ化して、認証のための一般ルールを 1 つ作成し、そのルールのレスポンスをグループ化できます。次の図に、このタイプの構成を示します。



レスポンスグループを使用すると、環境への RADIUS デバイスの追加も簡単になります。たとえば、この図に示したような環境では、Shiva RAS を RADIUS エージェントグループに迅速に追加できます。この新しい RAS は、適切なルールとレスポンスで自動的に設定されます。

RADIUS のトラブルシューティングとテスト

RADIUS 認証サーバとして動作するようにポリシー サーバを設定する作業を完了してから、以降のトピックで説明するツールを使用してポリシーをテストし、トラブルシューティングできます。

アカウントिंगとデバッグ用の RADIUS ログの生成

RADIUS ログは、ポリシー サーバによって生成されるデバッグ情報とアカウントング情報を追跡します。RADIUS ログファイルを使用すると、次の情報を追跡できます。

- ポリシー サーバのステータス
- 接続試行とセッション作成
- 各ポリシー サーバアクションに関する情報

ログのオンとオフは、ポリシー サーバ管理コンソールの[デバッグ]タブから切り替えます。

ポリシー サーバはログ ファイルに、ログ ファイルが作成された日時を示すタイムスタンプを付けます。たとえば、ファイルの名前が「log.txt」だとします。ポリシー サーバが再起動してログ ファイルを作成すると、このファイル名に次のように日時が追加されます。

log.txt.08Dec1999_13_30_57

同じファイルにログ情報を追加している場合、ファイル名の後の日時によって、ファイルが作成された日時を確認することができます。タイムスタンプは、ポリシー サーバが再起動したときのみ更新されます。

smreadclog による RADIUS ログ ファイルの読み込み

このツールは、ポリシー サーバによって生成された RADIUS ログ ファイルの読み込みに使用します。ポリシー サーバを RADIUS 認証サーバとして使用している場合、このツールは、トラブルシューティングに役立ちます。NAS と SiteMinder の間で交換された個別の RADIUS 属性を表示することもできます。

smreadclog では、次の引数を使用して、RADIUS ログ ファイルの読み込みに必要な情報を提供します。

-i<入力ファイル名>

ログファイルのファイル名を指定します。

-o<出力ファイル名>

出力ファイルのファイル名を指定します。

-s<秘密キー>

RADIUS パスワードの複号化に使用できる共有秘密キーを指定します。

-r

RADIUS パケット全体を 16 進数ダンプ表示することを示します。

-a

RADIUS 属性を個別に表示することを示します。

-d

RADIUS 属性をポリシーストアの定義に従って表示することを示します。このオプションを指定すると、実際の属性名と属性値が、属性のタイプに基づいた形式で表示されます。このオプションを指定しない場合は、属性名と値 (16 進数の文字列) だけが表示されます。

-p<RADIUS サーバ名>

使用している RADIUS サーバに対するポリシー サーバサービスの RADIUS アクティビティを記録したり、再生したりできるようになります。

-m<認証ポート番号>

デフォルトポート (1645) 以外のポートを RADIUS 認証に使用する場合に、そのポートを指定します。

-n<アカウントングポート番号>

デフォルトポート (1646) 以外のポートを RADIUS アカウンティングに使用する場合に、そのポートを指定します。

smreadclog を使用する方法

1. 次のいずれかのディレクトリに移動します。

- Windows NT の場合: <siteminder_installation>\Bin

ここで、<siteminder_installation> は、SiteMinder がインストールされている場所です。

- UNIX の場合: <siteminder_installation>/bin

ここで、<siteminder_installation> は、SiteMinder がインストールされている場所です。

2. 以下のコマンドを入力します。

```
smreadclog -i<input-file> -o<output-file>
-s<secret> -r -a -d -p<radius-server> -m<portnumber>
-n<portnumber>
```

例を以下に示します。

```
smreadclog -iradiuslog.txt -oradiuslog2.txt
-ssecret -r -a -d -p123.123.12.12
```

SiteMinder テスト ツールを使用してテストする方法

SiteMinder テスト ツールは、RADIUS 認証サーバの動作をシミュレートします。テストツールを使用すると、RADIUS ユーザを認証するポリシーをテストして、設定したレスポンス属性が適切なデータを返すことを確認できます。

RADIUS ポリシーをテストするには、次の手順に従います。

1. RADIUS ポリシーを作成します。
2. 「[ポリシー サーバ管理コンソールの設定 \(P. 852\)](#)」の説明にあるように、RADIUS を使用するようにポリシー サーバ管理コンソールを設定します。
3. 「[RADIUS ポリシーのテスト \(P. 853\)](#)」の説明にあるように、RADIUS ポリシーをテストするように SiteMinder テスト ツールを設定します。

ポリシー サーバ管理コンソールの設定

ポリシー サーバ管理コンソールを設定する方法

1. ポリシーサーバ管理コンソールを起動します。

重要: Windows Server 2008 上でこのグラフィカル ユーザ インターフェースにアクセスする場合は、管理者としてシステムにログインしている場合でも、管理者権限でショートカットを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。
 2. [設定] タブを選択します。
 3. [設定] タブで、以下の操作を実行します。
 - a. [RADIUS UDP ポート] グループボックスで、[有効] ラジオボタンをオンにします。
 - b. [認証] フィールドに、1645 と入力します。
 - c. [監査] フィールドに、1646 と入力します。
 4. [ステータス] タブで、ポリシー サーバを再起動してポリシー サーバ設定の変更内容を有効にします。
- これで、テストツールを使用して RADIUS ポリシーをテストできます。

RADIUS ポリシーのテスト

RADIUS ポリシーをテストする方法

1. テストツールを起動します。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行している場合は、管理者としてシステムにログインしている場合でも、管理者権限でコマンドライン ウィンドウを開きます。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

2. [SiteMinder Agent] グループ ボックスで、以下の操作を実行します。
 - a. [RADIUS] ラジオ ボタンをオンにします。
 - b. [Secret] フィールドに、SiteMinder 管理ユーザ インタフェースで RADIUS エージェント用に定義した共有秘密キーを入力します。
3. [User Information] グループ ボックスで、以下の操作を実行します。
 - a. [User] フィールドに、ユーザの認証をテストする RADIUS ユーザディレクトリのユーザの名前を入力します。
 - b. [Password] フィールドに、ユーザのパスワードを入力します。
 - c. RADIUS CHAP 認証方式を使用する場合は、[CHAP Password] チェック ボックスをオンにします。
4. [Command] グループ ボックスで、[IsAuthenticated] をクリックします。

ポリシー サーバがテストされ、レスポンス属性が属性グループボックスに表示されます。

付録 F: 属性および式のリファレンス

このセクションには、以下のトピックが含まれています。

[データ型](#) (P. 855)

[式の構文の概要](#) (P. 858)

[貼り付け](#) (P. 860)

[オペレータ](#) (P. 862)

[式の中に使用できる関数](#) (P. 879)

データ型

定数データはすべて、3 つの主なリテラル データ型 (文字列、数値、ブール) のいずれかです。文字列データ型には、セットおよび LDAP 識別名という 2 つのサブタイプが含まれます。数値データ型には、日付という 1 つのサブタイプが含まれます。すべての関数および演算の結果は、これらのデータ型またはそのサブタイプのいずれかになります。

- 文字列
 - セット
 - LDAP 識別名
- 数値
 - 日付
- ブール

リテラル データ型のほかに、変数として機能する以下のデータ型があります。

- ユーザ属性
- 名前付き式

文字列

文字列は、1 組の引用符または二重引用符で囲まれた 0 文字以上の一連の文字として、文字データを表します。文字列値は、組み込みの演算子および関数で利用できる定数です。たとえば、文字列を、別のデータ型に変換したり、連結したりできます。

「0」～「9」の文字を含む文字列（正または負の符号が先頭に付いている場合があります）は、数値に変換できます。「TRUE」および「YES」（または「true」および「yes」）という文字列は、ブール値 TRUE に変換できます。他の文字列値はすべて、FALSE に変換されます。2 つの文字列は連結できます。連結では、2 番目の文字列の先頭が、1 番目の文字列の末尾につながれます。

セットおよび LDAP 識別名 (DN) は、文字列データ型の特例です。セットは、キャレット文字で区切られた一連の要素（「element1^element2」など）です。セット内の各要素は文字列です。

LDAP DN は、LDAP ディレクトリでエントリを一意に識別する単純な文字列であり、その形式は LDAP 仕様で定義されています。

数値

数値は、整数である必要があります。また、正または負の符号が先頭に付いている場合があります。4 バイトの整数、または、 $-2^{31} \sim 2^{31}$ の範囲の整数はサポートされています。小数点は無視されます。数値は、組み込みの演算子および関数で利用できる定数です。たとえば、数値を別のデータ型に変換できます。

数値は、「0」～「9」の文字のみを含む文字列（負の符号が先頭に付いている場合があります）に変換できます。数値はブール値にも変換できます。ゼロ以外の数値は TRUE に、ゼロは FALSE に変換されます。

日付は、数値データ型の特例です。日付は、1970 年 1 月 1 日から経過した秒の数として表されます。

日付は、多くの関数で使用されます。たとえば、DOW 関数は、数値で表された日付を受け取り、曜日に相当する 0 ～ 6 という数値のいずれかを返します。また、文字列形式による日付を数値に変換する関数や、数値形式による日付を文字列に変換する関数もあります。

ブール

ブールは、2つの値、TRUE および FALSE のいずれかです。ブール値は、組み込みの演算子および関数で比較および使用できる定数です。たとえば、ブールを別のデータ型に変換できます。

ブール値を数値に変換すると、TRUE は 1 に、FALSE は 0 に変換されます。ブール値を文字列値に変換すると、TRUE は「TRUE」に、FALSE は「FALSE」に変換されます。

ユーザ属性

ユーザ属性には、名前と値があります。ユーザ属性の *名前* は、引用符を付けずに使用します。ユーザ属性の *値* は文字列であり、引用符を付けて使用します。ユーザ属性の名前は、以下のルールに従っている必要があります。

- 大文字と小文字は区別されません。
- 先頭の文字はアルファベットです。
- また、この名前には、US-ASCII アルファベット文字、数字、およびアンダースコア文字のみが含まれます。

ユーザ属性名は、リテラルデータ型ではなく、変数データ型として機能します。ユーザ属性名が見つかり、対応する属性値がユーザ ディレクトリから取得されます。属性に複数の値があるときは、それらがセットとして返されます。セットは、キャレット文字で区切られた一連の要素（「value1^value2」など）です。セット内の各要素は文字列です。

名前付き式

名前付き式には、仮想ユーザ属性およびユーザ クラスという 2 つのタイプがあります。

仮想ユーザ属性は、ユーザ属性とは異なります。文字列としてユーザ ディレクトリ内に格納されるユーザ属性とは異なり、仮想ユーザ属性は、実行時に計算されて結果が文字列、数値、またはブール値となる式に名前を付けます。さらに、仮想ユーザ属性は、読み取り専用である点でもユーザ属性と異なっています。

ユーザクラスは、仮想ユーザ属性の特例です。仮想ユーザ属性のように、ユーザクラスは、実行時に計算される式に名前を付けます。ユーザクラスが仮想ユーザ属性と異なるのは、ユーザグループまたはユーザディレクトリでメンバシップをチェックして、ブール値のみを返す式に名前を付ける点です。

仮想ユーザ属性名およびユーザクラス名は両方とも、以下のルールに従っている必要があります。

- 大文字と小文字は区別されません。
- 仮想ユーザ属性の場合は、先頭の文字が番号記号(#)である必要があります。
- ユーザクラスの場合は、先頭の文字がアットマーク(@)である必要があります。
- 2番目の文字は、文字またはアンダースコアである必要があります。
- 残りの文字は、US-ASCII アルファベット文字、数字、またはアンダースコア文字である必要があります。

注: アクティブな式と名前付き式は同じではありません。どちらのタイプの式もランタイムに評価されるのは同じですが、以下の点で異なります。

- アクティブな式はブール式ですが、名前付き式は文字列、数値、ブール値を返すことができます。
- アクティブな式はそのまま参照されるため、使用するたびに再入力する必要がありますが、名前付き式は名前で参照されるため、どこからでも参照でき、再利用できます。

式の構文の概要

この付録では、内部 SiteMinder 式エバリュエータが使用する構文について説明します。管理 UI でロールまたは資格を定義するときに、この構文を含むデータ型、演算子、および組み込み関数を式に使用できます。名前付きでない式は、定義している特定のロールまたは資格に固有のものです。また、グローバルに定義された名前付き式を使用できます。

名前付き式には、仮想ユーザ属性(名前が # で始まる)およびユーザクラス(名前が @ で始まる)が含まれます。

必要な情報をユーザのディレクトリ エントリから直接読み取ることができない場合は、仮想ユーザ属性が値を計算します。仮想ユーザ属性は文字列、数値、またはブール値を返します。たとえば、並べ替えるときに頻繁に使用できるように、名前情報の形式を指定するに、以下のように、「#SortName」と呼ばれる仮想ユーザ属性をユーザ インターフェイスで定義できます。

```
UCase(RTrim(LastName + "," + FirstName + " " + Initial))
```

この例では、2 つの組み込み関数、UCASE および RTRIM を使用しています。これらの名前では、大文字と小文字は区別されません。

ユーザ クラスは、マネージャや管理者などのユーザ タイプに基づく特定のカテゴリにユーザが属するかどうかを決定する式です。ユーザは、特定のユーザ クラスのメンバであるか、そうでないかのいずれかです。したがって、ユーザ クラス式の結果は常にブール値です。

仮想ユーザ属性またはユーザ クラスを定義するときに、それがプライベートであることを指定できます。これは、他の名前付き式からのみ呼び出すことができることを意味します。同様に、一部の組み込み関数は、特権のある関数として示されます。これは、別の名前付き式の内部からのみ呼び出すことができることを意味します。特権のある関数は、[注釈]セクションに「特権」という注記があります。1 つ以上の LDAP 識別名をパラメータとして受け取る関数は、[注釈]セクションに「LDAP のみ」という注記があります。

貼り付け

式は、ポリシー ストア内にオブジェクトとして格納すると、他の式を含め、どこからでも名前でも参照できます。どの式も、プレースホルダ %1 ~ %9 および特別のプレースホルダ %0 を使用して、名前付き式に値を渡すことができます。これは「貼り付け」と呼ばれます。

名前付き式には、仮想ユーザ属性およびユーザ クラスという 2 つのタイプがあります。仮想ユーザ属性名の先頭の文字は番号記号 (#) で、ユーザ クラス名の先頭の文字は符号です。この 2 つのタイプの名前付き式にはいずれも、その後 9 つまでのパラメータを指定できます。構文は、関数の構文と同様です。

```
#virtual_user_attribute(P1, P2, P3, P4, P5, P6, P7, P8, P9)
```

```
@user_class(P1, P2, P3, P4, P5, P6, P7, P8, P9)
```

ポリシー ストア内で名前付き式を作成するときには、組み込みの演算子、関数、リテラル データ型、プレースホルダ、および他の名前付き式を使用できます。変数データを名前付き式に渡すには、プレースホルダを使用します。以下の例では、URL が、参照されるたびに更新されます。したがって、URL をプレースホルダ %1 で表す必要があります。

例:

URL を受け取ってファイル名を返す、#URLFile という名前の仮想属性を、次のように作成できます。

```
#URLFile := { FIND(%1, '/')=0 ? %1 : #URLFile(AFTER(%1, '/')) }
```

```
Return_value=#URLFile('C:\My Documents\expression_syntax.doc')
```

```
Return_value='expression_syntax.doc'
```

この例では、URL が、プレースホルダ %1 経由で組み込み関数 FIND に渡されます。FIND は URL 内で「/」を検索し、最初に見つかった「/」の位置を返します。「/」が見つからない場合は、0 が返されます。また、#URLFile はファイル名を返します。それ以外の場合は、URL が、プレースホルダ %1 経由で組み込み関数 AFTER に渡されます。AFTER は、URL の、「/」に続く部分を返します。その後、短縮された URL が #URLFile に渡されます。再帰がサポートされています。

以下の表は、この例が繰り返されるたびに返される、位置および URL の値を示しています。

繰り返し	位置	URL
1	3	'My Documents¥expression_syntax.doc'
2	16	'expression_syntax.doc'

ENUMERATE や LOOP などの特定の組み込み関数が同じ名前付き式を複数回 (セット内の要素ごとに 1 回) 呼び出す場合は、特別のプレースホルダ %0 を使用して、その名前付き式を作成する必要があります。たとえば、セット要素のすべての数値から後続スペースを削除する、#RTrimset という名前の式を作成できます。

```
#RTrimset := RTrim(%0)
```

その後に、組み込み関数 ENUMERATE を使用して、#RTrimset にセットを渡すことができます。

```
Return_value=ENUMERATE('First_name ^Middle_name ^Last_name ',#RTrimset)  
Return_value='First_name^Middle_name^Last_name'
```

この例では、名、ミドル ネーム、および姓という 3 つの要素でセットが構成されています。ENUMERATE は、この各要素を #RTrimset に渡します。#RTrimset は後続スペースを削除し、短縮された名前を ENUMERATE に返します。

ENUMERATE は、返されるそれぞれの要素を演算結果に含め、それらをキャレット文字で区切ります。

オペレータ

このトピックでは、サポートされているすべての演算子の、カテゴリごとのリストを示します。

比較演算子

- 等価 (= および ~=)
- 不等価 (!= および ~=)
- より大きい (> および ~>)
- より小さい (< および ~<)
- 以上 (>= および ~>=)
- 以下 (<= および ~<=)

文字列演算子

- BEGINS_WITH および ~BEGINS_WITH
- ENDS_WITH および ~ENDS_WITH
- CONTAINS および ~CONTAINS
- パターン一致 (LIKE)
- 連結 (+)

演算子の設定

- 包括の設定 (IN および ~IN)
- INTERSECT および ~INTERSECT
- UNION および ~UNION
- インデックス作成 ([..])

論理演算子

- AND、&、および &&
- NOT
- OR、|、および ||
- XOR

算術演算子

- 加算(+)
- 減算(-)
- 乗算(*)
- 除算(/)

その他の演算子

- 条件付き決定(? および :)

等価演算子

等価演算子(=)は、2つの値を比較します。値が等しい場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。2つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

等価演算子(~=)は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

1 = 1
結果 = TRUE

2 = 1
結果 = FALSE

"sparrow" = "SPARROW"
結果 = FALSE

"sparrow" ~= "SPARROW"
結果 = TRUE

不等価演算子

不平等演算子(`!=`)は、2つの値を比較します。値が等しくない場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。2つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

不等価演算子(`~!=`)は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

```
1 != 1  
結果 = FALSE
```

```
2 != 1  
結果 = TRUE
```

```
"sparrow" != "SPARROW"  
結果 = TRUE
```

```
"sparrow" ~!= "SPARROW"  
結果 = FALSE
```

「より小さい」演算子

「より小さい」演算子(`<`)は2つの値を比較します。1番目の値が2番目の値より小さい場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。2つの値がブールである場合は、**TRUE** が **FALSE** より大きくなります。2つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

「より小さい」演算子(`~<`)は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

`1 < 2`

結果 = TRUE

`2 < 1`

結果 = FALSE

`'crow' < 'CROW'`

結果 = FALSE

`'crow' ~< 'CROW'`

結果 = FALSE

「より大きい」演算子

「より大きい」演算子(>)は 2 つの値を比較します。1 番目の値が 2 番目の値より大きい場合は、演算結果が TRUE になります。それ以外の場合は、演算結果が FALSE になります。2 つの値がブールである場合は、TRUE が FALSE より大きくなります。2 つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

「より大きい」演算子(~>)は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

`1 > 2`

結果 = FALSE

`2 > 1`

結果 = TRUE

`'crow' > 'CROW'`

結果 = TRUE

`'crow' ~> 'CROW'`

結果 = FALSE

「以下」演算子

「以下」演算子 ($\leq$) は、2 つの値を比較します。1 番目の値が 2 番目の値より小さいか、等価である場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。2 つの値がブールである場合は、**TRUE** が **FALSE** より大きくなります。2 つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

「以下」演算子 ($\sim\leq$) は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

```
1 <= 2  
結果 = TRUE
```

```
2 <= 1  
結果 = FALSE
```

```
'junco' <= 'JUNCO'  
結果 = FALSE
```

```
'junco' ~<= 'JUNCO'  
結果 = TRUE
```

「以上」演算子

「以上」演算子 ($\geq$) は、2 つの値を比較します。1 番目の値が 2 番目の値より大きい、等価である場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。2 つの値がブールである場合は、**TRUE** が **FALSE** より大きくなります。2 つの値が文字列の場合は、演算で、大文字と小文字が区別されます。

「以上」演算子 ($\sim\geq$) は、文字列値のみを比較します。このとき、大文字と小文字は区別されません。

例:

```
1 >= 2  
結果 = FALSE
```

```
2 >= 1  
結果 = TRUE
```

```
'junco' >= 'JUNCO'  
結果 = TRUE
```

```
'junco' ~>= 'JUNCO'  
結果 = TRUE
```

先頭演算子

2 つの先頭演算子 (BEGINS_WITH および ~BEGINS_WITH) は、文字列値と共に使用するよう設計されています。1 番目の文字列の先頭が 2 番目の文字列である場合は、演算結果が TRUE になります。それ以外の場合は、演算結果が FALSE になります。

「BEGINS_WITH」演算子では、大文字と小文字が区別されます。
「~BEGINS_WITH」演算子では、大文字と小文字が区別されません。

例:

```
'SiteMinder' BEGINS_WITH 'site'  
結果 = FALSE
```

```
'SiteMinder' ~BEGINS_WITH 'site'  
結果 = TRUE
```

末尾演算子

2 つの末尾演算子 (ENDS_WITH および ~ENDS_WITH) は、文字列値と共に使用するよう設計されています。1 番目の文字列の末尾が 2 番目の文字列である場合は、演算結果が TRUE になります。それ以外の場合は、演算結果が FALSE になります。

「ENDS_WITH」演算子では、大文字と小文字が区別されます。「~ENDS_WITH」演算子では、大文字と小文字が区別されません。

例:

```
'SiteMinder' ENDS_WITH 'DER'  
結果 = FALSE
```

```
'SiteMinder' ~ENDS_WITH 'DER'  
結果 = TRUE
```

包含演算子

2 つの包含演算子 (**CONTAINS** および **~CONTAINS**) は、文字列値と共に使用するよう設計されています。1 番目の文字列に 2 番目の文字列が含まれる場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。

CONTAINS 演算子では、大文字と小文字が区別されます。**~CONTAINS** 演算子では、大文字と小文字が区別されません。

例:

```
'SiteMinder' CONTAINS 'EMI'  
結果 = FALSE
```

```
'SiteMinder' ~CONTAINS 'EMI'  
結果 = TRUE
```

セット包含演算子

セット包含演算子 (**IN** および **~IN**) は、1 番目のオペランド (文字列) が、2 番目のオペランド (セット) の要素であるかどうかをチェックします。セットの要素が文字列である場合は、演算結果が **TRUE** になります。それ以外の場合は、演算結果が **FALSE** になります。

IN 演算子では、大文字と小文字が区別されます。**~IN** 演算子では、大文字と小文字が区別されません。

例:

```
'MON' IN 'Sun^Mon^Tue^Wed^Thu^Fri^Sat'
```

結果 = FALSE

```
'MON' ~IN 'Sun^Mon^Tue^Wed^Thu^Fri^Sat'
```

結果 = TRUE

パターン一致演算子

パターン一致演算子 LIKE は、「'abc' LIKE '???'」のように、文字列値を文字のパターン(文字列)と比較します。文字列値がパターンと一致する場合は、演算結果が TRUE になります。それ以外の場合は、演算結果が FALSE になります。パターン一致演算では、大文字と小文字が区別されます。

パターンは、単一の文字、一連の文字、またはその両方を組み合わせて作成します。一連の文字は、「0-9」のように、その最初の文字、ハイフン、およびその最後の文字を連結して指定します。有効な文字は、数字、大文字および小文字のアルファベット、および特別な意味を持つ予約文字です。文字セットは、1文字以上の文字、一連の文字、またはその両方を含み、角かっこで囲まれています。たとえば、[0-9A-Za-z]、[0-2ABC]、[789x-z] はすべて、有効な文字セットです。

注: 一連の文字は常に、文字セットの一部です。

以下の表は、予約文字とその意味のリストです。

文字	意味
' または "	'abc' や "abc" のように、文字列を指定します。
-	「0-9」のように、一連の文字を指定します。
?	任意の 1 文字と一致します。
*	任意の数 (1 または 0 を含む) の文字と一致します。
[set]	指定したセット内の任意の 1 文字と一致します。
[!set] または [^set]	指定したセットにない任意の 1 文字と一致します。
[set]?	指定したセット内または空の文字列にある任意の単一の文字と一致します。
[set]*	指定したセット内の、任意の数 (1 または 0 を含む) の一連の文字と一致します。

文字	意味
¥	¥* などの予約済みの文字をすべて、標準文字として扱います。

「?」の使用例

`" LIKE '?'`

結果 = FALSE

`'a' LIKE '?'`

結果 = TRUE

`'ab' LIKE '?'`

結果 = FALSE

`'abc' LIKE '???'`

結果 = TRUE

`'191' LIKE '1??'`

結果 = TRUE

`'201' LIKE '1??'`

結果 = FALSE

「*」の使用例

`" LIKE '*'`

結果 = TRUE

`'a' LIKE '*'`

結果 = TRUE

`'a1b2c3' LIKE '*'`

結果 = TRUE

`'robin' LIKE 'r*n'`

結果 = TRUE

`'room' LIKE 'r*n'`

結果 = FALSE

「[set]」の使用例

" LIKE '[abcde]'
結果 = FALSE

'f' LIKE '[abcde]'
結果 = FALSE

'c' LIKE '[abcde]'
結果 = TRUE

'abc' LIKE '[abcde]' **Compare:** 'abc' LIKE '[abcde]*'
結果 = FALSE

「[!set]」または「[^set]」の使用例

" LIKE '[!abcde]'
結果 = FALSE

'f' LIKE '[^abcde]'
結果 = TRUE

'c' LIKE '[!abcde]'
結果 = FALSE

'xyz' LIKE '[^abcde]'
結果 = FALSE

「[set]?」の使用例

" LIKE '[abcde]?'
結果 = TRUE

'a' LIKE '[abcde]?'
結果 = TRUE

'ab' LIKE '[abcde]?'
結果 = FALSE

'z' LIKE '[abcde]?'
結果 = FALSE

'[set]*' の使用例

" LIKE '[abcde]*'

結果 = TRUE

'a' LIKE '[abcde]*'

結果 = TRUE

'aabbccdde' LIKE '[abcde]*'

結果 = TRUE

'abcdef' LIKE '[abcde]*'

結果 = FALSE

'abc' LIKE '[abcde]*' **Compare:** 'abc' LIKE '[abcde]'

結果 = TRUE

'\' の使用例

'123-456-7890' LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

結果 = TRUE

'_!^*_?_' LIKE '_!_^_*_?_'

結果 = TRUE

大文字と小文字の区別のチェック例

'a' LIKE '[a-z]'

結果 = TRUE

'A' LIKE '[a-z]'

結果 = FALSE

'A' LIKE '[A-Za-z]'

結果 = TRUE

'Robin' LIKE '[A-Za-z]*'

結果 = TRUE

'Robin' LIKE '[A-Z][a-z]*'

結果 = TRUE

'robin' LIKE '[A-Z][a-z]*'

結果 = FALSE

セット共通演算子

セット共通演算子 (**INTERSECT** および **~INTERSECT**) は、2 つのセットを比較します。セットは、キャレット文字で区切られた一連の文字列です。演算結果のセットは、両方のセットに含まれている要素のみを含む文字列です。**INTERSECT** 演算子では、大文字と小文字が区別されます。**~INTERSECT** 演算子では、大文字と小文字が区別されません。

重要: 結果セット内の要素の順序は予測不可能です。演算で大文字と小文字が区別されない場合は、結果セット内の要素の大文字/小文字も予測不可能です。

例:

```
'BLUE JAY^ORIOLE^WREN' INTERSECT 'BLUE JAY^wren'  
結果 = 'BLUE JAY'
```

```
'BLUE JAY^ORIOLE^WREN' ~INTERSECT 'BLUE JAY^wren'  
結果 = 'BLUE JAY^WREN'
```

セット結合演算子

セット結合演算子 (**UNION**) は、2 つのオペランドセット (2 つのセットの結合) の一意の要素をすべて含むセットを返します。重複する要素は削除されます。

先頭がチルダ文字 (~) の **UNION** 演算子 (**~UNION**) の場合は、大文字/小文字のみが異なる 2 つの要素を同一と見なします。

重要: 結果セットの順序は予測不可能です。また、**~UNION** 演算子を指定した場合は、共通する要素が大文字になるか、小文字になるかも予測不可能です。

例:

```
"JUAN^BART^CHUCK" UNION "CHUCK^Bart"  
結果 = "JUAN^BART^CHUCK^Bart"
```

```
"JUAN^BART^CHUCK" ~UNION "CHUCK^Bart"  
結果 = "JUAN^BART^CHUCK"
```

```
"JUAN^BART^JUAN^CHUCK" UNION "JUAN^BART^JUAN^CHUCK"  
結果 = "JUAN^BART^CHUCK"
```

NOT 演算子

NOT 演算子は、単一のブール オペランドを受け取り、その値を逆にします。つまり、FALSE を TRUE に、TRUE を FALSE に変更します。この演算子は通常、比較演算子の結果を逆にするために使用されます。

例:

```
NOT ("SiteMinder" ENDS_WITH "R")  
結果 = TRUE
```

```
NOT ("SiteMinder " ~ENDS_WITH "R")  
結果 = FALSE
```

AND 演算子

AND 演算子(& および && とも書かれます)は、2 つのブール オペランドを受け取り、両方のオペランドが TRUE の場合に TRUE を返します。この演算子は通常、2 つの比較を接続するために使用されます。

最初のブール オペランドが FALSE の場合は、2 番目のブール オペランドがエバリュエータで評価されません(結果が必ず FALSE になるためです)。

例:

```
(1 > 2) AND ("JUAN" ~= "JUAN")  
結果 = FALSE
```

```
(1 < 2) AND ("JUAN" ~= "JUAN")  
結果 = TRUE
```

OR 演算子

OR 演算子(| および || とも書かれます)は、2 つのブール オペランドを受け取り、そのいずれかが TRUE である場合に TRUE を返します。この演算子は通常、2 つの比較を接続するために使用されます。

最初のブール オペランドが TRUE の場合は、2 番目のブール オペランドがエバリュエータで評価されません(結果がすでに TRUE になっているためです)。

例:

```
(1 > 2) OR ("JUAN" ~= "JUAN")  
結果 = TRUE
```

```
(1 < 2) OR ("JUAN" ~= "JUAN")  
結果 = TRUE
```

排他的 OR 演算子

排他的 OR (XOR) 演算子は、2 つのブール オペランドを受け取り、そのいずれか一方のみが TRUE である場合に TRUE を返します。このオペレータは通常、2 つの比較を接続するために使用されます。

例:

```
(1 > 2) XOR ("JUAN" ~= "JUAN")  
結果 = TRUE
```

```
(1 < 2) XOR ("JUAN" ~= "JUAN")  
結果 = FALSE
```

文字列連結演算子

文字列連結演算子 (+) は、その 2 つの文字列オペランドを組み合わせた文字列を返します。

1 番目のオペランドが文字列であるが、2 番目のオペランドは数値またはブールである場合は、2 番目のオペランドが文字列に変換されます。1 番目のオペランドが数値である場合は、演算子 (+) が、連結ではなく、加算を表します。

例:

```
"JUAN" + " " + "Jones"  
結果 = "JUAN Jones"
```

```
"JUAN" + 2  
結果 = "JUAN2"
```

算術加算演算子

算術加算 (+) 演算子は、2 つの数値オペランドの合計を返します。

1 番目のオペランドが数値であるが、2 番目のオペランドは文字列またはブールである場合は、2 番目のオペランドが数値に変換されます。

例:

1 + 2
結果 = 3

1 + "JUAN"
結果 = 1

1 + "32JUAN"
結果 = 33

算術減算演算子

算術減算 (-) 演算子は、2 つの数値オペランドの差を返します。

1 番目のオペランドが数値であるが、2 番目のオペランドは文字列またはブールである場合は、2 番目のオペランドが数値に変換されます。

例:

1 ~ 2
結果 = -1

1 - "JUAN"
結果 = 1

100 - "32JUAN"
結果 = 68

算術乗算演算子

算術乗算 (*) 演算子は、2 つのオペランドの積を返します。

1 番目のオペランドが数値であるが、2 番目のオペランドは文字列またはブールである場合は、2 番目のオペランドが数値に変換されます。

例:

```
1 * 2  
結果 = 2
```

```
1 * "JUAN"  
結果 = 0
```

```
100 * "32JUAN"  
結果 = 3200
```

算術除算演算子

算術除算 (/) 演算子は、2 つのオペランドの商を返します。

1 番目のオペランドが数値であるが、2 番目のオペランドは文字列またはブールである場合は、2 番目のオペランドが数値に変換されます。

除算はすべて、整数による除算です。

注: ゼロによる除算は、算術として定義されていないため、この環境では常に、結果がエラーになります。

例:

```
1 / 2  
結果 = 0
```

```
1 / "JUAN"  
結果 = 0
```

```
100 / "32JUAN"  
結果 = 3
```

条件付き演算子

条件付き演算子は、ブール式(1 番目のオペランド)を評価し、その結果に基づいて、他の 2 つのオペランドのいずれかを返します。

ブール オペランドが **TRUE** である場合は、演算結果が 2 番目のオペランド (**THEN** 節)になります。それ以外の場合は、3 番目のオペランド (**ELSE** 節)が返されます。2 番目と 3 番目のオペランドは同じ型である必要があります。

この演算子で評価されるのは常に、1 つの節のみです。つまり、**THEN** 節(オペランド)が評価される場合は、**ELSE** 節(オペランド)は評価されません。(この逆の場合もあります。)

例:

```
"JUAN" = "juan" ? "YES" : "NO"  
結果 = "NO"
```

```
"JUAN" ~= "juan" ? "YES" : "NO"  
結果 = "YES"
```

インデックス演算子

インデックス演算子は、セットおよび数値(インデックス)という 2 つの引数を受け取ります。セットが要素に分解され、指定したインデックスに対応する要素が返されます(従来の配列と同様)。インデックスが範囲外である場合は、空の文字列が返されます。

セット内の 1 番目の要素はインデックス ゼロ(0)にあります。

例:

```
"Sun^Mon^Tue^Wed^Thu^Fri^Sat"[2]  
結果 = "Tue"
```

```
"Sun^Mon^Tue^Wed^Thu^Fri^Sat"[0]  
結果 = "Sun"
```

式の中に使用できる関数

このトピックでは、使用領域ごとに、すべての関数のリストを記載しています。

数値関数

- ABS
- ALL
- ANDBITS
- ANY
- HEX
- MAX
- MIN
- MOD
- NOTBITS
- ORBITS
- SIGN
- XORBITS

文字列関数

- AFTER
- BEFORE
- CENTER
- CHAR
- FIND
- LCASE
- LEFT
- LEN
- LPAD
- LTRIM
- MID
- PCASE

- RIGHT
- RPAD
- RPT
- RTRIM
- SPACE
- TRANSLATE
- UCASE

セット関数

- COUNT
- ENUMERATE
- FILTER
- LOOP
- SORT

日付関数

- DATE (形式 1)
- DATE (形式 2)
- DATEFROMSTRING
- DATETOSTRING
- DAY
- DOW
- DOY
- HOUR
- HOUR24
- MINUTE
- MONTH
- NOW
- NOWGMT
- SECOND

- YEAR
- YEAR4

変換関数

- BOOLEAN
- NUMBER
- STRING

汎用ユーザ ディレクトリ I/O 関数

- GET
- SET

LDAP 関数

- ABOVE
- AT
- BELOW
- COMMONDN
- EXPLODEDN
- PARENTDN
- RDN
- RELATIONDN

URL/パス処理関数

- QS
- URL
- URLDECODE
- URLENCODE

ログ記録関数

- ERROR
- INFO
- TRACE
- WARNING

ファイル I/O 関数

- EXISTS
- KEY
- LOG

エラー処理関数

- MAYBE
- THROW
- VEXIST

ユーザ コンテキスト管理関数

- WITH

その他の関数

- EVALUATE

ABOVE 関数 -- 指定した LDAP DN より上にあるユーザ

指定されたユーザ (`user_DN`) が、指定されたコンテナ (`root_DN`) より上にあるコンテナ内に存在する場合、ABOVE 関数は TRUE を返します。

構文

ABOVE 関数の構文は以下のとおりです。

```
ABOVE(root_DN, user_DN)
```

パラメータ

ABOVE 関数は、以下のパラメータを受け取ります。

`root_DN` (文字列)

`user_DN` (文字列)

戻り値

ABOVE 関数はブール値を返します。

注釈

LDAP のみ: はい

ABS 関数 -- 絶対値の取得

ABS 関数は、数値の絶対値を検索します。

構文

ABS 関数の構文は以下のとおりです。

ABS(*number*)

パラメータ

ABS 関数は、以下のパラメータを受け取ります。

number (数値)

戻り値

ABS 関数は数値を返します。

例

```
Return_value=ABS (3)  
Return_value=3
```

```
Return_value=ABS (-2)  
Return_value=2
```

AFTER 関数 -- 文字列の取得

AFTER 関数は、ソース文字列中に存在する、指定された検索文字列を検索します。そして、その検索文字列の後に続く部分を返します。検索文字列が見つからない場合、AFTER 関数は空の文字列を返します。

構文

AFTER 関数の構文は以下のとおりです。

```
AFTER(source_string, search_string[, not_case_sensitive][, n])
```

パラメータ

AFTER 関数は、以下のパラメータを受け取ります。

source_string (文字列)

search_string (文字列)

not_case_sensitive (ブール)

(オプション) 大文字と小文字を区別するかどうかを指定します。

not_case_sensitive フラグを省略するか、FALSE に設定した場合は、ソース文字列中で、正確に一致する文字列が検索されます。*not_case_sensitive* フラグを TRUE に設定した場合は、大文字と小文字が区別されません。

n (数値)

(オプション) ソース文字列中で検索する検索文字列を指定します。*n* をゼロまたは 1 に設定するか、省略した場合は、ソース文字列にある最初の検索文字列が返されます。それ以外の場合は、ソース文字列内の *n* 番目の検索文字列が返されます。*n* が負の数値の場合は、ソース文字列の末尾から検索を開始します。

戻り値

AFTER 関数は文字列を返します。

例

```
Return_value=AFTER('EricEric', 'r')
Return_value='icEric'

Return_value=AFTER('EricEric', 'R')
Return_value=""

Return_value=AFTER('EricEric', 'R', TRUE)
Return_value='icEric'

Return_value=AFTER('EricEric', 'r', -1)
Return_value='ic'

Return_value=AFTER('EricEric', 'R', -1)
Return_value=""

Return_value=AFTER('EricEric', 'R', TRUE, -1)
Return_value='ic'

Return_value=AFTER('EricEric', 'r', 2)
Return_value='ic'

Return_value=AFTER('EricEric', 'R', 2)
Return_value=""

Return_value=AFTER('EricEric', 'R', TRUE, 2)
Return_value='ic'
```

ALL 関数 -- 設定されたすべてのビット

ALL 関数は 2 つの数値を受け取り、その内の 2 番目の数値の中に設定されたビットがすべて、1 番目の数値の中にも設定されている場合に、TRUE を返します。

構文

ALL 関数の構文は以下のとおりです。

```
ALL(number1, number2)
```

パラメータ

ALL 関数は、以下のパラメータを受け取ります。

number1 (数値)

number2 (数値)

戻り値

ALL 関数はブール値を返します。

例

```
Return_value=ALL(7, 2)
```

```
Return_value=TRUE
```

```
Return_value=ALL(7, 15)
```

```
Return_value=FALSE
```

ANDBITS 関数 -- ビット単位の AND 演算の実行

ANDBITS 関数は、2 つの数値に対してビット単位の処理を実行します。

構文

ANDBITS 関数の構文は以下のとおりです。

```
ANDBITS (number1,number2)
```

パラメータ

ANDBITS 関数は、以下のパラメータを受け取ります。

number1 (数値)

number2 (数値)

戻り値

ANDBITS 関数は数値を返します。

例

```
Return_value=ANDBITS(7,2)
```

```
Return_value=2
```

```
Return_value=ANDBITS(7,15)
```

```
Return_value=7
```

ANY 関数 -- 設定されたいずれかのビット

ANY 関数は 2 つの数値を受け取り、その内の 2 番目の数値の中に設定されたビットのいずれかが、1 番目の数値の中にも設定されている場合に、TRUE を返します。

構文

ANY 関数の構文は以下のとおりです。

```
ANY(number1, number2)
```

パラメータ

ANY 関数は、以下のパラメータを受け取ります。

number1 (数値)

number2 (数値)

戻り値

ANY 関数はブール値を返します。

例

```
Return_value=ANY(7, 2)
```

```
Return_value=TRUE
```

```
Return_value=ANY(7, 15)
```

```
Return_value=TRUE
```

AT 関数 -- 指定された LDAP DN にあるユーザ

指定されたユーザ (`user_DN`) が、指定されたコンテナ (`root_DN`) 内に存在する場合、AT 関数は `TRUE` を返します。

構文

AT 関数の構文は以下のとおりです。

```
AT(root_DN, user_DN)
```

パラメータ

AT 関数は、以下のパラメータを受け取ります。

root_DN (文字列)

user_DN (文字列)

戻り値

AT 関数はブール値を返します。

注釈

LDAP のみ: はい

BEFORE 関数 -- 文字列の取得

BEFORE 関数は、ソース文字列中に存在する、指定された検索文字列を検索します。そして、その検索文字列の前にある部分を返します。検索文字列が見つからない場合、BEFORE 関数はソース文字列全体を返します。

構文

BEFORE 関数の構文は以下のとおりです。

```
BEFORE(source_string, search_string[, not_case_sensitive][, n])
```

パラメータ

BEFORE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

search_string (文字列)

not_case_sensitive (ブール)

大文字と小文字を区別することを指定します。*not_case_sensitive* フラグを **FALSE** に設定するか、省略した場合は、ソース文字列中で、正確に一致する文字列が検索されます。*not_case_sensitive* フラグを **TRUE** に設定した場合は、大文字と小文字が区別されません。

n (数値)

ソース文字列中で検索する検索文字列を指定します。*n* をゼロまたは **1** に設定するか、省略した場合は、ソース文字列にある最初の検索文字列が返されます。それ以外の場合は、ソース文字列内の *n* 番目の検索文字列が返されます。*n* が負の数値の場合は、ソース文字列の末尾から検索を開始します。

戻り値

BEFORE 関数は文字列を返します。

例

```
Return_value=BEFORE('EricEric', 'r')
```

```
Return_value='E'
```

```
Return_value=BEFORE('EricEric', 'R')
```

```
Return_value='EricEric'
```

```
Return_value=BEFORE('EricEric', 'R', TRUE)
```

```
Return_value='E'
```

```
Return_value=BEFORE('EricEric', 'r', -1)
```

```
Return_value='EricE'
```

```
Return_value=BEFORE('EricEric', 'R', -1)
```

```
Return_value='EricEric'
```

```
Return_value=BEFORE('EricEric', 'R', TRUE, -1)
```

```
Return_value='EricE'
```

```
Return_value=BEFORE('EricEric', 'r', 2)
Return_value='EricE'
```

```
Return_value=BEFORE('EricEric', 'R', 2)
Return_value='EricEric'
```

```
Return_value=BEFORE('EricEric', 'R', TRUE, 2)
Return_value='EricE'
```

BELOW 関数 -- 指定した LDAP DN より下にあるユーザ

指定されたユーザ (*user_DN*) が、指定されたコンテナ (*root_DN*) より下にあるコンテナ内に存在する場合、BELOW 関数は TRUE を返します。

構文

BELOW 関数の構文は以下のとおりです。

```
BELOW(root_DN, user_DN)
```

パラメータ

BELOW 関数は、以下のパラメータを受け取ります。

root_DN (文字列)

user_DN (文字列)

戻り値

BELOW 関数はブール値を返します。

注釈

LDAP のみ: はい

BOOLEAN 関数 -- 「TRUE」または「FALSE」への切り替え

BOOLEAN 関数は、数値または文字列値を、「TRUE」または「FALSE」のいずれかの文字列値に変換します。ゼロの数値は「FALSE」に変換されます。他の数値はすべて、「TRUE」に変換されます。「TRUE」または「YES」の文字列値は「TRUE」に変換されます。他の文字列値はすべて、「FALSE」に変換されます。BOOLEAN 関数では、大文字と小文字は区別されません。

構文

BOOLEAN 関数の構文は以下のとおりです。

```
Boolean(number | string)
```

パラメータ

BOOLEAN 関数は、次の 2 つのパラメータのいずれかを受け取ります。

number (数値)

string (文字列)

戻り値

BOOLEAN 関数は、次の文字列値のいずれかを返します。

- TRUE
- FALSE

例

```
Return_value=BOOLEAN('Phoebe')  
Return_value="FALSE"
```

```
Return_value=BOOLEAN('Yes')  
Return_value="TRUE"
```

```
Return_value=BOOLEAN(123)  
Return_value="TRUE"
```

```
Return_value=BOOLEAN(0)  
Return_value="FALSE"
```

CHAR 関数 -- ASCII 値の変換

CHAR 関数は、指定された ASCII 値を、1 文字の文字列に変換します。

構文

CHAR 関数の構文は以下のとおりです。

CHAR(*ASCII_value*)

パラメータ

CHAR 関数は、以下のパラメータを受け取ります。

ASCII_value (数値)

ASCII 値を指定します。*ASCII_value* がゼロの場合は、空の文字列が返されます。*ASCII_value* が 255 を超える場合は、*ASCII_value* の係数 256 を、1 文字の文字列に変換します。

注: 値に係数 256 を実行することは、ビット単位 AND を 255 で実行することと同じです。

戻り値

CHAR 関数は、1 文字の文字列を返します。

例

```
Return_value=CHAR(0)  
Return_value=""
```

```
Return_value=CHAR(36)  
Return_value='$'
```

```
Return_value=CHAR(299)  
Return_value='+'
```


CENTER 関数 -- ソース文字列の長さの調整

CENTER 関数は、指定した文字をソース文字列の前後に追加して、指定した長さの文字列にします。追加しても長さが合わない場合は、文字列の後に、さらに 1 文字を追加します。

構文

CENTER 関数の構文は以下のとおりです。

```
CENTER(source, length[, padding])
```

パラメータ

CENTER 関数は、以下のパラメータを受け取ります。

source (文字列または数値)

ソース文字列を指定します。関数によって自動的に、数値が文字列に変換されます。

length (数値)

文字を追加した後の文字列の長さを指定します。

padding (文字列)

(オプション) ソース文字列の前後に追加される文字を指定します。

- *padding* が複数の文字の場合は、最初の文字が使用されます。
- *padding* が空の文字列の場合は、ソースに文字が追加されません。
- *padding* が省略されており、ソースが文字列の場合は、その前後にスペースが追加されます。
- *padding* が省略されており、ソースが数値の場合は、その前後にゼロが追加されます。

戻り値

CENTER 関数は文字列を返します。

例

```
Return_value=CENTER('Robin', 9, '*')
Return_value='**Robin**'

Return_value=CENTER('Robin', 7, 'ooo')
Return_value='oRobino'

Return_value=CENTER('Robin', 7, '')
Return_value='Robin'

Return_value=CENTER('Robin', 11)
Return_value=' Robin'

Return_value=CENTER(123, 9)
Return_value='000123000'
```

COMMONDN 関数 -- 共通ルートの取得

COMMONDN 関数は、LDAP サーバを呼び出すことなく、2 つの LDAP 識別名 (DN) の共通ルートを返します。

構文

COMMONDN 関数の構文は以下のとおりです。

```
COMMONDN(ldapdn1, ldapdn2)
```

パラメータ

COMMONDN 関数は、以下のパラメータを受け取ります。

ldapdn1 (文字列)

LDAP 識別名 (DN) を指定します。

ldapdn2 (文字列)

LDAP 識別名 (DN) を指定します。

注: *ldapdn1* と *ldapdn2* のいずれかが有効な LDAP DN でない場合、または、2 つの DN が共通ルートを共有していない場合は、空の文字列が返されます。LDAP DN では、大文字と小文字が区別されません。

戻り値

COMMONDN 関数は文字列を返します。

注釈

LDAP のみ: はい

例

```
Return_value=COMMONDN('uid=Vincent,o=NDS.com', 'ou=People,o=nds.com')
Return_value='o=NDS.com'
```

COUNT 関数 -- セット内の要素の数

COUNT 関数で、セット内の要素の数がわかります。

構文

COUNT 関数の構文は以下のとおりです。

```
COUNT(set[, case_sensitive])
```

パラメータ

COUNT 関数は、以下のパラメータを受け取ります。

set (文字列)

「element1^element2」のように、一連の要素をキャレット文字で区切って指定します。各要素は文字列です。

case_sensitive (ブール)

(オプション) 大文字と小文字を区別するかどうかを指定します。

- *case_sensitive* フラグを省略した場合は、すべての要素の数が返されます。
- *case_sensitive* フラグを指定した場合は、一意の要素のみの数が返されます。
- *case_sensitive* フラグが TRUE の場合は、大文字と小文字が区別されます。
- *case_sensitive* フラグが FALSE の場合は、大文字と小文字が区別されません。

戻り値

COUNT 関数は数値を返します。

例

```
Return_value=COUNT('phoebe^PHOEBE^robin^robin')  
Return_value=4
```

```
Return_value=COUNT('phoebe^PHOEBE^robin^robin', FALSE)  
Return_value=2
```

```
Return_value=COUNT('phoebe^PHOEBE^robin^robin', TRUE)  
Return_value=3
```

DATE 関数 -- 午前 0 時(形式 1)への設定

DATE 関数(形式 1)は、数値で表された日付と時間を受け取り、時間を、指定した日付の午前 0 時に設定します。

構文

DATE 関数(形式 1)の構文は以下のとおりです。

```
DATE(date_time)
```

パラメータ

DATE 関数(形式 1)は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。

date_time が、日付と時間を表す有効な数値でない場合は、-1 が返されます。

戻り値

DATE 関数(形式 1)は数値を返します。

DATE 関数 -- 年、月、日、時、分、および秒(形式 2)の変換

DATE 関数(形式 2)は、年、月、日、時、分、および秒を表す 6 つの数値を受け取り、数値で表した日付と時間に変換します。日付と時間は、1970 年 1 月 1 日から経過した秒の数として、数値で表されます。3 つの時間パラメータを省略した場合は、指定した日付の真夜中が時間として設定されます。

構文

DATE 関数(形式 2)の構文は以下のとおりです。

DATE(year, month, day[, hours, minutes, seconds])

パラメータ

DATE 関数(形式 2)は、以下のパラメータを受け取ります。

year(数値)

4 桁で表した年を指定します。

例: 2007

month(数値)

月を指定します。

範囲: 1 ~ 12

day(数値)

日付を指定します。

範囲: 1 ~ 31

hours(数値)

(オプション) 時間の数値を指定します。

範囲: 0 ~ 23

minutes(数値)

(オプション) 分の数値を指定します。

範囲: 0 ~ 59

seconds(数値)

(オプション) 秒の数値を指定します。

範囲: 0 ~ 59

戻り値

DATE 関数(形式 2) は数値を返します。

DATEFROMSTRING 関数 -- 数値への文字列の変換

DATEFROMSTRING 関数は、文字列で表された日付、時間、またはその両方を受け取り、その文字列を数値に変換します。日付と時間は、1970 年 1 月 1 日から経過した秒数として、数値で表されます。

構文

DATEFROMSTRING 関数の構文は以下のとおりです。

DATEFROMSTRING(*date_time*[, *format_string*])

パラメータ

DATEFROMSTRING 関数は、以下のパラメータを受け取ります。

date_time (文字列)

format_string (文字列)

(オプション) *date_time* の形式を指定します。たとえば、「YYYYMMDD」という *format_string* は、「20020223」のような形式を指定します。*format_string* を省略した場合は、「YYYYMMDDhhmmss」というデフォルトの形式が使用されます。*date_time* が無効な場合は、-1 が返されます。

戻り値

DATEFROMSTRING 関数は数値を返します。

例

```
Return_value=DATEFROMSTRING('20020223')
Return_value=1024804800
```

DATETOSTRING 関数 -- 文字列への数値の変換

DATETOSTRING 関数は、数値で表された日付、時間、またはその両方を受け取り、数値を文字列に変換します。

構文

DATETOSTRING 関数の構文は以下のとおりです。

```
DATETOSTRING(date_time[, format_string])
```

パラメータ

DATETOSTRING 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付、時間、またはその両方を、**1970 年 1 月 1 日**から経過した秒の数として指定します。*date_time* が無効な場合は、値「INVALID DATE」が返されます。

format_string (文字列)

(オプション) テーブル内のリストに含まれる形式コードを使用して、返される文字列にある日付、時間、またはその両方の形式を指定します。テーブルのリストに含まれない文字、または文字の組み合わせはいずれも、返される文字列内に、そのまま挿入されます。*format_string* を省略した場合は、ポリシー サーバに現在格納されているロケールに適した形式の日付および時間が、返される文字列の形式として使用されます。形式コードと、返される日時の形式は、次のとおりです。

コード	返される日時の形式
%a	曜日 (短縮表示)

コード	返される日時の形式
%A	曜日 (完全表示)
%b	月 (短縮表示)
%B	月 (完全表示)
%c	現在のロケールに適した形式の日付および時間
%#c	現在のロケールに適した完全な形式の日付および時間
%d	日付 (10 進表記、01 ~ 31)
%H	時 (24 時間形式、00 ~ 23)
%I	時 (12 時間形式、01 ~ 12)
%j	日付 (10 進表記、001 ~ 366)
%m	月 (10 進表記、01 ~ 12)
%M	分 (10 進表記、00 ~ 59)
%P	各地域の午前/午後表示 (12 時間形式の場合)
%S	秒 (10 進表記、00 ~ 59)
%U	週 (10 進表記、00 ~ 53、週の初日は日曜日)
%w	曜日 (10 進表記、0 ~ 6、0 は日曜日)
%W	週 (10 進表記、00 ~ 53、週の初日は月曜日)
%x	現在のロケールに適した形式の日付
%#x	現在のロケールに適した完全な形式の日付
%X	現在のロケールに適した形式の時間
%y	年 (2 桁の 10 進表記、00 ~ 99)
%Y	年 (4 桁の 10 進表記)
%z、%Z	タイムゾーン名またはその短縮名 (わかっている場合)
%%	パーセント記号

注: 番号記号(#)を使用すると、形式コードの意味が、以下のように変わります。

- 形式コード **%#c** で、現在のロケールに適した完全な形式の日付および時間が指定されます。
- 形式コード **%#x** で、現在のロケールに適した完全な形式の日付が指定されます。
- 上記以外の場合はすべて、番号記号(形式コードでパーセント記号の後に入れられた)を使用すると、先頭のゼロが削除されます。
- 番号記号は、月の名前など、アルファベット文字の形式にするコードには効果がありません。

戻り値

DATETOSTRING 関数は文字列を返します。

例

```
Return_value=DATETOSTRING(1024804800, '%Y%m%d')  
Return_value='20020223'
```

```
Return_value=DATETOSTRING(1024804800, '%Y%#m%#d')  
Return_value='2002223'
```

DAY 関数 -- 月の日付の取得

DAY 関数は、数値で表された日付と時間を受け取り、その日付(月の)に対応する数値を返します。

構文

DAY 関数の構文は以下のとおりです。

```
DAY(date_time)
```

パラメータ

DAY 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。日付が無効な場合は、-1 が返されます。

戻り値

DAY 関数は、1 ～ 31 のいずれかの数値を返します。

例

```
Return_value=DAY(1024804800)  
Return_value=23
```

DOW 関数 -- 曜日の取得

DOW 関数は、数値で表された日付と時間を受け取り、その曜日に対応する数値を返します。

構文

DOW 関数の構文は以下のとおりです。

```
DOW(date_time)
```

パラメータ

DOW 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。日付が無効な場合は、-1 が返されます。

戻り値

DOW 関数は、0 (日曜) ～ 6 (土曜) のいずれかの数値を返します。

例

```
Return_value=DOW(1024804800)  
Return_value=0
```

DOY 関数 -- 年内の日付の取得

DOY 関数は、数値で表された日付と時間を受け取り、その日付(年内の)に対応する数値を返します。

構文

DOY 関数の構文は以下のとおりです。

```
DOY(date_time)
```

パラメータ

DOY 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。日付が無効な場合は、-1 が返されます。

戻り値

DOY 関数は、1 ～ 366 のいずれかの数値を返します。

例

```
Return_value=DOY(1024804800)  
Return_value=173
```

ENUMERATE 関数 -- セット要素のチェック

ENUMERATE 関数は、貼り付け機能を使用して、指定したセット内の各要素を、パラメータ *#virtual_user_attribute* または *@user_class* で指定された名前付き式に渡します。名前付き式が値を返す場合は、返される文字列中に要素が含まれています。

構文

ENUMERATE 関数の構文は以下のとおりです。

```
ENUMERATE(set, #virtual_user_attribute | @user_class)
```

パラメータ

ENUMERATE 関数は、以下のパラメータを受け取ります。

set (文字列)

「*element1*^*element2*」のように、一連の要素をキャレット文字で区切って指定します。各要素は文字列です。

#virtual_user_attribute (名前付き式)

ユーザ属性を計算する名前付き式を指定します。

@user_class (名前付き式)

ユーザ ディレクトリ内またはグループ内にメンバシップがあるかをテストする名前付き式を指定します。

戻り値

ENUMERATE 関数は文字列を返します。

例

以下の事項が真であることを前提としています。

- *#virtual_user_attribute* は *#RTrimset* です。
- *#RTrimset* で指定した名前付き式は組み込み関数 *RTRIM* です。
- *RTRIM* は、1 つのパラメータを受け取ります。
- このパラメータは、プレースホルダ *%0* で表されます。
- したがって、*#RTrimset =RTRIM (%0)* です。

```
Return_value=ENUMERATE('First ^Middle ^Last ',#RTrimset)
```

```
Return_value='First^Middle^Last'
```

ERROR 関数 -- コンソール ログへのエラー メッセージの書き込み

ERROR 関数は、指定したエラー メッセージを SiteMinder コンソール ログに書き込みます。

構文

ERROR 関数の構文は以下のとおりです。

```
ERROR(error_message)
```

パラメータ

ERROR 関数は、以下のパラメータを受け取ります。

error_message (文字列)

戻り値

ERROR 関数は空の文字列を返します。ブール コンテキスト中で ERROR 関数を使用すると、値 **TRUE** が返されます。

例

```
Return_value=ERROR('Invalid Access')  
Return_value=""
```

EVALUATE 関数 -- 式の評価

EVALUATE 関数は、現在のユーザのコンテキストで式を評価し、その結果を文字列で返します。(オプション) ユーザ パスを指定した場合は、指定したユーザのコンテキストで式が評価されます。

構文

EVALUATE 関数の構文は以下のとおりです。

```
EVALUATE([user_path, ]expression)
```

パラメータ

EVALUATE 関数は、以下のパラメータを受け取ります。

user_path (文字列)

(オプション) 現在のユーザ以外のユーザを指定します。

expression (文字列)

評価対象の式を指定します。

戻り値

EVALUATE 関数は文字列を返します。

注釈

特権: はい

例

```
Return_value=EVALUATE("sn + ',' + givenname")
```

```
Return_value="Hood, Robin"
```

```
Return_value=EVALUATE(manager, "sn + ',' + givenname")
```

```
Return_value="Webb, Charlotte"
```

EXISTS 関数 -- ファイル名の検索

EXISTS 関数は、指定したファイルを検索し、そのファイルが存在する場合に TRUE を返します。それ以外の場合は、FALSE が返されます。

構文

EXISTS 関数の構文は以下のとおりです。

```
EXISTS(filename)
```

パラメータ

EXISTS 関数はファイル名を受け取ります。

filename (文字列)

戻り値

EXISTS 関数はブール値を返します。

注釈

特権: はい

例

```
Return_value=EXISTS('SmUtilities.dll')
Return_value=TRUE
```

EXPLODEDN 関数 -- セットへの LDAP DN の変換

EXPLODEDN 関数は、LDAP サーバを呼び出さずに、LDAP 識別名 (DN) をセットに変換します。

構文

EXPLODEDN 関数の構文は以下のとおりです。

```
EXPLODED(ldapdn [, remove_attribute_names])
```

パラメータ

EXPLODEDN 関数は、以下のパラメータを受け取ります。

ldapdn (文字列)

LDAP 識別名 (DN) を指定します。

remove_attribute_names (ブール)

(オプション) *remove_attribute_names* フラグを指定します。フラグが TRUE の場合は、LDAP 識別名 (DN) から属性名が削除されます。

戻り値

EXPLODEDN 関数は、「*element1*^*element2*」のように、キャレット文字で区切られた一連の要素を返します。各要素は文字列です。

注釈

LDAP のみ: はい

例

```
Return_value=EXPLODEDN('uid=hawk,o=NDS.com', FALSE)
Return_value='uid=hawk^o=NDS.com'

Return_value=EXPLODEDN('uid=hawk,o=NDS.com', TRUE)
Return_value='hawk^NDS.com'
```

FILTER 関数 -- セット要素のチェック

FILTER 関数は、指定したセット内の各要素を、指定したパターンと比較し、パターンと一致する要素のみを含む新しいセットを返します。オプションの **NOT** 演算子が含まれている場合は、パターンに一致しない要素のみを含む新しいセットが返されます。

構文

FILTER 関数の構文は以下のとおりです。

```
FILTER(set, [NOT ]pattern)
```

パラメータ

FILTER 関数は、以下のパラメータを受け取ります。

set (文字列)

「element1^element2」のように、一連の要素をキャレット文字で区切って指定します。各要素は文字列です。

pattern (文字列)

文字のパターンを指定します。パターンには、単一の文字、一連の文字、またはその両方を含めることができます。一連の文字は、ハイフンで区切った 2 文字として表記します。「0-9」、「a-z」、および「A-Z」は、有効な一連の文字の例です。パターンパラメータには、予約済みの、特別な意味を持つ文字を含めることもできます。予約済みの文字は次のとおりです。

文字	意味
?	任意の 1 文字と一致します。
*	任意の数 (1 または 0 を含む) の文字と一致します。
[set]	指定したセット内の任意の 1 文字と一致します。

文字	意味
[!set] または [^set]	指定したセットにない任意の 1 文字と一致します。
¥	¥* などの予約済みの文字をすべて、標準文字として扱います。

NOT (演算子)

(オプション) NOT 演算子が含まれている場合は、パターンに一致しない要素のみを含む新しいセットが返されます。

戻り値

FILTER 関数は、「element1^element2」のように、キャレット文字で区切られた一連の要素を返します。各要素は文字列です。

例

```
Return_value=FILTER('Faith^Earl^Emilie^Fred', 'E*')
Return_value='Earl^Emilie'
```

```
Return_value=FILTER('Faith^Earl^Emilie^Fred', NOT 'E*')
Return_value='Faith^Fred'
```

FIND 関数 -- 文字列内の位置の取得

FIND 関数は、ソース文字列中で、指定した検索文字列を検索して、その位置を返します。検索文字列が見つからない場合は、ゼロが返されます。

構文

FIND 関数の構文は以下のとおりです。

```
FIND(source_string, search_string[, not_case_sensitive][, n])
```

パラメータ

FIND 関数は、以下のパラメータを受け取ります。

source_string (文字列)

search_string (文字列)

not_case_sensitive (ブール)

(オプション) 大文字と小文字を区別するかどうかを指定します。

not_case_sensitive フラグを省略するか、**FALSE** に設定した場合は、ソース文字列中で、正確に一致する文字列が検索されます。*not_case_sensitive* フラグを **TRUE** に設定した場合は、大文字と小文字が区別されません。

n (数値)

(オプション) ソース文字列中で検索する検索文字列を指定します。*n* をゼロまたは **1** に設定するか、省略した場合は、ソース文字列にある最初の検索文字列が返されます。それ以外の場合は、ソース文字列内の *n* 番目の検索文字列が返されます。*n* が負の数値の場合は、ソース文字列の末尾から検索を開始します。

戻り値

FIND 関数は数値を返します。

例

```
Return_value=FIND('PhoebePhoebe', 'oe', FALSE, 1)  
Return_value=3
```

```
Return_value=FIND('PhoebePhoebe', 'OE', FALSE, 1)  
Return_value=0
```

```
Return_value=FIND('PhoebePhoebe', 'OE', TRUE, 1)  
Return_value=3
```

```
Return_value=FIND('PhoebePhoebe', 'oe', FALSE, -1)  
Return_value=9
```

```
Return_value=FIND('PhoebePhoebe', 'OE', FALSE, -1)  
Return_value=0
```

```
Return_value=FIND('PhoebePhoebe', 'OE', TRUE, -1)  
Return_value=9
```

```
Return_value=FIND('PhoebePhoebe', 'oe', FALSE, 2)  
Return_value=9
```

```
Return_value=FIND('PhoebePhoebe', 'OE', FALSE, 2)  
Return_value=0
```

```
Return_value=FIND('PhoebePhoebe', 'OE', TRUE, 2)  
Return_value=9
```

GET 関数 -- ユーザ ディレクトリでの属性の検索

GET 関数は、指定した属性 (1 つ以上の) をユーザ ディレクトリ内で検索し、属性値を返します。複数の属性値は、キャレット文字で区切られます。属性が見つからない場合は、空の文字列が返されます。

構文

GET 関数の構文は以下のとおりです。

```
GET(user_attribute_name | user_attributes_string)
```

パラメータ

GET 関数は、次のパラメータのいずれかを受け取ります。

user_attribute_name (引用符を含まない文字列)

単一のユーザ属性を指定します。

user_attributes_string (文字列)

文字で区切った、一連のユーザ属性名を指定します。返される文字列内の各属性値は、この文字で区切られています。

戻り値

GET 関数は文字列を返します。

注釈

特権: はい

LDAP のみ: はい

例

```
Return_value=GET('sn,givenname')  
Return_value='Finch,Robin'
```

HEX 関数 -- 16 進数への変換

HEX 関数は、10 進数を 16 進数に変換し、それを文字列として返します。

構文

HEX 関数の構文は以下のとおりです。

```
HEX(decimal_number)
```

パラメータ

HEX 関数は、以下のパラメータを受け取ります。

decimal_number (数値)

戻り値

HEX 関数は文字列を返します。

例

```
Return_value=HEX(16)  
Return_value='10'
```

hour 関数 -- 時間への変換

hour 関数は、指定した日付と時間を、1 ～ 12 の時間の 1 つに変換します。

構文

hour 関数の構文は以下のとおりです。

```
hour(date_time)
```

パラメータ

hour 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。

date_time が、日付と時間を表す有効な数値でない場合は、-1 が返されます。

戻り値

hour 関数は、1 ～ 12 のいずれかの数値を返します。

hour24 関数 -- 時間への変換

hour24 関数は、指定した日付と時間を、0 ～ 23 の時間の 1 つに変換します。

構文

hour24 関数の構文は以下のとおりです。

```
hour24(date_time)
```

パラメータ

HOUR24 関数は、以下のパラメータを受け取ります。

date_time (数値)

日付と時間を、1970 年 1 月 1 日から経過した秒の数として指定します。

date_time が、日付と時間を表す有効な数値でない場合は、-1 が返されます。

戻り値

HOUR24 関数は、0 ～ 23 のいずれかの数値を返します。

INFO 関数 -- コンソール ログへの INFO メッセージの書き込み

INFO 関数は、INFO メッセージとして、SiteMinder コンソール ログに文字列引数を書き込みます。

構文

INFO 関数の構文は以下のとおりです。

INFO(*source_string*)

パラメータ

INFO 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

INFO 関数はブール値を返します。この値は常に TRUE です。

例

```
Return_value=INFO("86% Complete")
Return_value=TRUE
```

KEY 関数 -- キーの検索

KEY 関数は、指定したファイル内の指定したアプリケーション セクション内で、指定したキーの名前を検索して、キーの値を返します。キー、アプリケーション、およびファイルがいずれも見つからない場合は、空の文字列が返されます。

構文

KEY 関数の構文は以下のとおりです。

```
KEY(filename, [application_name, ]key_name)
```

パラメータ

KEY 関数は、以下のパラメータを受け取ります。

filename (文字列)

ファイルを指定します。指定したファイルでは、先頭がセミコロン、番号記号 (#)、または 2 つのスラッシュである行はコメント行です。コメント行、空白行、先頭スペース、および後続スペースは無視されます。

application_name (文字列)

(オプション)アプリケーション セクションの名前を指定します。指定したファイルで、アプリケーション名は角かっこで囲まれ (*[application_name]*)、アプリケーション セクションの先頭であることが指定されます。アプリケーション名では、大文字と小文字が区別されます。

key_name (文字列)

キーの名を指定します。指定したファイルでは、キーの名前と値が鍵かっこによって囲まれ、等号でつながれてペアを形成します (*<key_name>=<key_value>*)。各行には 1 つのペアが含まれます。キーの名前では、大文字と小文字が区別されます。

戻り値

KEY 関数は文字列を返します。

注釈

特権: はい

LDAP のみ: いいえ

例

```
Return_value=KEY('application.dat', 'login user')
Return_value='key_value'
```

LCASE 関数 -- 小文字への変換

LCASE 関数は、指定した文字列にある大文字をすべて、小文字に変換します。

構文

LCASE 関数の構文は以下のとおりです。

```
LCASE(specified_string)
```

パラメータ

LCASE 関数は、以下のパラメータを受け取ります。

specified_string (文字列)

戻り値

LCASE 関数は文字列を返します。

例

```
Return_value=LCASE('BARRED OWL')
Return_value='barred owl'
```


LEFT 関数 -- 文字列の一部を取得

LEFT 関数は、文字列内の、指定された数の文字を返します。文字列が、指定された文字数より短い場合は、文字列全体が返されます。

構文

LEFT 関数の構文は以下のとおりです。

```
LEFT(source_string, length)
```

パラメータ

LEFT 関数は、以下のパラメータを受け取ります。

source_string (文字列)

length (数値)

戻り値

LEFT 関数は文字列を返します。

例

```
Return_value=LEFT('JuanJuan', 2)  
Return_value='Ju'
```

```
Return_value=LEFT('JuanJuan', 10)  
Return_value=('JuanJuan')
```

```
Return_value=LEFT('JuanJuan', 0)  
Return_value=''
```

LEN 関数 -- 文字列の長さの取得

LEN 関数は、文字列の長さを返します。

構文

LEN 関数の構文は以下のとおりです。

```
LEN(source_string)
```

パラメータ

LEN 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

この関数は数値を返します。

例

```
Return_value=LEN("JuanJuan")  
Return_value=8
```

LOG 関数 -- ファイルへの文字列の書き込み

LOG 関数は、指定したファイルに文字列の引数を書き込みます。

構文

LOG 関数の構文は以下のとおりです。

LOG(filename, source_string)

パラメータ

LOG 関数は、以下のパラメータを受け取ります。

filename (文字列)

source_string (文字列)

戻り値

LOG 関数はブール値を返します。この値は常に TRUE です。

注釈

特権: はい

例

```
Return_value=LOG("auditlog.txt", "Accessing Realm XXX")
Return_value=TRUE
```

LOOP 関数 -- ループでの仮想属性の呼び出し

LOOP 関数は、指定された値から、指定された値まで、ループ内の数値ごとに仮想属性を呼び出します。手順の値を指定した場合は、数値が手順の値でインクリメントされます。仮想属性で、空でない値が返される場合は、値が結果セット内に含まれています。

構文

LOOP 関数の構文は以下のとおりです。

```
LOOP(start_value, end_value, [step,] #virtual_user_attribute)
```

パラメータ

LOOP 関数は、以下のパラメータを受け取ります。

start_value (数値)

end_value (数値)

step (数値)

(オプション) デフォルトは 1 です。負の値も指定できます。

#virtual_user_attribute (名前付き式)

定義された仮想属性の名前。仮想属性で、**%0** への参照を使用して、現在のループ カウンタにアクセスできます。

戻り値

LOOP 関数はセットを返します。

例

以下は、仮想ユーザ属性が `#Padset := LPAD(%0, 2)` の場合の例です。

```
Return_set=LOOP(1, 5, #Padset)
Return_set="001^002^003^004^005"
```

```
Return_set=LOOP(1, 5, 2, #Padset)
Return_set="001^003^005"
```

```
Return_set=LOOP(5, 1, -1, #Padset)
Return_set="005^004^003^002^001"
```

LPAD 関数 -- ソース文字列の左側での長さの調整

LPAD 関数は、ソース文字列が、指定した長さになるように、その左側に、指定した文字列の最初の文字を追加します。

構文

LPAD 関数の構文は以下のとおりです。

```
LPAD(source_string, length[, padding])
```

パラメータ

LPAD 関数は、以下のパラメータを受け取ります。

source_string (文字列)

このパラメータには数値も指定できます。この数値は自動的に、文字列に変換されます。

length (数値)

文字を追加した後の、文字列の文字数。

padding (文字列)

(オプション) 複数の文字を指定した場合は、最初の文字のみが使用されます。この文字列の長さがゼロの場合は、文字が追加されません。ソースが文字列の場合に、追加する文字の指定を省略すると、スペースが追加されます。ソースが数値の場合に、追加する文字の指定を省略すると、ゼロが追加されます。

戻り値

LPAD 関数は文字列を返します。

例

```
Result_value=LPAD('Juan', 5)  
Result_value=' Juan'
```

```
Result_value=LPAD('Juan', 5, 'X')  
Result_value='XJuan'
```

```
Result_value=LPAD('Juan', 6, 'XY')  
Result_value='XXJuan'
```

```
Result_value=LPAD(5, 2)  
Result_value='05'
```

```
Result_value=LPAD(5, 2, ' ')  
Result_value=' 5'
```

LTRIM 関数 -- 文字列の先頭にあるスペースの削除

LTRIM 関数は、*source_string* を表す文字列を、その先頭のスペースを削除して返します。

構文

LTRIM 関数の構文は以下のとおりです。

```
LTRIM(source_string)
```

パラメータ

LTRIM 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

LTRIM 関数は文字列を返します。

例

```
Return_value=LTRIM(' Juan ')  
Return_value='Juan '
```

MAX 関数 -- 2 つのうち、より大きい値の特定

MAX 関数は、2 つの数値の引数のうち、より大きい方を返します。

構文

MAX 関数の構文は以下のとおりです。

```
MAX(int_1, int_2)
```

パラメータ

MAX 関数は、以下のパラメータを受け取ります。

int_1 (数値)

int_2 (数値)

戻り値

MAX 関数は数値を返します。

例

```
Return_value=MAX(-2, 4)  
Return_value=4
```

MAYBE 関数 -- 不確定結果のレポート

式を作成して、条件をチェックできます。この式は、チェックに必要な情報が不足している、または不明な場合に、**MAYBE** 関数を呼び出します。式エバリュエータは、**MAYBE** を見つけると、必要な情報が不足している式を解決しようとしません。これは、**MAYBE** が複合式の一部である場合にのみ可能です。

たとえば、演算子が **AND** で、オペランドのいずれかが **FALSE** である場合は、エバリュエータが、演算結果は **FALSE** であると決定できます。2 つのオペランドの一方が **TRUE** で、他方が未定義の場合、または両方が未定義の場合は、エバリュエータが演算結果を決定できません。オペランドが両方とも **TRUE** の場合は、**AND** 演算の結果が **TRUE** になります。

AND 演算子	True	False	未定義
True	True	False	不確定
False	False	False	False
未定義	不確定	False	不確定

同様に、演算子が **OR** で、オペランドのいずれかが **TRUE** である場合は、エバリュエータが、演算結果は **TRUE** であると決定できます。2 つのオペランドの一方が **FALSE** で、他方が未定義の場合、または両方が未定義の場合は、エバリュエータが演算結果を決定できません。オペランドが両方とも **FALSE** の場合は、**OR** 演算の結果が **FALSE** になります。

OR 演算子	True	False	未定義
True	True	True	True
False	True	False	不確定
未定義	True	不確定	不確定

エバリュエータが式を解決できない場合は、処理が停止します。また、指定したメッセージが、コンテキストに応じて、コンソール ログまたはレポートに出力されます。**MAYBE** は通常、ポリシーのコンテキストおよびレポート生成時のいずれかで、ロール評価中に呼び出されます。

条件は通常、時刻または IP アドレスによって変わります。また、仮想ユーザ属性（# 記号で指定された）、ユーザ クラス（@ 記号で指定された）、コンテキスト変数（% 記号で指定された）、およびユーザ属性のいずれかの値によっても変わります。

注: LDAP ユーザ ディレクトリの場合は、ユーザ属性が定義されているかどうかをエバリュエータが判定できません。

構文

MAYBE 関数の構文は以下のとおりです。

MAYBE(message)

パラメータ

MAYBE 関数は、以下のパラメータを受け取ります。

message (文字列)

式で条件を評価するために必要な、不足している情報を指定します。

戻り値

MAYBE 関数は値を返しません。

例

VEXIST(%ClientIP) ? #CheckIP : MAYBE('Client IP address is not defined.')

コンソール ログまたはレポートに出力されるメッセージ: 「Client IP address is not defined.」

MID 関数 -- 文字列の一部を取得

MID 関数は、*source_string* 中の、*開始位置* (1 から番号が付けられた) から、指定した *長さ* までの文字を返します。 *長さ* を指定しない場合は、*source_string* の残りの文字 (*開始位置* の後の) が返されます。

構文

MID 関数の構文は以下のとおりです。

MID(*source_string*, *start* [, *length*])

パラメータ

MID 関数は、以下のパラメータを受け取ります。

source_string (文字列)

start (数値)

length (数値) (オプション)

戻り値

MID 関数は文字列を返します。

例

```
Return_value=MID('JuanJuan', 2, 3)  
Return_value='uan'
```

```
Return_value=MID('JuanJuan', 2)  
Return_value='uanJuan'
```

MIN 関数 -- 2 つのうち、より小さい値の特定

MIN 関数は、2 つの数値の引数のうち、より小さい方を返します。

構文

MIN 関数の構文は以下のとおりです。

```
MIN(int_1, int_2)
```

パラメータ

MIN 関数は、以下のパラメータを受け取ります。

int_1 (数値)

int_2 (数値)

戻り値

MIN 関数は数値を返します。

例

```
Return_value=MIN(-2, 4)  
Return_value=-2
```

MINUTE 関数 -- 日付の「分」要素の取得

MINUTE 関数は、指定した *date_time* (1970 年 1 月 1 日から経過した秒数で表された) の「分」の要素を表す数値を返します。

構文

MINUTE 関数の構文は以下のとおりです。

MINUTE(*date_time*)

パラメータ

MINUTE 関数は、以下のパラメータを受け取ります。

date_time (数値)

秒の数で表された日付。

戻り値

MINUTE 関数は、0 ～ 59 のいずれかの数値を返します。*date_time* が無効な場合は、-1 が返されます。

MOD 関数 -- 除算の剰余の取得

MOD 関数は、2 番目の数値による 1 番目の数値の除算の係数 (剰余) を返します。2 番目の数値がゼロの場合は、ゼロが返されます。

構文

MOD 関数の構文は以下のとおりです。

MOD(*int_1*, *int_2*)

パラメータ

MOD 関数は、以下のパラメータを受け取ります。

int_1 (数値)

除算演算の被除数。

int_2 (数値)

除算演算の除数。

戻り値

MOD 関数は数値を返します。

例

Return_value=MOD(3, 2)

Return_value=1

Return_value=MOD(6, 3)

Return_value =0

MONTH 関数 -- 日付の「月」要素の取得

MONTH 関数は、指定した *date_time* (1970 年 1 月 1 日から経過した秒数で表された) の「月」の要素を表す数値を返します。

構文

MONTH 関数の構文は以下のとおりです。

MONTH(*date_time*)

パラメータ

MONTH 関数は、以下のパラメータを受け取ります。

date_time (数値)

秒の数で表された日付。

戻り値

MONTH 関数は、1 ～ 12 のいずれかの数値を返します。 *date_number* が無効な場合は、-1 が返されます。

NOTBITS 関数 -- ビット単位の NOT の実行

NOTBITS 関数は、ビット単位の NOT 演算を数値に実行します。

構文

NOTBITS 関数の構文は以下のとおりです。

NOTBITS(*number*)

パラメータ

NOTBITS 関数は、以下のパラメータを受け取ります。

number (数値)

戻り値

NOTBITS 関数は数値を返します。

例

```
Return_value=NOTBITS(0)  
Return_value=-1
```

```
Return_value=NOTBITS(1)  
Return_value=-2
```

NOW 関数 -- 秒単位での現在の時間の取得

NOW 関数は、その呼び出し時に、1970 年 1 月 1 日から経過した秒の数を表す数値を返します。この時間は、現在のサーバシステムが存在する地域の時間です。

この値は、特定の式での演算の開始時に計算されます。同じ式の中で NOW 関数を複数回、参照した場合は、常に同じ結果となります。

構文

NOW 関数の構文は以下のとおりです。

NOW()

パラメータ

NOW 関数はパラメータを受け取りません。

戻り値

NOW 関数は数値を返します。

例

```
Return_value=NOW()  
Return_value=1024804800
```

NOWGMT 関数 -- 秒単位での現在の時間の取得

NOWGMT 関数は、その呼び出し時に、1970 年 1 月 1 日から経過した秒の数を表す数値を返します。この時間は、グリニッジ(ZULU)タイムゾーンの時間です。

この値は、特定の式での演算の開始時に計算されます。同じ式の中で NOWGMT 関数を複数回、参照した場合は、常に同じ結果となります。

構文

NOWGMT 関数の構文は以下のとおりです。

```
NOWGMT ( )
```

パラメータ

NOWGMT 関数はパラメータを受け取りません。

戻り値

NOWGMT 関数は数値を返します。

NUMBER 関数 -- 数値への変換

NUMBER 関数は、その引数を数値に変換します。文字列は、数字以外の最初の文字まで変換されます。値が **TRUE** のブールは **1** に、それ以外の値のブールはゼロに変換されます。

構文

NUMBER 関数の構文は以下のとおりです。

```
NUMBER(source_string | bool_val)
```

パラメータ

NUMBER 関数は、次のパラメータのいずれかを受け取ります。

source_string (文字列)

bool_val (ブール)

戻り値

NUMBER 関数は数値を返します。

例

```
Return_value=NUMBER('juan')
```

```
Return_value=0
```

```
Return_value=NUMBER('45juan')
```

```
Return_value=45
```

```
Return_value=NUMBER(TRUE)
```

```
Return_value=1
```

ORBITS 関数 -- ビット単位の OR 演算の実行

ORBITS 関数は、その 2 つの引数に、ビット単位の **OR** 演算を実行します。

構文

ORBITS 関数の構文は以下のとおりです。

```
ORBITS(int_1, int_2)
```

パラメータ

ORBITS 関数は、以下のパラメータを受け取ります。

int_1 (数値)

int_2 (数値)

戻り値

ORBITS 関数は数値を返します。

例

```
Result_value=ORBITS(6, 1)
```

```
Result_value=7
```

```
Result_value=ORBITS(7, 8)
```

```
Result_value=15
```

PARENTDN 関数 -- LDAP ツリー内の親の取得

PARENTDN 関数は、LDAP ディレクトリ情報ツリー内で、指定した識別名 (DN) の上にある次のレベルを返します。指定した DN が無効であるか、すでにツリーの最上部にある場合は、空の文字列が返されます。

構文

PARENTDN 関数の構文は以下のとおりです。

```
PARENTDN(source_string)
```

パラメータ

PARENTDN 関数は、以下のパラメータを受け取ります。

source_string (文字列)

LDAP 識別名 (DN)。

戻り値

PARENTDN 関数は文字列を返します。

注釈

LDAP のみ: はい

例

```
Return_value=PARENTDN("uid=juan,o=NDS.com")  
Return_value="o=NDS.com"
```

```
Return_value=PARENTDN("o=NDS.com")  
Return_value=""
```

PCASE 関数 -- 文字列の大文字への変換

PCASE 関数は、指定した文字列の先頭の文字を大文字に変換します。

構文

PCASE 関数の構文は以下のとおりです。

```
PCASE(source_string)
```

パラメータ

PCASE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

PCASE 関数は文字列を返します。

例

```
Return_value=PCASE("framingham, mass")  
Return_value="Framingham, Mass")
```


QS 関数 -- クエリ文字列からの項目の取得

QS 関数は、式の評価時に、ユーザがアクセスしているリソースに関連付けられたクエリ文字列から項目を取得します。

この関数に引数を指定しない場合は、クエリ文字列全体（およびクエリ文字列のみ）が返されます。クエリ文字列は、変更されることなく返されます。

引数の文字列に空の文字列("")を指定した場合は、クエリ文字列にある、名前のない引数がすべて返されます。値が複数ある場合は、それらが 1 セットとして返されます。

引数の文字列に、空でない文字列を指定した場合は、それに一致する名前の、クエリ文字列内の引数がすべて返されます。大文字と小文字の区別は、オプションのブールフラグによって制御されます。値が複数ある場合は、それらが 1 セットとして返されます。

構文

QS 関数の構文は以下のとおりです。

```
QS([input_string],[ not_case_sensitive])
```

パラメータ

QS 関数は、以下のパラメータを受け取ります。

input_string (文字列)

(オプション)クエリ文字列内の引数の名前。

not_case_sensitive (ブール)

(オプション)*not_case_sensitive* フラグを **FALSE** に設定するか、省略した場合は、クエリ文字列中で、正確に一致する文字列が検索されます。

not_case_sensitive フラグを **TRUE** に設定した場合は、大文字と小文字が区別されません。

戻り値

QS 関数は文字列を返します。

例

次のリソースを前提としています。

http://myserver.com/index.jsp?Test=A&X&TEST=D&c&Dbg

```
Return_value=QS()
```

```
Return_value='Test=A&X&TEST=D&c&Dbg'
```

```
Return_value=QS("")
```

```
Return_value='X^c'
```

```
Return_value=QS("Test")
```

```
Return_value= 'A^D'
```

```
Return_value=QS("Test", false)
```

```
Return_value= 'A'
```

```
"Dbg" IN QS("")
```

```
Return_value=TRUE
```

RDN 関数 -- LDAP DN の最初の要素の取得

RDN 関数は、指定した LDAP 識別名 (DN) の最初の要素を返します。オプションのブール引数が TRUE (デフォルト) の場合は、属性名が削除され、値のみが返されます。

指定した DN が無効な場合は、空の文字列が返されます。この関数は、LDAP サーバを呼び出しません。

構文

RDN 関数の構文は以下のとおりです。

```
RDN(DN_string[, remove_name])
```

パラメータ

RDN 関数は、以下のパラメータを受け取ります。

DN_string (文字列)

LDAP 識別名

remove_name

(オプション) **TRUE** (デフォルト) に設定すると、返される文字列から属性名が削除されます。 **FALSE** に設定すると、返される文字列中に属性が含まれます。

戻り値

RDN 関数は文字列を返します。

注釈

LDAP のみ: はい

例

```
Return_value=RDN("uid=juan,o=NDS.com")  
Return_value="juan"
```

```
Return_value=RDN("uid=juan,o=NDS.com", TRUE)  
Return_value="juan"
```

```
Return_value=RDN("uid=juan,o=NDS.com", FALSE)  
Return_value="uid=juan"
```

RELATIONDN 関数 --2 つの識別名の比較

RELATIONDN 関数は、指定した 2 つの LDAP 識別名 (DN) を比較して、それらのリレーションシップを示す文字列を返します。

2 つの DN のいずれかが無効か、それらが完全に無関係な場合は、空の文字列が返されます。

2 つの DN に関係がある場合は、それらの相違点のレベル(ディレクトリ情報ツリーの)が文字列として返されます。1 番目の DN が 2 番目の DN の先祖である場合は、正の数値です。1 番目の DN が 2 番目の DN の子孫である場合は、負の数値です。2 つの DN が同等または同種のものである場合は、0(レベルなしを示す)が返されます。

注: この関数は、LDAP サーバ関数を呼び出しません。

構文

RELATIONDN 関数の構文は以下のとおりです。

```
RELATIONDN(dn_1, dn_2)
```

パラメータ

RELATIONDN 関数は、以下のパラメータを受け取ります。

dn_1 (文字列)

dn_2 (文字列)

戻り値

RELATIONDN 関数は文字列を返します。

注釈

LDAP のみ: はい

例

```
Return_value=RELATIONDN("uid=eric,o=NDS.com", "o=NDS.com")  
Return_value="-1"
```

```
Return_value=RELATIONDN("o=NDS.com", "uid=eric,o=NDS.com")  
Return_value="1"
```

```
Return_value=RELATIONDN("uid=dave,o=NDS.com", "uid=eric,o=NDS.com")  
Return_value="0"
```

```
Return_value=RELATIONDN("uid=dave,o=XYZ.com", "uid=eric,o=NDS.com")  
Return_value=""
```

RIGHT 関数 -- 文字列からの文字の取得

RIGHT 関数は、文字列の末尾から、指定した数の文字を返します。文字列がこの数より短い場合は、文字列全体が返されます。

構文

RIGHT 関数の構文は以下のとおりです。

```
RIGHT(source_string, length)
```

パラメータ

RIGHT 関数は、以下のパラメータを受け取ります。

source_string (文字列)

length (数値)

文字列の末尾から数えた、抽出する文字の数。

戻り値

RIGHT 関数は文字列を返します。

例

```
Return_value=RIGHT('JuanJuan', 2)  
Return_value='an'
```

```
Return_value=RIGHT('JuanJuan', 10)  
Return_value='JuanJuan'
```

```
Return_value=RIGHT('JuanJuan', 0)  
Return_value=''
```

RPAD 関数 -- 右側の文字列の長さの調整

RPAD 関数は、ソース文字列が指定した長さになるまで、指定した文字列の最初の文字を、文字列の末尾に追加します。

指定した文字列が複数の文字である場合は、最初の文字のみが使用されます。この文字列の長さがゼロの場合は、文字が追加されません。

ソースが文字列の場合に、追加する文字の指定を省略すると、スペースが追加されます。ソースが数値の場合に、追加する文字の指定を省略すると、ゼロが追加されます。

構文

RPAD 関数の構文は以下のとおりです。

```
RPAD(source_string|number, length[, padding])
```

パラメータ

RPAD 関数は、以下のパラメータを受け取ります。

source_string (文字列)

このパラメータには数値も指定できます。この数値は、文字列に変換されます。

length (数値)

padding (文字列)

(オプション)

戻り値

RPAD 関数は文字列を返します。

例

```
Return_value=RPAD('Juan', 5)
Return_value='Juan '
```

```
Return_value=RPAD('Juan', 5, 'X')
Return_value='JuanX'
```

```
Return_value=RPAD('Juan', 6, 'XY')
Return_value='JuanXX'
```

```
Return_value=RPAD(5, 2)
Return_value='50'
```

```
Return_value=RPAD(5, 2, ' ')
Return_value='5 '
```

RPT 関数 -- 文字列の反復

RPT 関数は、ソース文字列の、指定した回数の繰り返しである文字列を返します。

構文

RPT 関数の構文は以下のとおりです。

```
RPT(source_string|number, repeat_count)
```

パラメータ

RPT 関数は、以下のパラメータを受け取ります。

source_string (文字列)

このパラメータには数値も指定できます。この数値は、単一の文字に変換されます。

repeat_count (文字列)

戻り値

RPT 関数は文字列を返します。

例

```
Return_value=RPT('Juan', 3)
Return_value='JuanJuanJuan')

Return_value=RPT('*', 10)
Return_value="*****')
```

RTRIM 関数 -- 文字列の後続スペースの削除

RTRIM 関数は、ソース文字列から後続スペースを削除して、その結果を返します。

構文

RTRIM 関数の構文は以下のとおりです。

```
RTRIM(source_string)
```

パラメータ

RTRIM 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

RTRIM 関数は文字列を返します。

例

```
Return_value=RTRIM(' JuanJuan ')
Return_value=' JuanJuan'
```


SECOND 関数 -- 日付の秒数の取得

SECOND 関数は、1970 年 1 月 1 日から経過した秒の数で表された、日付の秒の要素を表す値を返します。

構文

SECOND 関数の構文は以下のとおりです。

```
SECOND(date_time)
```

パラメータ

SECOND 関数は、以下のパラメータを受け取ります。

date_time (数値)

秒数を指定します。

戻り値

SECOND 関数は、0 ～ 59 のいずれかの数値を返します。

SET 関数 -- 属性値の設定

SET 関数は、指定した属性に、指定した値を割り当てます。複数值属性には、複数の値がセットで指定されます。この関数は、**SiteMinder** でサポートされているすべてのユーザ ディレクトリに有効です。

SiteMinder から変更の成功が返された場合は、SET 関数が **TRUE** を返します。

SiteMinder が属性を認識できない場合は、この関数が失敗します。セキュリティ上の理由 (ユーザ ディレクトリのセキュリティ) により、属性が認識されない場合があります。また、**ODBC** ディレクトリの場合は、設定したクエリにないため、または、**SiteMinder** クエリ方式のセット プロパティ設定のリストに含まれない属性であるために認識されないことがあります。

構文

SET 関数の構文は以下のとおりです。

```
SET(attr_name, value)
```

パラメータ

SET 関数は、以下のパラメータを受け取ります。

attr_name (文字列)

value (文字列)

1 つ以上の値を指定します。複数の値はキャレット文字で区切られます。

戻り値

SET 関数はブール値を返します。

注釈

特権: はい

例

```
Return_value=SET("Retries", STRING(NUMBER(Retries) + 1))
```

SIGN 関数 -- 数値の符号の取得

SIGN 関数は数値を受け取ります。数値が負の場合は、負符号が返されます。数値がゼロの場合は、ゼロが返されます。数値が正の場合は、正符号が返されます。

構文

SIGN 関数の構文は以下のとおりです。

```
SIGN(number)
```

パラメータ

SIGN 関数は、以下のパラメータを受け取ります。

number (数値)

戻り値

SIGN 関数は、数値 -1、0、または +1 を返します。

例

```
Return_value=SIGN(-40)  
Return_value=-1
```

```
Return_value=SIGN(0)  
Return_value=0
```

```
Return_value=SIGN(999)  
Return_value=1
```

SORT 関数 -- セットの並べ替え

SORT 関数は、指定したセットを並べ替えます。オプションのブール パラメータを使用すると、大文字と小文字の区別が指定されます。重複する項目は、返されるセットでは削除されています。

構文

SORT 関数の構文は以下のとおりです。

```
SORT(source_set[, not_case_sensitive])
```

パラメータ

SORT 関数は、以下のパラメータを受け取ります。

source_set (文字列)

並べ替えられるセット。

not_case_sensitive (ブール)

(オプション)このパラメータを省略するか、**FALSE** に設定した場合は、大文字と小文字の違いを除いて同じである項目が、同一として並べ替えられます。このパラメータを **TRUE** に設定した場合は、大文字と小文字の区別が無視されます。

戻り値

SORT 関数はセットを返します。

例

```
Return_value=SORT("Eric^Bart^Chuck^BART^Chuck")
Return_value="BART^Bart^Chuck^Eric"

Return_value=SORT("Eric^Bart^Chuck^BART^Chuck", FALSE)
Return_value="BART^Bart^Chuck^Eric"

Return_value=SORT("Eric^Bart^Chuck^BART^Chuck", TRUE)
Return_value="Bart^Chuck^Eric"
```

SPACE 関数 -- 一連のスペースの取得

SPACE 関数は、指定した数のスペースで構成された文字列を返します。

構文

SPACE 関数の構文は以下のとおりです。

```
SPACE(repeat_count)
```

パラメータ

SPACE 関数は、以下のパラメータを受け取ります。

repeat_count (数値)

文字列に含めるスペースの数。

戻り値

SPACE 関数は文字列を返します。

例

```
Return_value=SPACE(3) Return_value='   '
Return_value=SPACE(10) Return_value="          "
```

STRING 関数 -- 文字列への変換

STRING 関数は、数値またはブール値を文字列値に変換します。

構文

STRING 関数の構文は以下のとおりです。

```
STRING(num_value|bool_value)
```

パラメータ

STRING 関数は、以下のパラメータのいずれかを受け取ります。

num_value (数値)

bool_value (ブール)

戻り値

STRING 関数は文字列を返します。

例

```
Return_value=STRING(TRUEVAL)  
Return_value="TRUE"
```

```
Return_value=STRING(123)  
Return_value='123'
```

THROW 関数 -- 処理の停止およびカスタム エラーのレポート

式を作成して、エラーをチェックできます。この式は、エラーが発生している場合に、カスタム エラー メッセージを渡して **THROW** 関数を呼び出します。式エバリュエータは、**THROW** を見つけると、式の処理を停止して、コンソール ログにカスタム エラー メッセージを出力します。

構文

THROW 関数の構文は以下のとおりです。

```
THROW(error_message)
```

パラメータ

THROW 関数は、以下のパラメータを受け取ります。

error_message (文字列)

コンソール ログに出力されるカスタム エラー メッセージを指定します。

戻り値

THROW 関数は値を返しません。

例

```
EXISTS('MyFile') ? #Process('MyFile') : THROW('File does not exist.')
```

コンソール ログに出力されるメッセージ: 「File does not exist.」

```
VEXIST(#Sortname) ? #Sortname : THROW('Sortname is not defined.')
```

コンソール ログに出力されるメッセージ: 「Sortname is not defined.」

TRACE 関数 -- コンソール ログへのトレース エントリの書き込み

TRACE 関数は、トレース エントリとして、SiteMinder コンソール ログに文字列引数を書き込みます。

構文

TRACE 関数の構文は以下のとおりです。

```
TRACE(source_string)
```

パラメータ

TRACE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

TRACE 関数はブール値を返します。この値は常に TRUE です。

例

```
Return_value=TRACE("Executing Code")  
Return_value=TRUE
```

TRANSLATE 関数 -- 文字列値の置換

TRANSLATE 関数は、2 番目の文字列中で見つけた 1 番目の文字列をすべて、3 番目の文字列に置換します。文字列の検索では、大文字と小文字が区別されます。ただし、オプションのブールを TRUE に設定した場合は区別されません。

構文

TRANSLATE 関数の構文は以下のとおりです。

```
TRANSLATE(source_string, search_string, replace_string[, not_case_sensitive])
```

パラメータ

TRANSLATE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

search_string (文字列)

replace_string (文字列)

not_case_sensitive (ブール)

(オプション)このパラメータを省略するか、FALSE に設定した場合は、検索で、大文字と小文字が区別されます。TRUE に設定した場合は、大文字と小文字が区別されません。

戻り値

TRANSLATE 関数は文字列を返します。

例

```
Return_value=TRANSLATE('Eric','r','x')
Return_value='Exic'

Return_value=TRANSLATE('Eric','ri','x')
Return_value='Exc'

Return_value=TRANSLATE('Eric','r','xy')
Return_value='Exyic'

Return_value=TRANSLATE('Eric','R','x')
Return_value='Eric'

Return_value=TRANSLATE('Eric','R','x',TRUE)
Return_value='Exic'
```

UCASE 関数 -- 大文字への変換

UCASE 関数は、ソース文字列を大文字に変換します。

構文

UCASE 関数の構文は以下のとおりです。

```
UCASE(source_string)
```

パラメータ

UCASE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

UCASE 関数は文字列を返します。

例

```
Return_value=UCASE('framingham, mass')
Return_value='FRAMINGHAM, MASS'
```


URL 関数 -- URL 文字列の要素の取得

URL 関数は、指定した URL (またはパス) を解析し、指定した要素を返します。この関数は、URL エンコード文字を無視します。

注: URL 関数は、バックスラッシュではなく、スラッシュを使用する文字列のみを解析します。Windows を実行するシステムを使用している場合は、TRANSLATE 関数を使用してバックスラッシュをスラッシュに変換します。

構文

URL 関数の構文は以下のとおりです。

```
URL(url_string, component)
```

パラメータ

URL 関数は、以下のパラメータを受け取ります。

url_string (文字列)

次の構文の後に、URL またはパスを指定する必要があります。

```
[<protocol>:][//][<user>[:<password>]@]<server>[:<port#>]  
[/<directory>]/<file>][?<querystring>]
```

component (文字列)

要素の名前では、大文字と小文字は区別されません。以下から、任意の要素を指定できます。

- PROTOCOL、DRIVE、または NAMESPACE
- USER
- PASSWORD
- SERVER
- PORT
- DOMAIN
- URI
- PATH
- DIRECTORY
- FILENAME
- BASENAME
- EXTENSION
- QUERYSTRING

戻り値

URL 関数は文字列を返します。

例

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzyz.jsp?id=12",  
"PROTOCOL")  
Return_value="http:"  
  
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyz.zip", "USER")  
Return_value="joe"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"PASSWORD")  
Return_value="dog"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzzy.jsp?id=12",  
"SERVER")  
Return_value="www.myserver.com"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzzy.jsp?id=12",  
"PORT")  
Return_value="8080"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzzy.jsp?id=12",  
"DOMAIN")  
Return_value="myserver.com"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip", "URI")  
Return_value="/dir1/xyzzy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip", "PATH")  
Return_value="/dir1/xyzzy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"DIRECTORY")  
Return_value="/dir1"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"FILENAME")  
Return_value="xyzzy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"BASENAME")  
Return_value="xyzzy"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"EXTENSION")  
Return_value="zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip",  
"QUERYSTRING")  
Return_value=""  
  
Return_value=URL("http://www.myserver.com:8080/app1/xyzzy.jsp?id=12",  
"QUERYSTRING")  
Return_value="id=12"
```

URLDECODE 関数 --URL 文字列のデコード

URLDECODE 関数は、指定した URL 文字列をデコードします。

構文

URLDECODE 関数の構文は以下のとおりです。

```
URLDECODE(url)
```

パラメータ

この関数は、以下のパラメータを受け取ります。

url (文字列)

戻り値

URLDECODE 関数は文字列を返します。

例

```
Return_value=URLDECODE("Framingham%2C+Mass")  
Return_value="Framingham, Mass"
```

URLENCODE 関数 -- 文字列のエンコード

URLENCODE 関数は、渡された文字列を URL 形式にエンコードします。

構文

URLENCODE 関数の構文は以下のとおりです。

```
URLENCODE(source_string)
```

パラメータ

URLENCODE 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

URLENCODE 関数は文字列を返します。

例

```
Return_value=URLENCODE('Framingham, Mass')  
Return_value='Waltham%2C+Mass'
```

VEXIST 関数 -- パラメータ定義の判定

VEXIST 関数は、名前付き式、コンテキスト変数、またはユーザ属性を受け取り、それが定義されているかどうかを判定します。定義されている場合は、**TRUE** が返されます。それ以外の場合は、**FALSE** が返されます。

注: LDAP ユーザ ディレクトリの場合は、ユーザ属性が定義されているかどうかを SiteMinder が判定できないため、**FALSE** が返されます。

構文

VEXIST 関数の構文は以下のとおりです。

```
VEXIST(#virtual_user_attribute | @user_class | %context_variable |  
user_attribute_name | user_attribute_string)
```

パラメータ

VEXIST 関数は、次のパラメータのいずれかを受け取ります。

#virtual_user_attribute (名前付き式)

ユーザ属性を計算する名前付き式を指定します。

@user_class (名前付き式)

ユーザ ディレクトリ内またはグループ内にメンバシップがあるかをテストする
名前付き式を指定します。

%context_variable (コンテキスト変数)

コンテキスト変数を指定します。

user_attribute_name (引用符を含まない文字列)

単一のユーザ属性を指定します。

user_attributes_string (文字列)

文字で区切った、一連のユーザ属性名を指定します。

戻り値

VEXIST 関数はブール値を返します。

例

```
Return_value=VEXIST(#Age)  
Return_value=TRUE
```

```
Return_value=VEXIST(@IsDolphin)  
Return_value=FALSE
```

```
Return_value=VEXIST(%ClientIP)  
Return_value=TRUE
```

```
Return_value=VEXIST(givenname)  
Return_value=FALSE
```

```
Return_value=VEXIST('last,first')  
Return_value=TRUE
```

TRACE 関数 -- コンソール ログへのトレース エントリの書き込み

WARNING 関数は、警告メッセージとして、SiteMinder コンソール ログに文字列引数を書き込みます。

構文

WARNING 関数の構文は以下のとおりです。

```
WARNING(source_string)
```

パラメータ

WARNING 関数は、以下のパラメータを受け取ります。

source_string (文字列)

戻り値

WARNING 関数はブール値を返します。この値は常に TRUE です。

例

```
Return_value=WARNING("Buffer Full")  
Return_value=TRUE
```

XORBITS 関数 -- ビット単位の XOR 演算の実行

XORBITS 関数は、その 2 つの引数に、ビット単位の XOR 演算を実行します。

構文

XORBITS 関数の構文は以下のとおりです。

```
XORBITS(num_1,num_2)
```

パラメータ

XORBITS 関数は、以下のパラメータを受け取ります。

num_1 (数値)

num_2 (数値)

戻り値

XORBITS 関数は数値を返します。

例

```
Return_value=XORBITS(7, 2)
```

```
Return_value=5
```

```
Return_value=XORBITS(7, 15)
```

```
Return_value=8
```

YEAR 関数 -- 日付の数値の「年」要素の取得

YEAR 関数は、1970 年 1 月 1 日から経過した秒の数で表された、日付の「年」の要素を示す数値を返します。

構文

YEAR 関数の構文は以下のとおりです。

```
YEAR(date_time)
```

パラメータ

YEAR 関数は、以下のパラメータを受け取ります。

date_time (数値)

秒数で表わされた日付。

戻り値

YEAR 関数は、70 ～ 138 のいずれかの数値を返します。

YEAR4 関数 -- 日付の「年」要素の取得(4 桁)

YEAR4 関数は、1970 年 1 月 1 日から経過した秒の数で表された、日付の「年」の要素(4 桁)を返します。

構文

YEAR4 関数の構文は以下のとおりです。

YEAR4(*date_time*)

パラメータ

YEAR4 関数は、以下のパラメータを受け取ります。

date_time(数値)

戻り値

YEAR4 関数は、1970 ～ 2038 のいずれかの数値を返します。

索引

.

.fcc ファイルによるインパーソネーションの有効化 - 707

1

- 1 つのディレクトリを持つ異機種環境でユーザが認証される仕組み - 839
- 1 つのディレクトリを持つ異機種環境のエージェントの定義 - 842
- 1 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザの認証 - 838

2

- 2 つのディレクトリを持つ異機種環境のエージェントの定義 - 847
- 2 つのポリシードメインの作成 - 847
- 2 つのユーザ ディレクトリのマッピングの確立 - 620
- 2 つのユーザ ディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法 - 843

4

- 4.x Web エージェント識別情報を作成するためのエージェントオブジェクトの設定 - 146
- 4.x アフィリエイト エージェントのレルムの設定 - 479, 492

A

- ABOVE 関数 -- 指定した LDAP DN より上にあるユーザ - 882
- ABS 関数 -- 絶対値の取得 - 883
- Active Directory グローバル カタログ ディレクトリ接続の設定 - 211
- Active Directory グローバル カタログ ユーザ ディレクトリ接続を設定する方法 - 210
- Active Directory 接続の設定 - 209

Active Directory 接続のための AD ネームスペース - 208

Active Directory 接続のための LDAP ネームスペース - 206

Active Directory ディレクトリ接続を設定する方法 - 203

Active Directory と NetBIOS 名 - 120

Active Directory に関する考慮事項 - 203

Active Directory の概要 - 181

Active Directory ユーザがパスワードを変更できない - 754

ADAM インスタンスおよびユーザ ストア接続の作成 - 199

ADAM ユーザ ストア ディレクトリ接続の設定 - 202

ADAM ユーザ ストアに接続するための SiteMinder ユーザの作成 - 200

ADAM ユーザ ストアへのユーザの読み込み - 201

ADAM ユーザ ディレクトリ接続を設定する方法 - 198

addCert オプション - 662

addPrivKey オプション - 661

addRevocationInfo オプション - 663

AffiliateXMLSignatureImplementation の設定 - 653

AFTER 関数 -- 文字列の取得 - 884

ALL 関数 -- 設定されたすべてのビット - 885

ANDBITS 関数 -- ビット単位の AND 演算の実行 - 886

AND 演算子 - 874

[AND ユーザ/グループ]チェック ボックス - 563

ANY 関数 -- 設定されたいずれかのビット - 887

Apache Web Server 証明書のインストール - 783

Apache Web サーバの前提条件 - 723

Apache のクライアント証明書に対する Web サーバの信頼 - 782

AT 関数 -- 指定された LDAP DN にあるユーザ - 888

AuthValidate ディレクトリ マッピングを設定する方法 - 303

B

BEFORE 関数 -- 文字列の取得 - 888

BELOW 関数 -- 指定した LDAP DN より下にあるユーザ - 890

BOOLEAN 関数 -- 「TRUE」または「FALSE」への切り替え - 891

C

CA Directory ユーザ ストアのキャッシュの有効化 - 186

CA Directory ユーザ ディレクトリ接続の設定 - 185

CA Directory ユーザ ディレクトリ接続を設定する方法 - 184

CA SOA Security Manager SOA エージェント - 136

CA SSO/WAC の統合 - 795

CA SSO から SiteMinder へのシングル サインオンの設定 - 808

CA SSO クライアントから SiteMinder へのシングル サインオンの設定 - 807

CA SSO 前の SiteMinder で保護されたリソースへのユーザ アクセス - 797

CA Technologies 製品リファレンス - iii

CA ディレクトリ キャッシュ設定の確認 - 187

CA への連絡先 - iii

CENTER 関数 -- ソース文字列の長さの調整 - 893

CGI ベースのパスワード サービス HTML テンプレートの変更 - 740

CGI ベースのパスワードサービス HTML フォームテンプレート - 738

CGI ベースのパスワード サービスの HTML フォームのカスタマイズ - 738

CGI ベースのパスワード サービスのプロパティ ファイル - 741

CGI ベースのパスワード サービスのローカライズ - 742

CGI またはサーブレットの場合のカスタム パスワード サービス ディレクトリの注意事項 - 724

changePassword オプション - 663

CHAP の概要 - 365

CHAR 関数 -- ASCII 値の変換 - 892

COMMONDN 関数 -- 共通ルートの取得 - 894

COUNT 関数 -- セット内の要素の数 - 895

createDB オプション - 660

CRL 署名の検証 - 417

CRL のサイズ制限 - 416

CRL の理由コード要件 - 415

CRL を特定するための CRL 配布ポイント - 418

D

DATEFROMSTRING 関数 -- 数値への文字列の変換 - 898

DATETOSTRING 関数 -- 文字列への数値の変換 - 899

DATE 関数 -- 午前 0 時 (形式 1) への設定 - 896

DATE 関数 -- 年、月、日、時、分、および秒 (形式 2) の変換 - 897

DAY 関数 -- 月の日付の取得 - 901

DBLocation の設定 - 653

DBUpdateFrequencyMinutes の設定 - 654

deleteDB オプション - 664

deleteRevocationInfo オプション - 664

delete オプション - 664

DN 属性 - 827

Domino ディレクトリ接続の設定 - 192

Domino のユーザ ディレクトリがポリシー サーバの要件を満たすことを確認 - 191

Domino ユーザ ディレクトリをユーザ ストアとして設定する方法 - 191

Domino ユーザの指定 - 153

DOW 関数 -- 曜日の取得 - 902

DOY 関数 -- 年内の日付の取得 - 903

E

EnableCustomExprOnly レジストリ キー - 408

EncryptedPassword の設定 - 654
Enterprise Log Manager 統合 - 815
ENUMERATE 関数 -- セット要素のチェック - 903
EPM によるアプリケーションの保護 - 601
ERROR 関数 -- コンソール ログへのエラー メッセージの書き込み - 905
eTelligent ルール - 629
eTelligent ルールのコンポーネント要件 - 630
eTelligent ルールの設定 - 631
eTelligent ルールのプロパティ ファイル - 631
EVALUATE 関数 -- 式の評価 - 905
EXISTS 関数 -- ファイル名の検索 - 906
EXPLODEDN 関数 -- セットへの LDAP DN の変換 - 907
export オプション - 665

F

FCCCompatMode の効果 - 710
FCC ファイルにおける名前/値ペアの生成方法 - 334
FCC を使用したセッション仕様の取得 - 709
Federation セキュリティ サービス認証方式 - 400
FILTER 関数 -- セット要素のチェック - 908
findAlias オプション - 665
FIND 関数 -- 文字列内の位置の取得 - 909
FSS 管理 UI 権限 - 71

G

GET 関数 -- ユーザ ディレクトリでの属性の検索 - 911

H

help オプション - 667
HEX 関数 -- 16 進数への変換 - 912
HOUR24 関数 -- 時間への変換 - 913
HOUR 関数 -- 時間への変換 - 913
HTML フォームに対応した SecurID 認証方式の前提条件 - 378
HTML フォーム認証テンプレート - 328
HTML フォーム認証方式 - 324

HTML フォーム認証方式の設定 - 340
HTML フォーム認証方式の前提条件 - 327
HTML フォームベース認証方式 - 314
HTTP プロキシを介した CRL へのアクセス - 422
HTTP プロキシを介した OCSP レスポンダへのアクセス - 432

I

IBM Directory Server ユーザ ディレクトリ接続の設定 - 190
IBM Directory Server ユーザ ディレクトリ接続を設定する方法 - 189
ICAS で使用する SiteMinder キー データベースを設定する方法 - 356
ICAS で使用するポリシー サーバを設定する方法 - 354
ICAS で使用するユーザ ディレクトリの設定 - 357
ICAS に関する要件 - 352
ICAS のインスタンスの作成 - 358
ICAS の概要 - 348
ICAS の用語 - 349
ICAS ファイル - 352
ICAS プロパティ ファイルの設定 - 354
ICAS 用の Java Runtime Environment (JRE) の設定 - 353
IIS Web サーバ証明書のインストール - 781
IIS プロキシ ユーザの指定 - 152
importDefaultCACerts オプション - 666
Information Card 認証方式 - 346
INFO 関数 -- コンソール ログへの INFO メッセージの書き込み - 914
IP アドレスの範囲の追加 - 574
IXMLEncryptDecryptImplementation の設定 - 654
IXMLSignatureImplementation の設定 - 654

J

JDBC データソースの展開 - 82
JVMOptions.txt ファイル - 631

K

Kerberos のサポート - 342
KEY 関数 -- キーの検索 - 915

L

LanMan ディレクトリ接続の設定 - 792
LanMan ディレクトリ接続の前提条件 - 792
LanMan ディレクトリ接続用のレジストリ キーの設定 - 792
LanMan ユーザ ディレクトリ - 791
LanMan ユーザディレクトリ - 224
LanMan ユーザ ディレクトリ接続の設定 - 793
LanMan ユーザ ディレクトリの概要 - 791
LanMan ユーザディレクトリの検索条件 - 794
LCASE 関数 -- 小文字への変換 - 916
LDAP 管理者ストア接続の設定 - 85
LDAP ディレクトリのポリシー バインド - 577
LDAP ディレクトリのユーザの明確化 - 168
LDAP に関する全般情報 - 167
LDAP の概要 - 166
LDAP ユーザが無効にならない - 754
LDAP リフェラル - 175
LDAP ロード バランシングおよびフェイルオーバー - 238
LEFT 関数 -- 文字列の一部を取得 - 917
LegacyCertMapping レジストリ キーの有効化 - 409
LEN 関数 -- 文字列の長さの取得 - 917
listCerts オプション - 666
listRevocationInfo オプション - 666
LoggerConfig.properties ファイルの変更 - 632
LOG 関数 -- ファイルへの文字列の書き込み - 918
LOOP 関数 -- ループでの仮想属性の呼び出し - 919
LPAD 関数 -- ソース文字列の左側での長さの調整 - 920
LTRIM 関数 -- 文字列の先頭にあるスペースの削除 - 921

M

MAX 関数 -- 2 つのうち、より大きい値の特定 - 922
MAYBE 関数 -- 不確定結果のレポート - 922
Microsoft SQL Server または Oracle ユーザ ディレクトリのポリシー バインド - 586
MID 関数 -- 文字列の一部を取得 - 924
MINUTE 関数 -- 日付の「分」要素の取得 - 926
MIN 関数 -- 2 つのうち、より小さい値の特定 - 925
MOD 関数 -- 除算の剰余の取得 - 926
MONTH 関数 -- 日付の「月」要素の取得 - 927
MS Passport 認証方式 - 361
MS Passport 認証方式の設定 - 363

N

NativeDBName の設定 - 653
Netscape Web サーバ証明書のインストール - 778
Netscape 認証局の信頼の確立 - 780
Netscape のクライアント証明書に対する Web サーバの信頼 - 780
NetWare の設定 - 194
NOTBITS 関数 -- ビット単位の NOT の実行 - 928
NOT 演算子 - 874
Novell eDirectory LDAP ディレクトリ接続を設定する方法 - 193
Novell eDirectory における匿名 LDAP アクセスの設定 - 194
Novell eDirectory の LDAP ディレクトリ接続の設定 - 197
NOWGMT 関数 -- 秒単位での現在の時間の取得 - 929
NOW 関数 -- 秒単位での現在の時間の取得 - 928
NSS ユーティリティのインストール - 229
NUMBER 関数 -- 数値への変換 - 930

O

OCSP 設定ファイルの作成 - 425
OCSP チェックの設定 - 431

OCSP と CRL 間のフェイルオーバー - 435
OCSP と CRL 間のフェイルオーバーの設定 - 437
OCSP に関する要件 - 423
OCSP リクエストの署名 (オプション) - 433
OCSP リクエストの署名の設定 - 434
ODBC 接続プーリング - 252
ODBC ディレクトリ接続の設定 - 215
ODBC データソース フェイルオーバーの設定 - 244
ODBC データベースの概要 - 179
ODBC ユーザ ディレクトリ接続への SQL クエリ方式の追加 - 246
ODBC ユーザ ディレクトリ接続を設定する方法 - 215
OID ディレクトリ用の組織単位を作成します。 - 213
OnAccessReject ルールのポリシーに関する考慮事項 - 509, 679
Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法 - 212
Oracle Internet Directory ユーザ ディレクトリの LDAP リフェラル制限 - 212
Oracle インターネット ディレクトリ接続の設定 - 214
ORBITS 関数 -- ビット単位の OR 演算の実行 - 930
OR 演算子 - 874

P

PAP の概要 - 365
PARENTDN 関数 -- LDAP ツリー内の親の取得 - 931
Passport 認証の検索仕様のマッピング - 364
Passport 認証の保護レベルの設定 - 362
Passport 認証方式の前提条件 - 363
PCASE 関数 -- 文字列の大文字への変換 - 932
printCert オプション - 667

Q

QS 関数 -- クエリ文字列からの項目の取得 - 933

R

RADIUS CHAP/PAP 認証方式 - 365
RADIUS CHAP/PAP 認証方式の設定 - 366
RADIUS CHAP/PAP 認証方式の前提条件 - 366
RADIUS CHAP/PAP 方式の概要 - 366
RADIUS エージェント - 135
RADIUS エージェント グループの概要 - 847
RADIUS エージェント グループのセットアップ - 848
RADIUS エージェントにより保護されたレルムの設定 - 491
RADIUS エージェントの設定 - 157
RADIUS エージェントのレルムの設定 - 478
RADIUS エージェント レスポンス属性 - 529
RADIUS 環境のポリシー - 820
RADIUS 環境への SiteMinder の展開 - 834
RADIUS クライアント/サーバアーキテクチャ - 818
RADIUS サーバとしてのポリシー サーバの使用 - 817
RADIUS サーバとしてのポリシー サーバの使用 方法 - 817
RADIUS サーバ認証方式 - 367
RADIUS サーバ認証方式の設定 - 368
RADIUS サーバ認証方式の前提条件 - 367
RADIUS 属性 - 827
RADIUS デバイスの保護に関するガイドライン - 835
RADIUS のトラブルシューティングとテスト - 850
RADIUS ポリシー ドメインのレスポンス - 825
RADIUS ポリシーのテスト - 854
RADIUS リソースと非 RADIUS リソース - 822
RADIUS レスポンス - 526
RADIUS レスポンス属性の設定 - 533
RADIUS レスポンスのグループ化 - 849
RDB 管理者ストア接続の設定 - 87
RDN 関数 -- LDAP DN の最初の要素の取得 - 934
RELATIONDN 関数 -- 2 つの識別名の比較 - 935
renameAlias オプション - 667
RIGHT 関数 -- 文字列からの文字の取得 - 937

RPAD 関数 -- 右側の文字列の長さの調整 - 938
RPT 関数 -- 文字列の反復 - 939
RTRIM 関数 -- 文字列の後続スペースの削除 - 940

S

SafeWord サーバ認証方式 - 368
SafeWord サーバ認証方式と HTML フォーム認証方式 - 370
SafeWord サーバ認証方式と HTML フォーム認証方式の設定 - 371
SafeWord サーバ認証方式と HTML フォーム認証方式の前提条件 - 370
SafeWord サーバ認証方式の設定 - 369
SafeWord サーバ認証方式の前提条件 - 369
SafeWord 認証によるアクセスのリクエスト - 444
SAML アフィリエイトエージェント - 132
SAML アフィリエイト エージェントと Federation セキュリティ サービスの設定 - 135
SECOND 関数 -- 日付の秒数の取得 - 941
SecurID 認証によるアクセスのリクエスト - 444
SecurID 認証方式 - 372
SecurID 認証方式と HTML フォーム認証方式の設定 - 379
SecurID 認証方式の設定 - 377
SecurID 認証方式の前提条件 - 376
SecurID ユーザの再アクティブ化および確認に使用するフォームのサポート - 380
Selectlogin.fcc の設定の詳細 - 448
SET 関数 -- 属性値の設定 - 941
SIGN 関数 -- 数値の符号の取得 - 942
SiteMinder eTelligent ルールのメリット - 630
SiteMinder Information Card 認証方式 (ICAS) - 347
SiteMinder Web エージェントで保護されたレルムの設定 - 490
SiteMinder Web エージェントのレルムの設定 - 477
SiteMinder キー データベース管理 - 651
SiteMinder で生成されるレスポンス属性の可用性 - 545

SiteMinder と CA SSO の統合の前提条件 - 803
SiteMinder .FCC ファイル - 330
SiteMinder アプリケーション ロール - 56
SiteMinder エージェントによるリソースの保護 - 164
SiteMinder エージェントの概要 - 130
SiteMinder が生成するユーザ属性 - 540
SiteMinder 仮想ディレクトリごとに、次の設定を確認する - 785
SiteMinder から CA SSO へのシングル サインオンの設定 - 804
SiteMinder がレスポンスを処理する方法 - 524
SiteMinder 管理者 - 68
SiteMinder 管理者による特別なアクセス - 196
SiteMinder 管理者ユーザ インターフェース - 59
SiteMinder 管理用の Novell eDirectory ユーザ アカウントの作成 - 196
SiteMinder グローバル ポリシーの概念 - 674
SiteMinder セキュリティ ポリシー - 45
SiteMinder テストツール - 757
SiteMinder テスト ツールを使用してテストする方法 - 853
SiteMinder での UID の使用方法 - 256
SiteMinder でユーザの識別に証明書データを使用する方法 - 383
SiteMinder にアクセスする前の eTrust WAC で保護されたリソースへのユーザ アクセス - 801
SiteMinder によるユーザセッションの管理 - 107
SiteMinder の概要 - 29
SiteMinder のコンポーネント - 29
SiteMinder ポリシーによってアクセスが許可されるはずですが、SSL 認証失敗のメッセージを受信しました - 788
SiteMinder ユーザへの管理者権限の割り当て - 200
smauthetsso カスタム認証方式の設定 - 812
smetssockie Web エージェント アクティブ レスポンス属性の設定 - 810
Smkeydatabase のエイリアス - 652

Smkeytool コマンドの構文およびオプション - 659

Smkeytool によるキー データベースの作成および管理 - 655

smkeytool を使用したキー データベースの変更 - 657

smreadclog による RADIUS ログ ファイルの読み込み - 851

SOA Security Manager 認証方式 - 400

SOA エージェント アクション - 501

Solaris で動作する Oracle 8 に非同期呼び出しを設定 - 251

SORT 関数 -- セットの並べ替え - 943

SPACE 関数 -- 一連のスペースの取得 - 944

SQL Server ユーザ ストアの英文字/小文字の区別なしに関する問題および末尾に余分な空白のあるパスワードに関する問題 - 216

SQL クエリへのポリシーのバインド - 589

SQL クエリ方式 - 244

SQL クエリ方式の設定 - 245

SSL 構成 - 778

SSL 接続機能の決定 - 777

SSL 接続の検証 - 238

SSL 接続を設定する前に - 228

SSL 認証方式のトラブルシューティング - 777

SSL の考慮点 - 81

SSL のトラブルシューティング - 783

SSL を介した基本認証方式 - 322

SSL を介した基本認証方式の設定 - 323

SSL を介した基本認証方式の前提条件 - 323

SSL を介した認証 - 317

SSL を使用する IIS Web サーバの設定 - 781

SSL を使用する Netscape Web サーバの設定 - 779

SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法 - 227

startimp.fcc を使用するエージェントの設定 - 706

STRING 関数 -- 文字列への変換 - 945

Sun Java System ユーザ ディレクトリ接続の設定 - 188

Sun Java System ユーザ ディレクトリ接続を設定する方法 - 188

T

TCP/IP 接続 - 124

THROW 関数 -- 処理の停止およびカスタム エラーのレポート - 945

TRACE 関数 -- コンソール ログへのトレース エントリの書き込み - 946, 955

TRANSLATE 関数 -- 文字列値の置換 - 947

U

UCASE 関数 -- 大文字への変換 - 948

UNIX プラットフォームでの smkeytool の使用例 - 668

URLDECODE 関数 -- URL 文字列のデコード - 952

URLENCODE 関数 -- 文字列のエンコード - 952

URL 関数 -- URL 文字列の要素の取得 - 949

V

validateCert オプション - 667

VEXIST 関数 -- パラメータ定義の判定 - 953

W

WebServiceConfig.properties のサンプル構成 ファイル - 643

Web エージェント - 131

Web エージェント アクション - 501

Web エージェント アクションのルールの設定 - 506

Web エージェント コンポーネント - 138

Web エージェント設定の概要 - 137

Web エージェントとポリシー サーバの時間の計算方法 - 103

Web エージェントに関連するポリシー サーバ オブジェクト - 139

Web エージェントに対するトラステッド ホスト - 121

Web エージェントの設定方法 - 140

Web エージェントの中央設定 - 141

Web エージェントの有効化 - 155

Web エージェントレスポンス - 525
Web エージェントレスポンス属性 - 526
Web エージェントレスポンス属性の設定 - 532
Web エージェントを中央で設定した場合の利点
- 137
Web サーバ - 320
Web サーバの証明書の有効期限の確認 - 785
Web サービスの変数リゾルバの設定 - 642
Web サービス変数 - 636, 639
Web サービス変数のコンポーネント要件 - 641
Web サービス変数の作成 - 649
Web サービス変数を解決する場合のセキュリ
ティ要件 - 641
Web ポータルの識別とユーザ ディレクトリの選
択 - 605
Web ポータル ポリシーの作成 - 609
Web ポータル リソースの作成 - 607
Web ポータル ロールの作成 - 608
Web ポータルを保護するアプリケーション セ
キュリティ ポリシー - 604
Windows CardSpace - 347
Windows NT ユーザディレクトリのフェイルオー
バー - 224
Windows ディレクトリ接続の設定 - 222
Windows ディレクトリ接続を設定する方法 -
220
Windows ディレクトリの概要 - 180
Windows 認証によるアクセスのリクエスト - 443
Windows 認証方式 - 314, 342
Windows 認証方式に関する考慮事項 - 345
Windows 認証方式の設定 - 345
Windows 認証方式の前提条件 - 343
Windows のクライアント証明書に対する Web
サーバの信頼 - 781
Windows プラットフォームでの smkeytool の使
用例 - 669
Windows ユーザ セキュリティ コンテキスト -
115
Windows ユーザ セキュリティ コンテキストの要
件 - 118
Windows ユーザ ディレクトリのフェイルオー
バー - 794

WinNTドメイン接続の要件 - 221
WS-Security ドキュメントのキー データベースに
保管されるアイテム - 643

X

X.509 クライアント証明書および HTML フォーム
認証方式 - 390
X.509 クライアント証明書および HTML フォーム
の認証方式の前提条件 - 391
X.509 クライアント証明書および基本認証方式
- 385
X.509 クライアント証明書および基本認証方式
の前提条件 - 386
X.509 クライアント証明書認証方式 - 315, 381
X.509 クライアント証明書認証方式の前提条件
- 383
X.509 クライアント証明書または HTML フォーム
認証方式 - 393
X.509 クライアント証明書または HTML フォーム
の認証方式の前提条件 - 395
X.509 クライアント証明書または基本認証方式
- 387
X.509 クライアント証明書または基本認証方式
の前提条件 - 389
X.509 クライアント認証方式の証明書マッピン
グ - 405
X.509 証明書および HTML フォーム認証方式
の設定 - 392
X.509 証明書および基本認証方式の設定 - 386
X.509 証明書認証方式の設定 - 384
X.509 証明書の証明書マッピングおよび有効性
チェック - 405
X.509 証明書または HTML フォーム認証方式
の設定 - 396
X.509 証明書または基本認証方式の設定 - 389
XMLDocumentOpsImplementation の設定 -
653
XORBITS 関数 -- ビット単位の XOR 演算の実行
- 955

Y

YEAR 関数 -- 日付の数値の「年」要素の取得 - 956

YEAR4 関数 -- 日付の「年」要素の取得(4 桁) - 957

あ

アイデンティティ セレクタの概要 - 346

アカウントティングとデバッグ用の RADIUS ログの生成 - 851

アカウント ステータス ユース ケース - 298

アクセス制御要件の定義 - 55

アクセス制御リスト - 44

アクティブ グローバル ルールの設定 - 682

アクティブなポリシーの設定 - 576

アクティブ ルールの設定 - 516

アクティブ レスポンス属性 - 827

新しい言語のサポート - 749

新しい文字列の追加 - 748

新しいユーザ パスワードが拒否される - 752

アフィリエイト エージェント アクション - 502

アフィリエイト エージェントのルールの設定 - 511

アフィリエイト エージェント レスポンス - 526

アフィリエイト エージェント レスポンス属性 - 529

アフィリエイト エージェント レスポンス属性の設定 - 534

アフィリエイト ドメイン - 481

アプリケーション サーバ エージェント - 135

アプリケーション セキュリティ ポリシーを作成するための管理権限 - 603

アプリケーション セキュリティ ポリシーを作成する方法 - 602

アプリケーションにデータを渡すレスポンスの使用 - 615

アプリケーションの詳細なポリシー コンポーネント - 627

アプリケーション リソースの指定 - 613

アプリケーションを説明するメタデータの指定 - 626

アプリケーションを説明するメタデータを含める - 617

アプリケーションを保護する場合の EPM ユース ケース - 604

安全な HTML フォーム認証テンプレート - 330

「以下」演算子 - 866

「以上」演算子 - 866

インデックス演算子 - 878

インパーソネーション(偽装) - 695

インパーソネーション イベント - 505

インパーソネーション イベント アクションのルールの設定 - 510

インパーソネーション イベントの設定 - 711

インパーソネーション結果フォーム - 717

インパーソネーション処理 - 696

インパーソネーションセッションの開始方法 - 702

インパーソネーションでの .fcc ファイルの基本要件 - 707

インパーソネーションでのセキュリティに関する考慮事項 - 698

インパーソネーションに使用される FCC 命令文 - 708

インパーソネーション認証方式の設定 - 402, 706

インパーソネーション認証方式の前提条件 - 401

インパーソネーションの概要 - 695

インパーソネーションのサンプル実装 - 714

インパーソネーションのサンプルフォーム - 716

インパーソネーションの制限 - 699

インパーソネーションプロセスを開始するためのフォーム - 716

インパーソネーション用の .fcc ファイルの例 - 710

インパーソネーション用のポリシー サーバオブジェクト - 701

インパーソネーション用のポリシーの設定 - 711

インパーソネーションレルムおよびイベント - 700

インパーソネーションログオフフォーム - 718

インフラストラクチャ オブジェクト - 34

英語以外の言語のデフォルト言語としての設定 - 749

エイリアス - 277

エイリアス属性のユース ケース - 278

エイリアス属性マッピングの作成 - 279

エージェント、レルム、およびルールによるリソースの識別 - 484

エージェント API のサポート - 392, 396

エージェント オブジェクトの作成による Web エージェント ID の確立 - 145

エージェントおよびエージェントグループ - 121

エージェント キー管理とセッション タイムアウトを整合させる方法 - 114

エージェント グループ - 159

エージェント グループの設定 - 160

エージェント グループへのエージェントの追加 - 161

エージェント識別情報用のカスタム エージェント オブジェクトの作成 - 163

エージェント設定オブジェクトでの設定パラメータの設定 - 148

エージェント設定オブジェクトの概要 - 148

エージェント設定オブジェクトのコピー - 149

エージェント設定オブジェクトの作成 - 150

エージェント設定オブジェクトの必須パラメータ - 152

エージェント設定パラメータの変更 - 153

エージェント タイプの属性の作成 - 829

エージェント名の指定 - 153

エンコーディング仕様の設定 - 763

演算を含む条件の作成 - 266

エンタープライズ ポリシー管理 (EPM) - 601

エンド ユーザエクスペリエンスの管理 - 47

オペレータ - 862

オンライン証明書状態プロトコル チェック (OCSP) - 422

オンライン ショッピング アプリケーションの保護 - 621

か

階層構造のポリシーでのディレクトリ マッピング - 600

階層構造のポリシーに対する許可処理 - 598

階層構造のポリシーに対する認証処理 - 596

外部管理者ストア クレデンシャルの更新 - 90

外部管理者ストア接続の変更 - 94

外部管理者ストアの考慮事項 - 79

外部管理者ストアを設定する方法 - 78

概要 - 777, 795

拡張 LDAP リフェラル処理 - 178

カスタム エージェント - 162

カスタム エージェント タイプの設定 - 162

カスタム オブジェクト クラスへのポリシーのバインド - 583

カスタム証明書マッピングの作成 - 775

カスタム ディレクトリ接続を設定します - 226

カスタム ディレクトリの概要 - 182

カスタム認証方式 - 316, 399

カスタム認証方式の設定 - 399

カスタム認証方式の前提条件 - 399

カスタム認証方式ライブラリの作成とインストール - 328

カスタム マッピング: 例 1 - 410

カスタム マッピング: 例 2 - 411

カスタム マッピング: 例 3 - 411

カスタム マッピング式 - 407

カスタム マッピングを必要とする証明書属性 - 774

カスタム ユーザ ディレクトリ接続を設定する方法 - 225

仮想ユーザ属性 - 258

仮想ユーザ属性の定義 - 260

仮想ユーザ属性ユース ケース - 259

関数を含む条件の作成 - 265

管理 UI の概要 - 60

管理 UI の起動 - 60

管理者 - 69

管理者アクセスの復元 - 103

管理者ストア オプション - 73

管理者に関する注意事項 - 95

管理者の作成 - 95

管理者のタイプ - 69

管理者の無効化 - 101

管理者ユース ケース - 97

管理者レコードの作成 - 74
管理者を作成する方法 - 94
管理ユーザ インタフェースの管理 - 59
キー データベースの概要 - 651
キー データベースのプロパティ ファイル - 652
偽装認証方式 - 401
既存属性の変更 - 833
既存のポリシー サーバ オブジェクトの変更 - 64
既存のレルムへの認証/検証ディレクトリ マッピングの割り当て - 304
機能テストの結果 - 767
機能テストの実行 - 764
基本認証方式 - 314, 320
基本認証方式の設定 - 321
基本認証方式の前提条件 - 321
許可イベント - 504
許可イベント アクションのルールの設定 - 508
許可イベント用グローバル ルールの作成 - 680
許可イベント用のグローバル ルール - 679
許可されたユーザに対するポリシーの処理 - 599
許可ディレクトリのレルムへの割り当て - 302
拒否された認証 - 598
グループ属性へのポリシーのバインド - 581
グループ名 - 280
グループ名属性マッピングの作成 - 282
グループ名のユース ケース - 280
グループ メンバシップの属性マッピングの作成 - 612
クレーム値を取得するアクティブ レスポンスの設定 - 359
クレジット限度をチェックするための名前付き式の定義 - 620
クレデンシャル セレクタ ソリューションのテスト - 472
クレデンシャル セレクタの概要 - 441
クレデンシャル セレクタのユースケース - 442
グローバル Web エージェント レスポンス属性の設定 - 685
グローバル アクティブ ポリシーの設定 - 688
グローバル オブジェクト - 38

グローバル ポリシー - 671
グローバル ポリシー、ルール、レスポンス - 671
グローバル ポリシー オブジェクトの特性 - 672
グローバル ポリシー オブジェクトを設定する方法 - 686
グローバル ポリシー処理 - 675
グローバル ポリシーで許容される IP アドレス - 689
グローバル ポリシーに対する単一 IP アドレスの指定 - 690
グローバル ポリシーの IP アドレスの範囲の追加 - 692
グローバル ポリシーの作成 - 686
グローバル ポリシーのサブネット マスクの追加 - 691
グローバル ポリシーの時間制限の追加と削除 - 692
グローバル ポリシーのホスト名の追加 - 690
グローバル ポリシーの有効と無効 - 688
グローバル ポリシーへのグローバル ルールの追加 - 686
グローバル ポリシーを設定する方法 - 676
グローバル ルール - 676
グローバル ルールとレスポンスの関連付け - 687
グローバル ルールの削除 - 683
グローバル ルールの有効と無効 - 681
グローバル ルールへの時間制限の追加 - 682
グローバル レスポンス オブジェクト - 683
グローバル レスポンス属性のタイプ - 684
グローバル レスポンスの設定 - 684
グローバル レスポンスのレスポンス属性 - 684
グローバル レスポンスへのルールの関連付け - 559
権限を確立する方法 - 47
検索機能によるユーザへのポリシーのバインド - 579, 585, 588
高度なパスワード オプションの設定 - 732
高度なルール オプション - 505, 515
後からアクティブ レスポンスで使用するクレームの保存 - 355
顧客ユース ケース - 297

顧客ロールの設定 - 624
このユースケースのクレデンシャル セレクタ
リョーション - 444
コンテキスト要求変数の作成 - 646
コンテンツをパーソナライズする方法 - 48

さ

サブレットベースのパスワードサービス JSP
フォームの概要 - 744
サブレットベースのパスワード サービス JSP
フォームの変更 - 746
サブレットベースのパスワード サービスの
ローカライズ - 747
サブレットベース パスワード サービスの JSP
形式のカスタマイズ - 744
サブネット マスクの追加 - 573
サブミット済みタスクのクリーンアップ - 66
サポートされているディレクトリ マッピング - 301
サポートされている認証方式 - 119
サポートされる認証方式とパスワード ポリシー
- 310
算術加算演算子 - 876
算術減算演算子 - 876
算術乗算演算子 - 877
算術除算演算子 - 877
サンプル selectlogin.fcc ファイル - 449
サンプル アプリケーションの保護 - 464
サンプル アプリケーションのレスポンスの設定
- 471
サンプル アプリケーションのレルムとルール -
465
サンプル アプリケーションを保護するポリシー
の設定 - 468
サンプル インパーソネーション実装の評価 -
716
式 - 291
式エディタの使用方法 - 264
式から条件を削除する - 267
式属性マッピングの作成 - 292
式で条件を変更する - 267
式でブール演算子を変更する - 269
式の構文の概要 - 858

式の中で条件をグループ化する - 268
式の中に使用できる関数 - 879
式のユース ケース - 291
式を編集する方法 - 267
システムとポリシードメインの設定 - 840
システムとポリシードメインを設定する方法 -
845
実行要件の定義 - 56
失敗した認証 - 598
失敗した認証の管理 - 455
従業員がエンジニアリング レルムのリソースに
アクセスする - 306
従業員とマネージャのロールの作成 - 614
手動設定によるユーザの追加 - 566
手動設定フィールドでのユーザへのポリシーの
バインド - 578, 584, 587
上級管理者による下級管理者の作成 - 99
条件付き演算子 - 878
情報カード (Information Card) の概要 - 346
証明書検証のトラブルシューティング - 438
証明書シリアル番号または発行者 DN への
マッピング - 412
証明書データベース内の証明書の一覧表示 -
235
証明書データベース ファイルの作成 - 230
証明書データベースへのサーバ証明書の追加
- 233
証明書データベースへのルート証明機関の追
加 - 231
証明書認証の証明書の抽出 - 382
証明書認証方式または基本認証方式を実行し
ているが、基本クレデンシャルを入力できま
せん。 - 789
証明書のプロンプトの表示後に認証失敗を受
け取る - 787
証明書の有効性チェック (任意) - 413
証明書破棄リスト (CRL) のチェック - 414
証明書破棄リスト (CRL) のチェックの設定 - 419
証明書ベースの認証テスト - 773
証明書マッピングの設定 - 405
証明書マッピングのテスト - 407
証明書用のプロンプトはありませんでした - 783

ショッピング アプリケーションに対するセキュリティ ポリシーの設定 - 626
新規ユーザアカウントをアクティブにするためのフォーム - 380
スーパーユーザによる上級管理者の作成 - 98
スタティック変数の作成 - 645
ストアド プロシージャを介して認証するように SQL クエリ方式を設定する方法 - 247
ストレス テスト - 770
すべての Netscape ブラウザが、毎回証明書を要求するように設定されていることを確認する - 784
すべての Web サーバが SSL を使用し、証明書を要求するように設定されていることを確認する - 784
スレッド制御ファイル - 770
正規表現照合の設定 - 730
正規表現の構文 - 729
成功した認証 - 597
静的変数 - 634
セキュリティ コンテキストでのシングル サインオン - 120
セキュリティとユーザを管理するための計画 - 44
セキュリティ モデルの実装 - 51
セキュリティ モデル要件の整理 - 52
セッションアイドルタイムアウト - 699
セッション情報を委任する方法 - 112
セッション タイムアウト - 113
セッションチケット - 106
セッションを管理する方法 - 49
セッションを再検証する方法 - 117
設定順 - 40
設定の概要 - 118
設定の保存とロード - 764
セット共通演算子 - 873
セット結合演算子 - 873
セット包含演算子 - 868
先頭演算子 - 867
ソート名のユース ケース - 296
属性および式のリファレンス - 855
属性タイプ - 826

属性の複数のインスタンスの定義 - 829
属性マッピングのしくみ - 275
属性マッピングの定義 - 277
組織セキュリティの要件を定義する方法 - 53
組織属性へのポリシーのバインド - 583
組織単位へのポリシーのバインド - 581
組織とリソースの要件に関する考慮事項 - 52
組織へのポリシーのバインド - 582
組織ロールへのポリシーのバインド - 580

た

ダイジェスト認証方式 - 316
タスク永続性データベースの管理 - 65
タスク評価要件の定義 - 55
単一 IP アドレスの指定 - 572
単一の認証方式の設定の複数のインスタンス - 320
中央設定とローカル設定の組み合わせ - 144
中央設定とローカル設定の組み合わせの使用例 - 144
追加属性情報を必要とする認証方式 - 750
追加属性の収集 - 337
常に RADIUS 属性を返す SiteMinder の設定 - 828
強い認証 - 441
定数 - 288
定数属性マッピングの作成 - 290
定数のユース ケース - 289
ディレクトリ サーバ情報の収集 - 81
ディレクトリ サーバのクレデンシャルの更新 - 91
ディレクトリ属性の概要 - 183
ディレクトリトポロジとLDAP リフェラル - 175
ディレクトリマッピング - 299
ディレクトリ マッピングとレスポンス - 308
ディレクトリ マッピングの概要 - 299
ディレクトリ マッピングの設定 - 302
ディレクトリ マッピングの大文字/小文字の区別 - 308
ディレクトリ マッピングの要件 - 301
ディレクトリ マッピングの例 - 305
データ型 - 855

データベース クレデンシャルの更新 - 92
データベース情報の収集 - 82
テスト環境エージェントの設定 - 758
テスト ツールの概要 - 757
テストツールを選択します。 - 761
テスト用のカスタム属性マッピング - 774
デフォルトの CGI リダイレクト URL の変更 - 725
デフォルトの管理者アカウント - 68
テンプレート文字列の使用 - 412
同一タイプの複数属性でのカスタム証明書マッピング - 412
等価演算子 - 863
同機種 RADIUS 環境でユーザを認証する方法 - 836
統合された SiteMinder および CA SSO のアーキテクチャ例 - 796
動作モード - 123
動作モードの実装 - 123
特殊な名前/値ペア - 334
特定のユーザに CGI でのパスワードの変更を可能にする - 734
特定ユーザにサブレットでのパスワードを変更を可能にする - 735
匿名認証方式 - 316, 397
匿名認証方式の設定 - 398
匿名認証方式の前提条件 - 398
ドメイン - 473
ドメインとユーザ メンバシップ - 475
ドメインのグローバル ポリシー処理の無効化 - 480
ドメインの削除 - 481
ドメインの変更 - 481
トラステッド ホスト オブジェクトの削除 - 125
トラステッド ホスト設定 - 122
トラステッド ホストホストのホスト設定オブジェクト - 126

な

名前付き式 - 256
名前付き式の定義 - 258
名前付き式の適用 - 271

名前付き式のメリット - 257
認証/検証ディレクトリ マッピングの設定 - 304
認証イベント - 502
認証イベントアクション用のルールの設定 - 507
認証イベント用のグローバル ルール - 677
認証イベント用のグローバル ルールの作成 - 678
認証機関と Web サービス変数 - 643
認証コンテキスト ポリシーの作成 - 462
認証時のポリシー サーバ検索を 1 つのユーザストアに限定 - 311
認証済み CA SSO クライアント ユーザの SiteMinder リソースへのアクセス - 799
認証と認可のディレクトリ マッピングを設定する方法 - 302
認証方式 - 309
認証方式処理 - 312
認証方式タイプ - 313
認証方式とクレデンシャル要件 - 318
認証方式の概要 - 309
認証方式の作成 - 838
認証方式保護レベルの有効性 - 699
ネストされたグループをポリシーに許可 - 562
ネストされたレルム - 487
ネストされたレルムとリソース - 592
ネストされたレルムの設定 - 494
ネストされたレルムを持つ SiteMinder を設定するときの例 - 590

は

バージョン 4 エージェントと RADIUS エージェント シミュレーションで使用するポリシー サーバのセットアップ - 759
バージョン 5 エージェントの場合に必要なポリシー サーバ情報 - 761
排他的 OR 演算子 - 875
パスワード/証明書の認証によるアクセスのリクエスト - 442
パスワードが誤っていることを示すメッセージが表示されない - 755
パスワード構成の設定 - 728

パスワード サービス CGI へのパスワードの変更リンクの追加 - 733
パスワード サービス サブレットへのパスワードの変更リンクの追加 - 735
パスワード サービスでリダイレクトされる際のログイン ID の削除 - 737
パスワード サービスの概要 - 719
パスワード サービスの機能 - 721
パスワード サービスの注意事項 - 723
パスワード正規表現 - 728
パスワードの制限の設定 - 731
パスワードの変更が強制される - 753
[パスワードの変更]リンクの追加 - 733
パスワードの有効期限の設定 - 727
パスワードポリシー - 719
パスワードポリシーの作成 - 726
パスワードポリシーのトラブルシューティング - 752
パターン一致演算子 - 869
バックエンド クレデンシャル選択のポリシーの設定 - 461
バックエンド処理で使用する認証方式のセットアップ - 456
バックエンド処理のセットアップ - 456
バックエンド処理のためのポリシードメインのセットアップ - 456
バックエンド ポリシードメインの AuthContext レスポンスの設定 - 459
バックエンド ポリシードメインのルールの作成 - 459
バックエンド ポリシードメインのレルムの設定 - 457
発行者 DN マッピング - 774
貼り付け - 860
反復タスクの削除 - 68
非 Web エージェントのエージェント識別情報の設定 - 155
非永続 Cookie と永続 Cookie - 105
非永続セッションと永続セッション - 106
非同期呼び出し対応の設定 - 250
標準的なリソース照合 - 513
ファースト ネーム(名)のユース ケース - 294

フェイルオーバーおよび接続プーリング時の非同期呼び出しの対応 - 250
フェイルオーバーの設定 - 240
フォームに表示される文字列の変更 - 747
フォーム認証情報コレクタ(FCC)の使用 - 446
フォーム ポスト変数 - 635
フォーム ポスト変数の作成 - 648
複数 cookie ドメインのサポート - 714
複数の cookie ドメインにまたがるセッションを維持する方法 - 111
複数のレルムにまたがるセッションを維持する方法 - 110
複数ポリシー ストアの同じユーザ ディレクトリ接続の定義 - 253
不等価演算子 - 864
不必要な属性のマッピング - 413
ブラウザ以外のクライアントのサポートの有効化 - 341
ブラウザ証明書の有効性の確認 - 786
プロキシ認証方式 - 316
プロパティファイル - 748
フロントエンド認証のための selectlogin.fcc ファイルの設定 - 447
フロントエンド認証方式の確立 - 445
フロントエンド認証方式の設定 - 453
平均経過時間の計算 - 769
変数 - 629
変数タイプ - 634
変数の概要 - 634
変数の検索の使用による変数の選択 - 538
変数の作成 - 644
変数を含むレスポンス属性の設定 - 536
包含演算子 - 868
ポート番号に関する考慮事項 - 239
保護されていないレルム、ルールおよびポリシー - 486
保護ポリシーの作成 - 464
保護レベル - 317
保護を必要とするアプリケーションの識別 - 610
保護を必要とするリソースの指定 - 622
ホスト設定オブジェクトのコピー - 126

ホスト設定オブジェクトの作成 - 142
ホスト設定オブジェクトのポリシー サーバ クラスタの構成 - 129
ホスト設定オブジェクトへの複数のポリシーサーバの追加 - 127
ホスト名の指定 - 573
ポリシー - 551
ポリシーが使用できる IP アドレス - 572
ポリシー サーバ - 320
ポリシー サーバ オブジェクトの概要 - 33
ポリシー サーバ オブジェクトの管理 - 61
ポリシー サーバ オブジェクトの削除 - 64
ポリシー サーバ オブジェクトの複製 - 62
ポリシー サーバ オブジェクト プロパティの表示 - 63
ポリシー サーバが LDAP ユーザ ストアにバインドする方法 - 179
ポリシー サーバが変数进行处理する方法 - 637
ポリシー サーバがポリシー式に含まれている変数进行处理する方法 - 637
ポリシー サーバから証明書データベースへの参照の設定 - 237
ポリシー サーバがルール进行处理する方法 - 499
ポリシー サーバがレスポンスに含まれている変数进行处理する方法 - 639
ポリシー サーバ管理コンソールの概要 - 32
ポリシー サーバ管理コンソールの設定 - 853
ポリシー サーバでの Passport 認証のサポート - 362
ポリシー サーバと Web エージェントの設定の確認 - 787
ポリシー サーバの LDAP 許可パフォーマンスの向上 - 568
ポリシー サーバのオブジェクト タイプ - 33
ポリシー サーバの概要 - 29
ポリシー サーバの識別 - 759
ポリシー サーバへのトラステッド ホストの登録 - 121
ポリシー サーバ ユーザ インターフェイスでの認証方式オブジェクトのセットアップ - 319

ポリシー サーバを使用した RADIUS 認証の動作 - 818
ポリシーでの変数の使用 - 636
ポリシードメインのオブジェクト - 37
ポリシードメインの概要 - 473
ポリシードメインの作成 - 843
ポリシードメインの設定 - 476
ポリシードメインの設定方法 - 475
ポリシードメインのセットアップ - 837
ポリシーとレスポンス - 594
ポリシーにおける式 - 554
ポリシーにユーザを追加します。 - 557
ポリシーの概要 - 551
ポリシーの削除 - 589
ポリシーの作成 - 555
ポリシーの時間制限 - 574
ポリシーの詳細オプション - 571
ポリシーの処理 - 590
ポリシーの設定方法 - 555
ポリシーの説明 - 553
ポリシーのバインド (Windows NT ユーザディレクトリ用) - 584
ポリシーの有効化および無効化 - 570
ポリシー バインド - 553
ポリシー バインドの確立 - 576
ポリシー ベース セキュリティの概要 - 43
ポリシーベースのセキュリティの実装 - 43
ポリシーへの LDAP 式の追加 - 569
ポリシーへの時間制限の追加 - 575
ポリシーへの式の追加 - 560
ポリシーへのルールの追加 - 558

ま

前の手順に従っても、証明書のプロンプトが表示されない - 784
マスク - 283
マスク属性マッピングの作成 - 284
マスク属性マッピングのビット マスク - 286
マスクのユース ケース - 283
末尾演算子 - 867

「見つかりません」エラー メッセージの表示 - 788

無許可のユーザに対するポリシーの処理 - 599

文字列連結演算子 - 875

や

有効性チェックを実行するための前提条件 - 414

ユーザ/グループ間の AND/OR 関係の指定 - 565

ユーザ アカウントが誤って無効にされる - 753

ユーザ アカウントが途中で無効になる - 753

ユーザが開始するパスワード変更 - 733

ユーザ クラス - 261

ユーザ クラスの定義 - 263

ユーザ クラス ユース ケース - 262

ユーザ グループへのポリシーのバインド - 579, 586, 588

ユーザ クレデンシャルの指定 - 762

ユーザ コンテキスト変数 - 635

ユーザ コンテキスト変数の作成 - 647

ユーザ ストア DSA パラメータの有効化 - 186

ユーザ ストア システムに Ping を発行します。 - 185, 188, 189, 191, 197, 208, 211, 213, 215, 222, 226

ユーザ セキュリティ コンテキストの永続セッションのメンテナンス方法 - 116

ユーザ セッション - 105

ユーザ セッションの概要 - 105

ユーザ セッションを開始する方法 - 109

ユーザ セッションを検証する方法 - 111

ユーザ セッションを終了する方法 - 115

ユーザ属性 - 826

ユーザ属性へのポリシーのバインド - 579, 589

ユーザ属性マッピング - 273

ユーザ属性マッピングの概要 - 273

ユーザ属性マッピングの適用 - 293

ユーザ ディレクトリ - 165, 591

ユーザ ディレクトリ接続の概要 - 166

ユーザ ディレクトリの SSL 接続の設定 - 236

ユーザ ディレクトリの検索 - 255

ユーザ ディレクトリの設定 - 842

ユーザ ディレクトリのセットアップ - 837, 847

ユーザ ディレクトリの内容の表示 - 254

ユーザ ディレクトリの割り当て - 476

ユーザによるパスワードの変更 - 737

ユーザまたはグループをポリシーから除外 - 561

ユーザ マッピングと名前付き式を使用したアプリケーション セキュリティ ポリシー - 618

ユース ケース - ロード バランシングとフェイルオーバー - 243

ユニバーサル ID - 255

ユニバーサル ID によるディレクトリ マッピング - 307

要求コンテキスト変数 - 635

「より大きい」演算子 - 865

「より小さい」演算子 - 864

ら

ラスト ネーム(姓)のユース ケース - 295

リクエスト処理におけるレルム - 489

リクエスト タイムアウト - 122

リグレーション テストの実行 - 769

リソース キャッシュからのレルムのクリア - 495

リソース照合と正規表現 - 513

リソース照合のための正規表現 - 513

リソース情報の指定 - 762

リソースとロールの識別 - 54

臨時従業員が品質保証レルムのリソースにアクセスする場合 - 306

ルール - 497

ルール アクション - 500

ルールがポリシーの一部として動作する仕組み - 498

ルール グループ - 519

ルール グループの概要 - 519

ルール グループの削除 - 522

ルール グループの作成 - 520

ルール グループの変更 - 522

ルール グループへのルールの追加 - 521

ルールとネストされたレルム - 500

ルールの概要 - 497

ルールの削除 - 517
ルールの時間制限とポリシーの時間制限の関係 - 575
ルールの有効化および無効化 - 515
ルールへの時間制限の追加 - 515
例 - 489
レガシー管理者 - 70
レガシー管理者に関する注意事項 - 74
レガシー管理者の権限 - 70
レガシー管理者の権限の移行 - 89
レガシー管理者の無効化 - 101
レガシー管理者を作成する方法 - 73
レスポンス - 523
レスポンス グループ - 546
レスポンス グループの削除 - 549
レスポンス グループの設定 - 546
レスポンス グループの変更 - 548
レスポンス グループへのレスポンスの追加 - 548
レスポンス属性 - 526, 530
レスポンス属性キャッシングの設定 - 539
レスポンス属性に含めるユーザの選択 - 537
レスポンス属性のタイプ - 531
レスポンス タイプ - 525
レスポンスでの変数の使用 - 636
レスポンスとディレクトリ マッピング - 529
レスポンスとレスポンス グループ - 523
レスポンスにおける変数オブジェクト - 535
レスポンスによるアプリケーションのカスタマイズ - 625
レスポンスの WinNT 属性 - 224
レスポンスの機能 - 825
レスポンスの削除 - 540
レスポンスの設定 - 530
レスポンスの編集 - 539
レスポンスまたはレスポンス グループへのルールの関連付け - 558
レポートの表示 - 772
レルム - 483
レルムの概要 - 483
レルムの削除 - 493
レルムの作成 - 477

レルムの設定 - 489
レルムの変更 - 493
レルム ヒントの使用 - 823
ローカライズしたパスワード サービス プロパティ ファイルの作成と使用 - 743
ローカライゼーション用の名前/値ペア - 336
ローカルでの Web エージェントの設定 - 143
ロード バランシングとフェイルオーバーの設定 - 241
ロード バランシングの設定 - 241
ルールに基づいたポリシーの確立 - 617
ルールに基づくアプリケーション セキュリティ ポリシー - 609
ログインに失敗した理由をユーザに伝える - 339