

SiteMinder

ポリシー サーバ設定ガイド

12.52 SP1



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。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複写、譲渡、変更、開示、修正、複製することはできません。

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。

本書の制作者は CA および CA Inc. 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 © 2014 CA. All rights reserved. 本書に記載されたすべての商標、商号、サービス・マークおよびロゴは、それぞれの各社に帰属します。

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 DataMinder™ (旧 CA DLP)
- CA Single Sign-On
- SiteMinder
- 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。

第 1 章: ドキュメントの変更

SiteMinder の旧リリースで発見された問題の結果として、**12.52 SP1** のドキュメントに以下の更新が行われました。

- [SSL を介した LDAP ユーザディレクトリ接続を設定する方法](#) (P. 218) -- 更新された 3 つの手順に、認証機関によって署名されるサーバ証明書でも証明書データベースにインポートされたルート認証機関の証明書が必要であることが示されます。RTC の問題 **118828** を解決します。
- [DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定](#) (P. 787) -- アップグレード後にポリシー サーバ設定ウィザードを実行する手順が追加されました。これにより、STAR イシュー番号 **21807419:01** が解決されます
- [Kerberos 認証の設定方法](#) (P. 359) - 以前は付録にあった内容の書き直し。新しい Kerberos 認証方式テンプレートを含めるために更新されました。
- [リスク ベース認証に対して SecurID HTML フォーム認証サポートを設定する方法](#) (P. 413) -- 新しいセクションで、リスク ベース認証 (RBA) に対して SiteMinder SecurID HTML フォーム認証サポートを設定する方法について説明します。
- Active Directory に関する考慮事項 -
IgnoreDefaultRedirectOnADnativeDisabled レジストリ設定に関する間違った注を削除しました。この更新によって、CQ170141 および STAR イシュー番号 **21243447:03** が解決されます
- [OAuth 認証方式の設定方法](#) (P. 437) - 「OAuth プロバイダへのアプリケーションの登録」セクションで、フィールド名を修正しました。このフィールドは、リダイレクト URI を読み取るようになりました。
「OAuth フォーム認証情報コレクタの変更」と「OAuth プロバイダ設定ファイルの変更」で一致するように、サンプル OAuth プロバイダおよびアプリケーション名を変更しました。
- [オープン形式の Cookie を作成する Web エージェント レスポンスの作成](#) (P. 648) - SM_USERIPADDRESS および SM_USERSESSIONIP フィールドの説明をより分かりやすくしました。
- [OAuth 認証方式の設定](#) (P. 451) - 誤字を修正しました。この変更によって、CQ170232 が解決されます。
- SAML アフィリエイト エージェントに関するセクションを削除しました。このエージェントはサポートされなくなりました。
- Oracle 8 のサポートが停止されたため、トピック「非同期コール用の Solaris 上の Oracle 8 の設定」を削除しました。CQ 177973 を解決します。

- [パスワードポリシーの作成](#) (P. 764) - ユーザがパスワードポリシーによって無効と判断されるパスワードを入力した場合に、ユーザがリダイレクトされる先の特定の URL が追加されました。 CQ 172044 を解決します。
- [ホスト設定オブジェクトのコピー](#) (P. 119) - ホスト設定オブジェクト名の先頭をハイフン ("-") にできないという旨の注を追加しました。 CQ 177096 を解決します。
- [ユーザ コンテキスト変数処理における複数値のユーザ属性結果のサポートの設定](#) (P. 691) - この新しいトピックでは、ユーザ コンテキスト変数进行处理する場合に、複数の値を持つユーザ属性結果をサポートするための設定方法について説明します CQ 177120 および STAR イシュー 21316166-1 を解決します。
- [インパーソネーションに関するセキュリティの考慮事項](#) (P. 743) - r12.03 では利用可能だったが現在は不足している、インパーソネーションに関するトピックを追加しました。 CQ 176089 および STAR イシュー 21533517-01 を解決します。
- [リスク ベース認証に対する SecurID HTML フォーム認証サポートの設定方法](#) (P. 413) - リスク ベース認証に対して SecurID HTML フォーム認証サポートを設定する方法について説明するセクションを追加しました。。 CQ 174046 を解決します。
- [安全な HTML フォーム認証テンプレートの使用](#) (P. 324) - en-US ロケールに対する安全なフォームの可用性およびその使用方法を反映するために、変更されました。 CQ 173591 を解決します。
- [FIPS のみの環境でテスト ツールを使用する方法](#) (P. 823) - FIPS のみの環境で、テスト ツールを使用する方法について説明するトピックを追加しました。 CQ 154567 および STAR イシュー 20890864-1 を解決します。

- [インパーソネーションスキームの前提条件](#) (P. 467) -
smauthimpersonate.dll (Windows) と smauthimpersonate (UNIX) がポリシー サーバにインストールされていることを正しく示すために、これらのファイルが Web エージェントと共にインターネットされているというステートメントを更新しました。 CQ 172378 および STAR イシュー 21467135-1 を解決します。
- [OpenID 認証スキームの設定](#) (P. 465) - 事後処理チェーン フィールドを変更した場合、ポリシー サーバを再起動する必要があるという旨の注を追加しました。 この修正は CQ 178990 を解決します。
- [CA Directory パスワード ポリシーの制御](#) (P. 773) - これは、ポリシー サーバで CA Directory のパスワード ポリシーを遵守する方法について説明する新しいセクションです。 この修正は CQ 174815 を解決します。

このガイドの第 2 版では、以下の点が変更されています。

ドキュメントのみの変更

- 「SiteMinder の概要」の章が削除しました。 コンポーネントの詳細については、「SiteMinder 実装ガイド」を参照してください。 各ポリシー オブジェクトの概要については、このガイドの対応するセクションを参照してください。
- 「ポリシーベースのセキュリティの実行」の章が削除されました。 環境の導入の詳細については、「SiteMinder 実装ガイド」を参照してください。
- 「[DeviceDNA™ エンドポイントによる拡張セッション保証の作成](#) (P. 788)」 — 手順 1 のメニューの順序を修正しました。 この修正は CQ 181072 を解決します。

- 以下のトピックはドメインに関する情報を示しているため、[ユーザーディレクトリ] の章から削除されました。
 - LDAP に関する全般情報
 - LDAP ディレクトリのユーザの明確化
 - LDAP 検索フィルタ
 - ディレクトリトポロジと LDAP リフェラル
 - 拡張 LDAP リフェラル処理
- [EVALUATE 関数](#) (P. 984) -- 式の評価。Evaluate 関数の構文を修正するためにトピックが更新されました。ドキュメントは **12.52 SP1** で更新されています。この変更によって、CQ 179704 が解決されます。
- [CA SiteMinder® for Secure Proxy Server の使用](#) (P. 17) -- CA SiteMinder® for Secure Proxy Server を使用して 管理 UI を保護する手順について説明します。ドキュメントは **12.52 SP1** で更新されています。この変更によって、CQ 182613 が解決されます。

目次

第 2 章：管理 UI の起動

ポリシー サーバは、ユーザの管理者権限に従って 管理 UI を使用して管理します。

- 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、または別の場所に移動する前に、必ず変更内容を保存してください。
- 管理 UI を使用する際は、ブラウザの [リフレッシュ] または [戻る] ボタンは使用しないでください。これらのボタンを使用すると、無効な状態が生成されます。

以下の手順に従います。

1. Web ブラウザを開きます。

- 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、SSL を介しそれを登録した場合は、以下の場所に移動します。

`https://host.domain:8443/iam/siteminder/adminui`

注: ユーザが SSL を介して登録された 管理 UI を初めて起動すると、ブラウザにより、セキュリティ証明書は信頼できる会社によって発行されなかったことが警告されます。この警告は、SSL 登録中に生成される自己署名証明書に関連しています。証明書を承認し、続行してください。

- 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、SSL を介しそれを登録しなかった場合は、以下の場所に移動します。

`http://host.domain:8080/iam/siteminder/adminui`

- 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、以下のいずれかの場所に移動します。

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

`https://host.domain:port/iam/siteminder/adminui`

ホスト

管理 UI ホスト システムの名前を指定します。

ドメイン

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。

ポート

アプリケーション サーバがリクエストをリスンするポートを指定します。

2. 管理者の認証情報を入力します。
3. [ログイン] をクリックします。

このウィンドウの表示内容は、管理 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 を使用して、ポリシー サーバ オブジェクトを管理できます。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。

ポリシー サーバ オブジェクトのコピー

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。コピー オプションはすべてのオブジェクトに使用できるわけではありません。

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。

以下の手順に従います。

1. 複製するオブジェクトが含まれるサブコンポーネント タイプに移動します。

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. 複製するレポートの種類を選択します。

例: エージェント

-
3. [作成] をクリックします。
 4. [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。
 5. 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。
 6. 新しい名前と説明を入力します。
 7. 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。
- オブジェクトが作成されます。

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。

以下の手順に従います。

1. 表示するオブジェクトのサブコンポーネントを選択します。
例: [ポリシー] - [ドメイン] を選択します。
2. オブジェクトのタイプをクリックします。
例: ドメイン
3. 検索条件を指定し、[検索] をクリックします。
4. 表示するオブジェクトの名前をクリックします。
[表示]画面に、選択したオブジェクトに関する情報が表示されます。

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。

以下の手順に従います。

1. 変更するオブジェクトのサブコンポーネントに移動します。
例: [ポリシー] - [ドメイン] をクリックします。
2. 変更するオブジェクトをクリックします。
例: レルム
3. 検索条件を指定し、[検索] をクリックします。
4. 変更するオブジェクトの名前をクリックします。
5. ページの最後までスクロールし、[変更] をクリックします。
6. 必要な変更を行い、[サブミット] をクリックします。
オブジェクトが変更されます。

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。

以下の手順に従います。

1. 削除するオブジェクトのサブコンポーネントを選択します。
例: [インフラストラクチャ] - [認証] をクリックします。
2. オブジェクトのタイプをクリックします。
例: 認証方式
3. 検索条件を指定し、[検索] をクリックします。
4. 削除するオブジェクトを選択します。

-
5. [Delete] をクリックします。
 6. オブジェクトを削除することを確認します。
ポリシー サーバ オブジェクトが削除されます。

タスク永続データベースの管理によるランタイム パフォーマンスの向上

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、以下のタスクが含まれます。

- 正常に処理を完了したタスク。
- 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。

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。

以下の 2 つのオプションを使用して、クリーンアップ タスクをスケジュール、変更、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します。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。

注: システム マネージャとしてログインした場合には、サブミット済みタスクをクリーンアップするためのスケジュールされたジョブのみを実行できます。システム マネージャ アカウントは [管理認証の設定] オプションを使用して定義されます。このアカウントは 管理 UI の最初の登録時に使用され、外部管理者ストアから取得できます。サブミット済みタスクのクリーンアップ機能を、他の管理者（スーパー ユーザ許可を持つ管理者アカウントを含む）が使用することはできません。

以下の手順に従います。

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。
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
 - 既存ジョブの変更
3. [ジョブ名] フィールドにクリーンアップ タスクの名前を、を [スケジュールリング] セクションでスケジュールのタイプおよびスケジュールの詳細を指定します。 [次へ] をクリックします。

-
4. 以下のフィールドに値を入力します。

最短有効期間

削除する必要があ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。

以下の手順に従います。

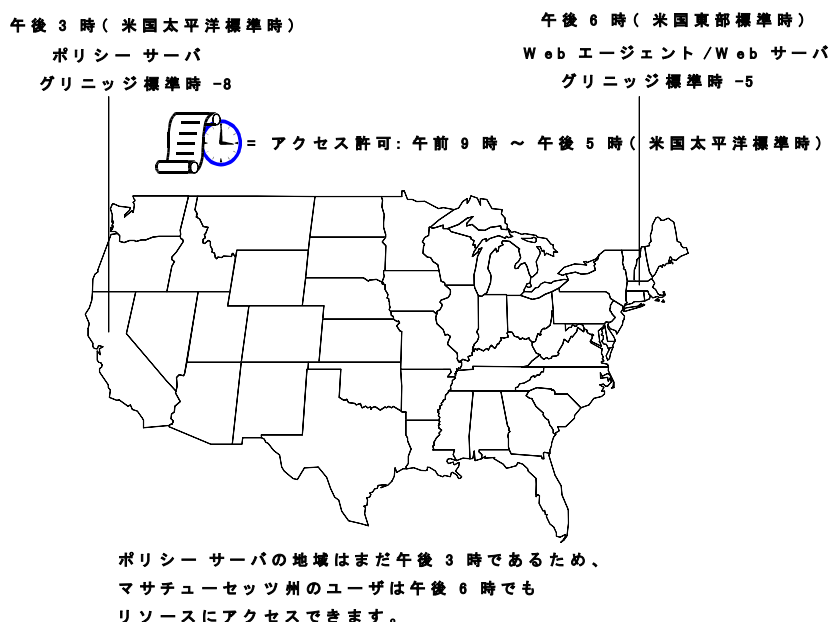
1. [管理]、[管理 UI]、[反復タスクの削除] をクリックします。
2. 削除する必要がある 1 つ以上のスケジュール済みクリーンアップタスクを選択します。
3. [サブミット] をクリックします。

削除タスクがサブミットされて、処理が実行されます。

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。

SiteMinder での管理 UI の保護

SiteMinder で管理 UI を保護するには、リバースプロキシサーバで機能するエージェントを設定し、かつ外部管理者ストアを設定する必要があります。アプリケーションサーバ上で管理 UI に直接アクセスするのではなく、リバースプロキシサーバによって管理 UI にアクセスします。

以下の点について考慮してください。

- 保護する管理 UI が複数ある場合、リバースプロキシサーバで機能する個別のエージェントによって各インスタンスを保護することを推奨します。
- 管理 UI が WebSphere にインストールされる場合、アプリケーションサーバホストシステムで SiteMinder 認証を有効にするには 500MB 以上の空きメモリが必要です。

CA SiteMinder® for Secure Proxy Server の使用

CA SiteMinder® for Secure Proxy Server を使用して管理 UI を保護できます。SiteMinder エージェントによる CA SiteMinder® for Secure Proxy Server の設定は、管理 UI を保護するためのプロキシサーバとして機能します。

CA SiteMinder® for Secure Proxy Server を使用して管理 UI を保護するための前提条件

以下の前提条件を考慮します。

- ポリシー サーバおよびポリシー ストアが設定されている。
- 管理 UI はポリシー サーバと共に正常に設定されている。
- ゲートウェイ（CA SiteMinder® for Secure Proxy Server）がポリシー サーバと共にインストールおよび設定されている。
- （オプション）SSL が設定された 管理 UI を保護するには、CA SiteMinder® for Secure Proxy Server と共にインストールされるホストで SSL を設定します。

注:

- 管理 UI と CA SiteMinder® for Secure Proxy Server のバージョンは同じである必要があります。
- SSL が設定された 管理 UI にアクセスするには、https リクエストを使用します。

CA SiteMinder® for Secure Proxy Server の設定

管理 UI のプロキシサーバとして機能する CA SiteMinder® for Secure Proxy Server を設定します。

以下の手順に従います。

1. 以下のプロキシルールを使用して、以下の場所で使用可能な Proxy Rules.xml ファイルを更新します。

```
<CA SiteMinder® for Secure Proxy Server のインストール場所>%proxy-engine%conf
<!-- Proxy Rules-->
<nete:proxyrules xmlns:nete="http://<管理 UI hostname:port>/">
<nete:cond criteria="beginswith" type="uri">
<nete:case value="/iam/siteminder/">
  <nete:forward>http(s)://<管理 UI hostname:port>$0</nete:forward>
</nete:case>
<nete:case value="/castylesr5.1.1/">
  <nete:forward>http(s):// <管理 UI hostname:port>$0</nete:forward>
</nete:case>
<nete:default>
<nete:forward>http://www.example.com$0</nete:forward>
</nete:default>
</nete:cond>
</nete:proxyrules>
```

2. 以下のパラメータを使用して、以下の場所で使用可能な server.conf ファイルの Default Virtual Host セクションを更新します。

```
<CA SiteMinder® for Secure Proxy Server のインストール場所>%proxy-engine%conf

■ enabledirectrewrite
■ redirectrewritablehostnames
```

3. 管理 UI が SSL ポート上で動作している場合は、その管理 UI をホストしているサーバの自己署名証明書をダウンロードします。ブラウザの一時ファイルを削除してから、管理 UI にアクセスします。
4. 以下の場所で使用可能な CA SiteMinder® SPS の認証局ファイルに、管理 UI の自己署名証明書を配置します。

```
<CA SiteMinder® for Secure Proxy Server のインストール場所>/SSL/certs
```

5. CA SiteMinder® for Secure Proxy Server を再起動します。

管理 UI 認証の有効化

基本認証を有効にするために管理認証を設定する際に、管理 UI を保護するエージェントとして **CA SiteMinder® for Secure Proxy Server** を選択できます。

以下の手順に従います。

1. リバース プロキシを使用して、エージェントを介して 管理 UI にアクセスします。

例：

`http://hostname.example.com:port/iam/siteminder`

ホスト名

CA SiteMinder® for Secure Proxy Server と共にインストールされるホストの名前を指定します。

ポート

CA SiteMinder® for Secure Proxy Server が実行されているポートを指定します。

2. 管理 UI のデフォルトのログイン認証情報を指定します。
3. 管理認証を設定するときに、管理 UI を保護するエージェントとして **CA SiteMinder® for Secure Proxy Server** を選択します。

注： 詳細については、「LDAP 管理者ストア接続の設定」および「RDB 管理者ストア接続の設定」を参照してください。

エージェントとして **CA SiteMinder® for Secure Proxy Server** を選択すると、対応するレルムおよびルールを含む **SiteMinderDomain** という名前のドメインが作成されます。新しく作成されたドメインは、デフォルトのログイン ページの代わりに基本認証を使用して 管理 UI を保護します。

アプリケーション サーバは **SiteMinder** 認証を適用するために自動的に再起動されます。

4. 設定されたユーザ ディレクトリ内で使用可能なユーザのユーザ名およびパスワードを指定します。

重要： ユーザ ディレクトリ接続を設定する際に作成されるスーパーユーザの認証情報を使用してください。

管理 UI ホーム ページが正常に表示されます。

注: アドレス バー内の URL は変わらず、プロキシ サーバの URL が表示されます。

HTML フォーム認証方式の使用

基本認証では、ユーザ認証情報を指定するためのポップアップ ウィンドウが表示されます。ユーザ認証情報を指定するためにログイン ページを使用したい場合は、HTML フォーム認証方式を使用します。

管理 UI に対して認証を有効にした後に、以下の手順を実行します。

以下の手順に従います。

1. admin フォルダをコピーします。

コピー元 :

<管理 UI のインストール場所>/siteminder/webagent-resource

コピー先 :

Webagent/samples

2. HTML フォーム認証方式を作成します。

注: 詳細については、「HTML フォーム認証の設定方法」を参照してください。

3. [スキーム セットアップ] セクションの [ターゲット] に、手順 1 でコピーした admin フォルダ内で使用可能な .fcc ファイルの場所を指定します。

例 :

/siteminderagent/admin/login.fcc

4. 新しく作成された HTML フォーム認証方式を SiteMinderDomain ドメインに追加します。
5. キャッシュをリフレッシュします。
6. 以下のように URL を使用して、管理 UI にアクセスします。

http://hostname.example.com:port/iam/siteminder/adminui

7. 権限のあるユーザのユーザ名とパスワードを入力します。

管理 UI ホームページが表示されます。管理 UI にアクセスするときには、アドレス バーの URL がプロキシ サーバであることがわかります。

リバース プロキシ サーバとしての Apache の使用

CA SiteMinder® for Secure Proxy Server に加えて、SiteMinder を使用して 管理 UI を保護するためのリバース プロキシ サーバとして、Apache Web サーバを使用することもできます。

Apache を使用した SiteMinder による 管理 UI の保護

リバース プロキシ サーバとして Apache Web サーバを使用して、SiteMinder により 管理 UI を保護できます。

以下の手順に従います。

1. リバース プロキシ サーバで動作するエージェントを設定します。

Apache などの、SiteMinder Web エージェントをサポートする Web サーバもまたリバース プロキシ サーバとして機能することが可能です。サポートされるサーバのサポート マトリックスを参照してください。

注: Apache の Web サーバをリバース プロキシ サーバとして機能させるには、Apache の Web サーバの設定ファイルを更新します。リバース プロキシ サーバの設定および設定ファイルの更新の詳細については、Web エージェント設定のドキュメントを参照してください。

重要: プロキシ サーバに対して設定されるルールに使用される URL は、管理 UI を最初に登録するために使用される URL と同じである必要があります。

例：

管理 UI が以下の URL で最初に登録された場合は、プロキシ サーバルールで同じ URL を指定します。

`http://host_name:8080/iam/siteminder/adminui`

2. [エージェント設定オブジェクト] (ACO) で、以下の例に示すように `LogOffUri` パラメータの値を設定します。

`/iam/siteminder/logout.jsp`

3. 外部管理者ストアを設定します。

注：外部管理者ストアを設定すると、アプリケーション サーバは自動的に再起動します。SiteMinder の 管理 UI は再起動後にのみ保護されます。

第 1 章: ドキュメントの変更 3

第 2 章: 管理 UI の起動 9

ポリシー サーバオブジェクトの管理	10
ポリシー サーバオブジェクトのコピー	10
ポリシー サーバオブジェクト プロパティの表示	11
既存のポリシー サーバオブジェクトの変更	12
ポリシー サーバオブジェクトの削除	12
タスク永続データベースの管理によるランタイム パフォーマンスの向上	13
サブミット済みタスクのクリーンアップ	14
反復タスクの削除	15
Web エージェントとポリシー サーバの時間の計算方法	16
SiteMinder での管理 UI の保護	17
CA SiteMinder® for Secure Proxy Server の使用	17
リバース プロキシ サーバとしての Apache の使用	22
管理 UI に対する SiteMinder 認証の無効化	47

第 3 章: SiteMinder 管理者 49

SiteMinder 管理者の概要	49
デフォルトのスーパーユーザ管理者	50
管理者アカウント	50
レガシー管理者アカウント	51
管理者ストア オプション	52
外部管理者ストアを設定する方法	53

外部管理者ストアの考慮事項.....	54
SSL の考慮点	56
ディレクトリ サーバ情報の収集.....	56
データベース情報の収集.....	57
JDBC データ ソースの展開	57
LDAP 管理者ストア接続の設定	60
RDB 管理者ストア接続の設定	62
レガシー管理者の権限の移行.....	64
外部管理者ストア認証情報の更新.....	65
外部管理者ストア接続の変更.....	69
管理者を作成する方法.....	70
管理者に関する注意事項.....	71
管理者アカウントの作成.....	71
管理者に付与された権限が適正かどうか確認する	72
スコープされた管理者の作成方法.....	73
スコープされた管理者に関するガイドライン	74
ワークスペースの作成.....	75
管理者の作成とワークスペースの割り当て	77
管理者がスコープされていることを確認する	79
管理者ユース ケース.....	79
スーパーユーザによる上級管理者の作成	80
スーパーユーザがワークスペースの作成および割り当てを行い、上級管理者をスコープする	81
上級管理者による下級管理者の作成	82
上級管理者 がワークスペースの作成および割り当てを行い、さらに下級管理者をスコープする	83
レガシー管理者を作成する方法.....	83
レガシー管理者に関する注意事項.....	83
レガシー管理者レコードの作成.....	84
管理 UI 権限の委任	86
管理者の無効化.....	87
レガシー管理者の無効化.....	87
管理者アクセスの復元.....	88
管理 UI 用のアクセシビリティ モードを設定する方法	89
ポリシー サーバオブジェクトの変更	90
管理者タイプの選択.....	91
管理者用のアクセシビリティ モードの設定	93
 第 4 章: ユーザ セッション.....	 95
ユーザ セッションの概要	95

非永続 Cookie と永続 Cookie.....	95
非永続セッションと永続セッション.....	95
セッションチケット.....	96
SiteMinder によるユーザセッションの管理.....	97
ユーザ セッションを開始する方法.....	100
複数のレルムにまたがるセッションを維持する方法.....	101
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.....	102
ユーザ セッションを検証する方法.....	103
セッション情報を委任する方法.....	103
セッション タイムアウト.....	104
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.....	105
ユーザ セッションを終了する方法.....	106
Windows ユーザ セキュリティ コンテキスト.....	106
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.....	107
セッションを再検証する方法.....	108
Windows ユーザ セキュリティ コンテキストの要件.....	109

第 5 章: エージェントおよびエージェントグループ 113

Web エージェントに対するトラステッド ホスト.....	113
ポリシー サーバへのトラステッド ホストの登録.....	113
トラステッド ホスト設定.....	114
トラステッド ホスト オブジェクトの削除.....	117
トラステッド ホストホストのホスト設定オブジェクト.....	118
ホスト設定オブジェクトのコピー.....	119
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.....	120
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.....	121
SiteMinder エージェントの概要.....	124
Web エージェント.....	125
RADIUS エージェント.....	126
アプリケーション サーバ エージェント.....	126
Web サービス セキュリティ (WSS) エージェント.....	127
Web エージェント設定の概要.....	128
Web エージェントを中央で設定した場合の利点.....	128
Web エージェント コンポーネント.....	129
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.....	130
Web エージェントの設定方法.....	132
Web エージェントの中央設定.....	133
ホスト設定オブジェクトの作成.....	133
ローカルでの Web エージェントの設定.....	135

中央設定とローカル設定の組み合わせの使用例	136
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	137
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	138
エージェント設定オブジェクトの概要	139
エージェント設定オブジェクトのコピー	140
エージェント設定オブジェクトの作成	141
エージェント設定オブジェクトの必須パラメータ	143
エージェント設定パラメータの変更	145
Web エージェントの有効化	146
RADIUS エージェントの設定	147
エージェント グループ	148
エージェント グループの設定	149
エージェント グループへのエージェントの追加	150
カスタム エージェント	150
カスタム エージェント タイプの設定	151
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	151
エージェント ディスカバリのご紹介	153
エージェント インスタンスの表示	154
ポリシー サーバハートビート間隔の設定	155
エージェント ディスカバリの無効化	156

第 6 章: ユーザ ディレクトリ 157

ユーザ ディレクトリ接続の概要	158
LDAP の概要	158
ODBC データベースの概要	159
Active Directory の概要	159
カスタム ディレクトリの概要	160
ディレクトリ属性の概要	160
CA Directory ユーザ ディレクトリ接続を設定する方法	162
ユーザ ストア システムに Ping を発行します。	162
CA Directory ユーザ ディレクトリ接続の設定	163
ユーザ ストア DSA パラメータの有効化	163
CA Directory ユーザ ストアのキャッシュの有効化	164
CA ディレクトリ キャッシュ設定の確認	165
z/OS ユーザ ディレクトリ接続向けの CA LDAP サーバを設定する方法	165
CA LDAP Server for z/OS の概要	166
CA LDAP Server for z/OS (TSS) でサポートされていない SiteMinder 機能	166
CA Top Secret r12 (TSS) バックエンドセキュリティ オプション	167
CA LDAP Server r15 for z/OS (ACF2) バックエンドセキュリティ オプション	172

CA LDAP Server r15 for z/OS (RACF) バックエンドセキュリティ オプション	175
ポリシー サーバレジストリ エントリの RACF 用の設定	177
ポリシー サーバから CA LDAP Server for z/OS への接続の設定	178
Oracle Directory Server Enterprise Edition ユーザ ディレクトリ接続を設定する方法	179
ユーザ ストア システムに Ping を発行します。	179
Oracle Directory Server Enterprise Edition ユーザ ディレクトリ接続の設定	179
IBM Directory Server ユーザ ディレクトリ接続を設定する方法	180
ユーザ ストア システムに Ping を発行します。	181
IBM Directory Server ユーザ ディレクトリ接続の設定	181
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	182
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	182
ユーザ ストア システムに Ping を発行します。	182
Domino ディレクトリ接続の設定	183
Novell eDirectory LDAP ディレクトリ接続を設定する方法	184
NetWare の設定	184
Novell eDirectory における匿名 LDAP アクセスの設定	185
SiteMinder 管理者による特別なアクセス	187
SiteMinder 管理用の Novell eDirectory ユーザ アカウントの作成	187
ユーザ ストア システムに Ping を発行します。	188
Novell eDirectory の LDAP ディレクトリ接続の設定	188
Active Directory LDS 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
Active Directory LDS ユーザ ストア ディレクトリ接続の設定	189
Active Directory ユーザ ストア接続を設定する方法	191
前提条件	192
AD ネームスペースおよび LDAP ネームスペースの情報の確認	193
ユーザ ストア情報の収集	195
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	197
Active Directory 統合の拡張の有効化	198
ユーザ ストア接続の作成	199
EnableADEnhancedReferrals レジストリ キーの無効化	201
SASL バインド レジストリ キーの有効化	202
ユーザ ストア接続のテスト	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	203
ユーザ ストア システムに Ping を発行します。	204
Active Directory グローバル カタログ ディレクトリ接続の設定	204
Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法	205
Oracle Internet Directory ユーザ ディレクトリの LDAP リフェラル制限	205
ユーザ ストア システムに Ping を発行します。	205
OID ディレクトリ用の組織単位を作成します。	206

Oracle インターネット ディレクトリ接続の設定	206
OpenLDAP Server ユーザ ディレクトリ接続を設定する方法	207
ユーザ ストアを作成	207
OpenLDAP Directory Server ユーザ ディレクトリ接続の設定	208
Red Hat Directory Server ユーザ ディレクトリ接続を設定する方法	208
ユーザ ストア システムに Ping を発行します。	209
Red Hat Directory Server ユーザ ディレクトリ接続 の設定	209
ODBC 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。	210
ODBC ディレクトリ接続の設定	210
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	212
カスタム ユーザ ディレクトリ接続を設定する方法	216
ユーザ ストア システムに Ping を発行します。	217
カスタム ディレクトリ接続を設定します	217
SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法	218
SSL 接続を設定する前に	218
証明書データベース ファイルの作成	219
ルート認証機関の証明書データベースへの追加	221
証明書データベースへのサーバ証明書の追加	223
証明書データベース内の証明書の一覧表示	224
ユーザ ディレクトリの SSL 接続の設定	226
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	226
SSL 接続の検証	227
SSL を介した Oracle ユーザ ディレクトリ接続の設定	227
ポリシー サーバが、SSL を介し Oracle データベースに接続する方法	228
LDAP ロード バランシングおよびフェイルオーバー	230
ポート番号に関する考慮事項	231
フェールオーバーの設定	231
ロード バランシングの設定	232
ロード バランシングとフェイルオーバーの設定	233
ユース ケース - ロード バランシングとフェイルオーバー	234
ODBC データ ソース フェイルオーバーの設定	235
SQL クエリ方式	235
SQL クエリ方式の設定	235
ODBC ユーザ ディレクトリ接続への SQL クエリ方式の追加	237
ストアド プロシージャを使用して認証するように SQL クエリ方式を設定する方法	238
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	240
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	243
ユーザ ディレクトリの内容の表示	244

ユーザディレクトリの検索	244
ユニバーサル ID	245
SiteMinder での UID の使用方法	245
名前付き式	246
名前付き式のメリット	247
名前付き式の定義	247
ユーザ属性マッピング	262
ユーザ属性マッピングの概要	263
属性マッピングのしくみ	264
属性マッピングの定義	266
上級ユーザ属性マッピングの例	282

第 7 章: ディレクトリマッピング 289

ディレクトリ マッピングの概要	289
許可 ID マッピングおよび検証 ID マッピング	291
ID マッピングの設定	291
ID マッピングがサポートするディレクトリ	292
ID マッピング エントリ タイプ	293
複雑なユーザ検索式の使用	293
許可 ID マッピングを設定する方法	294
検証 ID マッピングを設定する方法	296
デフォルトのグローバル検証ディレクトリ マッピングの設定	297
従来のディレクトリ マッピング設定	298
認証と認可のディレクトリ マッピングを設定する方法	298
AuthValidate ディレクトリ マッピングを設定する方法	300
レルムからのディレクトリ マッピングの削除	301
ディレクトリ マッピングの例	301
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	302
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	303
ユニバーサル ID によるディレクトリ マッピング	304
ディレクトリ マッピングの大文字/小文字の区別	305
複雑なユーザ 検索条件による ID マッピング	305
ディレクトリ マッピングとレスポンス	306

第 8 章: 認証方式 307

認証方式の概要	307
サポートされる認証方式とパスワードポリシー	308
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	309

認証方式処理.....	310
永続認証コンテキスト データ	311
保護レベル.....	312
認証方式と認証情報要件.....	313
単一の認証方式の設定の複数のインスタンス	315
基本認証方式の設定方法.....	315
基本認証方式の前提条件を確認する	316
基本認証方式の設定.....	317
SSL を介した基本認証方式の設定方法.....	317
SSL を介した基本認証方式の前提条件の確認.....	318
SSL を介した基本認証方式の設定.....	319
HTML フォーム認証の設定方法.....	320
HTML フォーム認証が必要なことをエージェント所有者に伝える	323
HTML フォーム認証テンプレート ファイルの設定.....	323
HTML フォーム認証方式の設定	336
FCC の国際化を有効にする方法.....	339
Windows 認証方式.....	355
Windows 認証の前提条件が満たされていることを確認してください。	356
Windows 認証方式の設定.....	357
Kerberos 認証の設定方法	359
ドメイン コントローラでの Kerberos 認証の設定.....	359
ポリシー サーバでの Kerberos 認証の設定.....	366
Web サーバでの Kerberos 認証設定.....	369
Windows ワークステーションでの Kerberos 認証の設定.....	369
Kerberos 認証方式の設定	370
Windows ホスト上での Kerberos 外部レルムの設定.....	371
SiteMinder Kerberos 認証のトラブルシューティング.....	372
Information Card 認証方式	376
アイデンティティ セレクタの概要.....	376
SiteMinder Information Card 認証方式 (ICAS)	377
RADIUS CHAP/PAP 認証方式.....	397
PAP の概要	397
CHAP の概要.....	398
RADIUS CHAP/PAP 方式の概要.....	398
RADIUS CHAP/PAP 認証方式の前提条件.....	398
RADIUS CHAP/PAP 認証方式の設定.....	399
RADIUS サーバ認証方式.....	399
RADIUS サーバ認証方式の前提条件.....	400
RADIUS サーバ認証方式の設定.....	400
SafeWord サーバ認証方式.....	401

SafeWord サーバ認証方式の前提条件	401
SafeWord サーバ認証方式の設定	401
SafeWord サーバ認証方式と HTML フォーム認証方式	402
SafeWord サーバ認証方式と HTML フォーム認証方式の前提条件	403
SafeWord サーバ認証方式と HTML フォーム認証方式の設定	404
SecurID 認証方式	404
SecurID 認証方式を設定する方法	408
SecurID HTML フォーム認証方式を設定する方法	410
SecurID ユーザを再アクティブ化し、確認するための SecurID HTML フォーム サポート	412
新しいユーザ アカウントをアクティブにするための SecurID HTML フォーム サポート	412
リスク ベース認証をサポートする SecurID HTML フォーム認証を設定する方法	413
X.509 クライアント証明書認証方式	416
証明書認証の証明書の抽出	417
SiteMinder でユーザの識別に証明書データを使用する方法	418
X.509 クライアント証明書認証方式の前提条件	418
X.509 証明書認証方式の設定	419
X.509 クライアント証明書および基本認証方式	420
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	421
X.509 証明書および基本認証方式の設定	422
X.509 クライアント証明書または基本認証方式	423
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	424
X.509 証明書または基本認証方式の設定	425
X.509 クライアント証明書および HTML フォーム認証方式	426
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	427
エージェント API のサポート	428
X.509 証明書および HTML フォーム認証方式の設定	428
X.509 クライアント証明書または HTML フォーム認証方式	429
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	431
エージェント API のサポート	432
X.509 証明書または HTML フォーム認証方式の設定	432
匿名認証方式	433
匿名認証方式の前提条件	434
匿名認証方式の設定	434
カスタム認証方式	435
カスタム認証方式の前提条件	435
カスタム認証方式の設定	435
OAuth 認証方式を設定する方法	437
OAuth 認証プロセス	439
OAuth プロバイダへのアプリケーションの登録	440
OAuth プラグインの有効化	443

OAuth フォーム認証情報コレクタの変更.....	444
OAuth プロバイダ設定ファイルの変更.....	446
エージェントでの FCCCompatMode の無効化.....	451
OAuth 認証方式の設定	451
OpenID 認証方式.....	452
SiteMinder での OpenID 認証の動作の仕組み	453
OpenID 認証方式を設定する方法.....	455
レガシー フェデレーション認証方式	466
偽装認証方式.....	467
インパーソネーション認証方式の前提条件	467
インパーソネーション認証方式の設定	468

第 9 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 471

X.509 クライアント証明書認証方式の証明書マッピング	471
証明書マッピングの設定.....	471
証明書マッピングのテスト.....	473
カスタム マッピング式.....	474
同一タイプの複数属性でのカスタム証明書マッピング	479
X.509 クライアント証明書認証方式における証明書の有効性チェック	480
有効性チェックを実行するための前提条件	481
証明書破棄リスト (CRL) のチェック	481
オンライン証明書状態プロトコル チェック (OCSP)	489
OCSP と CRL 間のフェイルオーバー.....	503
証明書検証のトラブルシューティング	507

第 10 章: 強い認証 509

クレデンシャル セレクタの概要	509
クレデンシャル セレクタのユースケース	510
パスワード/証明書の認証によるアクセスのリクエスト.....	511
Windows 認証によるアクセスのリクエスト	512
SecurID 認証によるアクセスのリクエスト.....	512
SafeWord 認証によるアクセスのリクエスト	512
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.....	513
フロントエンド認証方式の確立	513
フォーム認証情報コレクタ (FCC) の使用	514
フロント エンド認証のための selectlogin.fcc ファイルの設定	515
フロントエンド認証方式の設定	521
失敗した認証の管理.....	523

バックエンド処理のセットアップ	524
バックエンド処理で使用する認証方式のセットアップ	524
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	524
バックエンド ポリシー ドメインのレルムの設定	525
バックエンド ポリシー ドメインのルール作成	527
バックエンド ポリシー ドメインの AuthContext レスポンスの設定	528
バックエンド クレデンシャル選択のポリシーの設定	530
サンプル アプリケーションの保護	532
サンプル アプリケーションのレルムとルール	533
サンプル アプリケーションを保護するポリシーの設定	536
サンプル アプリケーションのレスポンスの設定	537
クレデンシャル セレクタ ソリューションのテスト	539

第 11 章: アプリケーション オブジェクトでの Web アプリケーションのセキュリティ ポリシーの定義 541

アプリケーション オブジェクトを使用してリソースを保護する利点	541
アプリケーション オブジェクトでの Web アプリケーションのセキュリティ ポリシーの定義方法	543
管理者権限の確認	545
アプリケーション オブジェクトの作成およびセキュリティ ポリシーの一般プロパティの定義	546
アプリケーション リソースの指定	547
保護されているリソースにアクセスできるユーザを識別するロールの作成	548
(オプション) レスポンスの設定による Web アプリケーションのカスタマイズ	549
(オプション) レスポンス グループの設定による Web アプリケーションのカスタマイズ	550
ユーザ ロールとリソースを関連付けるポリシーの設定	551
(オプション) 詳細なアプリケーション オプションの設定	552
アプリケーション オブジェクトを使用して、アプリケーション セキュリティ ポリシーを定義するためのユース ケース	556
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	556
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	562
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	570
CA DataMinder コンテンツ分類に基づくアプリケーション セキュリティ ポリシー	580

第 12 章: ドメイン 587

ポリシー ドメインの概要	587
ドメインとユーザ メンバシップ	589
ポリシー ドメインの設定方法	589
ポリシー ドメインの設定	590

ユーザディレクトリの割り当て	590
レルムの作成	591
ドメインへの Identity Manager 環境の追加	593
ドメインのグローバル ポリシー処理の無効化	594
ドメインの変更	594
ドメインの削除	594

第 13 章: レルム 595

レルムの概要	595
信頼レベルの導入	596
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	597
保護されていないレルム、ルールおよびポリシー	599
ネストされたレルム	600
リクエスト処理におけるレルム	602
例	602
レルムの設定	602
SiteMinder Web エージェントで保護されたレルムの設定	603
RADIUS エージェントにより保護されたレルムの設定	604
レルムの変更	605
レルムの削除	605
ネストされたレルムの設定	606
リソース キャッシュからのレルムのクリア	607

第 14 章: ルール 609

ルールの概要	609
ルールがポリシーの一部として動作する仕組み	610
ポリシー サーバがルールを処理する方法	611
ルールとネストされたレルム	612
ルール アクション	612
高度なルール オプション	617
Web エージェント アクションのルールの設定	618
認証イベントアクション用のルールの設定	619
許可イベントアクションのルールの設定	620
OnAccessReject ルールのポリシーに関する考慮事項	621
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	622
リソース照合と正規表現	623
標準的なリソース照合	623
リソース照合のための正規表現	623

ルールの有効化および無効化	625
高度なルール オプション	625
ルールへの時間制限の追加	626
アクティブ ルールの設定	627
ルールの削除	627

第 15 章: ルール グループ 629

ルール グループの概要	629
ルール グループの作成	630
ルール グループへのルールの追加	631
ルール グループの変更	632
ルール グループの削除	632

第 16 章: レスポンスとレスポンス グループ 633

レスポンス	633
レスポンス タイプ	634
レスポンス 属性	635
レスポンスとディレクトリ マッピング	636
レスポンスの設定	636
レスポンス 属性の設定	637
レスポンスにおける変数オブジェクトの使用	641
レスポンス 属性キャッシングの設定	644
レスポンスの編集	645
レスポンスの削除	645
オープン形式の Cookie を生成する Web エージェント レスポンスの作成方法	646
ポリシー サーバ オブジェクトの変更	647
オープン形式 Cookie を生成する Web エージェント レスポンスの作成	648
ルールへの Web エージェント レスポンスの追加	653
ポリシーへのルールの追加	654
サンプルをガイドとして使用したアプリケーションのカスタマイズ	655
SiteMinder が生成するユーザ属性	655
SiteMinder で生成されるレスポンス 属性の可用性	661
レスポンス グループ	662
レスポンス グループの設定	663
レスポンス グループへのレスポンスの追加	664
レスポンス グループの変更	664
レスポンス グループの削除	665

第 17 章: ポリシー 667

ポリシーの概要.....	667
ポリシー バインド.....	669
ポリシーにおける式.....	670
ポリシーの信頼レベル.....	671
ポリシーの設定方法.....	671
ポリシーの作成.....	672
ポリシーにユーザを追加します。.....	673
ポリシーへのルールを追加.....	674

第 18 章: 変数 675

eTelligent ルール.....	675
eTelligent ルールのコンポーネント要件.....	675
eTelligent ルールのメリット.....	676
eTelligent ルールの設定方法.....	677
eTelligent ルールのプロパティ ファイル.....	677
変数の概要.....	680
変数タイプ.....	680
ポリシーでの変数の使用.....	682
レスポンスでの変数の使用.....	683
Web サービス変数.....	683
Web サービス変数のコンポーネント要件.....	685
Web サービス変数を解決する場合のセキュリティ要件.....	685
Web サービスの変数リゾルバの設定.....	686
WebServiceConfig.properties のサンプル構成ファイル.....	687
認証機関と Web サービス変数.....	687
変数の作成.....	687
スタティック変数の作成.....	688
コンテキスト要求変数の作成.....	689
ユーザ コンテキスト変数の作成.....	690
ユーザ コンテキスト変数进行处理する場合の複数値のユーザ属性結果のサポートの設定.....	691
フォーム ポスト変数の作成.....	692
Web サービス変数の作成.....	693

第 19 章: キーおよび証明書管理 695

証明書および秘密キーの使用.....	695
署名および検証操作.....	698
暗号化/復号化の操作.....	698

SSL 接続用の証明書	699
Artifact バック チャネルをセキュリティ保護する証明書	699
証明書データ ストアでサポートされている形式	701
信頼された証明書およびキー/証明書ペアのインポート	701
既存のファイルからキー/証明書のペアをインポートする	702
キー/証明書ペアの作成方法	703
新規の証明書署名リクエストの生成	705
証明書データ ストアの証明書の更新	706
証明書とキー データのエクスポート	707
認証機関 (CA) 証明書の使用	708
CA 証明書のインポート	709
CRL による証明書の有効性チェック	710
証明書管理用の CRL の追加	711
CRL の更新	712
OCSP による証明書の有効性チェック	712
OCSP と CRL チェックとの間のフェールオーバー	713

第 20 章: グローバル ポリシー、ルール、レスポンス 715

グローバル ポリシー	715
グローバル ポリシー オブジェクトの特性	716
SiteMinder グローバル ポリシーの概念	719
グローバル ポリシー処理	720
グローバル ポリシーを設定する方法	720
グローバル ルール	721
グローバル レスポンス オブジェクト	728
グローバル レスポンスのレスポンス属性	729
グローバル ポリシー オブジェクトを設定する方法	731
グローバル ポリシーの有効と無効	733
グローバル アクティブ ポリシーの設定	733
グローバル ポリシーで許容される IP アドレス	734
グローバル ポリシーに対する単一 IP アドレスの指定	734
グローバル ポリシーのホスト名の追加	735
グローバル ポリシーのサブネット マスクの追加	736
グローバル ポリシーの IP アドレスの範囲の追加	736
グローバル ポリシーの時間制限の追加と削除	737

第 21 章: インパーソネーション(偽装) 739

インパーソネーションセッションの開始方法	740

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	743
認証方式保護レベルの有効性	744
セッションアイドルタイムアウト	744
インパーソネーションの制限	744
インパーソネーションレルムおよびイベント	745
インパーソネーションの設定	746
startimp.fcc を使用するエージェントの設定	747
インパーソネーション認証方式の設定	747
.fcc ファイルによるインパーソネーションの有効化	748
インパーソネーション イベントの設定	752
インパーソネーション用のポリシーの設定	753
複数 cookie ドメインのサポート	755
インパーソネーションのサンプル実装	755
サンプル インパーソネーション実装の評価	757
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	757

第 22 章: パスワード ポリシー 761

パスワードサービスの概要	761
パスワードポリシーを設定する方法	762
パスワードポリシーの考慮事項	763
パスワードポリシーの作成	764
パスワードの有効期限の設定	765
パスワード構成の設定	766
パスワード正規表現	767
パスワードの制限の設定	770
パスワードの詳細オプションの設定	771
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	772
CA Directory パスワードポリシー コントロール	773
ユーザが開始するパスワード変更	774
[パスワードの変更] リンクの追加	774
ユーザによるパスワードの変更	774
パスワード変更失敗メッセージの有効化	775
パスワードポリシーのトラブルシューティング	776
新しいユーザ パスワードが拒否される	776
ユーザ アカウントが誤って無効にされる	776
ユーザ アカウントが途中で無効になる	777
パスワードの変更が強制される	777
LDAP ユーザが無効にならない	777
Active Directory ユーザがパスワードを変更できない	778

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	778

第 23 章: DeviceDNA™ による拡張セッション保証 781

DeviceDNA™ による拡張セッション保証を設定する方法	781
DeviceDNA™ による拡張セッション保証の制限	784
DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定	786
DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定	787
DeviceDNA™ エンドポイントによる拡張セッション保証の作成	788
レルムへのエンドポイントの追加	790
アプリケーション コンポーネントへのエンドポイントの追加	791
パートナーシップにおける DeviceDNA™ による拡張セッション保証の有効化	792
SM--関連するログ ファイル	793
アップグレード中に DeviceDNA™ による拡張セッション保証を設定する方法	794
DeviceDNA™ による拡張セッション保証の制限	798
Windows ポリシー サーバの停止	800
UNIX ポリシー サーバの停止	801
監査ストア アップグレード スクリプト	801
Windows ポリシー サーバの起動	808
UNIX ポリシー サーバの起動	809
DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定	810
DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定	811
DeviceDNA™ エンドポイントによる拡張セッション保証の作成	812
レルムへのエンドポイントの追加	814
アプリケーション コンポーネントへのエンドポイントの追加	815
パートナーシップにおける DeviceDNA™ による拡張セッション保証の有効化	816
SM--関連するログ ファイル	817

第 24 章: SiteMinder テストツール 819

テスト ツールの概要	819
テスト環境エージェントの設定	820
Windows でのテスト ツールの起動	822
UNIX でのテスト ツールの起動	822
FIPS 専用環境でテスト ツールを使用する方法	823
ポリシー サーバの識別	826
テストツールを選択します。	829
リソース情報の指定	829
ユーザ認証情報の指定	830
エンコーディング指定の設定	830

テスト ツール設定ファイルへのテスト設定の保存とロード	831
(オプション) ポリシー サーバへのテスト ツール接続の制限	832
機能テストの実行.....	833
(オプション) リグレッションテストおよびストレステスト用のコマンド スクリプト ファイルへのテストの記録.....	835
機能テストの結果.....	836
平均経過時間の計算.....	838
コマンド スクリプト ファイルに記録されたテストの再生によるリグレッション テストの実行	838
ストレステストの実行	839
スレッド制御ファイルの設定	839
高度な再生モードでスレッド制御ファイルを実行することによるストレステストの実行.....	842
(オプション) テストでのユーザ名の処理方法の設定	843
(オプション) 安定したロードをシミュレートするためのポリシー サーバリクエスト間のスリープ時間の設定.....	844
(オプション) CSV 形式でファイルに再生テストの結果を書き込むためのテスト ツールの設定	845
レポートの表示.....	845
証明書ベースの認証テスト	847
カスタム マッピングを必要とする証明書属性	847
テスト用のカスタム属性マッピング	848

第 25 章: エージェント設定のトラブルシューティング 851

英語以外の入力文字にジャンク文字が含まれる	851
Advanced Authentication Server からのエラー メッセージ	852
"Token not Found" エラー	853

付録 A: SSL 認証方式のトラブルシューティング 855

概要.....	855
SSL 接続機能の決定	856
SSL 設定	856
Netscape のクライアント証明書に対する Web サーバの信頼.....	857
Windows のクライアント証明書に対する Web サーバの信頼.....	858
Apache のクライアント証明書に対する Web サーバの信頼.....	860
SSL のトラブルシューティング	860
証明書用のプロンプトはありませんでした	861
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	861
証明書のプロンプトの表示後に認証失敗を受け取る	864
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.....	865

「見つかりません」エラーメッセージの表示	865
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。	866

付録 B: LanMan ユーザ ディレクトリ 867

LanMan ユーザ ディレクトリの概要	867
LanMan ディレクトリ接続の前提条件	868
LanMan ディレクトリ接続の設定	868
LanMan ディレクトリ接続用のレジストリ キーの設定	869
LanMan ユーザ ディレクトリ接続の設定	870
Windows ユーザ ディレクトリのフェイルオーバー	870
LanMan ユーザディレクトリの検索条件	870

付録 C: CA SSO/WAC の統合 871

CA SSO/WAC の統合の概要	872
統合された SiteMinder および CA SSO のアーキテクチャ例	873
CA SSO 前の SiteMinder で保護されたリソースへのユーザ アクセス	874
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	876
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	878
SiteMinder と CA SSO の統合の前提条件	880
SiteMinder から CA SSO へのシングル サインオンの設定	881
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	883
CA SSO から SiteMinder へのシングル サインオンの設定	885
smetssockie Web エージェント アクティブ レスポンス属性の設定	887
smauthetssso カスタム認証方式の設定	888

第 26 章: CA User Activity Reporting Module の統合 890

付録 D: RADIUS サーバとしてのポリシー サーバの使用方法 893

RADIUS サーバとしてのポリシー サーバの使用	893
RADIUS クライアント/サーバアーキテクチャ	894
ポリシー サーバを使用した RADIUS 認証の動作	894
RADIUS 環境のポリシー	896
RADIUS リソースと非 RADIUS リソース	898
レルム ヒントの使用	899
RADIUS ポリシー ドメインのレスポンス	901
レスポンスの機能	901

属性タイプ	902
常に RADIUS 属性を返す SiteMinder の設定	905
エージェント タイプの属性の作成	905
RADIUS 環境への SiteMinder の展開	910
RADIUS デバイスの保護に関するガイドライン	911
同機種 RADIUS 環境でユーザを認証する方法	912
ユーザディレクトリのセットアップ	913
ポリシー ドメインのセットアップ	914
認証方式の作成	914
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	914
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	915
システムとポリシー ドメインの設定	916
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	918
ユーザディレクトリの設定	918
ポリシー ドメインの作成	919
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	919
システムとポリシー ドメインを設定する方法	921
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	923
ユーザディレクトリのセットアップ	923
2 つのポリシー ドメインの作成	923
RADIUS エージェント グループの概要	923
RADIUS エージェント グループのセットアップ	924
RADIUS レスポンスのグループ化	925
RADIUS のトラブルシューティングとテスト	926
監査とデバッグのための RADIUS ログの生成	927
smreadclog による RADIUS ログ ファイルの読み込み	927
SiteMinder テスト ツールを使用してテストする方法	929

付録 E: 属性および式のリファレンス 931

データ型	931
式の構文の概要	934
貼り付け	936
演算子	938
等価演算子	939
不等価演算子	940
「より小さい」演算子	940
「より大きい」演算子	941
「以下」演算子	942
「以上」演算子	943

先頭演算子.....	943
末尾演算子.....	944
包含演算子.....	944
セット包含演算子.....	945
パターン一致演算子.....	945
セット共通演算子.....	950
セット結合演算子.....	950
NOT 演算子.....	951
AND 演算子.....	951
OR 演算子.....	952
排他的 OR 演算子.....	952
文字列連結演算子.....	953
算術加算演算子.....	953
算術減算演算子.....	954
算術乗算演算子.....	954
算術除算演算子.....	955
条件付き演算子.....	955
インデックス演算子.....	956
式の中に使用できる関数.....	956
ABOVE 関数 -- 指定した LDAP DN より上にあるユーザ.....	960
ABS 関数 -- 絶対値の取得.....	961
AFTER 関数 -- 文字列の取得.....	961
ALL 関数 -- 設定されたすべてのビット.....	963
ANDBITS 関数 -- ビット単位の AND 演算の実行.....	964
ANY 関数 -- 設定されたいずれかのビット.....	964
AT 関数 -- 指定された LDAP DN にあるユーザ.....	965
BEFORE 関数 -- 文字列の取得.....	966
BELOW 関数 -- 指定した LDAP DN より下にあるユーザ.....	967
BOOLEAN 関数 -- 「TRUE」または「FALSE」への切り替え.....	968
CHAR 関数 -- ASCII 値の変換.....	969
CENTER 関数 -- ソース文字列の長さの調整.....	970
COMMONDN 関数 -- 共通ルートの取得.....	971
COUNT 関数 -- セット内の要素の数.....	972
DATE 関数 -- 午前 0 時（形式 1）への設定.....	974
DATE 関数 -- 年、月、日、時、分、および秒（形式 2）の変換.....	974
DATEFROMSTRING 関数 -- 数値への文字列の変換.....	976
DATETOSTRING 関数 -- 文字列への数値の変換.....	977
DAY 関数 -- 月の日付の取得.....	979
DOW 関数 -- 曜日の取得.....	980
DOY 関数 -- 年内の日付の取得.....	980

ENUMERATE 関数 -- セット要素のチェック	981
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	983
EVALUATE 関数 -- 式の評価	984
EXISTS 関数 -- ファイル名の検索	985
EXPLODEDN 関数 -- セットへの LDAP DN の変換	986
FILTER 関数 -- セット要素のチェック	987
FIND 関数 -- 文字列内の位置の取得	988
GET 関数 -- ユーザディレクトリでの属性の検索	990
HEX 関数 -- 16 進数への変換	991
HOURL 関数 -- 時間への変換	991
HOURL24 関数 -- 時間への変換	992
INFO 関数 -- コンソール ログへの INFO メッセージの書き込み	993
KEY 関数 -- キーの検索	993
LCASE 関数 -- 小文字への変換	995
LEFT 関数 -- 文字列の一部を取得	995
LEN 関数 -- 文字列の長さの取得	996
LOG 関数 -- ファイルへの文字列の書き込み	997
LOOP 関数 -- ループでの仮想属性の呼び出し	997
LPAD 関数 -- ソース文字列の左側での長さの調整	998
LTRIM 関数 -- 文字列の先頭にあるスペースの削除	1000
MAX 関数 -- 2 つのうち、より大きい値の特定	1000
MAYBE 関数 -- 不確定結果のレポート	1001
MID 関数 -- 文字列の一部を取得	1003
MIN 関数 -- 2 つのうち、より小さい値の特定	1004
MINUTE 関数 -- 日付の「分」要素の取得	1005
MOD 関数 -- 除算の剰余の取得	1005
MONTH 関数 -- 日付の「月」要素の取得	1006
NOTBITS 関数 -- ビット単位の NOT の実行	1007
NOW 関数 -- 秒単位での現在の時間の取得	1007
NOWGMT 関数 -- 秒単位での現在の時間の取得	1008
NUMBER 関数 -- 数値への変換	1009
ORBITS 関数 -- ビット単位の OR 演算の実行	1010
PARENTDN 関数 -- LDAP ツリー内の親の取得	1010
PCASE 関数 -- 文字列の大文字への変換	1011
QS 関数 -- クエリ文字列からの項目の取得	1012
RDN 関数 -- LDAP DN の最初の要素の取得	1014
RELATIONDN 関数 -- 2 つの識別名の比較	1015
RIGHT 関数 -- 文字列からの文字の取得	1016
RPAD 関数 -- 右側の文字列の長さの調整	1017
RPT 関数 -- 文字列の反復	1018

RTRIM 関数 -- 文字列の後続スペースの削除	1019
SECOND 関数 -- 日付の秒数の取得	1020
SET 関数 -- 属性値の設定	1020
SIGN 関数 -- 数値の符号の取得	1021
SORT 関数 -- セットの並べ替え	1022
SPACE 関数 -- 一連のスペースの取得	1023
STRING 関数 -- 文字列への変換	1024
THROW 関数 -- 処理の停止およびカスタム エラーのレポート	1025
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	1026
TRANSLATE 関数 -- 文字列値の置換	1026
UCASE 関数 -- 大文字への変換	1027
URL 関数 -- URL 文字列の要素の取得	1028
URLDECODE 関数 -- URL 文字列のデコード	1031
URLENCODE 関数 -- 文字列のエンコード	1032
VEXIST 関数 -- パラメータ定義の判定	1033
WARNING 関数 -- コンソール ログへの WARNING メッセージの書き込み	1034
XORBITS 関数 -- ビット単位の XOR 演算の実行	1035
YEAR 関数 -- 日付の数値の「年」要素の取得	1035
YEAR4 関数 -- 日付の「年」要素の取得（4 桁）	1036

詳細情報:

[外部管理者ストアを設定する方法 \(P. 53\)](#)

認証方式の変更

管理 UI の保護に使用されるデフォルトの SiteMinder 認証方式は、基本的なユーザ名とパスワードです。デフォルトの認証方式を、SAML および WS-Fed 認証以外のあらゆる SiteMinder サポートの認証方式に変更できます。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [レルム] をクリックします。
3. 以下のレルムを検索して、名前をクリックして開きます。

SiteMinder_ims_realm

注: このレルムは SiteMinderDomain という名前のドメインと関連付けられます。

4. [変更] をクリックして設定を有効にします。
5. [認証方式] リストから認証方式を選択します。
6. 必要な場合は、追加の設定を入力します。
7. [サブミット] をクリックします。

管理 UI は選択された認証方式を使用して保護されます。

詳細情報:

[認証方式](#) (P. 307)

管理 UI に対する SiteMinder 認証の無効化

SiteMinder の管理 UI を保護する必要がない場合、SiteMinder 認証を無効にできます。管理 UI に対する SiteMinder の保護を削除した後であっても、リバースプロキシサーバによって管理 UI にアクセスできます。

アプリケーションサーバ上で管理 UI に直接アクセスするには、データディレクトリを削除し、ポリシーサーバに再度管理 UI を登録します。

以下の手順に従います。

1. 管理 UI にログインします。
2. [管理認証]ウィザードを実行して、SiteMinder 認証を使用する管理 UI の保護を必要としなくなったことを指定します。

注: 外部管理者ストアの使用を続行する場合は、既存のディレクトリサーバまたはデータベース接続情報を残します。

3. 管理 UI ホスト システムにログインします。
4. 管理 UI データ ディレクトリを削除します。管理 UI を展開したアプリケーションサーバのタイプによって、データディレクトリの場所が特定されます。
 - 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、データディレクトリの場所は以下のとおりです。
`install_dir/adminui/server/default/data`
 - 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、データディレクトリのデフォルトの場所は以下のとおりです。
 - (JBoss): `<JBoss Install>/server/default/data`
 - (WebSphere):
`<WebSphere>%AppServer%profiles%<your_profile>%data`
 - (WebLogic): `<welblogic install>%user_projects%domains%<user_domain>%data`
5. ポリシーサーバホストシステムにログインし、XPSRegClient ユーティリティを使用して管理 UI 登録ウィンドウをリセットします。
6. ポリシーサーバに管理 UI を登録します。

第 3 章: SiteMinder 管理者

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 管理者の概要 \(P. 49\)](#)
- [外部管理者ストアを設定する方法 \(P. 53\)](#)
- [管理者を作成する方法 \(P. 70\)](#)
- [スコープされた管理者の作成方法 \(P. 73\)](#)
- [管理者ユース ケース \(P. 79\)](#)
- [レガシー管理者を作成する方法 \(P. 83\)](#)
- [管理者の無効化 \(P. 87\)](#)
- [レガシー管理者の無効化 \(P. 87\)](#)
- [管理者アクセスの復元 \(P. 88\)](#)
- [管理 UI 用のアクセシビリティ モードを設定する方法 \(P. 89\)](#)

SiteMinder 管理者の概要

管理者は、ポリシー サーバ オブジェクトおよびツールにアクセスできるユーザです。

異なる管理者が組織内でのそれぞれのロールに応じた権限を持つように、複数の管理者アカウントを作成できます。

このモデルを使用することにより、ポリシー サーバ オブジェクトおよびツールの管理を他のユーザに委任できます。

管理者 ID のデフォルトのソースであるポリシー ストアを設定するとき、完全なシステム権限を持ったデフォルトの **SiteMinder** スーパーユーザ アカウントが作成されます。このデフォルト設定では、ソフトウェア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。

外部管理者ユーザ ストアを設定することをお勧めします。

デフォルトのスーパーユーザ管理者

ポリシー ストアを設定すると、デフォルトのスーパーユーザ 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、以下の操作に使用します。

- 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
- トラストドホストの管理
- ポリシー管理 API の管理

デフォルトのスーパーユーザ アカウントには以下の認証情報があります。

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。

管理者アカウント

管理者アカウントを使用して、以下の SiteMinder 管理タスクを実行できます。

- 管理 UI を使用した SiteMinder オブジェクトの管理
- XPSImport や XPSExport などのポリシー サーバ ツールの使用

管理者アカウントを作成して、権限をきめ細かく委任し、該当する管理者が使用できる管理機能を特定します。特に、管理者アカウントは以下のプロパティを定義します。

スコープ

管理者がアクセスできるのは、すべての SiteMinder データか、または割り当てられた管理ワークスペースで定義されたオブジェクトに限定されるかを指定します。

アクセス方法

SiteMinder データにアクセスし、それを管理するために管理者が使用できる方法を指定します。

権利

管理者がアクセスできる SiteMinder オブジェクトのカテゴリを指定します。また、それらを表示するだけか、またはオブジェクトを表示し、かつ修正も可能かを指定します。

このきめ細かい指定で、管理者を作成し、組織内の管理ロールに一致する権限を割り当てることができます。

注: 外部管理者ストアの管理ユーザと関連付けられる新規の管理者アカウントのみを作成できます。ただし、管理者アカウントは、それらの管理者が管理 UI にアクセスできるようにするために、ポリシーストアに格納されているレガシー管理者レコードに対して自動的に生成されます。

レガシー管理者アカウント

レガシー管理者アカウントは、以下の管理タスクを実行するために使用できます。

- smobjimport や smobjexport などのポリシー サーバ ツールの使用

- 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 の使用

注: ユーザの環境にポリシー管理 API を使用するスクリプトまたはプログラムが含まれる場合、レガシー管理者アカウントが必要です。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。

注: ポリシーストアが管理者 ID のソースとして設定されている（デフォルト）場合、レガシー管理者は管理 UI にアクセスすることもできます。いったん外部管理者ストアが設定されると、レガシー管理者アカウントは管理 UI にアクセスできません。

詳細情報:

[管理者ストア オプション](#) (P. 52)

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。ただし、管理者アカウントの追加に、企業ディレクトリなどの外部管理者ユーザ ストアを使用することを推奨します。

管理者 ID を格納する場所を決める場合は、以下の要素を考慮します。

- 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、外部管理者ストアを使用する必要があります。
- ポリシー ストアに管理者 ID を格納する場合、レガシー管理者を作成することによりポリシー ストアに新規の管理者レコードのみを確立できます。

- 外部ストアに管理者 ID を格納する場合、管理者アカウントを作成して外部ストアで管理者レコードを探します。

注: 外部管理者ストアが一度設定されると、管理 UI アクセスを許可するために、新規レガシー 管理者を作成したり、管理者アカウントをレガシー管理者レコードに関連付けることはできません。

- 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。

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、以下の手順に従います。

1. (オプション) SiteMinder で 管理 UI を保護する場合は、リバース プロキシサーバで機能するようにエージェントを設定します。

注: リバース プロキシサーバ 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。

2. 外部管理者ストアに関する注意事項を確認します。
3. SSL の注意事項を確認します。
4. 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。

注: WebSphere にデータ ソースを展開している場合は、JNDI 名は、必ずデータ ソース プロパティの下で、以下のプレフィックスを付けます。

jdbc/

例: データ ソース名が **abc** である場合、JNDI 名は **jdbc/abc** です。

5. 外部管理者ストアへの接続を設定します。
6. (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。外部管理者ストアの設定は、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、継続して以下のアクションを実行できます。
 - 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
 - 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、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 を使用して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。WebSphere にデータ ソースを展開している場合は、データ ソースプロパティ下の JNDI 名が以下のテキストから始まっていることを確認してください。

jdbc/

例: データ ソース名が abc である場合、JNDI 名は jdbc/abc です。

以下の手順に従います。

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する前に、管理者権限でコマンドラインウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドラインウィンドウを開きます。

- (UNIX)

`smjdbcsetup.sh`

ユーティリティは一意的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、以下の手順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. 以下のいずれかを実行します。

- サービスを自動的に開始するには、以下の手順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

接続を設定することにより、ポリシー ストアから外部ストアに管理者 ID のソースを変更します。

以下の手順に従います。

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。
2. (オプション) 本製品で管理 UI を保護するには、ドロップダウン リストからエージェントを選択し、[次へ] をクリックします。

CA SiteMinder® for Secure Proxy Server を使用して設定されているエージェント、またはリバース プロキシサーバで機能するように設定されているエージェントを必ず選択します。

3. [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。

4. 以下のタスクを実行します。

- 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、SSL 対応ポートを入力してください。そうしないと、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] 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 サーバは SSL を介して通信する必要があります。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。

5. 以下のタスクを実行します。

- 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。

6. 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。

重要: LDAP または他のアプリケーションによって使用されるまたは書き込まれる *任意*の属性を提示しないでください。この状況が発生した場合は、/logout.jsp ページにリダイレクトされ、管理 UI にログインできなくなります。

7. 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。
8. 1 人のユーザを選択し、[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。接続が設定されると、このユーザはスーパーユーザになります。

9. 接続詳細を確認し、[完了] をクリックします。

重要: 管理者の新しい認証情報でログインする *前に*、アプリケーションサーバを手動で再起動してください。

RDB 管理者ストア接続の設定

接続を設定することにより、ポリシー ストアから外部ストアに管理者 ID のソースを変更します。

以下の手順に従います。

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。
2. (オプション) 本製品で管理 UI を保護するには、ドロップダウン リストからエージェントを選択し、[次へ] をクリックします。

リバース プロキシ サーバで機能するように設定されているエージェントを選択します。

3. ディレクトリ タイプリストから以下のいずれかの項目を選択します。
 - 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、リレーショナルデータベース（[フル ネーム] 列あり）テンプレートを選択します。
 - ユーザのフル ネームを計算する必要がある場合は、リレーショナルデータベース（[フル ネーム] 列なし）テンプレートを選択します。
4. [次へ] をクリックします。

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。
5. データ ソースを選択し、[次へ] をクリックします。
6. ユーザ テーブルを選択し、[次へ] をクリックします。
7. 以下のタスクのいずれかを実行します。
 - リレーショナルデータベース（[フル ネーム] 列あり）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ます。[次へ] をクリックします。
 - 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、以下のようになります。
 - 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. フル ネームの取得クエリを構築します。##FIRST_NAME、##LAST_NAME、および##PRIMARY_KEY をかユーザ属性にマップする列名で置き換えます。

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. [Next] をクリックします。
8. [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。

9. ユーザを選択し、[次へ] をクリックします。

注: このユーザはスーパーユーザになります。

10. 接続詳細を確認し、[完了] をクリックします。

重要: 外部管理者ストアを設定した後、管理者の新しい認証情報でログインする前に、アプリケーションサーバを手動で再起動します。

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、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。

以下の手順に従います。

注: 管理者が外部ストアに存在していることを確認します。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。

1. [管理] - [管理者] をクリックします。
2. [管理者] をクリックします。
[管理者] ページが表示されます。
3. 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。
4. 変更する管理者の名前をクリックします。
[管理者の表示] ページが表示されます。ユーザパスはポリシーストアを指しています。
5. [変更] をクリックします。
設定とコントロールがアクティブになります。

6. [一般] 内の [検索] をクリックします。
[ユーザの選択] ページが表示されます。
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。

外部管理者ストア認証情報の更新

外部管理者ストアへの接続で管理 UI が使用する認証情報を変更する場合は、新しい認証情報を管理 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) **smjndisetaup** ユーティリティ。ディレクトリ サーバユーザ アカウントの認証情報を更新します。

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。**smjndisetaup** 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) **smjdbcsetap** ユーティリティ。データベース ユーザ アカウントの認証情報を更新します。

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場合は、**smjdbcsetap** 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、以下の事項を考慮してください。

- (LDAP) 管理認証ウィザードを使用して、管理 UI がディレクトリ サーバにアクセスするために使用する認証情報を更新します。

重要: 認証情報を更新するためにウィザードを使用した後、ディレクトリ サーバ上の認証情報をできるだけ早く更新してください。ディレクトリ サーバの認証情報がウィザードを使用して提供した認証情報に一致するよう更新されるまで、管理者は 管理 UI にログインできません。

- (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。

詳細情報:

[JDBC データ ソースの展開 \(P. 57\)](#)

ディレクトリ サーバの認証情報の更新

ディレクトリ管理者認証情報を更新するには、smjndisetaup ユーティリティを使用します。

注: smjndisetaup 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。 smjndisetaup ユーティリティを使用して接続認証情報を作成することはできません。

ディレクトリ サーバ認証情報を更新する方法

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)

```
smjndisetaup.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する前に、管理者権限でコマンドラインウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドラインウィンドウを開きます。

■ (UNIX)

```
smjndisetaup.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実行します。

■ 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。

ユーティリティは認証情報を確認し、ディレクトリ接続認証情報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ます。また、このユーティリティは新しいディレクトリ接続詳細も更新します。

■ 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。

データベース認証情報の更新

JNDI データ ソースのデータベース ユーザ認証情報を更新するには、`smjdbcsetup` ユーティリティを使用します。

データベース認証情報を更新する方法

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。
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する前に、管理者権限でコマンドラインウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドラインウィンドウを開きます。

- (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 ソース認証情報を確認し、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 ソース認証情報を確認します。

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. 以下のタスク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と、データソースが展開されます。
- (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

再度管理認証ウィザードを実行し、管理 UI が管理者認証のために接続する外部ストアを変更します。

詳細情報:

[外部管理者ストアを設定する方法](#) (P. 53)

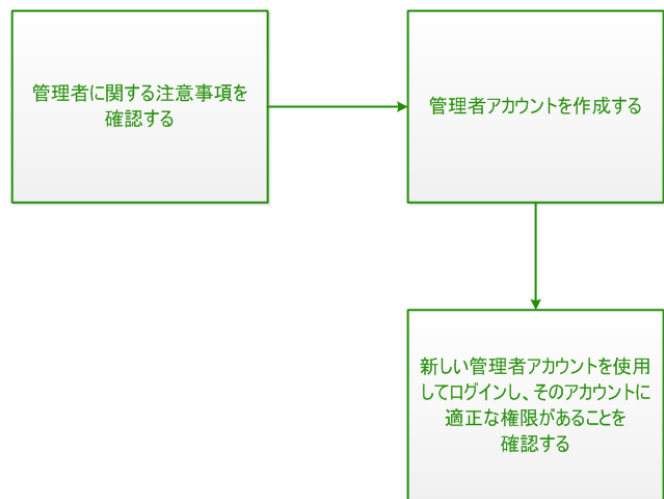
管理者を作成する方法

SiteMinder 管理者アカウントの権限をきめ細かく設定し、該当する管理者が使用できる管理機能を指定できます。

SiteMinder 管理者は、1 つ以上のセキュリティ カテゴリに権限を割り当てられます。それらのカテゴリによって認証方式の管理など 管理 UI でその管理権限が定義されます。デフォルトで管理者は、割り当てられたセキュリティ カテゴリに関連したすべての SiteMinder オブジェクトにアクセスできます。



スーパーユーザまたは必要な権限を持つ他の管理者



1. [管理者に関する注意事項を確認する](#) (P. 71)
2. [管理者アカウントを作成する](#) (P. 71)
3. [新規管理者アカウント権限を確認する](#) (P. 72)

管理者に関する注意事項

管理者を設定する前に、以下のガイドラインを確認してください。

- このプロセスは、管理 UI で管理者 ID のソースとしてのみ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する場合は、レガシー管理者を作成する。
- 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。 これら権限を特定することは、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。
- 管理者がアプリケーションセキュリティ ポリシーに責任を負うかどうか。

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、管理者が GUI 権限とレポート権限を持つ場合、その管理者は、これらの権限を持つ別の管理者を作成できますが、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。同様に、管理者は、割り当てられたワークスペースにより定義された範囲と同等またはそれ以下の範囲を持つ管理者のみを作成できます。

管理者アカウントの作成

管理者アカウントを作成することにより管理者を作成します。

以下の手順に従います。

1. スーパーユーザまたは適切な権限を持った他の管理者アカウントを使用して、管理 UI にログインします。
2. [管理] - [管理者] をクリックします。
3. [管理者] をクリックします。
4. [管理者の作成] をクリックします。
5. [一般] の下の [検索] をクリックします。
6. 検索条件を指定し、[検索] をクリックします。
7. 目的の管理者を選択し、[選択] をクリックします。

8. 以下のタスクのいずれかを実行します。
 - 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。
9. 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]セクションで指定します。それぞれのロールにおいて管理者に必要な方法を選択します。

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。
10. [権限] セクションで [追加] をクリックします。
11. 管理者に付与するセキュリティ カテゴリを選択し、[OK] をクリックします。

注：セキュリティ カテゴリは、特定の SiteMinder オブジェクトに相当する 1 つ以上のタスクを含みます。
12. 各セキュリティ カテゴリに対し、管理者に付与する権限を [権限] テーブルで指定します。

注：テーブル内のセキュリティ カテゴリに適用可能な権限のみが利用可能です。
13. [サブミット] をクリックします。

管理者が作成されます。

管理者に付与された権限が適正かどうか確認する

管理者アカウントを作成した後、そのアカウントが適正な権限であることを確認します。

以下の手順に従います。

1. 管理者アカウントを使用して、管理 UI にログインします。
2. アカウントが権限を持つセキュリティ カテゴリのみが表示されることを、管理 UI で確認します。

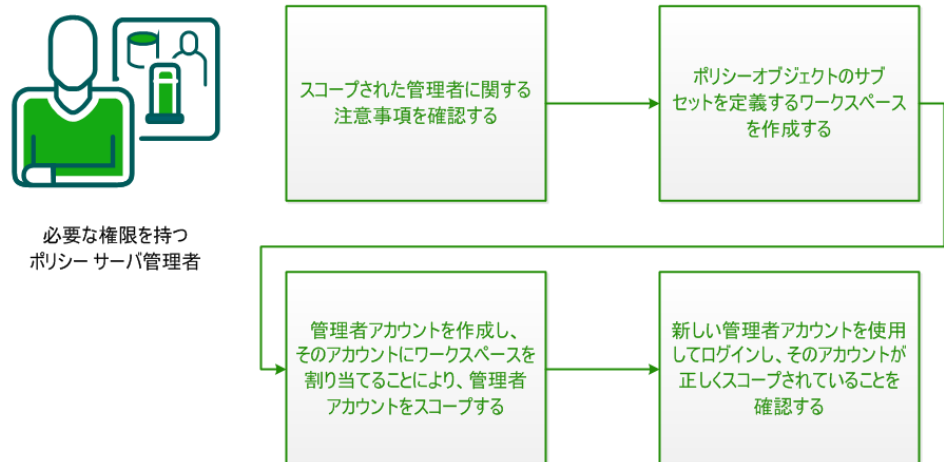
管理者アカウントの作成に必要なタスクが完了しました。

スコープされた管理者の作成方法

SiteMinder 管理者アカウントの権限をきめ細かく設定し、該当する管理者が使用できる管理機能を指定できます。

SiteMinder 管理者アカウントは、1つ以上のセキュリティ カテゴリに権限を割り当てられます。それらのカテゴリによって認証方式の管理など管理 UI でその管理権限が定義されます。デフォルトで管理者アカウントは、割り当てられたセキュリティ カテゴリに関連したすべての SiteMinder オブジェクトにアクセスできます。

ワークスペースは、SiteMinder オブジェクトのサブセットを定義します。ワークスペースを1つ以上の管理者アカウントに割り当て、使用できるオブジェクトをフィルタし、その管理権限のスコープをさらに制御します。割り当てられたワークスペースによって権限が制限される管理者アカウントは、*スコープされた管理者*と呼ばれます。



1. [スコープされた管理者に関する注意事項を確認する](#) (P. 74)
2. [SiteMinder オブジェクトのサブセットを定義するワークスペースを作成する](#) (P. 75)
3. [管理者アカウントを作成し、スコープする](#) (P. 77)
4. [新規管理者アカウントのスコープを確認する](#) (P. 79)

スコープされた管理者に関するガイドライン

スコープされた管理者は、**権限**を有するポリシー ストアのすべてのオブジェクトを管理することはできません。管理 UI は、割り当てられたワークスペースで定義されているオブジェクトのみが含まれているように見えます。

スコープされた管理者が新規のトップレベル オブジェクトを追加した場合、そのオブジェクトは直ちに同様にスコープされた他の管理者のすべてに利用可能になります。

スコープされた管理者に、新規の管理者を作成する権限がある場合、同じワークスペースまたはより**限定的な**ワークスペースでの管理者のみ作成できます。スコープされた管理者が新規ワークスペースを作成して新規管理者をさらにスコープする場合、この新規のワークスペース オブジェクトが現在のワークスペースに追加されます。その後、管理者は、新規管理者にその現在のワークスペースまたは新しいワークスペースを割り当てることができます。

新規管理者がオブジェクトを追加する場合、元の管理者もそれを表示できます。元の管理者が表示できる有効なオブジェクトセットには、自分で作成したワークスペースに追加された新規オブジェクトが含まれます。

スコープされた管理者は、**権限**を有するポリシー ストアのすべてのオブジェクトを管理することはできません。管理 UI は、割り当てられたワークスペースのオブジェクトのみが含まれているように見えます。

スコープされた管理者が新規のトップレベル オブジェクトを追加した場合、そのオブジェクトは直ちに同様にスコープされた他の管理者のすべてに利用可能になります。

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、管理者が **GUI** 権限とレポート権限を持つ場合、その管理者は、これらの権限を持つ別の管理者を作成できますが、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。同様に、管理者は、割り当てられたワークスペースにより定義された範囲と同等またはそれ以下の範囲を持つ管理者のみを作成できます。

新規管理者がオブジェクトを追加する場合、元の管理者もそれを表示できます。元の管理者が表示できる有効なオブジェクトセットには、自分で作成したワークスペースに追加された新規オブジェクトが含まれます。

注: ワークスペースを使用してスコープされるのは**管理者アカウント**のみです。レガシー管理者をスコープすることはできません。ただし、ポリシーストアでレガシー管理者レコードと関連付けられた管理者アカウントは、スコープすることができます。

スコープされた管理者を設定する際には、その管理者の以下の属性を考慮してください。

- 管理者のスコープ。管理者はすべてのポリシー データを管理できるのか、またはワークスペースで定義されたポリシー データのサブセットを管理できるのかという点です。
- 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。
- 管理者がアプリケーション セキュリティ ポリシーに責任を負うかどうか。

ワークスペースの作成

ワークスペースを作成して、スコープされた管理者が管理者権限を持つトップレベルのポリシー オブジェクト（たとえば、ドメイン、認証方式など）のサブセットを定義します。

注: ワークスペースの実際の内容は、トップレベルのコンテンツに加えて、任意の子オブジェクトから構成されます。例には、ドメインの下のレルム、およびすべての必要なオブジェクト（自動的に含まれるオブジェクト）が含まれています。

ワークスペースのコンテンツは以下に示すように動的です。

- ワークスペースのコンテンツはいつでも変更できます。ワークスペースに割り当てられたすべての管理者は、新規オブジェクトを直ちに利用できます。ワークスペースに割り当てられた管理者は、削除されたすべてのオブジェクトを利用できなくなります。
- スコープされた管理者にオブジェクト タイプを変更する権限がある場合、その管理者はワークスペースでオブジェクトを作成できます。これらの新規オブジェクトは、ワークスペースに割り当てられたすべての管理者に直ちに利用可能です。

以下の手順に従います。

1. SiteMinder スーパーユーザまたは適切な権限を持った他の管理者アカウントを使用して、管理 UI にログインします。
2. [管理] - [管理者] - [ワークスペース] - [ワークスペースの作成] をクリックします。
[ワークスペースの作成] ページが表示されます。
3. [一般] セクションのフィールドにワークスペースの名前および説明を入力します。
4. [メンバ] セクションで、必要なポリシー データのサブセットを定義するオブジェクトをワークスペースに追加します。

注: 一般的に使用されている一部のオブジェクトをワークスペースに追加して、[メンバ] リストにデフォルトで表示されるようにします。必要に応じてそれを削除することもできます。

- a. [検索] をクリックします。
[ワークスペース コンテンツの選択] ページが表示されます。
- b. タイプのドロップダウン メニューの [検索対象] オブジェクトからワークスペースに追加するオブジェクトのタイプを選択します。オプションで、[名前] または [説明] (または両方) によって検索を特定のオブジェクトに絞り込みます。

- c. [検索] をクリックします。

一致したオブジェクトが一覧表示されます。

注: ログインしている管理者アカウント自体がスコープされる場合は、検索条件に一致したオブジェクトの一覧は使用できるオブジェクトに限定されます。

- d. ワークスペースに追加するオブジェクトを選択し（複数選択可）、[選択] をクリックします。

[ワークスペースの作成] ページが再度表示されます。

- 5. (オプション) 対応する [読み取り専用] チェック ボックスをオンにしてワークスペース メンバの管理に設定します。

- 6. [サブミット] をクリックします。

ワークスペースの作成タスクがサブミットされて、処理が実行されます。SiteMinder では、ワークスペースに矛盾がない（ワークスペースのオブジェクトに関連する必要なすべてのオブジェクトがワークスペースに存在する）ことを確認します。矛盾する場合、不在のオブジェクトが追加され、情報ダイアログが表示されて、一部のオブジェクトが自動的に追加されてワークスペースに矛盾がなくなったことを示します。

管理者の作成とワークスペースの割り当て

管理者アカウントを作成し、管理できるオブジェクトのスコープを定義するワークスペースを割り当てることによって、スコープされた管理者を作成します。

以下の手順に従います。

1. スーパーユーザまたは適切な権限を持った他の管理者アカウントを使用して、管理 UI にログインします。
2. [管理] - [管理者] をクリックします。
3. [管理者] をクリックします。
4. [管理者の作成] をクリックします。
5. [一般] の下の [検索] をクリックします。
6. 検索条件を指定し、[検索] をクリックします。

7. 目的の管理者を選択し、[選択] をクリックします。
8. オブジェクトのサブセットを定義するワークスペースを選択し、管理者を[ワークスペース]のドロップダウン リストからスコープします。
9. 以下のタスクのいずれかを実行します。
 - 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。
10. 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] セクションで指定します。それぞれのロールにおいて管理者に必要な方法を選択します。

例：管理者が **XPSTImport** と **XPSTExport** 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。
11. [権限] セクションで [追加] をクリックします。
12. 管理者に応じたセキュリティ カテゴリを選択し、[OK] をクリックします。

注：セキュリティ カテゴリは、特定のオブジェクトに相当する 1 つ以上のタスクを含みます。
13. 権限（[読み取り] - [権限] - [変更] - [伝達]）を選択して、[権限] セクションで追加したセキュリティ カテゴリに適用します。
14. [サブミット] をクリックします。

スコープされた管理者が作成されます。

管理者がスコープされていることを確認する

管理者アカウントにワークスペースを割り当てた後、アクセスできるのはスコープされたオブジェクトのサブセットのみであることを確認します。

以下の手順に従います。

1. スコープされた管理者アカウントを使用して、管理 UI にログインします。
2. ワークスペースでスコープされたオブジェクトのサブセットのみが表示されることを、管理 UI で確認します。

スコープされた管理者アカウントの作成に必要なタスクが完了しました。

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。

- SiteMinder のデフォルトの管理者（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。

- スーパーユーザによる上級管理者の作成
- スーパーユーザがワークスペースの作成および割り当てを行い、上級管理者をスコープする
- 上級管理者が下級管理者を作成する
- 上級管理者 がワークスペースの作成および割り当てを行い、さらに下級管理者をスコープする

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

管理者権限がすべてのポリシー オブジェクトに有効なのか、または割り当てられたワークスペースで定義されたオブジェクトのサブセットに有効なのかを示します。

権限

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

管理者の完全権限セットを定義します。権利はセキュリティ カテゴリ、スコープ、および権限に基づきます。

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」と呼ばれる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	権限 *
エージェント管理	V, M

セキュリティカテゴリ	権限 *
アプリケーション管理	V, M, P
ポリシー管理	V, M, P

* 権限：表示、管理、伝達、実行（レポート実行の場合のみ）

注：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ができます。

この段階で、上級管理者が管理 UI にログインすると、その上級管理者はポリシーストアのエージェント、アプリケーション、ドメインをすべて表示および変更できます。上級管理者が管理者アカウントを作成すると、その上級管理者は、「アプリケーション管理」および「ポリシー管理」のセキュリティ カテゴリのみをそのアカウントに割り当てるができます。

スーパーユーザがワークスペースの作成および割り当てを行い、上級管理者をスコープする

スーパーユーザは上級管理者のスコープを制限して、上級管理者がポリシーデータのサブセットのみ管理できるようにしたい場合があります。これを行うには、スーパーユーザは、まず以下のメンバを持つ **Workspace1** という名前のワークスペースを作成します。

タイプ	名前
エージェント	Agent1
アプリケーション	Application1
アプリケーション	Application2
ドメイン	DomainB

スーパーユーザは次に、**Workspace1** を割り当てるために 上級管理者の [管理者] レコードを編集します。

この段階で、上級管理者がログインすると、DomainB 内の Agent1、Application1、Application2、およびポリシーのみを表示して変更できます。ポリシー ストアの他のエージェント、アプリケーションおよびドメインは管理 UI に表示されません。

上級管理者による下級管理者の作成

このユース ケースの後半の部分では、特定の権限のセットを持つ管理者が別の管理者を作成する方法を示します。

「上級管理者」が、「下級管理者」と呼ばれる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 カテゴリを追加するとき、（上級管理者が伝達権限を有している）以下の 2 つの権限のみ利用できます。

- アプリケーション管理
- ポリシー管理

注: 伝達権限により、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

セキュリティ カテゴリ	権限 *
アプリケーション管理	V
ポリシー管理	V, M

* 権限：表示、管理、伝達、実行（レポート実行の場合のみ）

この段階で、下級管理者が管理 UI にログインすると、その下級管理者は Application1 と Application2 を表示できます。さらに DomainB を表示し変更できます（デフォルトで上級管理者のスコープに制限される）。

上級管理者がワークスペースの作成および割り当てを行い、さらに下級管理者をスコープする

上級管理者が下級管理者のスコープをさらに限定して、それらの下級管理者がポリシー データのより小さいサブセットのみを管理できるようにします。これを行うには、上級管理者は、まず以下のメンバを持つ **Workspace2** という名前のワークスペースを作成します。

タイプ	名前
アプリケーション	Application2
ドメイン	DomainB

上級管理者は次に、**Workspace2** を割り当てるために下級管理者の「管理者」レコードを編集します。

この段階で、上級管理者がログインすると、**DomainB** の **Application2** のみを表示し、**DomainB** のポリシーを変更できます。ポリシー ストアの他のアプリケーションおよびドメインは 管理 UI に表示されません。

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](#) (P. 83)。
2. レガシー管理者レコードを作成します。
3. 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。

- ポリシー管理 API を使用する SiteMinder オブジェクトを管理するためにレガシー管理者が必要な場合
- トラストド ホスト管理者として機能するためにレガシー管理者が必要な場合

注: ポリシー ストアが管理者 ID のソースとして設定されている（デフォルト）場合、レガシー管理者は 管理 UI にアクセスすることもできます。いったん外部管理者ストアが設定されると、レガシー管理者アカウントは 管理 UI にアクセスできません。

レガシー管理者レコードの作成

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

以下の手順に従います。

1. [管理] - [管理者] をクリックします。
2. [レガシー管理者] をクリックします。
3. [レガシー管理者の作成] をクリックします。
4. [OK] をクリックします。
5. [一般] セクションで、以下のタスクを実行します。

- 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。

- 管理者が新しい認証情報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 の表示名は以下の形式で表示されます。

ログイン名: Joe Smith

6. 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。
7. 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。

8. 以下のいずれかの項目を行います。
 - 以下のタスクみを実行するための権限を持つ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
 - 上記のタスクの権限を持つレガシー管理者を作成している場合は次の手順に進み、以下のいずれかの項目を行います。
 - 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

9. [管理者の権限] セクションから [システムまたはドメイン] オプションを選択します。

注: 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、ポリシー管理 API 内のすべてのポリシー ドメインにアクセスできます。[システム] を選択すると、[タスク] セクションが表示されます。

ドメイン

管理者は、ポリシー管理 API 内の一部のポリシー ドメインにアクセスできます。[ドメイン] を選択すると、[タスクと範囲] セクションが表示されます。[タスク] セクションには、実行できる管理タスクがリスト表示されます。[範囲] セクションには、管理できるドメインがリスト表示されます。

注: [レガシー管理者の作成] 画面の [範囲] セクションは、ワークスペースを使用した管理者アカウント スコープとは関係ありません。

10. 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、以下のタスクを実行します。
 - 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] セクションで選択します。

11. [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。

注: (ポリシー ストアで [レガシー管理者] レコードと関連付けられた) 管理者アカウントもまた、管理 UI アクセスを提供するために作成されます。

管理 UI 権限の委任

レガシー管理者が作成されると、ポリシー ストアで同じレコードと関連付けられた管理者アカウントもまた生成されます。この管理者アカウントは、レガシー管理者に対して指定された設定に相当する 管理 UI アクセス権限で設定されます。

アクセス権限に影響するレガシー管理者設定への変更は、関連する管理者アカウントに自動的に伝達されます。

また、管理者アカウントに利用可能なきめ細かい 管理 UI アクセス権限を利用するために関連する管理者設定を直接変更できます。

注: 関連する管理者アカウント設定への変更は、レガシー管理者に伝達されません。さらに、いったん関連する管理者の変更が行なわれると、レガシー管理者への変更は管理者アカウントにはもう伝達されません。

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はありません。

管理者を無効にする方法

1. [管理] - [管理者] をクリックします。
2. [管理者] をクリックします。
[管理者] ページが表示されます。
3. 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。
4. 無効にするユーザの名前をクリックします。
[管理者の表示] ページが表示されます。
5. [変更] をクリックします。
設定とコントロールがアクティブになります。
6. [無効] を選択します。
7. [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] オプションをクリアに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はありません。

以下の手順に従います。

1. [管理] - [管理者] をクリックします。
2. [レガシー管理者] をクリックします。
3. 検索条件を指定し、[検索] をクリックします。

4. 無効にするユーザの名前をクリックします。
5. [変更] をクリックし、設定をアクティブにします。
6. 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、ポリシー管理 API 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。
7. (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] をクリックします。
 - b. [管理者] をクリックします。
 - c. 検索条件を入力し、[検索] をクリックします。
 - d. 権限を削除した [レガシー管理者] の名前をクリックします。
 - e. [変更] をクリックし、設定をアクティブにします。
 - f. [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。

以下を行うことで [レガシー管理者] を有効にできます。

- レガシー管理者アカウントを変更しシステムまたはドメインに固有のタスクを含める。
- 該当する場合、それぞれの [管理者] レコードの [無効] チェック ボックスからチェックを外します。

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。

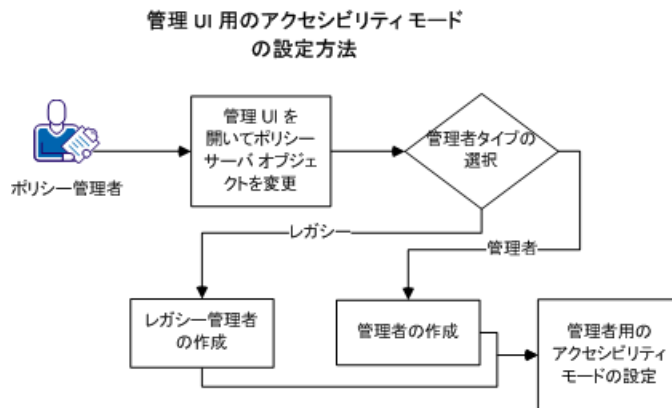
- smreg ユーティリティを使用して、デフォルトの SiteMinder スーパーユーザアカウントのパスワードを変更します。
- XPSSecurity ユーティリティを使用して、外部管理者ストアに存在しているユーザをスーパーユーザにします。

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。

管理 UI 用のアクセシビリティモードを設定する方法

本製品では、アクセシビリティを設定できます。

以下の図は、管理 UI 用のアクセシビリティモードを設定する方法を示しています。



以下の手順に従います。

1. [管理 UI を開いてポリシー サーバ オブジェクトを変更します](#) (P. 90)。
2. (以下のリストから) [管理者タイプを選択します](#) (P. 91)。
 - [管理者を作成します](#) (P. 91)。
 - [レガシー管理者を作成します](#) (P. 92)。
3. [管理者用のアクセシビリティモードを設定します](#) (P. 93)。

ポリシー サーバ オブジェクトの変更

管理 UI を開いて、ポリシー サーバ上のオブジェクトを変更します。

以下の手順に従います。

1. ブラウザで以下の URL を入力します。

`https://host_name:8443/iam/siteminder/adminui`

`host_name`

管理 UI ホスト システムの完全修飾名を指定します。

2. [ユーザ名] フィールドに **SiteMinder** スーパーユーザ名を入力します。
3. [パスワード] フィールドに **SiteMinder** スーパーユーザ アカウントのパスワードを入力します。

注: スーパーユーザ アカウントのパスワードにドル記号 (\$) 文字が含まれている場合は、ドル記号文字の各インスタンスを **\$DOLLAR\$** に置換します。たとえば、**SiteMinder** スーパーユーザ アカウント パスワードが **\$password** の場合は、パスワード フィールドに「**\$DOLLAR\$password**」と入力します。

4. 適切なサーバ名または IP アドレスが [サーバ] 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。
5. [ログイン] を選択します。

管理者タイプの選択

使用可能な管理者のタイプは、以下のとおりです。

- この製品以外の外部のサードパーティ データベースに格納されているアカウントおよび認証情報を持っている管理者。
- ポリシー ストア内部に格納されているアカウントおよび認証情報を持っているレガシー管理者。

任意のタイプの管理者をアクセシビリティ モードを使用するように設定できます。ただし、管理者を作成するには、[外部データベースをまず設定する必要があります](#) (P. 53)。

アクセシビリティ モードを設定する管理者タイプを以下のいずれかから選択します。

- [管理者](#) (P. 91)
- [レガシー管理者](#) (P. 92)

管理者の作成

管理 UI でレガシー管理者を作成します。このレガシー管理者は、本製品のアクセシビリティ モードを使用します。

以下の手順に従います。

1. [管理] - [管理者] を選択します。
2. 「管理者」を選択します。
3. [管理者の作成] を選択します。
4. [一般] の下の [検索] を選択します。
5. 検索条件を指定し、[検索] を選択します。
6. 目的の管理者を選択し、[選択] を選択します。
7. [サブミット] を選択します。
8. [この管理者用のアクセシビリティ モードを設定します](#) (P. 93)。

レガシー管理者の作成

管理 UI でレガシー管理者を作成します。このレガシー管理者は、本製品のアクセシビリティ モードを使用します。

以下の手順に従います。

1. 管理 UI から、[管理] - [管理者] - [レガシー管理者] を選択します。
2. [レガシー管理者の作成] を選択します。
3. 以下のオプション ボタンが選択されていることを確認します。

レガシー管理者タイプの新規オブジェクトの作成

4. [OK] を選択します。
5. [名前] フィールドを選択し、レガシー管理者のユーザ名を入力します。
6. 以下のオプション ボタンが選択されていることを確認します。

SiteMinder データベース

7. [パスワード] フィールドを選択し、レガシー管理者のパスワードを入力します。
8. [パスワードの確認入力] を選択し、手順 7 で使用したのと同じパスワードを入力します。
9. 以下のオプション ボタンを選択します。

システム

10. [サブミット] を選択します。
11. この管理者用のアクセシビリティ モードを設定します。

管理者用のアクセシビリティ モードの設定

管理者を作成した後、管理者用のアクセシビリティ モードを設定します。

以下の手順に従います。

1. 管理 UI から、[管理] - [管理者] - [管理者] を選択します。
2. アクセシビリティ モードを設定するレガシー管理者の右側にある [編集] アイコンを選択します。
3. 以下のチェック ボックスをオンにします。

GUI を許可

4. [追加] を選択します。
5. 以下の手順に従い、アクセシビリティ モードを設定します。
 - a. [権限の作成] 画面で、以下の表の「セキュリティ カテゴリ」列に表示されている項目のチェック ボックスをオンにします。
 - b. 「セキュリティ カテゴリ」列に対応するチェック ボックスをすべてオンにして、[OK] を選択します。
 - c. 以下の表に示すように、権限の列 (V、M、および X) のチェック ボックスをオンにします。
6. [サブミット] を選択します。

アクセシビリティ モードが設定されて、確認メッセージが表示されます。アクセシビリティ モードを必要とする管理者は、管理 UI にアクセスするためにこの管理者アカウントを使用できます。

セキュリティ カテゴリ	V	M	X	B	R	P
保守管理	X	X				
エージェント管理	X	X				
エージェント タイプ管理	X	X				
アプリケーション管理	X	X				
アプリケーション ロール管理	X	X				
認証管理	X	X				
ディレクトリ管理	X	X				
ドメイン管理	X	X				

セキュリティカテゴリ	V	M	X	B	R	P
式管理	X	X				
グローバル ポリシー管理	X	X				
ホスト管理	X	X				
レガシー ドメイン管理	X	X				
マッピング管理	X	X				
パスワード ポリシー管理	X	X				
ポリシー管理	X	X				
レポート： ユーザ別アクティビティ			X			
レポート： 管理操作			X			
レポート： アプリケーション			X			
レポート： ユーザ別アプリケーション			X			
レポート： 拒否された許可			X			
レポート： 拒否されたリソース			X			
レポート： ロール別ポリシー			X			
レポート： 保護されたリソース			X			
レポート： リソース アクティビティ			X			
レポート： ユーザ別リソース			X			
レポート： アプリケーション別ロール			X			
レポート： リソース別ロール			X			
レポート： リソース別ユーザ			X			
レポート： ロール別ユーザ			X			
ユーザ管理	X	X				
変数管理	X	X				

第 4 章: ユーザ セッション

ユーザ セッションの概要

本製品は、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オンラインでの取引が可能になり、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 ブラウザに cookie を保持するだけでなく、ハードディスクに書き込まれる cookie を使用するよう SiteMinder を設定できます。ハードディスク上で保持される SiteMinder cookie は永続 cookie といいます。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。

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ます。セッション チケットにより、シングルサインオン SiteMinder 環境内のすべてのサイトにわたっ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ます。また、それらを解読し、確認できるのはポリシー サーバのみです。SiteMinder 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: 非永続的および永続的 Cookie は、非永続的または永続的なユーザの SiteMinder セッション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。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。

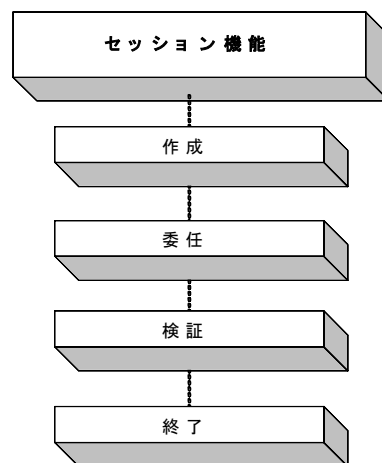
永続セッション

エージェントはセッションストア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、クライアントのオプションの**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。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

以下の画像に示すように、**SiteMinder** 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中に多くの機能を実行します。



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。

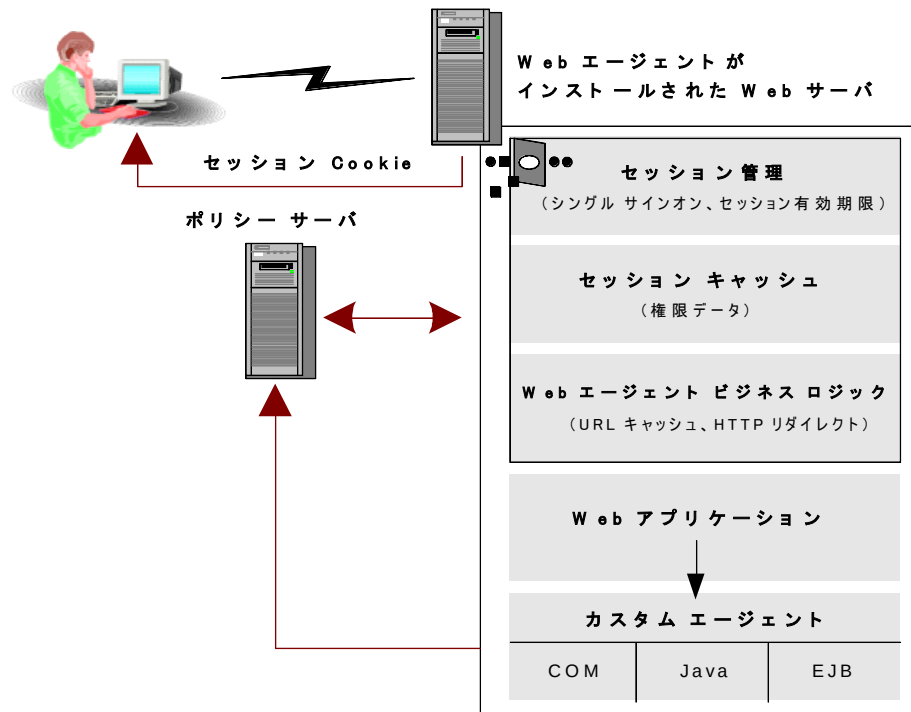
セッション終了

以下のいずれかのイベントが発生した場合に、ユーザセッションを終了します。

- ユーザがログアウトした場合。
- 設定されたセッション タイムアウトが期限切れになった場合。
- SiteMinder システム マネージャがユーザを無効にした場合。

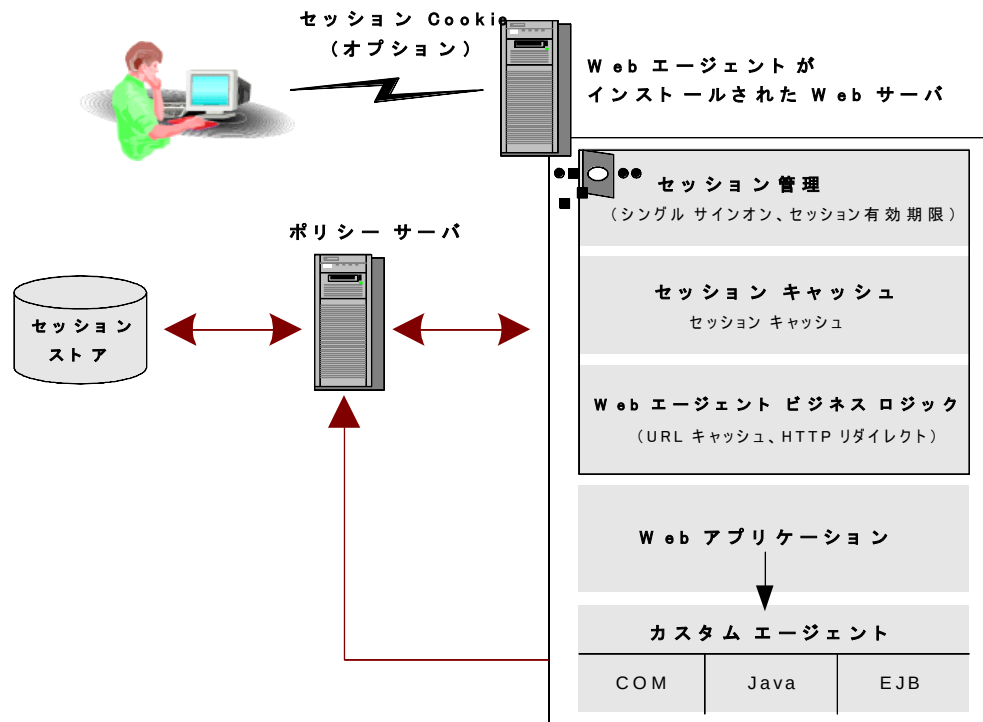
ログアウトした場合、またはユーザセッションが失効した場合には、ユーザは再度ログインして新しいセッションを作成する必要があります。手動で無効にされたユーザはセッションを再開できません。

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。



エージェントはアプリケーション層間で
セッション チケット情報を引き渡します。

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。



エージェントはアプリケーション層間でセッション チケット 情報を引き渡します。

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

リソースへのアクセスを要求するユーザは、そのリソースが含まれるレルムのコンテキスト内に作成されるセッションを保持します。レルムに関連付けられている認証方式によって、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、以下のコンポーネントを定義する独自の HTTP ヘッダを使用することにより、製品環境のすべての Web サーバで利用できます。

- 使用されている認証方式。
- ユーザが認証しているネームスペース。

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の追加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本製品が設定されている場合は、認証方式に保護レベルを割り当てることもできます。これらの保護レベルは管理者が割り当てます。レベルの範囲は 1 ～ 1000 です。1 は安全性が最も低いレベルで、1000 は安全性が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、別のレルムのセッションにおいても検証済みにすることができます。他のレルムの保護レベルは、元のレルムと比べて同等または低くなります。保護レベルが同等または低い場合、そのユーザは再認証を受ける必要はありません。元のユーザセッションは有効なままになります。より高い保護レベルの認証方式に関連付けられたリソースへのアクセスを要求するユーザは、認証情報を再入力するように求められます。元の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 \(P. 307\)](#)」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

本製品は、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、セッション情報は保持されます。複数の Cookie ドメインにわたってセッション情報を維持するには、シングルサインオンのエージェントを設定します。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。

これらのドメインにより、ユーザは 1 つの cookie ドメインでいったん認証を受けた後に、情報の入力を再度求められることなく、別の cookie ドメインにアクセスできます。

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لم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

エージェントは、まだアクティブであるかどうかを確認することによりユーザセッションを検証します。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。エージェントは **cookie** を読み取り、キャッシュに情報が存在する場合に、セッションを検証します。情報が存在しない場合、エージェントはポリシー サーバに問い合わせ、ユーザの識別情報を確認します。エージェントは、その他のセッション情報や許可情報も収集します。

使用される保護レベルよりも高い保護レベルのリソースにアクセスするユーザは、別の認証が実行されても、セッション情報を保持します。

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。各アプリケーションは、セッションチケットを取得した後でポリシー 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。

Web エージェントは、委任のためにセッションチケットを使用するだけでなく、一連のデフォルト **HTTP** ヘッダをセッション管理のために使用可能にします。これらのデフォルトヘッダは、**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、他の種類の一意の 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。ユニバーサル ID は、アプリケーションにかかわらず、ユーザ ID を自動的に送信します。さらに、その ID はアプリケーションに組み込まれます。組み込まれた ID は、常に変化するユーザディレクトリとは別個のユーザ識別法をアプリケーションに提供します。

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、タイムアウト値によって指定されます。セッション タイムアウトは、以下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。
Idle Timeout (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の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。

名前	目的
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

エージェントは、本製品が保護する環境内の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エージェントで同じ値に設定する必要があります。

製品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。製品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。

キーの更新とセッション タイムアウトを調整してください。そうしないと、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。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 回実行すると、最初のキーのロールオーバーの *前*に書き込まれた **cookie** はすべて無効になります。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、最大セッション タイムアウトを 6 時間に設定します。この設定により、複数のキー ロールオーバーによってセッション **cookie** が無効になることが防止されます。

ユーザ セッションを終了する方法

ユーザ セッションは、以下の方法のいずれかの方法で終了できます。

- ユーザがログアウトする。
- セッション タイムアウトが期限切れになる。
- システム マネージャがユーザを無効にする。

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。アプリケーション (Microsoft Exchange Server、SQL Server など) で Microsoft のアクセス制御リスト (ACL) またはその他のツールを使用してセキュリティを提供するには、ユーザのセキュリティ コンテキストが必要です。

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

エージェントは、Windows ユーザ セキュリティ コンテキストを提供し、IIS Web サーバ (Windows 2000 プラットフォーム) 上の Web リソースへのアクセスを可能にします。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実行できます。

本製品は、**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**

本製品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。

- 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。

本製品は、**HTTP 1.x** 要求、**HTTP Cookie**、および必要に応じて **SSL/TLS** をサポートするブラウザと連携することができます、

- 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。本製品は、リソースへのアクセスを許可するときに、認証情報収集のこのメカニズムと **Windows セキュリティ コンテキスト**を組み合わせて使用します。

- 本製品では、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 は汎用フォームの認証情報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

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します。セッションストアは永続セッションをメンテナンスします。

セッションストアは、ユーザの暗号化された認証情報を格納し、ユーザをセッション ID に関連付けます。セッションがクライアントと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パージされると、エージェントはセッションストアからユーザの認証情報を取得します。認証情報の取得により、エージェントは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伝達するために、セッションストアにはセキュリティ 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

ポリシー サーバを通じて **Web** エージェントを明示的に設定し、セッションの再検証のためにセッション ストアとコンタクトをとることができます。ユーザのブラウザに格納されたセッション **Cookie** にはセッション ID が含まれます。**Cookie** はセッション ID を使用して、セッション ストアからユーザの認証情報を再び取得します。

注: セッション **Cookie** が期限切れになると、その ID に関連付けられた認証情報は無効になります。そのため、ユーザは再認証する必要があります。

検証期間により、エージェントがセッションを再検証する間隔が決定します。検証期間は、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。検証期間が期限切れになると、エージェントはセッション ストアへの問い合わせを行います。

検証期間が短すぎると、エージェントがセッション ストアに問い合わせる回数が増えるため、本製品がリクエストを処理する速度に影響します。有効期間は長めに設定してください。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 値よりも少ない場合、エージェントはキャッシュ情報を使用します。

注: セッション サーバが動作していないと、検証期間は無限になり、エージェントはセッション ストアに問い合わせを行いません。

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。
- ユーザに Web サーバへのローカル ログイン権限があること。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インする権限が必要になります。
- World Wide Web 発行サービスを開始するアカウントについては、以下のガイドラインを考慮します。
 - システム アカウントには、追加の設定は必要ありません。
 - ドメインアカウントには、オペレーティング システムの一部として機能するための関連するユーザの権限が必要です。

設定の概要

手順	製品コンポーネント
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	ポリシー サーバ管理コンソールの [データ] タブ
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
サポートされている認証方式 (P. 110)のうちの 1 つを各レルムと関連付ける	[レルム] ウィンドウのリソース グループ ボックス
各レルム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	[レルム] ウィンドウのセッション グループ ボックス

手順	製品コンポーネント
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

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

ユーザ証明は提供されないため、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。ユーザ証明を収集するには、基本認証またはフォーム認証と証明認証を組み合わせる必要があります。

NTLM 認証方式には、本製品の Windows ユーザ セキュリティ コンテキストとの互換性はありません。Web サーバ上の一部のリソースに NTLM 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 コンテキストを使用します。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。

詳細情報:

[認証方式](#) (P. 307)

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。

本製品でシームレスな **Microsoft** セキュリティ コンテキストをサポートするには、**NetBIOS** 名が、**Microsoft** の推奨する **DNS** ドメイン名と一致している必要があります。

Active Directory ドメインの **DNS** 名が、その **NetBIOS** 名と異なる場合は、ユーザ ドメインを確立できません。本製品が **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、**Web** サーバはセキュリティ 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

本製品は、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、すべてのレلمは永続的である必要があります。そうでないと、本製品がセキュリティ コンテキストの永続化を試行する際に、ユーザは再び識別情報の入力を求められます。

第 5 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 \(P. 113\)](#)
- [トラステッド ホストホストのホスト設定オブジェクト \(P. 118\)](#)
- [SiteMinder エージェントの概要 \(P. 124\)](#)
- [Web エージェント設定の概要 \(P. 128\)](#)
- [Web エージェントの設定方法 \(P. 132\)](#)
- [エージェント設定オブジェクトの概要 \(P. 139\)](#)
- [Web エージェントの有効化 \(P. 146\)](#)
- [RADIUS エージェントの設定 \(P. 147\)](#)
- [エージェント グループ \(P. 148\)](#)
- [カスタム エージェント \(P. 150\)](#)
- [エージェント ディスカバリの紹介 \(P. 153\)](#)

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 ホストとは、1 つ以上のエージェントをインストールできるクライアントシステムです。

ポリシー サーバへのトラステッド ホストの登録

エージェントをインストールするシステムからトラステッド ホストを登録します。ホストの登録は、エージェント設定プロセスの一部です。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。

重要: ホストの登録は、**Web** エージェントをインストールまたは設定するたびに行うのではなく、1 回のみ行います。

初期設定時に、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。エージェントは、ホスト設定ファイル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。最初の試行に失敗した場合、トラステッドホストはリストされている次のポリシー サーバへの接続を試行します。

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。

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 **Smhost.conf** ファイル内でホスト設定オブジェクトを検索します。トラステッドホストは、このホスト設定オブジェクトから、その設定の詳細を取得します。

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 **PolicyServer** 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。

リクエスト タイムアウト

RequestTimeout 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。設定を使用することで、**Web** サーバの応答時間が最適化されます。

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。 使用できる動作モードは、以下のとおりです。

フェールオーバー モード

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**としてマークします。トラステッド ホストは、リスト内の次のポリシー サーバにリクエストを送信します。 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 モード

ラウンドロビンモードは動的なロードバランシング モードです。 エージェントは、[ホスト設定オブジェクト] にリストされているすべてのポリシー サーバ間でのリクエストを分散させます。

ラウンドロビンモードでは、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、それ以降も同様です。 このプロセスは、トラステッド ホストがすべての使用可能なポリシー サーバにリクエストを送信するまで続行します。 次のリクエストはリストの最初にあるポリシー サーバに戻り、サイクルが再開されます。

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。

この設定をお勧めします。理由は、複数のポリシー サーバを使用するときにダイナミック ロードバランシングによってスループットが向上し、ユーザの認証および許可がより効率的になるためです。 ダイナミック ロードバランシングまた、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。 ロードバランシング ポリシー サーバの 1つが利用不能になると、引き続きフェールオーバーが行われます。

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。

以下の手順に従います。

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 ストアを参照するようにします。
3. **EnableFailover** パラメータを設定します。
 - フェイルオーバー モードを有効にするには、**EnableFailover** パラメータを「はい」に設定します。
 - ラウンドロビン ダイナミック 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、および監査ポートにおいて使用可能なソケットの数により、使用可能な **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。個別の **Web** サーバスレッドにより、各ユーザのアクセス リクエストが処理されます。各スレッドには独自のソケットが必要です。**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。

いくつかの設定は、トラステッド ホストとポリシー サーバ間の TCP/IP 接続に影響します。

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。これは、トラフィックが少量から中程度の Web サイトに適しています。以下の状況では、この数を増やします。

- トラフィック量が多い Web サイトを管理している。
- 仮想サーバのエージェント ID を定義している。

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、この値を増やします。

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。より多くのソケットが必要な場合は、必要な各レベルで追加するソケットの数を増やします。

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。smreghost 登録ツールを使用して、再度ホストを登録します。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 smreghost 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。

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] をクリックします。
2. [トラステッドホスト] をクリックします。
[トラステッドホスト] ページが表示されます。

3. 検索条件を指定し、[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。
5. [トラステッドホストの削除] をクリックします。
ホストの削除を確認するメッセージが表示されます。
6. [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。

SmHost.conf ファイルの **hostconfigobject** パラメータは、エージェントが使用するホスト設定オブジェクトを識別します。エージェントは各 **Web** サーバの子プロセスの初期化フェーズ中に、**SmHost.conf** ファイル内の設定を使用します。**SmHost.conf** ファイルは起動時に使用されるだけでなく、新しい子プロセスを開始する **Web** サーバにより実行時にも使用されます。この初期化が実行された後には、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してください。

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を作成するには、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を推奨します。

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。

以下の手順に従います。

1. [インフラストラクチャ] - [ホスト] をクリックします。
2. [ホスト設定オブジェクト] をクリックします。
[ホスト設定オブジェクト] ページが表示されます。
3. [ホスト設定の作成] をクリックします。
[ホスト設定の検索] ページが表示されます。
4. [ホスト設定タイプのオブジェクトのコピーの作成] を選択し、[OK] をクリックします。

注: デフォルトでは、[DefaultHostSettings ホスト設定オブジェクト] が選択されています。

[ホスト設定の作成] ページが表示されます。

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。

5. 名前と説明を入力します。
注: ハイフン ("-") 文字でホスト設定オブジェクト名を開始しないでください。
6. [設定値とクラスタ] で、新規オブジェクトで異なるプロパティを変更します。
7. [サブミット] をクリックします。
ホスト設定オブジェクトが作成されます。

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、フェールオーバー、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 サーバを追加します。

注: 複数の仮想 IP アドレス (VIP) としてポリシー サーバを公開しているハードウェア ロードバランサの場合は、それらの VIP をフェールオーバーモードに設定することを推奨します。ハードウェア ロードバランサが同じ機能をより効率よく実行するため、ラウンドロビンモードは不必要です。

以下の手順に従います。

1. [インフラストラクチャ] - [ホスト] をクリックします。
2. [ホスト設定オブジェクト] をクリックします。
[ホスト設定オブジェクト] ページが表示されます。
3. [ホスト設定の作成] をクリックし、次に、[OK] をクリックします。
[ホスト設定の作成] ページが表示されます。

注: デフォルトでは、[ホスト設定タイプの新規オブジェクトの作成] オプションが選択されます。

4. 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. 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] ページが表示されます。
7. 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、ポートの隣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、クラスタ下の [追加] をクリックして手順 6 および 7 を繰り返します。

9. フェールオーバーしきい値のパーセンテージを入力します。

注: アクティブなサーバのパーセンテージが指定したパーセンテージ以下になると、クラスタはリスト内の次のクラスタにフェールオーバーします。

10. [サブミット] をクリックします。

ホスト設定オブジェクトはポリシー サーバ詳細により更新されます。

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

フェー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ールオーバーすることができます。

注: ハードウェア ロード バランサを使用して複数の仮想 IP アドレス (VIP) としてポリシー サーバを公開している場合は、それらの VIP をフェールオーバー設定に設定することを推奨します。ハードウェア ロード バランサが同じ機能をより効率よく実行するため、クラスタリングは unnecessary です。

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。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が作成されていることを前提とします。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。

以下の手順に従います。

1. [インフラストラクチャ] - [ホスト] をクリックします。
2. [ホスト設定オブジェクト] をクリックします。
[ホスト設定オブジェクト] ページが表示されます。
3. [ホスト設定の作成] をクリックし、次に、[OK] をクリックします。
[ホスト設定の作成] ページが表示されます。

注: デフォルトでは、[ホスト設定タイプの新規オブジェクトの作成] オプションが選択されます。

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. 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. ホスト設定を指定します。
6. [クラスタ] セクションで、[追加] をクリックします。
[クラスタの追加] ページが表示されます。
7. 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、ポートの隣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。

クラスタが追加されます。

注: クラスタを変更するには、最初の列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、最後の列のマイナス記号をクリックします。別のクラスタを追加するには、[追加] をクリックして手順 7 および 8 を繰り返します。

9. フェールオーバーのしきい値の割合を入力します。

SM は、クラスタに設定されたポリシー サーバの数と指定するフェールオーバーしきい値の割合の両方に基づいたフェールオーバーのしきい値を自動的に計算します。フェールオーバーのしきい値（各クラスタ内のサーバリストの右側の列に表示される）は、クラスタ内でアクティブであるポリシー サーバの総数であり、設定に基づく利用可能なクラスタを検討できます。クラスタ内のアクティブなサーバの数が、表示されたフェールオーバーのしきい値を下回ると、クラスタは、リスト内の次に使用可能なクラスタにフェールオーバーします。設定に基づいたフェールオーバーのしきい値の計算結果が整数でない場合、ポリシーサーバはその値を四捨五入して整数にします。

たとえば、5 つのポリシー サーバが設定されたクラスタを考えてみます。

フェールオーバーのしきい値パーセントが、

[0% ~ 19%] の範囲の場合、必要なアクティブ サーバの値（フェールオーバーのしきい値）は 1 に切り上げられます。

20% の場合、フェールオーバーのしきい値の計算値は 1 です。

[21% ~ 39%] の範囲の場合、フェールオーバーのしきい値は 2 に切り上げられます。

40% の場合、フェールオーバーのしきい値の計算値は 2 です。

[41% ~ 59%] の範囲の場合、フェールオーバーのしきい値は 3 に切り上げられます。

60% の場合、フェールオーバーのしきい値の計算値は 3 です。

[61% ~ 79%] の範囲の場合、フェールオーバーのしきい値は 4 に切り上げられます。

80% の場合、フェールオーバーのしきい値の計算値は 4 です。

[81% ~ 99%] の範囲の場合、フェールオーバーのしきい値は 5 に切り上げられます。

100% の場合、フェールオーバーのしきい値の計算値は 5 です。

[フェールオーバーのしきい値パーセント] を設定すると、その設定は、ホスト設定オブジェクトを使用するすべてのクラスタに適用されます。

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以下になると、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ールオーバーします。

10. [サブミット] をクリックします。

ホスト設定オブジェクトはポリシー サーバで設定されます。

重要: 設定値は、クラスタが指定されない場合に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ールオーバー操作を指定します。単純なフェールオーバー操作を優先するためにクラスタをすべて削除する場合は、ポリシー サーバ情報をすべて削除してください。

詳細情報:

[ポリシー サーバ オブジェクトのコピー](#) (P. 10)

SiteMinder エージェントの概要

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。これらは、ポリシー実行ポイント (PEP) と見なされます。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。ユーザは保護されているリソースを要求します。エージェントは、認証方式に基づく認証情報をユーザに対して求めます。エージェントは、ポリシー サーバに認証情報を送信します。

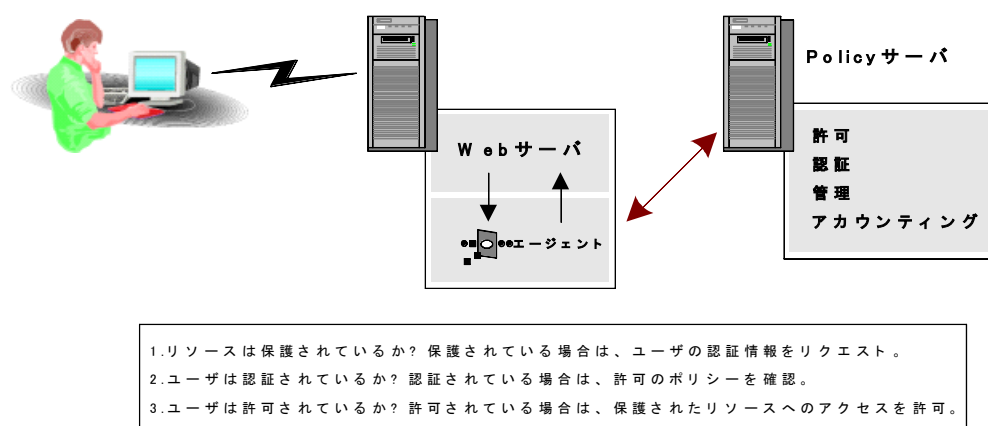
ポリシー サーバはその証明に基づいて、ユーザを認証するかどうか、また要求されたリソースへのアクセスを許可するかどうかを決定します。次にポリシー サーバはエージェントと通信し、要求されたリソースへのアクセスの許可もしくは拒否を行います。

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

エージェントは **Web** サーバと連携します。ユーザが **Web** サーバからのページを要求すると、**Web** エージェントはポリシー サーバと通信します。エージェントは認証リクエストおよび許可リクエストを処理し、ユーザはリソースにアクセスできるようになります。ポリシー サーバは、エージェントがパーソナライズされたコンテンツを提供するための情報を提供することもできます。

以下の図は、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、本製品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ができます。ユーザが **Web** サーバを介して、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。



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。保護されたリソースの場合、エージェントはユーザから認証情報を収集し、その情報をポリシー 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、エージェントはポリシー サーバに確認することにより、ユーザがリソースに対して許可されているかどうか判断します。ポリシー サーバが許可を確認すると、エージェントに通知されます。エージェントは、保護されたリソースをユーザに対して **Web** サーバが表示することを可能にします。

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。

詳細情報:

[エージェント グループ](#) (P. 148)

RADIUS エージェント

リモート認証ダイヤルイン ユーザ サービス (RADIUS) プロトコルは、ネットワーク アクセス サーバ (NAS) と RADIUS 認証サーバ間でのセッションの認証情報および設定情報のやり取りを行います。プロキシ サーバ、ファイアウォール、またはダイヤルアップセキュリティ デバイスでは、多くの場合に RADIUS プロトコルを使用します。

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。

詳細情報:

[RADIUS サーバとしてのポリシー サーバの使用](#) (P. 893)

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、本製品と J2EE プラットフォームを統合します。

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、単一のエージェントです。ポリシー サーバの観点からは、アプリケーション サーバリソースを保護する異なるエージェント タイプがあります。エージェント タイプにより、サーブレットおよび EJB コンポーネントを保護するための柔軟性を以下の方法で実現できます。

- サーブレットまたは EJB エージェントの使用
- エージェントの使用

Web サービス セキュリティ(WSS)エージェント

WSS（旧 SOA）エージェントは、Web サーバおよびアプリケーション サーバと統合され、それらサーバの SOAP/XML ベースの Web サービスのリソースへのアクセス リクエストを認証および許可します。

Web エージェント設定の概要

エージェントを設定するためのオプションは、以下のとおりです。

中央設定

ポリシー サーバからエージェント設定を行うことを示します。ポリシー ストアは、一群のエージェントによって共有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: エージェントは Web サーバ上の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。

Web エージェントを中央で設定した場合の利点

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の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により、共有秘密キーを生成できます。クライアント側は、この秘密キーを使用することにより、ポリシー サーバへの安全な接続を確立します。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

ネットワークのエージェント側には、エージェントの動作における複数の主なコンポーネントがあります。

エージェント

Web サーバへの仮想インターフェース。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。このホストは、ポリシー サーバへの接続を処理します。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、それぞれに一意の名前が必要です。

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します。

以下のデータにより、トラステッドホストが識別されます。

- 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。LocalConfig.conf ファイルは、各 Web エージェントのエージェント設定パラメータを保持します。WebAgent.conf ファイルは、エージェントを有効または無効にし、すべての関連するプラグインをロードします。

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、ホスト設定オブジェクト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。

詳細情報:

[Web エージェント設定の概要 \(P. 128\)](#)

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は、以下のオブジェクトがエージェント設定に関連しています。

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、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。

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。管理 UI を使用して、これらのオブジェクトを作成または変更します。

詳細情報:

[Web エージェント設定の概要](#) (P. 128)

Web エージェントの設定方法

エージェントを設定するには、いくつかのタスクを完了する必要があります。これらのタスクは、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: 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、エージェントの通信用のポリシーサーバをセットアップしてください。

以下の手順に従います。

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を登録する権限が必要です。

注: [トラステッドホストの登録] 権限のみを持つ管理者を作成した場合、その管理者は管理 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ます。このオブジェクトにより、エージェントの名前を定義します。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、エージェントをインストールします。
7. トラステッドホストを登録する。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設定する場合には、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。

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

ホスト設定オブジェクトを作成（または複製）できます。

以下の手順に従います。

1. [インフラストラクチャ] - [ホスト] をクリックします。
2. [ホスト設定オブジェクト] をクリックします。
3. [ホスト設定の作成] をクリックします。

4. 以下のタスクのいずれかを実行します。
 - （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します。ポリシー サーバインストール プログラムにより、**DefaultHostSettings** オブジェクトが作成されます。

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。
 - オブジェクトを作成します。
5. [OK] をクリックします。
6. 名前と説明を入力します。
7. [設定値] で、[ホスト設定] を指定します。
8. [サブミット] をクリックします。

ローカルでの Web エージェントの設定

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取ります。ローカル エージェント設定ファイル内の値は、エージェント設定オブジェクト内の値よりも優先されます。エージェントは、これらすべての設定をまとめて使用します。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、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。

以下の手順に従います。

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 エージェントを設定します。
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 (`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。

すべてのエージェントには、`<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。

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、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 **example.com** に設定します。ネットワーク内の 1 つのエージェントで、**CookieDomain** パラメータを **example.org** に設定します。エージェント設定オブジェクトで設定された他のパラメータ値をすべて使用します。

解決方法：

以下の手順に従います。

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 つのエージェントで、**CookieDomain** パラメータのみを **example.org** に変更します。他のパラメータはいずれも変更しないでください。

エージェント設定ファイル内の **CookieDomain** パラメータの値は、エージェント設定オブジェクト内の値よりも優先されます。エージェント設定オブジェクトにより、他のすべてのパラメータの設定が決まります。

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します。オブジェクト名は、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 **AgentName** パラメータまたは **DefaultAgentName** のパラメータの値と一致する必要があります。ポリシー サーバは、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プします。このマッピングにより、ポリシーがエージェントに関連付けられます。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。

以下の手順に従います。

1. [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. [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. [OK] をクリックします。
5. エージェントの名前および説明を入力します。

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。
6. エージェントの [スタイル] として **SiteMinder** を、[エージェントタイプ] として **Web エージェント** をそれぞれ選択します。
 7. [サブミット] をクリックします。

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 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します。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。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。

以下の手順に従います。

1. [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. [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。

[エージェント タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。

4. [OK] をクリックします。
5. エージェントの名前および説明を入力します。

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。

6. 以下の設定を確認します。
 - [SiteMinder] オプショ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。
7. [4.x エージェントをサポートします] オプションを選択します。
[信頼性の設定] ページが表示されます。

8. エージェントが存在するサーバの 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。

9. 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: 同じ Web サーバ上にある仮想サーバは同じ秘密キーを使用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。

10. [サブミット] をクリックします。

4.x Web エージェント オブジェクトが作成されます。

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、エージェントの設定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 LocalConfig.conf エージェント設定ファイルに相当するものです。

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、設定しているエージェントに関連付けられます。

詳細情報:

[エージェント設定オブジェクトの必須パラメータ](#) (P. 143)

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。 サポートされる **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後、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。

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コピーする方法

1. [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. [エージェント設定オブジェクト] をクリックします。
[エージェント設定オブジェクト] ページが表示されます。
3. [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] ページが表示されます。
4. [エージェント設定タイプのオブジェクトのコピーの作成] を選択し、以下のいずれかを行います。
 - 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 1 つ選択します。
5. [OK] をクリックします。
[エージェント設定の作成] ページが表示されます。
6. 新しい名前および説明を入力し、[サブミット] をクリックします。
パラメータを指定するかまたはコメントの入力を続行するよう促されます。
7. コメントを入力し [はい] をクリックします。
デフォルトのエージェント設定オブジェクトに基づいたエージェント設定オブジェクトが作成されます。

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。

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をコピーせずに、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、すべてのパラメータとパラメータ値を手動で入力します。

以下の手順に従います。

1. [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. [エージェント設定オブジェクト] をクリックします。
[エージェント設定オブジェクト] ページが表示されます。
3. [エージェント設定の作成] をクリックします。
[エージェント設定タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
[エージェント設定の作成] ページが表示されます。
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。
5. エージェントの名前および説明を入力します。
6. [追加] をクリックします。
[パラメータの作成] ページが表示されます。
7. パラメータの名前および値を入力します。

8. (オプション) 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] オプションを選択します。
- 1つのパラメータに対して複数の値を追加するには、以下を実行します。
 - a. [複数值] オプションを選択します。
 - b. パラメータに対する値を入力します。
 - 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. (オプション) 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] ページ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。単一の暗号化された値は、一連の記号として表示されます。

9. [OK] をクリックします。

リストにパラメータが追加されます。

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。

10. [サブミット] をクリックします。

エージェント設定オブジェクトが作成されます。

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では、以下のパラメータの設定が必要です。

- 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 エージェントのみ) **DominoDefaultUser**
- (Domino エージェントのみ) **DominoSuperUser**

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 エージェントでは、エージェント設定オブジェクト内の以下の設定が必要です。

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。

注: NTLM 認証方式または **Windows** ユーザセキュリティ コンテキスト機能を使用する場合は、これらのパラメータを空白のままにします。

これらのパラメータにより、本製品が保護する IIS Web サーバの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 アカウントが識別されます。アクセスを要求するユーザは、必要なサーバアクセス権限を持っていない可能性があります。エージェントでは、この NT ユーザ アカウントが必要です。管理者は、本製品におけるプロキシユーザ アカウントとしての役割を果たす、この NT アカウントを割り当てます。

Domino ユーザの指定

Domino サーバ上に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

Domino Web エージェントが Domino サーバに対してユーザを識別する名前を指定します。この状況は、そのユーザが Domino ディレクトリに存在せず、別のユーザディレクトリに対して本製品によって認証済みである場合に発生します。この値は暗号化できます。

DominoSuperUser

本製品に正常にログイン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

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

DefaultAgentName

エージェント ID が割り当てられていない IP アドレスに対して Web エージェントが使用するエージェント ID 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、暗号化または複数值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。

以下の手順に従います。

1. [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. [エージェント設定オブジェクト] をクリックします。
3. 検索条件を指定して、[検索] をクリックします。
4. 変更するエージェント設定オブジェクトの名前をクリックします。
5. [変更] をクリックします。
6. パラメータ名の左側の右方向矢印をクリックします。
7. 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。正方形記号(□)により複数のパラメータが区切られます。

8. (オプション) [暗号化] オプションを選択します。

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] ページに一連の記号として表示されます。

9. [OK] をクリックします。

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5〜8を繰り返します。

10. [サブミット] をクリックします。

Web エージェントの有効化

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。

エージェントを有効にするには、以下の前提条件を完了します。

- ポリシー サーバおよびエージェント通信において必要なオブジェクトをすべてセットアップします。
- エージェントをインストールおよび設定します。

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します。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。

以下の手順に従います。

1. [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. [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. [OK] をクリックします。
5. エージェントの名前および説明を入力します。

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。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。
6. [RADIUS] オプションを選択し、次に、[RADIUS ベンダー] を選択します。

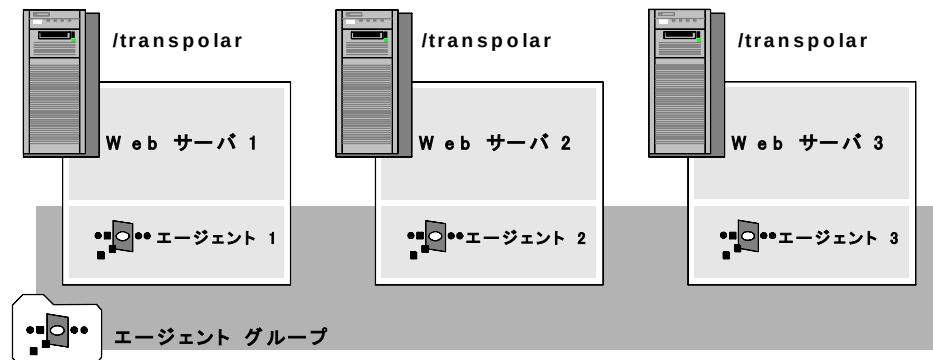
注: エージェント タイプとして Generic RADIUS を選択すると、すべてのタイプの RADIUS 機器を保護できます。Generic RADIUS エージェント タイプはベンダーに特有のレスポンス属性へのアクセスを提供しません。

7. 以下の信頼性の設定を指定します。
 - RADIUS クライアント (NAS デバイス) の IP アドレスを入力します。
 - 英数字の共有秘密キーの入力および確認を行います。
8. RADIUS レルム ヒントの数を入力します。
9. [サブミット] をクリックします。

エージェントグループ

エージェントグループとは、同じタイプのエージェントの集まりです。これらのエージェントグループにより、リソースが多くの Web サーバ上で複製されている場合に、より大きなユーザベースに対する同じリソースへのアクセスが可能になります。また、グループを使用することにより、すべての Web エージェントに対してポリシーを 1 つだけ定義して、時間を節約することもできます。エージェントグループにより、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。

以下の図では、1 セットのポリシーを使用して Web ファームを保護する方法について説明しています。この場合は、1 つのエージェントグループが一連のポリシーにバインドされます。



エージェントを作成する前に、エージェントグループを作成できます。エージェントグループを編集し、新しいエージェントをグループに追加します。

注: エージェントグループ内の設定は、エージェント設定オブジェクトと共有してください。

詳細情報:

[ポリシー](#) (P. 667)

エージェントグループの設定

エージェントグループを使用して、共有リソースをより効率よく保護するエージェントを管理します。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。

以下の手順に従います。

1. [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. [エージェントグループ] をクリックします。
3. [エージェントグループの作成] をクリックします。
4. [OK] をクリックします。
5. エージェントグループの名前および説明を入力します。
6. 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まれます。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。

7. [追加/削除] をクリックします。

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] にリスト表示されます。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。

8. [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。
9. [OK] をクリックします。
10. [サブミット] をクリックします。

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。

以下の手順に従います。

1. [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. [エージェント グループ] をクリックします。
3. 検索条件を指定し、[検索] をクリックします。
4. 変更するエージェント グループの名前をクリックします。
5. [変更] をクリックします。
6. [追加/削除] をクリックします。

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] にリスト表示されます。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。

7. [使用可能なメンバ] 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。
8. [OK] をクリックします。
9. [サブミット] をクリックします。

カスタム エージェント

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。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付属しています。

独自のエージェントを作成したら、新しいエージェント タイプを設定します。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。本製品に含まれている API を使用して、エージェント タイプを作成します。

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 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します。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が作成されていることを前提とします。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。

以下の手順に従います。

1. [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. [エージェント] をクリックします。
[エージェント] ページが表示されます。
3. [エージェントの作成] をクリックします。
[エージェント タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
[エージェントの作成] ページが表示されます。

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。

- エージェントの名前および説明を入力します。

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。

- エージェントスタイルとして **SiteMinder** を、エージェントタイプとして **Web** エージェントをそれぞれ選択し、**[4.x エージェントをサポートする]** オプションを選択します。

[信頼性の設定] フィールドが表示されます。

- エージェントが存在するサーバの IP アドレスを入力します。
- 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、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。

- [サブミット]** をクリックします。

エージェントオブジェクトが作成されます。

エージェント ディスカバリのご紹介

エージェント ディスカバリでは、さまざまなエージェント タイプおよびバージョンのインスタンスを検出できます。検出されると、バージョン、状態など、エージェントに固有の詳細を表示できます。また、ご自分の企業内のさまざまなホスト上で展開されたエージェントのリストを表示したり、リストから不要なエージェント インスタンス エントリを削除したりできます。

注: エージェント ディスカバリは従来のエージェントをサポートしません。

ハートビートメッセージは、**r12.5** エージェントの識別に役立ちます。エージェントがポリシー サーバにハートビート メッセージを送信する頻度は、ハートビート間隔と呼ばれます。ハートビート間隔を示す **MinTimeBetweenAgentStatusUpdates** パラメータは、ポリシー サーバがストアのエージェント インスタンス オブジェクトを更新する前に待機する最小時間を制御します。最小時間が経過する前にポリシー サーバがハートビート メッセージを受信しても、それは無視されます。このパラメータのデフォルト値は **24** 時間です。また、最小値は **1** 時間です。エージェントは **MinTimeBetweenAgentStatusUpdates** パラメータに設定した時間が経過したかどうかにかかわらず、起動時に自分自身を識別します。

エージェント ディスカバリは、エージェントのバージョンに関係なく、ポリシー サーバと定期的に通信するエージェントをすべて識別します。**r12.5** より古いエージェントを識別するために、エージェント ディスカバリは、エージェントの IP アドレスとトラステッドホストの組み合わせを使用します。この組み合わせに少しでも変更があると、同じエージェントに対して複数のエントリが追加されることになります。

エージェント ディスカバリでは、**r12.5** より古い **ASA** エージェントは識別されません。古いエージェントはポリシー サーバに情報を送信しません。**ASA** エージェントの属性は、エージェント インスタンス リストに不明として表示されます。**ASA** エージェント ホストに同じホストとトラステッドホストの組み合わせを使用する **r12.5** より古い別のエージェントがある場合、**ASA** エントリはリストに表示されません。

管理 UI で [インフラストラクチャ] の下の [エージェント インスタンス] を使用して、企業内で展開するエージェント インスタンスのリストを表示できます。

注: r12.5 より前のエージェントについては、IP アドレスがホスト名として表示されます。

エージェント インスタンスの表示

企業内に展開されたエージェントの数を知るためにエージェント インスタンスをリスト表示できます。また、提供する検索条件に基づいてエージェント インスタンスのリストを表示することもできます。エージェント インスタンスの任意の属性を検索条件にすることができます。

リストから、エージェント インスタンスに関連する詳細をすべて表示し、1 つ以上のエージェント インスタンス エントリを削除できます。r12.5 より古いエージェントを削除すると、エージェント属性はトラステッドホスト以外、いっさいリストに表示されません。削除したエージェントを再起動すると、すべての属性がリストに再度表示されます。

注: エージェント インスタンス エントリを削除しても、それがポリシーサーバと定期的に通信する限り、エージェント インスタンスはアクティブなままです。

エージェント インスタンスの属性を並べ替えて、選択した属性を使用するエージェント インスタンスのリストを表示することができます。

以下の手順に従います。

1. [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. [エージェント インスタンス] をクリックします。
3. [詳細の表示] の下のアイコンをクリックします。

注: r12.5 より古いエージェントについては、設定モードは [FIPS のみ] または [Compact] のいずれかです。管理 UI は、サーバエージェント間の通信で使用されている実際のモードに基づいて詳細を表示します。

4. (オプション) [検索] フィールドで検索条件を指定します。

エージェント インスタンス リストは検索条件に基づいてフィルタされます。

5. (オプション) 属性のテーブル見出し内の矢印マークをクリックすると、その属性に基づいてエージェント インスタンス リストを並べ替えることができます。

ポリシー サーバ ハートビート間隔の設定

ハートビート間隔は、エージェント ディスカバリがエージェントの状態を特定するのに便利です。エージェントが 24 時間以上ハートビート メッセージを送信しないと、エージェント インスタンスの状態は非アクティブになります。ユーザは、ハートビート間隔をデフォルト値 (24 時間) から変更できます。

以下の手順に従います。

1. ポリシー サーバをホストしているマシンでコマンドプロンプトを開きます。
2. 以下のコマンドを入力します。
`XPSConfig`
3. 「SM」と入力します。
4. `MinTimeBetweenAgentStatusUpdates` オプションに対する値を入力します。
5. 「C」と入力します。
6. 新しい値を入力します。
デフォルト：24 時間
最小：1 時間
7. 「Q」を 3 回入力します。

エージェント ディスカバリの無効化

一部の環境では、エージェント ディスカバリはデータベース マルチマスタ レプリケーションフローにおいて役立ちます。必要に応じて、エージェント ディスカバリを無効にできます。エージェント ディスカバリを無効にするには、XPSConfig ツールを使用して、グローバル設定パラメータの **AgentDiscoveryEnabled** に値を割り当てます。このパラメータは以下の値をサポートします。

0= 無効

1= 自動検出

2= 有効

デフォルト値は **1** の自動検出です。この場合、ポリシー ストアはすべてのエージェント インスタンス オブジェクトについて検索されます。オブジェクトが見つからない場合は値が **0** に更新され、エージェント インスタンス オブジェクトが見つかった場合は **2** に更新されます。エージェント ディスカバリが無効な場合は、トレース メッセージがポリシー サーバのトレース ログに表示されます。このパラメータの値を変更する場合は、変更を有効にするために必ずポリシー サーバを再起動してください。

第 6 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 \(P. 158\)](#)

[CA Directory ユーザ ディレクトリ接続を設定する方法 \(P. 162\)](#)

[z/OS ユーザ ディレクトリ接続向けの CA LDAP サーバを設定する方法 \(P. 165\)](#)

[Oracle Directory Server Enterprise Edition ユーザ ディレクトリ接続を設定する方法 \(P. 179\)](#)

[IBM Directory Server ユーザ ディレクトリ接続を設定する方法 \(P. 180\)](#)

[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 \(P. 182\)](#)

[Novell eDirectory LDAP ディレクトリ接続を設定する方法 \(P. 184\)](#)

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

[Active Directory ユーザ ストア接続を設定する方法 \(P. 191\)](#)

[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 \(P. 203\)](#)

[Oracle Internet Directory ユーザ ディレクトリ接続を設定する方法 \(P. 205\)](#)

[OpenLDAP Server ユーザ ディレクトリ接続を設定する方法 \(P. 207\)](#)

[Red Hat Directory Server ユーザ ディレクトリ接続を設定する方法 \(P. 208\)](#)

[ODBC ユーザ ディレクトリ接続を設定する方法 \(P. 210\)](#)

[カスタム ユーザ ディレクトリ接続を設定する方法 \(P. 216\)](#)

[SSL を使った LDAP ユーザ ディレクトリ接続を設定する方法 \(P. 218\)](#)

[SSL を介した Oracle ユーザ ディレクトリ接続の設定 \(P. 227\)](#)

[LDAP ロード バランシングおよびフェイルオーバー \(P. 230\)](#)

[ODBC データ ソース フェイルオーバーの設定 \(P. 235\)](#)

[SQL クエリ方式 \(P. 235\)](#)

[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 \(P. 243\)](#)

[ユーザ ディレクトリの内容の表示 \(P. 244\)](#)

[ユーザ ディレクトリの検索 \(P. 244\)](#)

[ユニバーサル ID \(P. 245\)](#)

[名前付き式 \(P. 246\)](#)

[ユーザ属性マッピング \(P. 262\)](#)

ユーザ ディレクトリ接続の概要

ユーザ識別情報を確認するため、ポリシー ストアはユーザ ディレクトリを使用して、組織情報、ユーザ属性、およびユーザ認証情報を格納します。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します。これらのディレクトリ接続により、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が解決されます。

ポリシー サーバは、以下のタイプのユーザ ディレクトリをサポートします。

- LDAP
- ODBC
- Oracle
- カスタム

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。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問い合わせます。
- 2 つ目の接続は、検索および更新用です。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、認証情報のテスト用です。ポリシー サーバは、提供された認証情報を使用して、ユーザストアにバインドを試みます。バインド試行の結果により、認証情報が承諾されるかどうかが決まります。

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。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

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、以下の手順に従います。

- 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、ユーザ ディレクトリ ウィンドウからアクセスできます。このペインのフィールドにより、ポリシー サーバと ODBC 互換データベース間で情報がマップされます。

注: クエリ方式ごとに一意のデータ ソースを作成します。

- 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。ODBC データソースはシステム 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。Active Directory (AD) および LDAP ネームスペースの設定は、いくつかの機能面の違いにおいて類似しています。

Active Directory ユーザストアを設定する場合に AD 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。

- 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の両方は、信頼できるものとして確立されます。

- 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

Active Directory ユーザストアを設定する場合に AD ネームスペースを使用すると、以下のようなデメリットがあります。

- 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: ポリシー サーバが UNIX オペレーティング システムにインストールされる場合、AD ユーザ ストアに接続するために AD ネームスペースを使用できません。

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 を使用して共有ライブラリを作成することにより、カスタム ユーザ ディレクトリ接続を作成できます。この C 言語 API は、別売のソフトウェア開発キットに付属しています。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。[ユーザ ディレクトリ] ペイン上でカスタム ネームスペースを設定します。

ディレクトリ属性の概要

一部の機能には、7 つの SiteMinder 属性に対する読み取りまたは書き込みアクセス権限が必要になります。属性値は、ポリシー サーバに接続されているユーザ ディレクトリにあります。[ユーザ属性] セクションで、ディレクトリ ユーザ属性を SiteMinder 属性にマップします。また、属性マッピングを使用して、独自の共通名を定義することもできます。それぞれの名前を、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。
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。

属性名	データタイプ	ディレクトリタイプ	説明
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」機能で使用され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: 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。これらの認証情報には、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あります。

詳細情報:

[ユーザ属性マッピング](#) (P. 262)

CA Directory ユーザ 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 CA Directory ユーザ ストア接続を作成する手順を示します。

1. CA Directory ユーザ ディレクトリ接続を設定します。
2. ユーザ ストア DSA パラメータを有効にします。
3. (オプション) CA Directory ユーザ ストアのキャッシュを有効にします。
4. (オプション) CA Directory キャッシュ設定を確認します。

ユーザ ストア システムに Ping を発行します。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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。

以下の手順に従います。

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] を選択します。
3. [ユーザ ディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
6. [管理者認証情報] 領域で [認証情報が必要] を選択します。
7. 管理者アカウントの認証情報を入力します。
8. (オプション) [ユーザ属性] 領域で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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
10. [サブミット] をクリックします。

ユーザ ディレクトリ接続が作成されます。

ユーザ ストア DSA パラメータの有効化

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。

以下の手順に従います。

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、以下のエントリを定義します。

```
#SiteMinder
set mimic-netscape-for-siteminder = true;
set concurrent-bind-user = DN;
set hold-ldap-connections = true;
```

3. .dxi 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、大規模なユーザストアの認証および許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。

以下の手順に従います。

1. dsa ユーザとして、DXI ファイル（たとえば <dxserver_install>%config%servers%eTrustDsa.dxi）の末尾に以下の行を追加します。

■ r12

```
# 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、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。

■ r12 SP1 以降

```
# cache configuration
set ignore-name-bindings = true;
```

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。

以下の手順に従います。

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。

例：

`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。

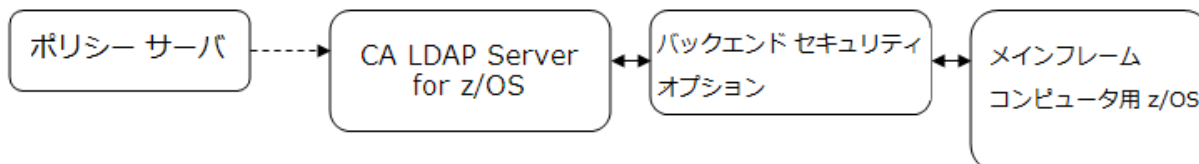
z/OS ユーザ ディレクトリ接続向けの CA LDAP サーバを設定する方法

以下に、ポリシー サーバへの CA LDAP Server for z/OS ユーザ ストア接続を設定する手順を示します。

1. 以下の情報を見直します。
 - CA LDAP Server for z/OS の概要。
 - CA LDAP Server for z/OS でサポートされていない SiteMinder 機能。
 - CA Top Secret r12 (TSS) バックエンドセキュリティ オプションおよびそのオブジェクト クラス階層。
 - CA LDAP Server r15 for z/OS (ACF2) バックエンドセキュリティ オプションおよびそのオブジェクト クラス階層。
 - CA LDAP Server r15 for z/OS (RACF) バックエンドセキュリティ オプションおよびそのネームスペース階層。
2. TSS、ACF2 または RACF 用のポリシー サーバ レジストリ エントリを設定します。
3. ユーザ ディレクトリ接続を設定します。

CA LDAP Server for z/OS の概要

ポリシー サーバから LDAP サーバへの接続を設定することにより、CA LDAP Server for z/OS をユーザ ストアとして設定できます。ポリシー サーバから LDAP サーバへの接続を設定する方法は、LDAP サーバを保護するために使用しているバックエンド オプションによって異なります。



CA は、CA LDAP サーバに対して以下のバックエンド セキュリティ オプションをサポートします。

- CA Top Secret r12 (TSS)
- CA LDAP Server r15 for z/OS (RACF)
- CA LDAP Server r15 for z/OS (ACF2)

ポリシー サーバから LDAP サーバに接続を設定する前に、これらのバックエンド セキュリティ オプションに関するオブジェクト クラス階層について理解します。また、LDAP ネームスペースでポリシー サーバ レジストリにバックエンド関連オブジェクト クラスを追加します。

注: z/OS はメインフレーム コンピュータ用の IBM オペレーティング システムです。

CA LDAP Server for z/OS (TSS) でサポートされていない SiteMinder 機能

CA LDAP Server for z/OS では、以下の機能がサポートされていません。

パスワード サービス

パスワード サービスはサポートされていません。

匿名バインド

ユーザ ストアとして CA LDAP Server r15 for z/OS を設定する場合、
[ユーザ ディレクトリの作成] ページで [管理者認証情報] の値を指定します。

ユーザ名でサポートされていない文字

ユーザ名では以下の文字がサポートされていません。

- スペース
- 一重引用符
- 左丸かっこ
- 右丸かっこ
- カンマ
- 円記号

ユーザ グループとポリシー

ユーザ グループをポリシーに追加し、そのグループ内のユーザの許可を試行すると失敗します。

負荷分散とフェールオーバー(TSS 向け)

負荷分散とフェールオーバーはサポートされていません。

LDAP フェールオーバーとレプリケーション(RACF および ACF2 向け)

LDAP フェールオーバーとレプリケーションはサポートされていません。

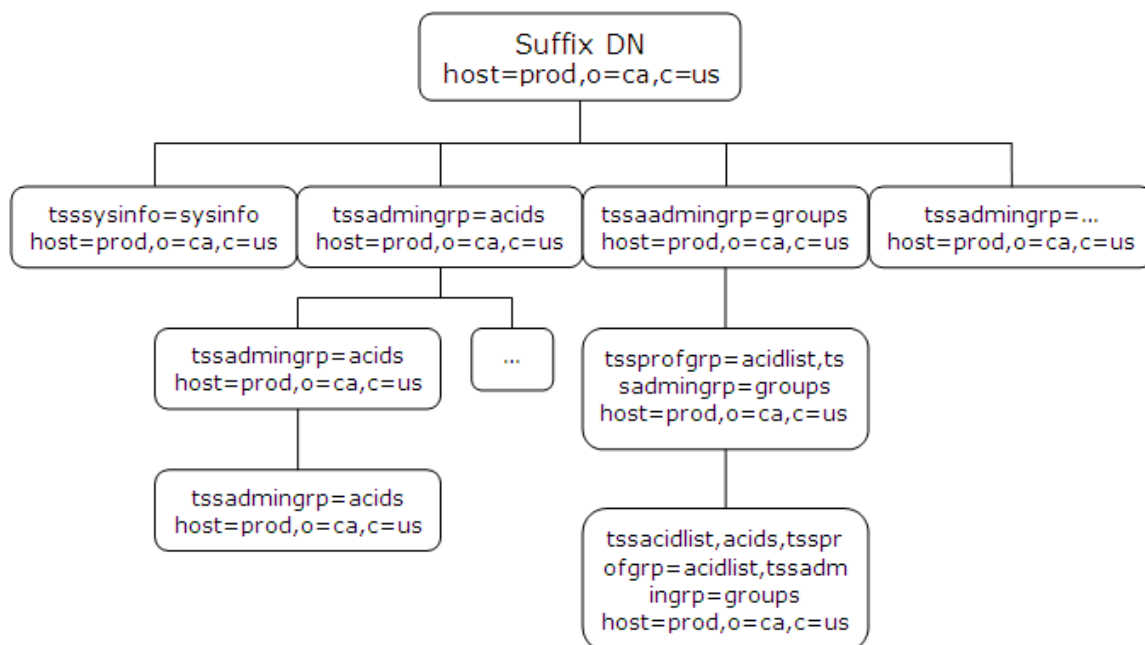
CA Top Secret r12 (TSS)バックエンド セキュリティ オプション

CA LDAP Server for z/OS を保護するために TSS を使用している場合は、ポリシー サーバから CA LDAP サーバへの接続を設定する前に、以下の手順を実行してください。

1. TSS オブジェクト クラス階層について理解します。
2. LDAP ネームスペースでポリシー サーバ レジストリに TSS オブジェクト クラスを追加します。

TSS オブジェクト クラス階層

以下の図は、CA Top Secret ディレクトリ情報ツリー（DIT）内のオブジェクト クラス エントリの階層を示しています。図の下に各オブジェクト クラスの説明があります。



オブジェクト クラス host

CA Top Secret データベースに関するオブジェクト クラス階層へのアクセスを開始するために使用されるオブジェクト クラス。

オブジェクト クラス tsssysinfo

`host` の下のオブジェクト クラス階層にブランチを作成するために使用されるオブジェクト クラス。

オブジェクト クラス tssadmingrp

host の下のオブジェクト クラス階層にブランチを作成するために使用されるオブジェクト クラス。

値 :

- acids
- profiles
- グループ
- departments
- divisions
- zones

オブジェクト クラス tssacid

すべてのユーザ タイプについて ACID レコードフィールドにアクセスするために使用されるオブジェクト クラス。

オブジェクト クラス tssacidgrp

acid の下のオブジェクト クラス階層にブランチを作成するために使用されるオブジェクト クラス。

ポリシー サーバレジストリ エントリの TSS 用の設定

CA LDAP Server for z/OS には他の LDAP サーバとは異なるオブジェクト クラスが含まれています。ポリシー サーバから CA LDAP サーバへの接続を設定する前に、LDAP ネームスペースで特定のポリシー サーバレジストリ エントリに TSS オブジェクト クラスを追加します。以下のポリシー サーバレジストリ エントリのデフォルト値に対する代替値を代入します。

registry_entry_home

以下のレジストリ エントリの場所を指定します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Netegrity¥SiteMinder
¥CurrentVersion¥Ds.

default_value

レジストリ エントリのデフォルト値を指定します。

replacement_value

レジストリ エントリの TSS オブジェクト クラスが含まれる新しい値を指定します。

- registry_entry_home¥ClassFilters
class_filters_default_value:
organization,organizationalUnit,groupOfNames,groupOfUniqueNames,group
class_filters_replacement_value:
class_filters_default_value,eTTSSAcidName,tssacidgrp,tssadmingrp
- registry_entry_home¥GroupClassFilters
group_class_filters_default_value:
groupOfNames,groupOfUniqueNames,group
group_class_filters_replacement_value:
group_class_filters_default_value,eTTSSAcidName,tssacidgrp,tssadmingrp
- registry_entry_home¥PolicyClassFilters
policy_class_filters_default_value:
organizationalPerson,inetOrgPerson,organization,organizationalUnit,
groupOfNames,groupOfUniqueNames,group
policy_class_filters_replacement_value:
policy_class_filters_default_value,eTTSSAcidName,tssacidgrp,tssadmingrp
- registry_entry_home¥PolicyResolution

このレジストリ エントリに以下の TSS オブジェクト クラスを追加します。

TSS オブジェクト クラス	レジストリ キー タイプ	データ
eTTSSAcidName	REG_DWORD	0x00000001(1)
tssacid	REG_DWORD	0x00000001(1)
tssacidgrp	REG_DWORD	0x00000002(2)
tssadmingrp	REG_DWORD	0x00000003(3)

注: ポリシー サーバが発行する LDAP クエリ（ユーザの全リストなど）には、完了までに最大 60 秒かかるものがあります。このような条件下では、ポリシー サーバ側からのクエリの大部分はタイムアウトします。接続性を改善するために、このレジストリ キー エントリを以下のように調節できます。

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\Debug]
LDAPPingTimeout = 300;          REG_DWORD
```

CA LDAP Server for z/OS ユーザ ディレクトリ接続の設定

ポリシー サーバがユーザ ストアと通信するように、ユーザ ディレクトリ 接続を設定します。

注: 負荷分散とフェールオーバーはこの LDAP サーバではサポートされていません。

以下の手順に従います。

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] を選択します。
3. [ユーザ ディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続 情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ 設定] を設定します。

注: [最大時間] には必ず「100」を入力します。ポリシー サーバが、この LDAP サーバからデータを取得するのにこれだけの時間を必要と します。

6. [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. [ユーザ名] に管理者の完全な DN を入力します。
 - c. [パスワード] に管理者パスワードを入力します。

注: TSS ではユーザ ストアへの匿名のバインドが許可されません。

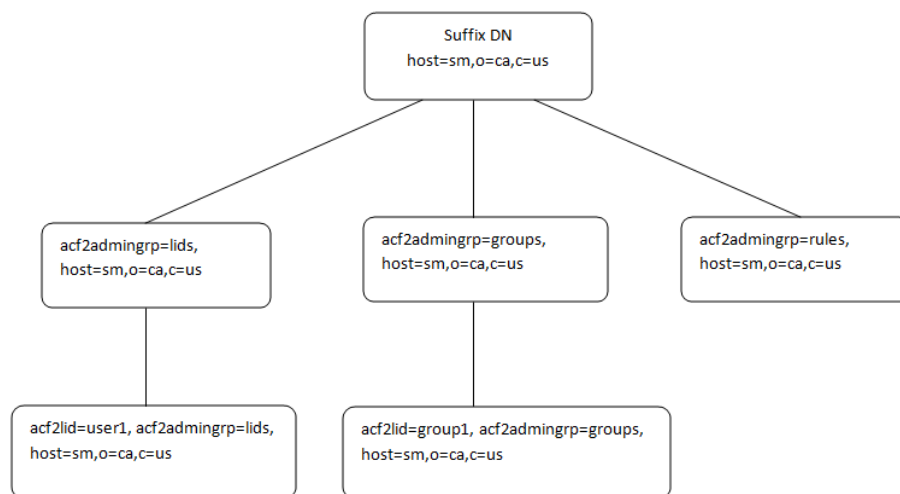
7. (オプション) [ユーザ属性] 領域で **SiteMinder** 用に予約されているユーザディレクトリ プロファイル属性を指定します。
8. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。
ユーザディレクトリ接続が作成されます。

CA LDAP Server r15 for z/OS (ACF2) バックエンド セキュリティ オプション

このセクションでは、ポリシー サーバでユーザストアとして **CA LDAP Server r15 for z/OS (ACF2)** を設定するのに必要な設定について説明します。

ACF2 オブジェクト クラス階層

以下の図は、**CA ACF2** ディレクトリ情報ツリー (DIT) 内のオブジェクトクラス エントリの階層を示しています。この図では、各オブジェクトクラスについて説明しています。



オブジェクト クラス host

CA ACF2 データベースに関するオブジェクトクラス階層へのアクセスを開始するために使用されるオブジェクトクラス。

オブジェクト クラス acf2admingrp

host の下のオブジェクト クラス階層にブランチを作成するために使用されるオブジェクト クラス。

値:

- lids
- グループ
- rules

オブジェクト クラス acf2lid

LID レコードフィールド (ユーザ レコード) にアクセスするために使用されるオブジェクト クラス。acf2lid は、追加、変更、削除が可能な唯一のオブジェクト クラスです。他のオブジェクト クラス オブジェクトはすべて読み取り専用です。

オブジェクト クラス acf2lidgrp

CA ACF2 内のグループをエミュレートするために使用されるオブジェクト クラス。

オブジェクト クラス acf2ruletype

ルール タイプのようにグループ化するために使用されるオブジェクト クラス。acf2ruletype オブジェクト クラスは読み取り専用なので、変更、追加、削除はできません。

ポリシー サーバレジストリ エントリの ACF2 用の設定

CA LDAP Server r15 for z/OS (ACF2) には、他の LDAP サーバとは異なるオブジェクト クラスのセットが含まれます。ポリシー サーバから CA LDAP サーバへのユーザ ディレクトリ接続を設定する前に、LDAP ネームスペースで特定のポリシー サーバ レジストリ エントリに ACF2 オブジェクト クラスを追加します。以下のポリシー サーバ レジストリ エントリのデフォルト値に対する代替値を代入します。

registry_entry_home

以下のレジストリ エントリの場所を指定します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Netegrity¥SiteMinder
¥CurrentVersion¥Ds.

default_value

レジストリ エントリのデフォルト値を指定します。

replacement_value

レジストリ エントリの ACF2 オブジェクト クラスが含まれる新しい値を指定します。

■ registry_entry_home¥ClassFilters

class_filters_default_value:

organization,organizationalUnit,groupOfNames,groupOfUniqueNames,group

class_filters_replacement_value:

class_filters_default_value,*

■ registry_entry_home¥GroupClassFilters

group_class_filters_default_value:

groupOfNames,groupOfUniqueNames,group

group_class_filters_replacement_value:

group_class_filters_default_value,*

■ registry_entry_home¥PolicyClassFilters

policy_class_filters_default_value:

organizationalPerson,inetOrgPerson,organization,organizationalUnit,groupOfNames,groupOfUniqueNames,group

policy_class_filters_replacement_value:

policy_class_filters_default_value,*

■ registry_entry_home¥PolicyResolution

このレジストリ エントリに以下の ACF2 オブジェクト クラスを追加します。

ACF2 オブジェクト クラス	レジストリ キー タイプ	データ
acf2lid	REG_DWORD	0x00000001(1)
acf2admingrp	REG_DWORD	0x00000002(2)
eTACFLidName	REG_DWORD	0x00000001(1)

- registry_entry_home¥Debug

UNIX で、このレジストリ エントリに以下の ACF2 オブジェクト クラスを追加します。

ACF2 オブジェクト クラス	レジストリ キー タイプ	データ
LDAPPingTimeout=	REG_DWORD	300;

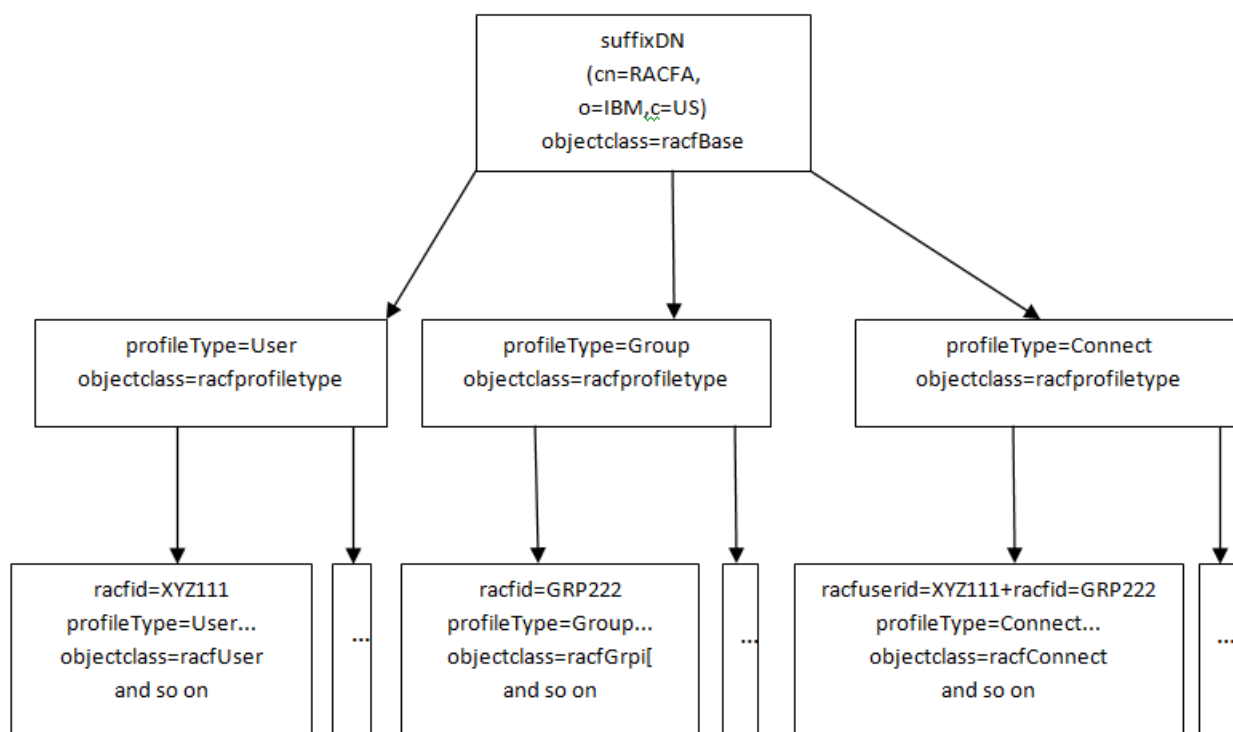
注: このレジストリ キーの値は、CA LDAP Server r15 for z/OS (ACF2) のレスポンス時間に基づいて変更できます。

CA LDAP Server r15 for z/OS (RACF) バックエンド セキュリティ オプション

このセクションでは、ポリシー サーバでユーザ ストアとして CA LDAP Server r15 for z/OS (RACF) を設定するのに必要な設定について説明します。

RACF ネームスペース階層

以下の図は、RACF ディレクトリ情報ツリー (DIT) 内のネームスペース エントリの階層を示しています。この図では、各ネームスペースについて説明しています。RACF ネームスペース階層は ACF2 オブジェクト クラス階層に似ています。



ACF2 サーバに似ているので、階層内の上位 4 つのエントリはサーバによって予約済みで、読み取り専用で、生成されます。これらの予約済みエントリは、RACF ユーザ、グループおよび接続を階層的に表すためのものです。

ポリシー サーバレジストリ エントリの RACF 用の設定

CA LDAP Server r15 for z/OS (RACF) には、他の LDAP サーバとは異なるオブジェクトクラスのセットが含まれます。ポリシー サーバから CA LDAP サーバへのユーザ ディレクトリ接続を設定する前に、LDAP ネームスペースで特定のポリシー サーバ レジストリ エントリに RACF オブジェクトクラスを追加します。以下のポリシー サーバ レジストリ エントリのデフォルト値に対する代替値を代入します。

registry_entry_home

以下のレジストリ エントリの場所を指定します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥Ds.

default_value

レジストリ エントリのデフォルト値を指定します。

replacement_value

レジストリ エントリの RACF オブジェクトクラスが含まれる新しい値を指定します。

■ registry_entry_home¥ClassFilters

class_filters_default_value:

organization,organizationalUnit,groupOfNames,groupOfUniqueNames,group

class_filters_replacement_value:

class_filters_default_value,*

■ registry_entry_home¥GroupClassFilters

group_class_filters_default_value:

groupOfNames,groupOfUniqueNames,group

group_class_filters_replacement_value:

group_class_filters_default_value,*

- registry_entry_home¥PolicyClassFilters
policy_class_filters_default_value:
organizationalPerson,inetOrgPerson,organization,organizationalUnit,group
OfNames,groupOfUniqueNames,group
policy_class_filters_replacement_value:
policy_class_filters_default_value,*
- registry_entry_home¥PolicyResolution
このレジストリ エントリに以下の RACF オブジェクト クラスを追加します。

RACF オブジェクト クラス	レジストリ キー タイプ	データ
eTRACUserid	REG_DWORD	0x00000001(1)
eTRACAdminGrp	REG_DWORD	0x00000002(2)

- registry_entry_home¥Debug
UNIX で、このレジストリ エントリに以下の RACF オブジェクト クラスを追加します。

RACF オブジェクト クラス	レジストリ キー タイプ	データ
LDAPPingTimeout=	REG_DWORD	300;

注: このレジストリ キーの値は、CA LDAP Server r15 for z/OS (RACF) のレスポンス時間に基づいて変更できます。

ポリシー サーバから CA LDAP Server for z/OS への接続の設定

ポリシー サーバから CA LDAP Server for z/OS (RACF) または CA LDAP Server for z/OS (ACF2) へのディレクトリ接続を設定するには、管理 UI 内の既存 ユーザ ディレクトリ オブジェクトを開きます。

以下の手順に従います。

1. [ユーザ ディレクトリ] ダイアログ ボックスが表示されます。
2. [ディレクトリのセットアップ] で、ネームスペースとして LDAP を選択します。

3. ご使用の LDAP ディレクトリの接続情報を入力します。

注: フェールオーバーは、この LDAP サーバではサポートされていません。

4. [LDAP 検索] セクションの [最大時間] フィールドで、値を 300 秒に指定します。

注: ポリシー サーバがこの LDAP サーバからデータを取得するのに時間がかかるため、タイムアウトの時間は長く設定する必要があります。

5. [証明と接続] で、ポリシー サーバがこの LDAP サーバに接続するために使用する管理者認証情報を指定します。

重要: 必ず管理者認証情報を指定する必要があります。それは、CA LDAP Server r15 for z/OS (RACF) および CA LDAP Server r15 for z/OS (ACF2) でのユーザストアへの匿名バインドが許可されていないためです。

Oracle Directory Server Enterprise Edition ユーザ ディレクトリ接続を設定する方法

以下のプロセスに、Oracle Directory Server Enterprise Edition ユーザストア接続を作成する手順を示します。

1. ユーザストアシステムに ping を発行します。
2. Oracle Directory Server Enterprise Edition 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。

ポリシーサーバとユーザディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザストアシステムに ping を必ず発行してください。

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。

Oracle Directory Server Enterprise Edition ユーザ ディレクトリ接続の設定

ポリシーサーバが Oracle Directory ユーザストアと通信できるように、接続を設定します。

以下の手順に従います。

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] を選択します。
3. [ユーザ ディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
6. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] を選択します。
 - b. 管理者アカウントの認証情報を入力します。

注: Oracle 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 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。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ユーザを作成して、[ユーザ名] にそのユーザを指定します。

7. (オプション) [ユーザ属性] 領域で 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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。

ユーザ ディレクトリ接続が作成されます。

IBM Directory Server ユーザ ディレクトリ接続を設定する方法

以下に、ポリシー サーバへの IBM Directory Server ユーザ ストア接続を作成する手順を示します。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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。

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 ストアと通信できるように、接続を設定します。

以下の手順に従います。

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] を選択します。
3. [ユーザディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
6. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [ユーザ属性] 領域で 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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。

ユーザディレクトリ接続が作成されます。

注: IBM Directory Server リフェラルは、SiteMinder と互換性がありません。

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、2 段階のプロセスです。

1. Domino 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. 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 ストアとして設定する前に、それが以下の前提条件を満たしていることを確認してください。

- 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 を共有します。

例：Lotus Notes においてアドレス帳（names.nsf）にグループ *marketing/myorg.org* を追加する場合は、[ユーザ ディレクトリ] 画面の [ルート] フィールドに「*o=myorg.org*」と入力します。

- 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しています。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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 ストアと通信できるように、ユーザ ディレクトリ接続を設定します。

以下の手順に従います。

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] を選択します。
3. [ユーザ ディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。

注: [ルート] に指定する値は、Lotus Notes で割り当てた組織名と一致します。

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 形式に従います。

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ールオーバーを設定します。
7. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
8. (オプション) [ユーザ属性] 領域で 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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
10. [サブミット] をクリックします。

ユーザ ディレクトリ接続が作成されます。

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. Novell eDirectory LDAP ディレクトリ接続を設定します。

NetWare の設定

この設定により、ポリシーサーバは 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できます。一部の拡張機能では、ポリシーサーバの書き込みアクセスを可能にするために Novell eDirectory を設定することが必要になる場合があります。

LDAP が Novell eDirectory のインストールに含まれている場合は、Novell eDirectory のサーバに「LDAP Server」という名前を付け、LDAP グループに「LDAP Group」という名前を付けます。LDAP Server は LDAP Group のメンバーです。

以下の手順に従います。

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。

詳細情報：

[ディレクトリ属性の概要](#) (P. 160)

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 と通信するためには、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します。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にする方法が最も簡単です。ユーザには、すべての機能を実行するのに十分な権限が割り当てられます。

以下では、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。LDAP ディレクトリに対するすべての匿名アクセスには、SiteMinder に付与するものと同じ権限が付与されます。

<fsp>

1. LDAP_Anonymous という名前のユーザを作成します。

以下の手順に例を示します。

- 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。

以下の手順に例を示します。

- 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. [続行] 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、管理 UI での「[Novell eDirectory LDAP ディレクトリ接続の設定 \(P. 188\)](#)」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、これらの手順の方が適切な場合があります。

1. SiteMinder 管理者を表す Novell eDirectory ユーザを作成します。たとえば、siteminder_admin などです。
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、ポリシー管理者のユーザ アカウントを作成できます。

以下の手順に従います。

1. NW 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。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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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。

以下の手順に従います。

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] を選択します。
3. [ユーザ ディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。

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. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [ユーザ属性] 領域で **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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。
ユーザ ディレクトリ接続が作成されます。

Active Directory LDS ユーザ ディレクトリ接続を設定する方法

Active Directory LDS をユーザ ストアとして使用することができます。以下の手順に従います。

1. ユーザ ストア システムに **ping** を発行します。
2. 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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに **ping** を必ず発行してください。

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 LDS 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 LDS** 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。

以下の手順に従います。

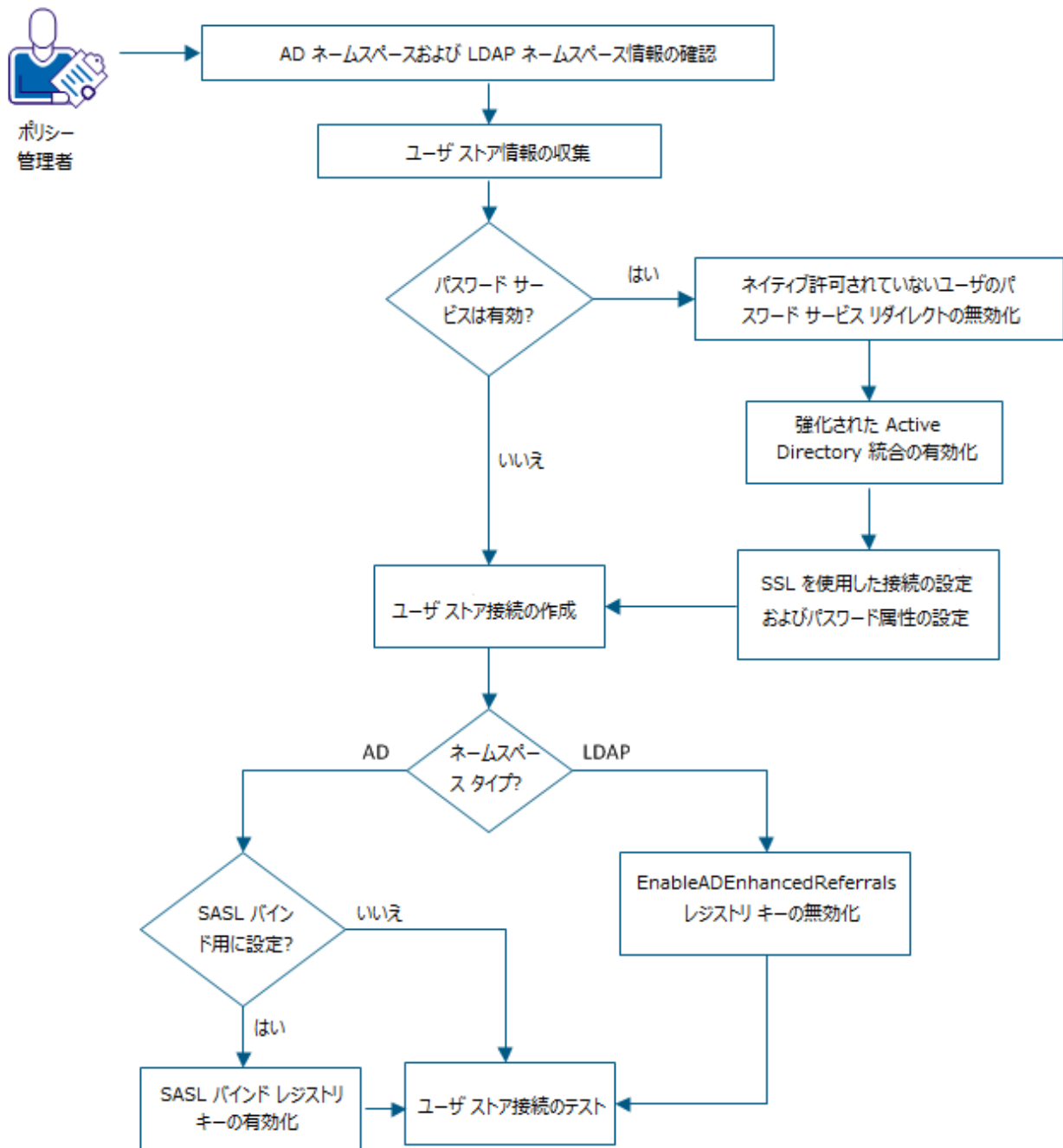
1. [インフラストラクチャ] - [ディレクトリ] をクリックします。
2. [ユーザ ディレクトリ] を選択します。

3. [ユーザディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
6. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [ユーザ属性] 領域で **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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。
ユーザディレクトリ接続が作成されます。

Active Directory ユーザ ストア接続を設定する方法

以下の図に、Active Directory ユーザ ストア接続を設定するための必要な手順を示します。

Active Directory ユーザ ストア接続の設定方法



AD ユーザ ストア接続を設定するには、以下の手順に従います。

1. AD ネームスペースおよび LDAP ネームスペースの情報の確認。
2. [ユーザ ストア情報の収集](#) (P. 195)。
3. [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](#) (P. 197)。
4. [拡張された Active Directory 統合の有効化](#) (P. 198)。
5. [ユーザ ストア接続の作成](#) (P. 199)。
6. [EnableADEnhancedReferrals レジストリ キーの無効化](#) (P. 201)。
7. [SASL バインド レジストリ キーの有効化](#) (P. 202)。
8. [ユーザ ストア接続のテスト](#) (P. 203)。

前提条件

AD ユーザ ストア接続を設定する前に、以下が揃っていることを確認します。

- 管理 UI がインストールされ、ポリシー サーバに登録されている。
- ユーザ ストア接続を作成するために必要な権限を持った管理者アカウント。

AD ネームスペースおよび LDAP ネームスペースの情報の確認

AD ネームスペースまたは LDAP ネームスペースとして、ユーザ ストア接続を設定できます。選択するネームスペースのタイプは、サポートされる機能および SiteMinder 機能のその他の領域に影響します。

以下の表では、AD ネームスペースと LDAP ネームスペースの利点および欠点について説明します。

ネームスペースタイプ	利点	欠点
AD	■ 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。 ■ 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	■ 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: 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。

ネームスペースタイプ	利点	欠点
LDAP	■ 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	■ ネイティブ Windows SASL はサポートされない。 ■ オブジェクトクラス属性はインデックス付けされない。 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 のオブジェクトクラス属性をインデックス付けします。 注: 詳細については、ベンダー固有のドキュメント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。 エージェントは、Windows ユーザセキュリティ コンテキストで稼働でき、IIS Web サーバ上の Web リソースへのアクセスを可能にします。SiteMinder で Windows ユーザセキュリティ コンテキストを提供するには、セッションストアを設定し、レルム単位で永続セッションを有効にします。この機能を有効にするには、管理 UI を使用してユーザディレクトリを作成する間に、[ディレクトリのセットアップ] で以下のオプションを選択します。 認証済みユーザのセキュリティ コンテキストを使用

ユーザ ストア情報の収集

ユーザ ストア接続を作成する前に、LDAP 設定およびユーザ属性に関する必要な情報を収集します。

ディレクトリ サーバ管理者に問い合わせ、この情報を収集してください。

サーバ

Active Directory ホスト システムの IP アドレスおよびポートを指定します。

LDAP 検索ルート

ポリシー サーバがディレクトリ接続の開始ポイントとして使用する LDAP ツリー内の場所を指定します。ポリシー サーバは、ルートからユーザの検索を開始します。

例：dc=domainname、dc=com

ユーザ DN の検索

LDAP ユーザ ストアでユーザを特定するための LDAP 検索式またはユーザ DN のテキスト文字列を指定します。完全な検索には、先頭文字列と終端文字列が必要です。[先頭] フィールドの文字列、ユーザ名、および[終端] フィールドの文字列の組み合わせを使用して、LDAP ユーザ ストアが検索されます。

例（先頭）：(sAMAccountName=

例（終端）：)

ユニバーサル ID

SiteMinder がユニバーサル ID として使用する属性の名前を指定します。

例：sAMAccountName

無効フラグ

ユーザの無効状態が格納されるユーザ ディレクトリ 属性の名前を指定します。

例：carLicense（または任意の整数属性）

パスワード

SiteMinder がユーザのパスワードの認証に使用するユーザ ディレクトリ 属性の名前を指定します。

例：unicodePwd

パスワード データ

SiteMinder がパスワード サービス用のデータとして使用できるユーザディレクトリ属性の名前を指定します。

例 : audio

[パスワード データ] には任意の大型のバイナリ属性を値として使用できます。値は基本パスワード サービスを使用している場合にのみ必要です。

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、これらのユーザがディレクトリ サーバで元から無効である場合、そのユーザに認証情報の再入力を求めます。**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、これらのユーザをパスワード サービスにリダイレクトします。

この動作を防ぐためには、以下のレジストリ キーが必要です。

`IgnoreDefaultRedirectOnADnativeDisabled`

ポリシー サーバ管理者およびリクエストに問い合わせ、キーを作成し、有効にするように依頼します。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. レジストリ エディタを開き、以下の場所に移動します。

`HKEY_LOCAL_MACHINE\Software\Netegrity\SiteMinder\CurrentVersion\Ds\LdapProvider`

3. レジストリ キー `IgnoreDefaultRedirectOnADnativeDisabled` を作成し、レジストリのタイプは `REG` にします。

値：0（無効）または1（有効）

デフォルト：0

4. 値を1として設定します。
5. レジストリ エディタを終了します。
6. ポリシー サーバを再起動します。

重要: パスワード サービスへのリダイレクトを指定するパスワード ポリシーが有効な場合、**SiteMinder** は、このレジストリ 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 統合の拡張の有効化

Active Directory 2008 には、Windows ネットワーク オペレーティング システム (NOS) に固有なユーザ属性とドメイン属性がいくつかあります。LDAP 標準では、これらのユーザおよびドメインの属性は必要ありません。パスワード サービスが有効な場合は、管理 UI を使用して、拡張された Active Directory 統合を有効にします。このオプションは、AD のユーザ属性と、SiteMinder でマップされるユーザ属性を同期することによって、ポリシー サーバのユーザ管理機能と AD のパスワード サービス間の統合を強化します。

以下の手順に従います。

1. 管理 UI にログインします。
2. [管理] - [ポリシー サーバ] - [グローバル ツール] をクリックします。
3. [Active Directory 統合を拡張] を選択します。
4. [サブミット] をクリックします。

拡張された Active Directory 統合が有効になります。

ユーザ ストア接続の作成

ユーザ ストア接続の設定により、ポリシー サーバは **Active Directory** と通信できます。環境でパスワード サービスが使用されている場合は、**SSL** 接続とパスワード属性（例： **uicodePWD**）が必要です。

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、ベンダー固有のドキュメントを参照してください。

以下の手順に従います。

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] - [ユーザ ディレクトリ] をクリックします。
3. [ユーザ ディレクトリの作成] をクリックします。
4. **Microsoft Active Directory** は **LDAP** に対応したユーザ ディレクトリです。**AD** ネームスペースまたは **LDAP** ネームスペースを使用して、接続を設定できます。以下のいずれかを実行します。
 - デフォルトの **LDAP** 設定のままにします。
 - [ネームスペース] リストから **[AD]** を選択します。
5. [一般] および [ディレクトリのセットアップ] の領域に残りの必要な接続情報を入力します。

注: ポリシー サーバと **Active Directory** ネームスペースが **SSL** を介して通信する場合は、[ディレクトリのセットアップ] の [サーバ] で **IP** アドレスおよびポートを指定してください。 **IP** アドレスが指定されていない場合、ユーザ ディレクトリ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 サーバおよびディレクトリ ストアが使用する認証情報は、以下の条件で **FIPS** 準拠している必要があります。

- ポリシー サーバが **FIPS** モードで動作している場合。
- ポリシー サーバと通信する際に、ディレクトリ接続では安全な **SSL** 接続が使用される場合。

6. (オプション) [ディレクトリのセットアップ] で [設定] をクリックしてロードバランシングおよびフェールオーバーを設定します。
7. [管理者認証情報] で、以下の手順に従います。
 - a. [認証情報が必要] を選択します。
 - b. 管理者アカウントの認証情報を入力します。

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は、[ユーザ名] フィールドで管理者の完全修飾ドメイン名 (FQDN) を指定します。しない場合、ユーザ認証に失敗する場合があります。

8. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
9. [ユーザ属性] 領域で 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。
10. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
11. [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作成されます。

EnableADEnhancedReferrals レジストリ キーの無効化

ユーザ ストア接続が LDAP ネームスペースを使用して設定されている場合は、EnableADEnhancedReferrals レジストリ キーを無効にします。このレジストリ キーを無効にすると、LDAP 接続エラーは発生しません。

ポリシー サーバ管理者およびリクエストに問い合わせ、キーを無効にするように依頼します。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. レジストリ エディタを開き、以下の場所に移動します。

```
HKEY_LOCAL_MACHINE\Software\Netegrity\SiteMinder\CurrentVersion\Ds\LDAPProvider
```

3. EnableADEnhancedReferral キーの値を 0 として設定します。

値：0（無効）または 1（有効）

デフォルト：1

4. レジストリ エディタを終了します。

SASL バインド レジストリ キーの有効化

Windows ベースのポリシー サーバは、SASL を使用して、Active Directory に対してユーザーを認証できます。SASL バインドの使用を有効にするには、EnableSASLBind レジストリ キーを作成し、有効にします。

注: この設定を有効にする場合、完全修飾識別名ではなく、ユーザーディレクトリ設定で管理者名を AD ログイン名に設定します。

重要: ポリシー サーバとユーザー ストア間での SSL 接続を設定する場合は、このレジストリ キーを有効にしないでください。

ポリシー サーバ管理者およびリクエストに問い合わせて、キーを作成し、有効にするように依頼します。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. レジストリ エディタを開き、以下の場所に移動します。

```
HKEY_LOCAL_MACHINE\Software\Netegrity\SiteMinder\CurrentVersion\Ds\LDAPProvider
```

3. レジストリ キー EnableSASLBind を作成し、値のタイプは REG_DWORD にします。
4. 値を 1 として設定します。
5. レジストリ エディタを終了します。
6. ポリシー サーバを再起動します。

ユーザ ストア接続のテスト

ユーザに対してクエリを実行することにより接続をテストします。

以下の手順に従います。

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] - [ユーザ ディレクトリ] をクリックします。
3. 作成したユーザ ストアの名前をクリックします。
4. [ディレクトリのセットアップ] で [内容の表示] をクリックします。
5. [検索] のタイプが [属性/値] として選択されていることを確認します。
6. [属性] にユニバーサル ID を入力します。
例：sAMAccountName
7. [値] に * を入力します。
8. [実行] をクリックします。

アカウント詳細が表示されます。ユーザ ストアに正常に接続しました。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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。

以下の手順に従います。

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] を選択します。
3. [ユーザディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
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ールオーバーを設定します。

注: ロード バランシングとフェー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ールオーバー」を参照してください。

7. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。

8. (オプション) [ユーザ属性] 領域で **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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
10. [サブミット] をクリックします。
ユーザ ディレクトリ接続が作成されます。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに **ping** を必ず発行してください。

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、以下の値を入力します。
 - 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。

以下の手順に従います。

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。

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. (オプション) [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [ユーザ属性] 領域で 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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。

ユーザ ディレクトリ接続が作成されます。

OpenLDAP Server ユーザ ディレクトリ接続を設定する方法

OpenLDAP Server をユーザ ストアとして使用することができます。ユーザディレクトリ接続を作成するには、以下の手順に従います

1. ユーザ ストアを作成します。
2. OpenLDAP Server ユーザ ディレクトリ接続を設定します。

ユーザ ストアを作成

OpenLDAP ディレクトリ サーバをユーザ ストアとして使用することができます。

ユーザ ストアを作成する方法

1. LDIF ファイルを使用して、ルート DN の下に ou=People を作成します。
2. 組織単位の下にユーザを作成します。

OpenLDAP Directory Server ユーザ ディレクトリ接続の設定

ポリシー サーバが **OpenLDAP 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。

以下の手順に従います。

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] を選択します。
 3. [ユーザ ディレクトリの作成] をクリックします。
 4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
 5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
 6. [管理者認証情報] 領域で、以下の手順を実行します。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
 7. (オプション) [ユーザ属性] 領域で **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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
 9. [サブミット] をクリックします。
- ユーザ ディレクトリ接続が作成されます。

Red Hat Directory Server ユーザ ディレクトリ接続を設定する方法

Red Hat Directory Server をユーザ ストアとして使用することができます。ユーザ ディレクトリ接続を作成するには、以下の手順に従います

1. ユーザ ストア システムに **ping** を発行します。
2. 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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに **ping** を必ず発行してください。

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。

Red Hat Directory Server ユーザ ディレクトリ接続 の設定

ポリシー サーバが Red Hat 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。

以下の手順に従います。

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] を選択します。
3. [ユーザディレクトリの作成] をクリックします。
4. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
5. [LDAP 設定] 領域の [LDAP 検索] および [LDAP ユーザ DN ルックアップ設定] を設定します。
6. [管理者認証情報] 領域で以下を行います。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [ユーザ属性] 領域で 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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。
9. [サブミット] をクリックします。

ユーザディレクトリ接続が作成されます。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに **ping** を必ず発行してください。

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 **Microsoft SQL Server** データベースを使用し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、パフォーマンスが低下することがあります。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。

以下の手順に従います。

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。
ユーザ ディレクトリと関係するオブジェクトが、左側に表示されます。
2. [ユーザ ディレクトリ] をクリックします。
[ユーザ ディレクトリ] 画面が表示されます。
3. [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] 画面が表示されて、**LDAP** 接続を設定するために必要な設定が表示されます。

4. [ネームスペース] リストから [ODBC] を選択します。

ODBC 設定が表示されます。

5. [一般] および [ディレクトリのセットアップ] の領域に残り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. SQL クエリ方式を選択します。

7. (オプション) [管理者認証情報] 領域で以下を行います。

- a. [認証情報が必要] を選択します。

- b. 管理者アカウントの認証情報を入力します。

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。

8. (オプション) [ユーザ属性] 領域で **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. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。

10. [サブミット] をクリックします。

ユーザディレクトリ接続が作成されます。

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「ABC」でなければならないように設定されていても、ユーザが「ABC」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/in_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、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。

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。

ポリシー サーバとユーザ ディレクトリまたは、データベースの間にネットワーク接続が存在することを確認するため、設定する前にユーザ ストア システムに ping を必ず発行してください。

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。

以下の手順に従います。

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] をクリックします。
3. [ユーザディレクトリの作成] 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。

4. [ネームスペース] リストから [カスタム] を選択します。
5. [一般] および [ディレクトリのセットアップ] の領域に必要な接続情報を入力します。
6. (オプション) [管理者認証情報] 領域で、以下の手順を実行します。
 - a. [認証情報が必要] オプションを選択します。
 - b. 管理者アカウントの認証情報を入力します。
7. (オプション) [属性マッピングリスト] 領域の [作成] をクリックして、ユーザ属性マッピングを設定します。

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します。

8. [サブミット] をクリックします。

ユーザディレクトリ接続が作成されます。

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 接続を行うには、証明書データベース ファイルを設定する必要があります。

以下の手順に従います。

1. 証明書データベース ファイルを作成します。
2. 証明書データベースに証明機関（CA）を追加します。
3. 証明書データベースにサーバ証明書を追加します。サーバ証明書が認証局によって発行された場合は、各認証局からのルート証明書も証明書データベースに追加されます。
4. 証明書データベース内の証明書を一覧表示します
5. ユーザ ディレクトリの SSL 接続を設定します。
6. 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
7. 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。
- データベース ファイルが Netscape データベース バージョン ファイル 形式（cert8.db）であること。ポリシー サーバは、Mozilla LDAP SDK を使用して、LDAP ディレクトリと通信します。

重要: cert8.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。

- (Active Directory) 以下の点を考慮してください。
 - ユーザ ディレクトリ接続が AD ネームスペースで設定される場合、後続のトピックで説明する SSL 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 証明書リポジトリを使用します。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。
 - 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。
 - Active Directory インスタンスをホストしているコンピュータで、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。

証明書データベース ファイルの作成

証明書データベース ファイルを作成するには、ポリシー サーバに含まれている Mozilla Network Security Services (NSS) certutil アプリケーションを使用します

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する前に、管理者権限でコマンドライン ウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドライン ウィンドウを開きます。

以下の手順に従います。

1. コマンドプロンプトから、インストール bin ディレクトリに移動します。

例 : C:¥Program Files¥CA¥SiteMinder¥bin

注: Windows には固有の certutil ユーティリティがあります。ポリシーサーバ bin ディレクトリから作業していることを確認してください。そうしないと、間違えて Windows certutil ユーティリティを実行する場合があります。

2. 以下のコマンドを入力します。

```
certutil -N -d certificate_database_directory
```

-N

cert8.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*

certutil ツール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。

- cert8.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
```


ルート認証機関の証明書データベースへの追加

ルート認証機関 (CA) を追加するには、Mozilla Network Security Services (NSS) certutil アプリケーションを使用します。これはポリシー サーバにあります。

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する前に、管理者権限でコマンドライン ウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドライン ウィンドウを開きます。

以下の手順に従います。

1. コマンドプロンプトから、ポリシー サーバインストール bin ディレクトリに移動します。

例: C:\Program Files\CA\SiteMinder\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。誤って 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 -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ってください。

-t *trust_arguments*

証明書に適用する信頼属性を指定します。使用可能な 3 つの信頼カテゴリは、「SSL、電子メール、オブジェクト署名」の順番で表記されます。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。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**、**.pem** などがあります。

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。

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

SSL を使用した通信を有効にするには、サーバ証明書を証明書に追加します。Mozilla Network Security Services (NSS) certutil アプリケーションを使用します。これはポリシー サーバで使用できます。

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。

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する前に、管理者権限でコマンドライン ウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドライン ウィンドウを開きます。

以下の手順に従います。

1. コマンドプロンプトから、ポリシー サーバインストール bin ディレクトリに移動します。

例: C:\Program Files\CA\SiteMinder\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。誤って 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 -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`

信頼引数を指定します。各証明書には、3 つの使用可能な信頼カテゴリがあります。これらのカテゴリを表記する順番は、「SSL、電子メール、オブジェクト署名」です。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。

重要: これは SSL 信頼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。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`、`.pem` などがあります。

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んでください。

`-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んでください。

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あることを確認するには、Mozilla Network Security Services (NSS) `certutil` アプリケーションを使用します。ポリシーサーバにはこのツールが含まれています。

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する前に、管理者権限でコマンドライン ウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドライン ウィンドウを開きます。

以下の手順に従います。

1. コマンドプロンプトから、ポリシー サーバインストール bin ディレクトリに移動します。

例: C:\Program Files\CA\SiteMinder\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。誤って 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んでください。

このコマンドによって、証明書を証明書データベースに追加する際に指定した、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 ストアが適切に通信できるように、ユーザ ストアへの SSL 接続を設定します。

以下の手順に従います。

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] をクリックします。
4. 対象のユーザ ディレクトリ接続の名前をクリックします。
5. 下へスクロールし、[変更] をクリックします。
6. [ディレクトリのセットアップ] 領域の [安全な接続] オプションを選択し、[サブミット] をクリックします。

ユーザ ディレクトリ接続では、SSL を介して通信できます。

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

SSL を使用してユーザ ディレクトリと通信するには、証明書データベースを参照するようにポリシー サーバを設定します。

以下の手順に従います。

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 インターフェイスにアクセスする場合は、管理者権限でショートカットを開きます。管理者としてシステムにログインしている場合でも、管理者権限を使用します。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、証明書データベース ファイルへのパスを入力します。

例 : C:\certdatabase\cert8.db

注: key3.db ファイルは、cert8.db ファイルと同じディレクトリ内にある引数があります。

4. ポリシー サーバを再起動します。
ポリシー サーバは、SSL を使用してユーザ ディレクトリと通信できます。

SSL 接続の検証

ユーザ ディレクトリとポリシー サーバが適切に通信しているかどうか判断するには、必ず SSL 接続を確認します。

以下の手順に従います。

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] をクリックします。
4. 対象のユーザ ディレクトリ接続の名前をクリックします。
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] 画面が表示され、ユーザ ディレクトリの内容がリスト表示されます。

SSL を介した Oracle ユーザ ディレクトリ接続の設定

SSL を介して通信するためのポリシー サーバと Oracle データベース間の接続を設定できます。接続を設定する前に、Oracle データベースが SSL に対して有効であることを確認してください。

ポリシー サーバが、SSL を介し Oracle データベースに接続する方法

以下プロセスでは、SSL を介したポリシー サーバと Oracle データベース間の接続がどのように確立されるかについて説明します。

1. ポリシー サーバが接続要求を行うと、Oracle データベース サーバはその公開証明書を提示します。ポリシー サーバは、Oracle データベース サーバが提示した証明書の信頼性を検証するように設定できます。

オプションで、証明書を検証するための設定をポリシー サーバに行わず、SSL を介した Oracle データベースと通信するための設定をポリシー サーバに行うことができます。

注: ポリシー サーバは、トラストストアを使用して証明書の信頼性を検証します。トラストストアは、認証局 (CA) の単一の公開証明書です。または、トラストストアは、信頼された CA からの公開証明書のリストが含まれる PKCS12 トラストストアです。公開証明書はパスワード保護されていません、しかし、PKCS12 トラストストアは暗号化されパスワード保護されています。

2. Oracle 証明書がトラストストア内の証明書と一致すると、暗号化された接続がポリシー サーバと Oracle データベースの間で確立されます。トラストストア内の証明書と一致しない場合、接続は失敗し、ポリシー サーバはエラーを生成します。

Windows での SSL の設定

ODBC データ ソース管理者コンソールを使用して、SSL を介し Oracle と通信するようポリシー サーバを設定できます。

以下の手順に従います。

1. [ODBC データ ソース管理者] コンソールを開きます。
2. [システム DSN] タブで、ご使用の SiteMinder Oracle データベースの DSN を選択します。
3. [構成] をクリックします。

ODBC Oracle Wire Protocol ドライバのセットアップ ダイアログ ボックスが表示されます。

4. [セキュリティ] タブをクリックします。

5. 以下の暗号化パラメータを指定します。

暗号化方法

ポリシー サーバと **Oracle** データベース サーバの間で送信されるデータを暗号化するためにポリシー サーバが使用する暗号化方式を指定します。

デフォルト：0 - 暗号化しない

必須値：1 - SSL オート

サーバ証明書の検証

(オプション) ポリシー サーバが **Oracle** データベース サーバが提示する証明書の信頼性を検証するということを指定します。

デフォルト：オン

ポリシー サーバを使用せずに、**Oracle** データベースが提示する証明書の信頼性を検証するよう **SSL** を設定するには、選択内容をクリアします。

トラストストア

トラストストア ファイルのパス名を定義します。 **Oracle** データベースが提示する証明書の信頼性を検証するには、ポリシー ストアが必要な場合に、以下の値を指定します。

必須値：トラストストアは、CA の公開証明書または 1 つ以上の証明書を含む **PKCS12** トラストストアのいずれかになります。公開証明書は、パスワード保護されていない単一の証明書です。**PKCS12** トラストストアはパスワード保護されています。

トラストストア パスワード

トラストストアにアクセスするために必要なパスワードを定義します。

証明書内のホスト名

証明書内のホスト名を定義します。証明書内のホスト名は、**Oracle** データベース サーバに接続するために使用されるホスト名と一致している必要があります。ホスト名が一致しないと、接続は失敗します。

注：キーストア、キーストアパスワードおよびキーパスワードパラメータは、この接続に適用されません。

6. [OK] をクリックします。

UNIX での SSL の設定

UNIX 上でポリシー サーバ用の SSL を設定し、ポリシー サーバが SSL を介し Oracle と通信できるようにします。

以下の手順に従います。

1. エディタを使用して、`system_odbc.ini` ファイルを編集します。
`system_odbc.ini` ファイルは `/nete_ps_root/db` ディレクトリにあります。
2. SSL を介し接続する Oracle DSN に以下のパラメータを追加します。

`ValidateServerCertificate=0` または `1`

サーバ証明書を検証する場合は、`1` を指定します。サーバ証明書を検証しない場合は、`0` を指定します。

`TrustStore=CA` 証明書または PKCS12 トラスト ストアへのパス

`TrustStorePassword=TrustStorePassword`

`HostNameInCertificate=hostname.domain.com`

例：

`ValidateServerCertificate=1`

`TrustStore=¥nete_ps_root¥db¥MyCAcert.cer` または

`¥nete_ps_root¥db¥MyCertTrustStore.p12`

`TrustStorePassword=abcd`

`HostNameInCertificate=mydbhost.abc.com`

3. `system_odbc.ini` ファイルを保存して閉じます。

LDAP ロード バランシングおよびフェイルオーバー

フェールオーバーおよびロード バランシングを実行するために、ポリシー サーバは複数の LDAP サーバに LDAP クエリを分散できます。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。ロードバランシングを使用すると、よりすばやく効率的に 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。

フェールオーバーの設定

フェールオーバーを設定して冗長性を確保します。両方のディレクトリは同じポート番号を共有するため、フェー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タイプの通信（SSL または SSL 以外）を使用します。

以下の手順に従います。

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。
2. [フェールオーバーの追加] をクリックします。
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: 最後のサーバにポートを指定し、グループ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。

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。

以下の手順に従います。

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。
2. [ロード バランシングの追加] をクリックします。
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

複数のサーバにリクエストを分散し、冗長性を確保するには、ロード バランシングとフェイルオーバーを設定します。

以下の手順に従います。

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。
2. フェイルオーバー用に指定されるポリシー サーバのホスト名とポートを入力します。

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. 他のフェイルオーバー サーバを定義するには、手順 **2** と **3** を繰り返します。

最後のサーバにポートを指定し、他の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他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. ロード バランシング用に指定されるポリシー 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、特定システムへの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. 手順 **5** および **6** を繰り返して、他のロード 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] フィールドには、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がリストされます。フェー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 バランシング用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、ユーザ ディレクトリ B にフェールオーバーし、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、ユーザ ディレクトリ A にフェイルオーバーし、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。

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。このアクションの結果、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には、フェイルオーバーを設定します。

以下の手順に従います。

1. [ディレクトリのセットアップ] セクションで [設定] をクリックします。
2. [追加] をクリックします。
3. フェイルオーバー用として指定され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、他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。

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。[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: 列名のプレフィックスである「SM_」は、特殊名のために予約されています。ユーザディレクトリの列名の先頭には、「SM_」というプレフィックスを使用しないでください。

SQL クエリ方式の設定

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。

以下の手順に従います。

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] をクリックします。
3. [SQL クエリ方式の作成] をクリックします。
4. [オブジェクトの新規作成] オプションが選択されていることを確認し、[OK] をクリックします。
5. [一般] 領域のフィールドに名前および説明を入力します。
6. 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。

リレーショナルデータベースを使用するには、各クエリを設定します。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
ユーザの Name パラメータは一意です。
- SmUser—table
- SmGroup—table
- Password
- SmUserGroup—table
- ID
- UserID
- FirstName
- LastName
- TelephoneNumber
- EmailAddress
- Mileage
- PIN

- GroupID

- Disabled

7. 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。

8. 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。

以下の手順に従います。

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。
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: 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" に置き換えてください。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 プロシージャを使用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、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

Oracle 関数内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。

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。

- ゼロの値は成功を指定します。
- 1 の値は失敗を指定します。

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。

- 1 の値は成功を指定します。
- ゼロの値は失敗を指定します。

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、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。ネットワーク障害に伴う遅延を回避するために、呼び出し元は非同期呼び出しを放棄できます。

以下のデータベースに対して、非同期呼び出しを行うことができます。

- SQLServer
- Windows NT および Solaris 上の Oracle 8 ～ Oracle 11g

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、生じるメモリ リークはわずかです。ネットワークが不安定な場合には、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。

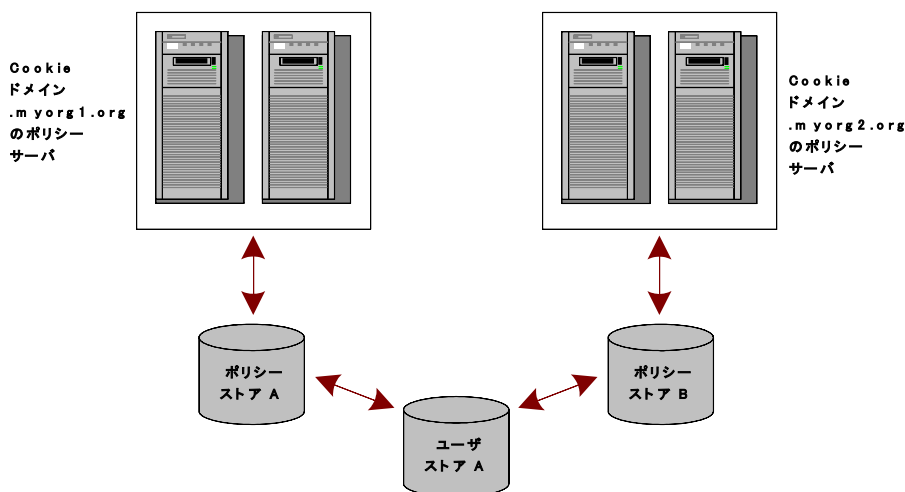
ODBC 接続プーリング

以下の基準は ODBC 接続プーリングに関する説明です。

- ODBC データ ソースは接続をプールします。複数のディレクトリがデータ ソースを使用する場合は、使用可能な接続の数がディレクトリおよびストアの合計に達することがあります。ポリシー ストアはデータ ソースを使用する必要がある、ポリシー サーバの上の別のストアはデータ ソースを使用する必要がある。
- 接続の共有については、SQL Server は Oracle よりも制限を受けます。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。同時アクティビティは制限されます（特にマルチプロセッサ マシンの場合）。接続数を増やせば、一般的に同時実行性が向上します。
- 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。
- 接続を共有すると、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。
- ポリシー サーバは、シャット ダウンするときに開いている接続を閉じます。

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。1つのポリシー ストアを参照するように複数のポリシー サーバを設定できます。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 ストアを含めることができます。



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。

以下の手順に従います。

1. 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. 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。

5. 同じユーザ ストアにアクセスする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できます。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 サインオンを使用できます。

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。

以下の手順に従います。

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。

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。

以下の手順に従います。

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、新規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、基になるユーザ リポジトリの変更を回避できます。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 番号を使用してユーザ情報を検索するレガシー アプリケーションを使用しているとします。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] ペインの [ユーザ属性] セクションで UID を指定できます。

詳細情報:

[CA Directory ユーザ ディレクトリ接続を設定する方法](#) (P. 162)

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、個別の認証情報などのユーザ属性を格納します。一部のユーザ属性値は、ユーザディレクトリから直接読み取られ、他の値は必要になるたびに計算する必要があります。これらの計算式は名前を付けて、または名前を付けないで保存できます。

名前付き式は、名前で参照するポリシーストアオブジェクトです。これらは、アプリケーションオブジェクトで定義されたセキュリティポリシーで再利用できます。名前のない式は、レスポンスなどのドメインオブジェクトに格納されます。これらは、従来のセキュリティポリシーで使用されるルールです。

注: 名前付き式は、アプリケーションオブジェクトでのみ使用できます。レスポンスやルールなどのドメインオブジェクトを使用して定義された従来のセキュリティポリシーでは、名前付き式は使用できません。

名前付きと名前なしの両方のすべての式を使用して、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

名前付き式には、以下のメリットがあります。

- 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。
- 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。
- 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。
- 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。
- 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、名前によって参照でき、アプリケーションオブジェクトで定義されたセキュリティポリシーで再利用できるポリシーストアオブジェクトです。

注: 名前付き式は、アプリケーションオブジェクトでのみ使用できます。レスポンスやルールなどのドメインオブジェクトを使用して定義された従来のセキュリティポリシーでは、名前付き式は使用できません。

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. 248)
- [ユーザクラス](#) (P. 251)

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。

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)
- ほかの名前付き式への参照
- 組み込み関数および式構文

注: 名前付き式は、アプリケーション オブジェクトでのみ使用できます。レスポンスやルールなどのドメイン オブジェクトを使用して定義された従来のセキュリティ ポリシーでは、名前付き式は使用できません。

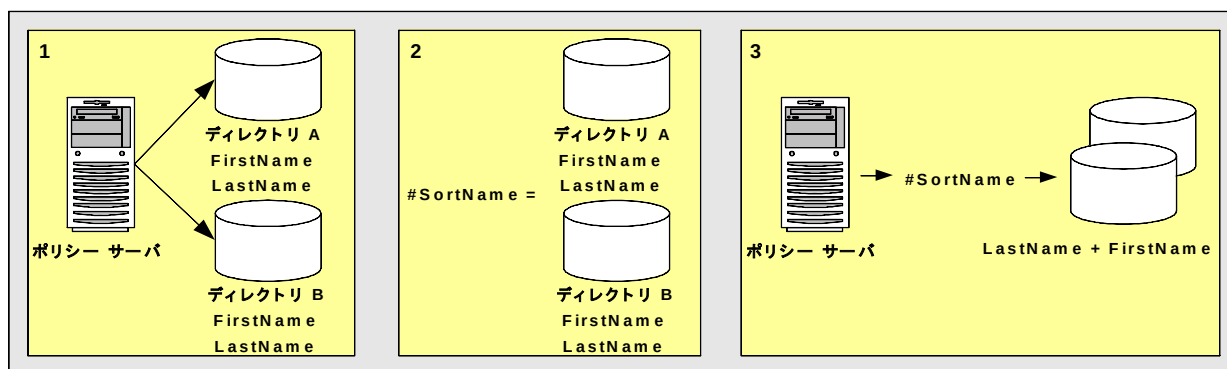
詳細情報:

[式の構文の概要](#) (P. 934)

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が作成されていることを前提とします。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。

以下の手順に従います。

1. [ポリシー] - [式] をクリックします。
2. [名前付き式] をクリックします。
3. [名前付き式の作成] をクリックします。
4. [オブジェクトの新規作成] オプションが選択されていることを確認し、[OK] 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. [仮想ユーザ属性] オプションを選択し、[一般] 領域に式の名前を入力します。
6. [名前付き式の追加] 領域の [式] フィールドに、式を入力します。
7. (オプション) [名前付き式の追加] 領域で [無効] オプションを選択し、式を無効にします。無効にした式は式エディタにリスト表示されません。また、別の名前付き式または名前のない式はそれ呼び出すことができません。
8. (オプション) [名前付き式の追加] 領域の [プライベート] オプションを選択します。他の名前付き式のみがプライベート式を呼び出すことができます。名前のない式はプライベート式を呼び出すことができません。
9. (オプション) [名前付き式の追加] 領域の [編集] をクリックし、[式エディタ] を開きます。
10. [サブミット] をクリックします。

名前付き式が作成されます。

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

注: 名前付き式は、アプリケーション オブジェクトでのみ使用できます。レスポンスやルールなどのドメイン オブジェクトを使用して定義された従来のセキュリティ ポリシーでは、名前付き式は使用できません。

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。

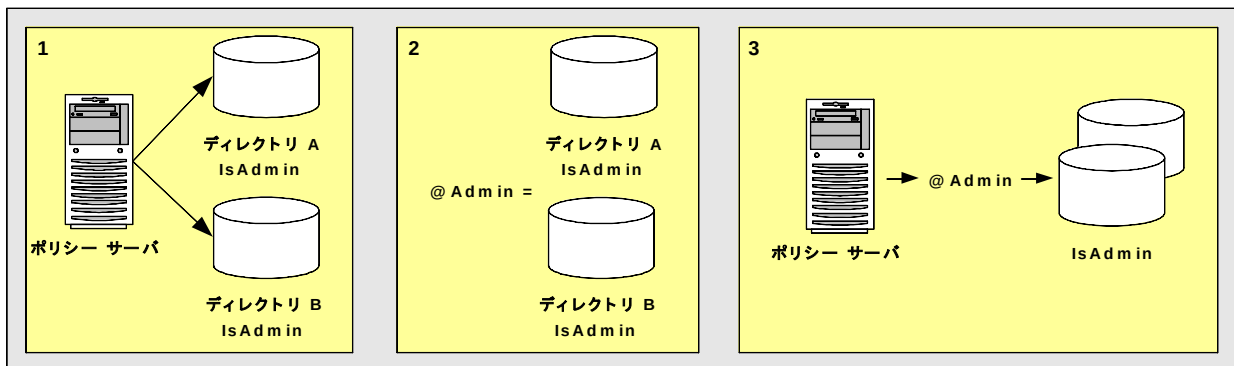
詳細情報:

[式の構文の概要 \(P. 934\)](#)

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が作成されていることを前提とします。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。

以下の手順に従います。

1. [ポリシー] - [式] をクリックします。
2. [名前付き式] をクリックします。
3. [名前付き式の作成] をクリックします。
4. [オブジェクトの新規作成] オプションが選択されていることを確認し、[OK] をクリックします。
5. [ユーザ クラス] オプションを選択し、[一般] 領域に式の名前を入力します。
6. [名前付き式の追加] 領域の [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。

7. (オプション) [名前付き式の追加] 領域で [無効] オプションを選択し、式を無効にします。無効にした式は式エディタにリスト表示されません。また、他の名前付き式または名前のない式はそれ呼び出すことができません。
8. (オプション) [名前付き式の追加] 領域の [プライベート] オプションを選択します。他の名前付き式のみがプライベート式を呼び出すことができます。名前のない式はプライベート式を呼び出すことができません。
9. (オプション) [名前付き式の追加] 領域の [編集] をクリックし、[式エディタ] を開きます。
10. [サブミット] をクリックします。
名前付き式が作成されます。

式エディタの使用法

式エディタを使用して、以下のタスクを実行できます。

- 関数と演算子、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。

以下の手順に従います。

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。
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。
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かつこと、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. 257)
 - [式から条件を削除する](#) (P. 257)
 - [式の中で条件をグループ化する](#) (P. 257)
 - [式でブール演算子を変更する](#) (P. 259)
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 ボックスでカッコ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

以下の手順に従います。

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、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、監査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。

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、以下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
- 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により、SiteMinder で共通名を定義して、それぞれの共通名をユーザ属性名にマップできます。属性名は、基礎スキーマが異なる複数のユーザ ディレクトリに含めることができます。

ユーザ属性マッピングの概要

ユーザ属性マッピングには、以下の 5 つのタイプがあります。

- 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、以下に示す相違点があります。

- アクセス
 - ユーザ属性マッピングの一部のタイプは、ユーザ属性値にマップされている読み取り専用 (R) タイプです。これらの属性は変更されません。ユーザ属性マッピングの他のタイプは、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) 可能タイプです。これらの属性は変更可能です。
 - 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	no
グループ名 (RW)	ブール値	ディレクトリ固有	no
マスク (RW)	ブール値	ディレクトリ固有	no
定数 (R)	文字列、数値、ブール	ディレクトリ固有	no
式 (R)	文字列、数値、ブール	ディレクトリ固有	no

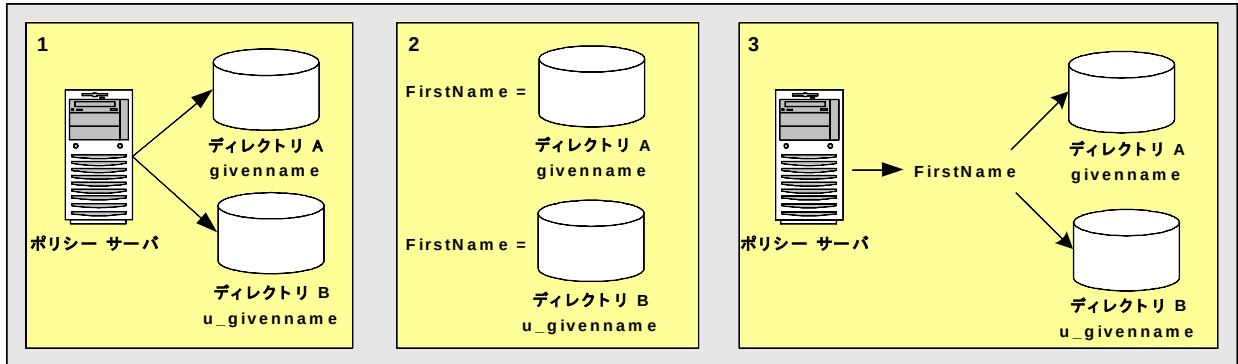
名前付き式タイプ	データ型	可視性	プレフィクス
仮想ユーザ属性 (R)	文字列、数値、ブール	グローバル	#
ユーザクラス (R)	ブール値	グローバル	@

属性マッピングのしくみ

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。このようなユニバーサルなスキーマにより、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。

このようなビューを作成することで、ディレクトリ タイプに関係なくユーザ属性を参照します。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、以下の図に示します。



- 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。

この設定の結果、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。ディレクトリは操作上、同一となります。

属性マッピングの定義

以下の 1 つ以上の属性マッピング タイプを使用できます。

- [エイリアス](#) (P. 266)
- [グループ名](#) (P. 268)
- [マスク](#) (P. 271)
- [定数](#) (P. 277)
- [式](#) (P. 279)

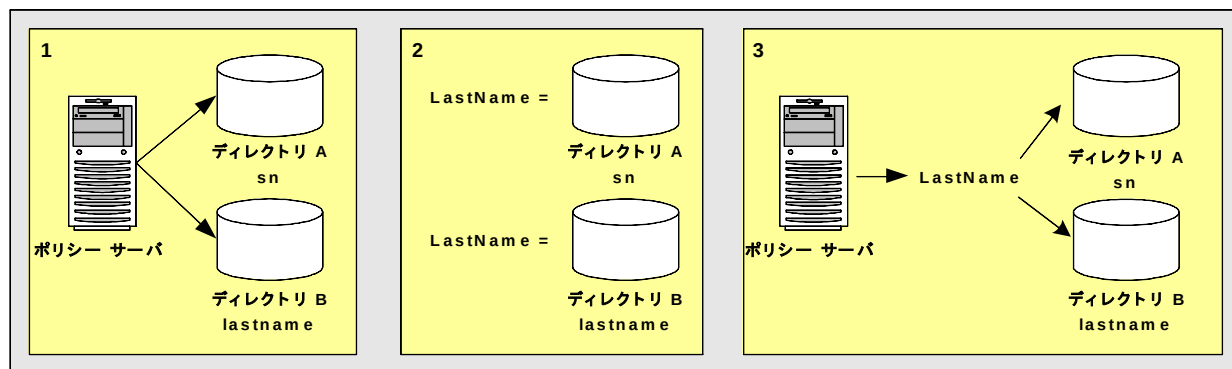
エイリアス

「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、ユーザのラスト ネーム (姓) を識別するが、基礎スキーマが異なる 2 つの LDAP ユーザディレクトリの基本的なシナリオを表します。以下の図に、2 つのエイリアス属性マッピングによって、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。



1. 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。

この設定の結果、同じユーザ情報の 2 つの異なるビューが生成されます。
2. **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。

LastName は、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使用します。

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。ユーザ属性マッピングはディレクトリ固有です。

以下の手順に従います。

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。

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ーであると、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。

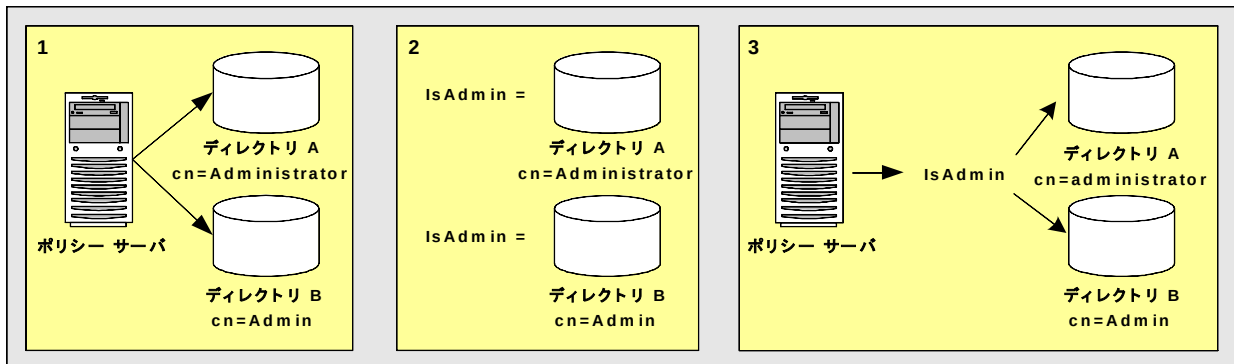
グループ名属性マッピングとユーザ クラスとの相違点は、以下のとおりです。

- 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「アット」マークで始まりません。

グループ名のユース ケース

このユース ケースでは、別々の基本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 2 つの LDAP ユーザ ディレクトリを示します。

以下の図に、2 つのグループ名属性マッピングによって、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。



1. 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。

この設定の結果、同じユーザ情報の 2 つの異なるビューが生成されます。

2. 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=groups,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。ユーザが IsAdmin を参照するのは、Administrator に適用されるポリシー、表現または他のオブジェクトを定義することがグループ化する場合です。

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。

以下の手順に従います。

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。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 ボックスのリストに表示されます。

マスク

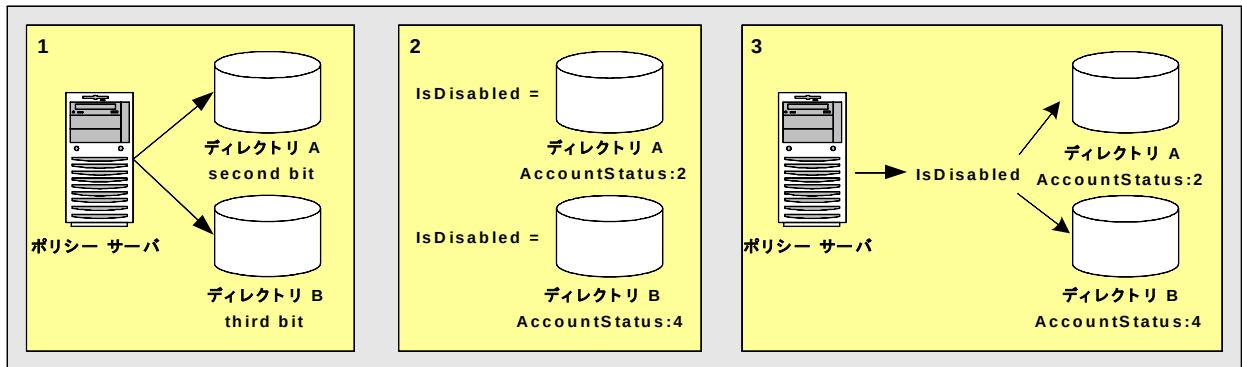
マスク属性マッピングでは、共通名を使用する名前にマッピングして、ビット パターンが格納されているユーザ属性を識別することができます。マスク属性マッピングにより、共通名は、読み取りまたは変更可能なユーザ属性値にマッピングされます。マスク属性マッピングでは、結果としてブール値が返されます。ユーザディレクトリ内のビット パターンが指定されたマスクと一致すると、マッピングの結果は TRUE 値になります。それ以外の場合、結果は FALSE になります。

マスクのユース ケース

アカウントの状態などのその属性に関する情報を提供するために、一部のディレクトリ実装では、属性の個別のビットを使用します。属性にビットマスクを適用できます。

このユース ケースでは、無効なユーザ アカウントを識別する 2 つの Active Directory ユーザ ストアを示します。各アカウントには、異なる基礎スキーマがあります。

以下の図に、2つのマスク属性マッピングによって、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。



1. 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。

この設定の結果、同じユーザ情報の 2 つの異なるビューが生成されます。

2. 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、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）によって判断します。
- ディレクトリ B では、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）によって判断します。

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。ユーザのアカウント ステータス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。

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

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 [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

ビットマスク属性マッピングでは、1 つまたは複数のビットの値を、ユーザ属性内の他のビットの値をマスクすることによってテストします。

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、ユーザ属性が `AccountStatus` と命名されると仮定します。属性 `AccountStatus` は、以下の 3 つのフラグの状態をビット パターンで格納します。

ビット パターン	フラグ
00?	アカウントは無効ですか?

ビット パターン	フラグ
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 TRUE になります。表に結果を説明します。

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なります。

AccountStatus:4

AccountStatus のビット単位の AND 演算はビットマスクで実行され、結果がビットマスクと等しいかどうかテストします。結果が等しい場合は、テストされたビットの値が 1 で、フラグが TRUE であることを意味します。以下の表に結果を示します。

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

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?
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 進数値をかつこ内に示します。

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`

`AccountStatus` のビット単位の AND 演算はビットマスクで実行され、結果がビットマスクと等しいかどうかテストします。結果が等しい場合は、テストされた両方のビットの値が **1** で、両方のフラグが **TRUE** であることを意味します。以下の表に結果を示します。

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

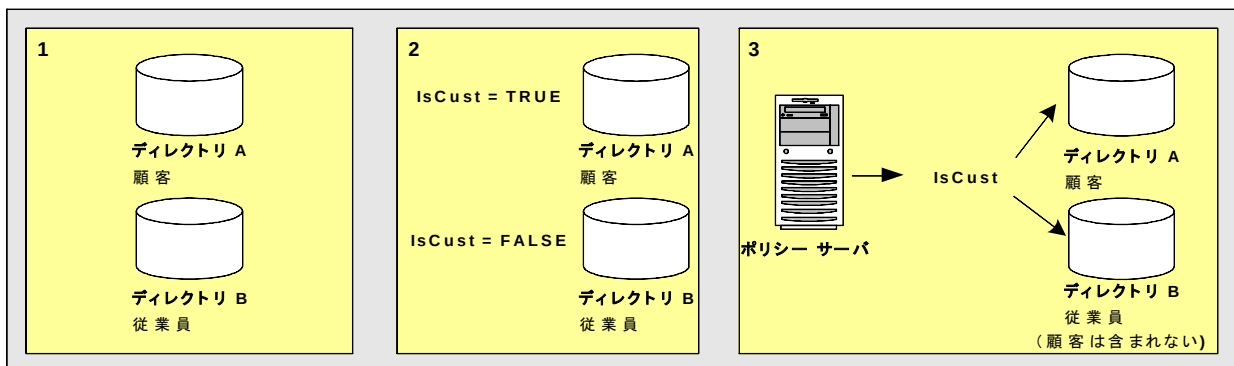
定数属性マッピングでは、共通名を、ディレクトリ内のどのユーザでも同一の値、つまり定数にマッピングすることができます。定数属性マッピングにより、共通名は、読み取り専用 (R) である定数にマッピングされるため、システム管理者以外は、このマッピングを変更できません。

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シナリオを表します。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. 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は、IsCust を参照します。共通名を使用することで、システムは、ユーザが格納される特定のディレクトリに関係なく、ユーザが顧客かどうかを判断できます。このマッピングは、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ことを示します。

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。

以下の手順に従います。

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。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 ボックスに表示されます。

式

式属性マッピングでは、式に共通名をマップできます。式には、基礎スキーマで指定した 1 つ以上のユーザ属性名を含めることができます。この式は、SiteMinder 式の構文ルールに従います。

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。
- いずれも読み取り専用 (R) です。
- いずれ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。

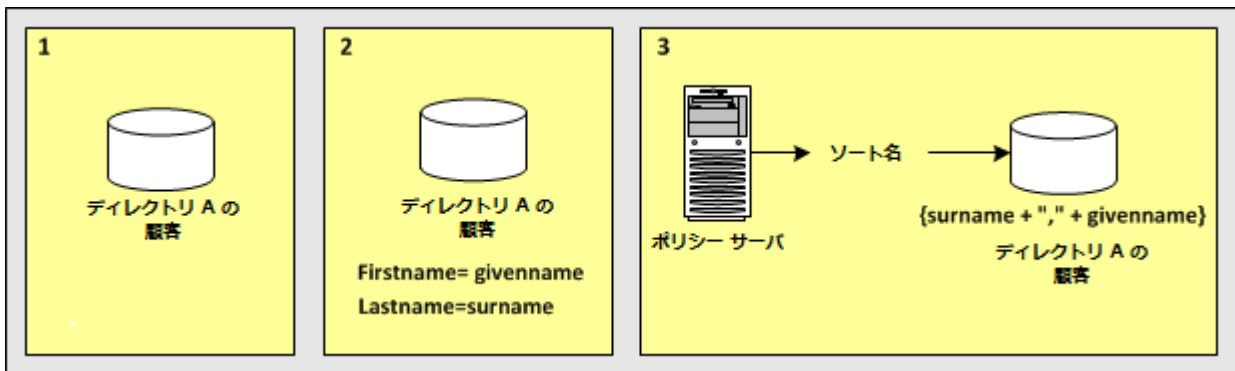
詳細情報:

[名前付き式](#) (P. 246)

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。保護されているリソースは、各ユーザの ソート名 (last name, first name) を必要とします。ユーザ ディレクトリは一意にこの属性を参照しません。代わりに、ディレクトリは、各ユーザのラスト ネーム (姓) を **surname** として、ファースト ネーム (名) を **givenname** として格納します。

以下の図に、式属性マッピングによって、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。



単一のユーザディレクトリで、共通名は、ディレクトリにユーザ属性名を使用してソート名を作成する式にマッピングされます。

- ディレクトリ A には、すべてのユーザレコードが含まれます。
- マッピングの名前は **SortName** です。
- **SortName** を定義する式は次のとおりです。
`{surname + "," + givenname}`
注: この式は、SiteMinder 式の構文ルールに従います。
- **SortName** は、**surname** および **givenname** 属性が含まれる式にマッピングされる共通名です。

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、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。

以下の手順に従います。

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。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 ボックスに表示されます。

上級ユーザ属性マッピングの例

以下の例では、より複雑なユーザ属性マッピングの設定を示します。

この展開例は、異なるタイプの 2 つのユーザ ディレクトリを使用する衣料品小売会社です。

ディレクトリ A

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

カスタマのみ用の 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 が同じユーザ情報を識別する方法を詳しく示します。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定義する方法について説明します。共通のビューはユニバーサルなスキーマとして役立ちます。これによってディレクトリは操作上、同一となります。

属性の説明	ディレクトリ A 属性(LDAP)	ディレクトリ B 属性(ODBC)
各ユーザのファースト ネーム (名)	givenname	u_first_name
各ユーザのラストネー ム (姓)	surname	u_last_name
各ユーザのソート名 (姓、名)	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
カスタマとしてのユー ザ	group:cn=customer,ou=groups,o=acme.com	ユーザは常に顧客です。
ユーザアカウントのス テータス	AccountStatus 属性 (1 セットのフラグ) 。 2 番目のビットは無効なアカウントです。	u_disabled

エイリアス マッピング タイプを持つ名の属性をマップする

ディレクトリ A およびディレクトリ B で名のユーザ属性を表すには 2 つのエイリアス属性マッピングを使用します。

展開

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。

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。

名前

FirstName

マッピング タイプ

エイリアス

定義

givenname

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。

名前

FirstName

マッピング タイプ

エイリアス

定義

u_first_name

ディレクトリ A でユーザを参照する場合、FirstName は givenname にマップされます。ディレクトリ B でユーザを参照する場合、FirstName は u_first_name にマップされます。

エイリアス マッピング タイプを持つ姓の属性をマップする

ディレクトリ A およびディレクトリ B で姓のユーザ属性を表すには 2 つのエイリアス属性マッピングを使用します。

展開

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。

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。

名前

LastName

マッピングタイプ

エイリアス

定義

surname

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。

名前

LastName

マッピングタイプ

エイリアス

定義

u_last_name

ディレクトリ A でユーザを参照する場合は、ユーザの姓が **surname** によって識別されるかどうかは共通のビューによって決定されます。ディレクトリ B でユーザを参照する場合は、ユーザの姓が **u_last_name** によって識別されるかどうかは共通のビューによって決定されます。

式およびエイリアス マッピングのタイプを持つソート名属性をマップする

ディレクトリ A およびディレクトリ B でユーザのソート名を表現するには、式属性マッピングとエイリアス属性マッピングを使用します。

展開

- ディレクトリ A は、ユーザごとにソート名を一意的に識別しません。ユーザごとに、ディレクトリ A では名が **givenname** として格納され、姓が各ユーザの **surname** として格納されます。
- ディレクトリ B は、ソート名を **sort_name** で識別します。

解決方法

1. ディレクトリ A の式属性マッピングを以下のように作成します。

名前

ソート名

マッピングタイプ

式

定義

`(surname + "," + givenname)`

注: この式は、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。

名前

ソート名

マッピングタイプ

エイリアス

定義

`sort_name`

ディレクトリ A でユーザを参照する場合、ソート名は指定した式に基づいて計算されます。ディレクトリ B でユーザを参照する場合、ソート名は属性 `sort_name` によって表現されます。

グループおよび定数マッピングのタイプを持つカスタマをマップする

ディレクトリ A およびディレクトリ B でカスタマを識別するには、グループおよび定数属性マッピングを使用します。

展開

- ディレクトリ A には、従業員が格納されます。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`

- 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。ディレクトリ B に格納されている以上、そのユーザは顧客です。

解決方法

1. ディレクトリ A のグループ属性マッピングを以下のように作成します。

名前

IsCustomer

マッピング タイプ

グループ

定義

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作成します。

名前

IsCustomer

マッピング タイプ

定数

定義

TRUE

ディレクトリ A を参照するとき、ユーザが

cn=Customers,ou=Groups,o=acme.com に属する場合、ユーザはカスタムと見なされます。ディレクトリ B を参照する場合、すべてのユーザはカスタムです。

マスクおよび式マッピングのタイプを持つアカウント ステータスをマップする

ディレクトリ A およびディレクトリ B で無効なユーザ アカウントを識別するには、マスク属性マッピングおよび式属性マッピングを使用します。

展開

- 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。
- 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作成します。

名前

IsDisabled

マッピング タイプ

マスク

定義

AccountStatus : 2

定義は、ビット パターンが AccountStatus に格納され、ビット マスクが 2 (10 進) であることを示します。

2. ディレクトリ B の式属性マッピングを以下のように作成します。

名前

IsDisabled

マッピング タイプ

式

定義

(u_disabled = "y")

u_disabled はブール式です。

ディレクトリ A を参照する場合は、ビット パターンによって、ユーザが無効かどうか判定されます。ディレクトリ B を参照する場合は、式によって、ユーザが無効かどうか判定されます。

第 7 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 \(P. 289\)](#)

[ID マッピングの設定 \(P. 291\)](#)

[従来のディレクトリ マッピング設定 \(P. 298\)](#)

[レルムからのディレクトリ マッピングの削除 \(P. 301\)](#)

[ディレクトリ マッピングの例 \(P. 301\)](#)

[ディレクトリ マッピングとレスポンス \(P. 306\)](#)

ディレクトリ マッピングの概要

ポリシー サーバでは、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。ただし、ユーザは1つのディレクトリに対して認証し、個別のディレクトリに対して許可できます。この機能をディレクトリ マッピングといいます。

許可情報を格納する個別の分散したユーザ ディレクトリに、認証情報を格納するセントラルディレクトリをマップできます。許可ディレクトリは特定のネットワーク アプリケーションと関連付けられます。マッピングにより、認証済みのユーザを別の許可ディレクトリで探します。

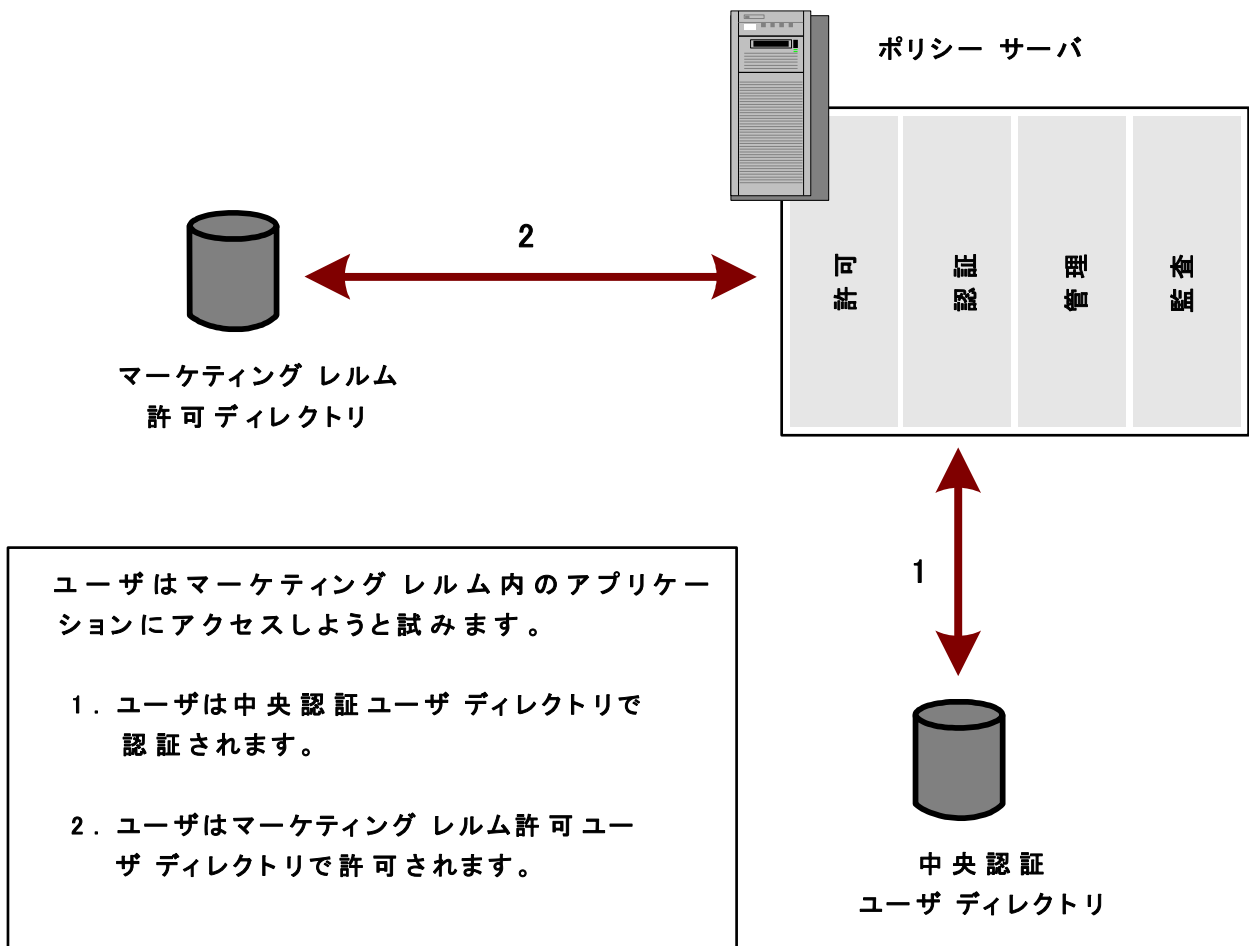
注: ディレクトリ マッピングはインパーソネーション（偽装）をサポートしません。偽装対象のユーザは、ドメインに関連付けられている認証ディレクトリ内で一意である必要があります。そうでない場合、インパーソネーションは失敗します。

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。
2. ディレクトリ マッピングを設定します。
3. レルムにディレクトリ マッピングを割り当てます。特定のネットワーク アプリケーションにおいて、ユーザは、このディレクトリに対して許可されます。

たとえば、会社のすべてのユーザは、1つの中央のユーザディレクトリに対して認証されます。マーケティング組織は、個別のユーザディレクトリに対して許可されます。認証ユーザディレクトリとマーケティング許可ユーザディレクトリの間でディレクトリ マッピングを設定してから、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。このレルムは、マッピングに定義された許可ディレクトリを使用します。ユーザがマーケティングアプリケーションへのアクセスを試みると、ユーザは中央のユーザディレクトリに対して認証され、マーケティングユーザディレクトリに対して許可されます。

この例を以下の図に示します。



許可 ID マッピングおよび検証 ID マッピング

ID マッピングでは複数のターゲット ユーザ ディレクトリを設定することができ、カスタム検索条件を使用することもできます。この設定では、環境を柔軟に設定できます。

ID マッピングには 2 つのタイプがあります。

許可 ID マッピング

あるディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを許可するためのディレクトリ マッピングです。

検証 ID マッピング

1 つのディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを検証するためのディレクトリ マッピングです。1 つのポリシー サーバに接続された認証ユーザ ディレクトリは、別のポリシー サーバに接続された検証ユーザ ディレクトリにマップされます。

ID マッピングには、既存のユーザ ディレクトリ接続は必要ありません。

従来の（認証/許可および認証/検証）ディレクトリ マッピングと呼ばれる、ディレクトリ マッピングのための従来の方法があります。この従来の方法は、このリリースにおいても使用でき、現在ある従来のマッピングはすべて同様に動作し続けます。

従来のディレクトリ 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、認証ディレクトリのほか、許可ディレクトリまたは検証ディレクトリに必要です。

ID マッピングの設定

ID マッピングでは、カスタム検索条件を使用して、ソース ディレクトリからターゲット ディレクトリにユーザをマッピングするための高度な方法が利用できます。ID マッピングはユーザ許可およびユーザ検証に使用することができます。

ID マッピングでは、以下の 2 つの方法を使用できます。

- 許可マッピング（デフォルト）
- 検証マッピング

ID マッピングではカスタム検索を有効にし、別の ID マッピング エントリ オブジェクトを使用して、マッピング ルールの順序を制御できます。ポリシー サーバは最初に、ID マッピングで定義されたマッピング メカニズムを使用して、ユーザを見つけようとします。マッピングが失敗した場合は、セッション ユーザ ディレクトリを使用してユーザを特定します。ポリシー サーバにおいて、このディレクトリをデフォルトに設定するには、セッション ディレクトリがポリシー ストア内にある必要があります。

注: 検証マッピングについては、認証ディレクトリはローカル ストア内になくても構いません。

ID マッピングがサポートするディレクトリ

以下の表に、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を示します。

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
LDAP	同一の DN	ユニバーサル ID
	ユニバーサル ID	カスタム検索
	カスタム検索	
AD	同一の DN	ユニバーサル ID
	ユニバーサル ID	カスタム検索
	カスタム検索	
リレーショナル データベース	ユニバーサル ID	同一の DN
	カスタム検索	ユニバーサル ID
		カスタム検索

ID マッピング エントリ タイプ

ID マッピングは、1 つ以上のエントリを含むことができます。ID マッピング エントリは、ターゲット ディレクトリでユーザを検索する方法を指定するルールを定義します。ターゲット ディレクトリでユーザを検索するために、ポリシー サーバは、セッション チケット情報に基づいた検索条件を使用します。

ID マッピングは 1 つ以上のターゲット ユーザ ディレクトリを含むことができます。同じ ID マッピング オブジェクト内の別のターゲット ユーザ ディレクトリのためにエントリを追加できます。ID マッピング エントリは、マッピングの順序づけられたリストとして処理されます。

2 つのタイプの ID マッピング エントリは、以下のとおりです。

許可 ID マッピング エントリ

ソース ディレクトリとターゲット ディレクトリへのリンクを指定します。リンクが利用できない場合、認証ディレクトリを使って許可を行います。同一の DN またはユニバーサル ID を選択するか、カスタム検索条件を指定できます。

検証 ID マッピング エントリ

ソース ディレクトリ名をテキスト文字列およびターゲット ディレクトリへのリンクとして指定します。ソース ディレクトリの名前を指定するか、デフォルト値の **SMSESSION User Directory** を選択できます。デフォルト値は、検証に、セッション チケット内のユーザ ディレクトリが使用されることを示します。ターゲット ディレクトリへのリンクがない場合、認証ディレクトリを使って検証を行います。同一の DN またはユニバーサル ID を選択するか、カスタム検索属性を指定できます。

複雑なユーザ検索式の使用

複雑なユーザ検索条件を使用して、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プできます。ユーザ検索条件は属性の組み合わせです。属性はソース ディレクトリまたはターゲット ディレクトリのものです。

通常、ユーザ検索条件はユーザ ディレクトリ固有のものです。たとえば、ODBC ベースのユーザ検索条件は、LDAP ベースのユーザ検索条件とは異なります。

別のネームスペースでユーザディレクトリをサポートするには、検索条件をユーザディレクトリごとに定義します。また、ターゲットユーザディレクトリ用にユーザディレクトリ属性マッピングを定義することもできます。その後、各ユーザディレクトリは属性マッピング用にそれぞれ固有の検索条件を定義する必要があります。ユーザディレクトリ属性マッピングでは、ユーザディレクトリ固有の検索条件を定義できます。

許可 ID マッピングを設定する方法

認証と許可の ID マッピングの設定は 2 段階のプロセスです。

1. 許可 ID マッピングの設定
2. 許可 ID マッピングのレルムへの割り当て

許可 ID マッピングの設定

1 つのディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを許可するには、ID マッピングを設定します。

以下の手順に従います。

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. [アイデンティティ マッピング] をクリックします。
3. [アイデンティティ マッピングの作成] をクリックします。
4. マッピングの名前および説明を指定します。
5. マッピングタイプとして [認証-許可] を選択します。
6. [エントリの作成] をクリックします。
7. ID マッピング エントリの名前を指定します。
8. それぞれのリストから、ソースとターゲットのディレクトリを選択します。
9. ユーザ検索条件として、以下のいずれかのオプションを選択します。

同一の DN

ユーザの DN（識別名）を認証ディレクトリから検証ディレクトリにマップします。

ユニバーサル ID

認証ディレクトリの [ユニバーサル ID] 属性の値と、検証ディレクトリの [ユニバーサル ID] の値を照合します。

カスタム検索

ターゲットディレクトリとソース ディレクトリの属性を指定します。ソース ディレクトリ属性は、ユーザ指定の属性または SiteMinder セッションです。

10. [OK] をクリックします。

ID マッピング エントリは許可 ID マッピング オブジェクトに追加されます。

11. [サブミット] をクリックします。

許可 ID マッピング オブジェクトが設定されます。

許可 ID マッピングのレルムへの割り当て

許可 ID マッピングをレルムに割り当てます。ポリシー サーバはレルム内で指定されている許可マッピングを使用してユーザの許可を行います。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
[レルム] ページが表示されます。
2. 変更するレルムを選択します。
3. [変更] をクリックします。
4. 許可マッピング リストから許可ディレクトリとして使用する ID マッピングを選択します。
5. [サブミット] をクリックします。

許可 ID マッピングが、選択されたレルムに割り当てられます。

詳細情報:

[アプリケーションの詳細なポリシー コンポーネントの設定 \(P. 555\)](#)

検証 ID マッピングを設定する方法

認証と検証の ID マッピングの設定は 2 段階のプロセスです。

1. 検証 ID マッピングを設定します。
2. 検証 ID マッピングをレルムに割り当てます。

注: 同じストア内のディレクトリに検証マッピングを作成できます。ソースディレクトリは、必ずしもローカルストアにある必要はありません。

検証 ID マッピングの設定

1 つのディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを検証するように、ID マッピングを設定します。

以下の手順に従います。

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. [アイデンティティ マッピング] をクリックします。
3. [アイデンティティ マッピングの作成] をクリックします。
4. 名前と説明を入力します。
5. マッピング タイプとして [認証-検証] を選択します。
6. [エントリの作成] をクリックします。
7. 名前を入力します。
8. ディレクトリがセッションの内部にない場合は、ソース ディレクトリを指定します。
9. ターゲット ディレクトリを選択します。
10. ユーザ検索条件を選択します。

カスタム検索を選択する場合は、ターゲット ディレクトリおよびソース ディレクトリの属性を指定します。

ソース ディレクトリ属性は、ユーザ指定の属性または SiteMinder セッションです。

11. [OK] をクリックします。

ID マッピング エントリは検証 ID マッピング オブジェクトに追加されます。

12. [サブミット] をクリックします。

検証 ID マッピング オブジェクトが設定されます。

検証 ID マッピングのレルムへの割り当て

検証 ID マッピングをレルムに割り当てます。ポリシー サーバはレルム内で指定されている検証ディレクトリ マッピングを使用してユーザの許可を行います。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. 変更するレルムを選択します。
3. [変更] をクリックします。
4. 検証マッピング リストから許可ディレクトリとして使用する ID マッピングを選択します。
5. [サブミット] をクリックします。

検証 ID マッピングが選択されたレルムに割り当てられます。

詳細情報:

[アプリケーションの詳細なポリシー コンポーネントの設定 \(P. 555\)](#)

デフォルトのグローバル検証ディレクトリ マッピングの設定

検証マッピングのグローバル デフォルトとして、単一の検証 ID マッピングを使用できます。グローバル検証 ID マッピングを設定すると、レルムごとに設定する必要がなくなるので時間を節約できます。ただし、グローバル検証 ID マッピングはローカル マッピングで上書きできます。

以下の手順に従います。

1. [ポリシー] - [グローバル] をクリックします。
2. [グローバル検証ディレクトリ マッピングの選択] をクリックします。
[グローバル検証ディレクトリ マッピングの選択] ページが開きます。
3. 対応するリストから検証 ID マッピング オブジェクトを選択します。

4. [サブミット] をクリックします。

選択された検証 ID マッピング オブジェクトは、検証マッピングのグローバルデフォルトとして設定されます。

従来のディレクトリ マッピング設定

従来のディレクトリ マッピングは、同一の DN またはユニバーサル ID を使用して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プします。

従来のディレクトリ マッピング方法の 2 つのタイプは、以下のとおりです。

- 認証/許可（許可マッピング）
- 認証/検証（検証マッピング）

認証/許可マッピングまたは認証/検証マッピングが設定されると、ポリシー サーバはまずセッションユーザディレクトリを使用してユーザを見つけようとします。次に、ユーザがセッションユーザディレクトリで見つからない場合にのみ、指定されたマッピング メカニズムを使用します。

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/許可のディレクトリ 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 マッピングを設定します。
2. 許可ディレクトリをレルムに割り当てます。

ディレクトリ マッピングの設定

1 つのディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを許可するには、ディレクトリ マッピングを設定します。

以下の手順に従います。

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] をクリックします。
3. [ディレクトリ マッピングの作成] をクリックします。
4. 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。

5. [同一の DN] または [ユニバーサル ID] を選択します。

重要: ユニバーサル ID が認可ディレクトリ内の 1 つのエントリを指している場合にのみ、ディレクトリ マッピングは成功します。

6. [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。

詳細情報:

[ユニバーサル ID](#) (P. 245)

許可ディレクトリのレルムへの割り当て

レルムにディレクトリ マッピングを割り当てます。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。

以下の手順に従います。

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] リストから、認証されたユーザを許可するためにレルムが使用できるユーザ ディレクトリを選択します。

デフォルト値は、ディレクトリ マッピングがないことを示します。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. 602)

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

1 つのディレクトリに対してユーザを認証し、別のディレクトリに対してユーザを検証するには、認証/検証ディレクトリ マッピングを設定します。

注: 認証/検証マッピングはグローバルです。

以下の手順に従います。

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/検証ディレクトリ マッピング] をクリックします。
3. [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。
4. [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。
5. [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。
6. 利用可能なオプションからマップされた DN を選択します。
7. [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 タスクが作成されます。

レルムからのディレクトリ マッピングの削除

許可または検証のディレクトリ マッピングを更新または削除するには、既存のレルムからディレクトリ マッピング関連付けを解除します。

以下の手順に従います。

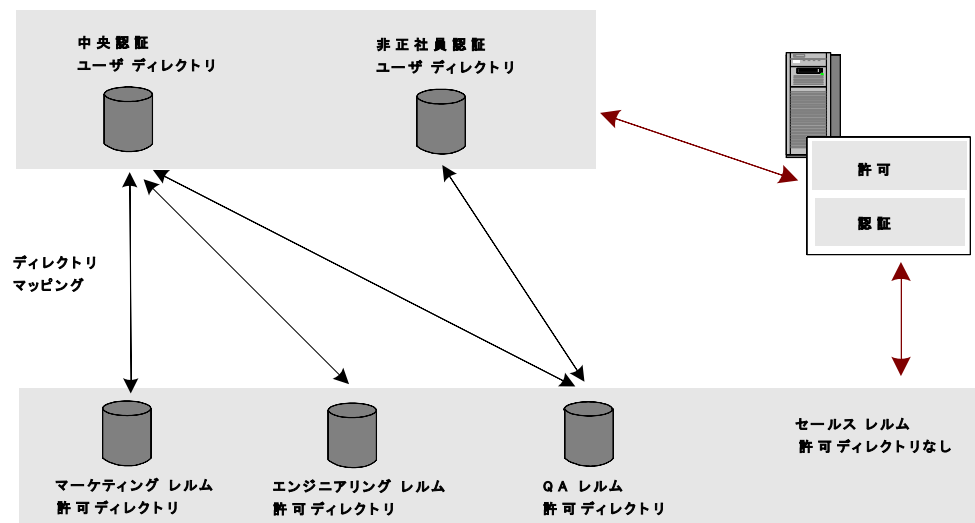
1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. 変更するレルムを選択します。
3. [変更] をクリックします。
4. 対応するリストから関連付けを解除するディレクトリ マッピングを選択します。
5. [サブミット] をクリックします。

ディレクトリ マッピングは選択されたレルムから削除されます。

同じ手順で、レルムから、許可および検証の ID マッピングの関連付けを解除することができます。

ディレクトリ マッピングの例

ネットワーク リソースにアクセス権があるユーザを認証し許可するためには複数のディレクトリ マッピングが必要です。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単純なケースを示します。



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。リクエストされたリソースが **Marketing**、**Engineering** または **Quality Assurance** レルムにある場合、ポリシー サーバは認証ディレクトリの 1 つを使用します。この決定は、セッションチケット情報のディレクトリに基づきます。

複雑なユーザ検索条件を使用したディレクトリをマップするために ID マッピングを使用できます。

詳細情報:

[レルム](#) (P. 595)

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

複数のユーザディレクトリを示している上記の図では、正規従業員の認証データはセントラル認証ユーザディレクトリに格納されています。従業員がエンジニアリングレルムのリソースへのアクセスを試みると、従業員は認証されます。

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。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の間のディレクトリ マッピングを検索します。次に、サーバはユーザ識別情報を許可ディレクトリにマップします。これで、ポリシー サーバは、その従業員が要求したレルムへのアクセス権を持っているかどうかを確認できます。

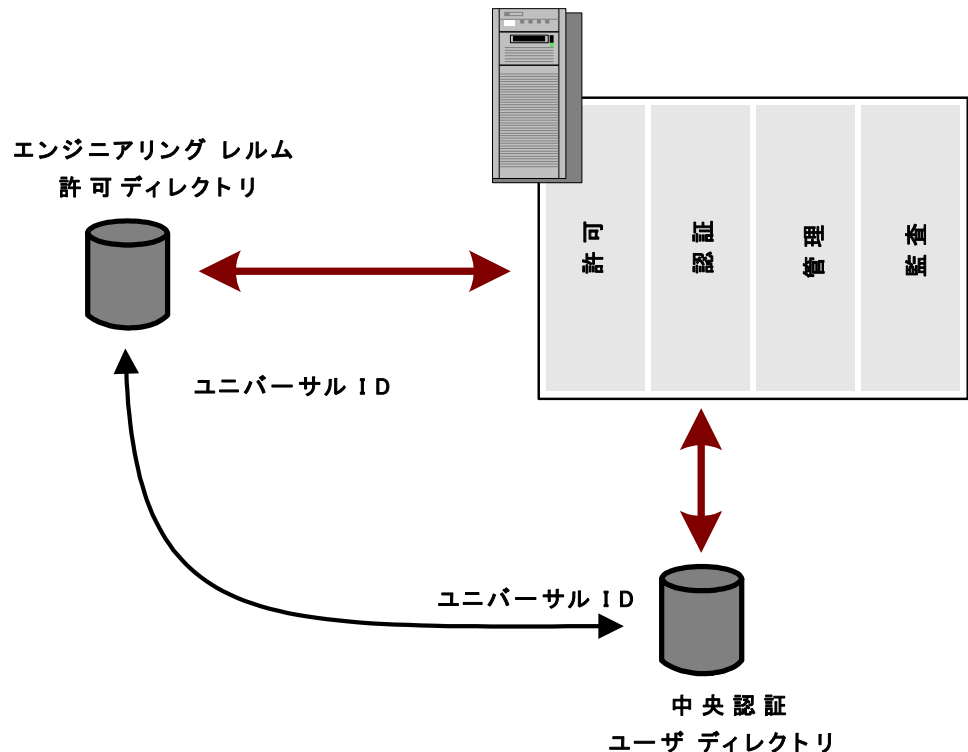
臨時従業員が品質保証レルムのリソースにアクセスする場合

複数のユーザディレクトリを示している上記の図では、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。

ポリシーサーバは、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、ポリシーサーバは、その従業員が認証された同じディレクトリに対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

複数のユーザディレクトリを示している上記の図では、従業員の認証データはセントラル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のリソースへのアクセスを試みます。従業員が認証されると、ポリシーサーバはディレクトリマッピングを使用して、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。ユーザは従業員ユニバーサルIDによって識別されます。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てします。許可ディレクトリで一致するユニバーサルIDを検索するために、ポリシーサーバは、認証ディレクトリで定義されているユニバーサルID属性を使用します。許可ディレクトリでユニバーサルIDを特定した後は、ポリシー処理を終了し、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判断できます。

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。認証ディレクトリでは大文字と小文字が区別されないが、許可ディレクトリでは大文字と小文字が区別される場合には、競合が発生する場合があります。

認証ディレクトリでは大文字と小文字が区別されるため、認証が失敗する場合があります。ただし、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。許可が大文字と小文字の区別により失敗する場合、ポリシー サーバには回復する方法がありません。

許可ディレクトリが大文字/小文字を区別する場合は、許可ディレクトリの必須形式と一致するように、認証したユーザ名の形式を変更します。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」に変更してから、ユーザを許可します。

複雑なユーザ 検索条件による ID マッピング

ID マッピングは、セッション チケット情報、および ID マッピング エントリの構成方法に基づいてユーザを見つけます。

XPS 関数 `IDENTITY_MAP` は名前によって ID マッピング オブジェクトを検索します。 エントリを解決してターゲット ユーザディレクトリでユーザを見つけると、2 つ目のパラメータで指定された属性の値を返します。

例:

「target」という名前の ID マッピングによって定義されたターゲット ユーザディレクトリのユーザの姓を取得するには、以下のように指定します。

```
IDENTITY_MAP ("target", last_name;
```

ディレクトリ マッピングとレスポンス

ユーザディレクトリ属性を組み込むことができ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してください。たとえば、**OnAuth** イベントの場合、ディレクトリ マッピングは認証ディレクトリから値を取得します。**OnAccess** イベントの場合、ディレクトリ マッピングは許可ディレクトリから値を取得します。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 \(P. 633\)](#)」を参照してください。

第 8 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 \(P. 307\)](#)
- [基本認証方式の設定方法 \(P. 315\)](#)
- [SSL を介した基本認証方式の設定方法 \(P. 317\)](#)
- [HTML フォーム認証の設定方法 \(P. 320\)](#)
- [Windows 認証方式 \(P. 355\)](#)
- [Kerberos 認証の設定方法 \(P. 359\)](#)
- [Information Card 認証方式 \(P. 376\)](#)
- [RADIUS CHAP/PAP 認証方式 \(P. 397\)](#)
- [RADIUS サーバ認証方式 \(P. 399\)](#)
- [SafeWord サーバ認証方式 \(P. 401\)](#)
- [SafeWord サーバ認証方式と HTML フォーム認証方式 \(P. 402\)](#)
- [SecurID 認証方式 \(P. 404\)](#)
- [X.509 クライアント証明書認証方式 \(P. 416\)](#)
- [X.509 クライアント証明書および基本認証方式 \(P. 420\)](#)
- [X.509 クライアント証明書または基本認証方式 \(P. 423\)](#)
- [X.509 クライアント証明書および HTML フォーム認証方式 \(P. 426\)](#)
- [X.509 クライアント証明書または HTML フォーム認証方式 \(P. 429\)](#)
- [匿名認証方式 \(P. 433\)](#)
- [カスタム認証方式 \(P. 435\)](#)
- [OAuth 認証方式を設定する方法 \(P. 437\)](#)
- [OpenID 認証方式 \(P. 452\)](#)
- [レガシー フェデレーション認証方式 \(P. 466\)](#)
- [偽装認証方式 \(P. 467\)](#)

認証方式の概要

一般的に、ネットワーク リソースの所有者は、これらのリソースにアクセスすることを試みるユーザの識別情報を確認したいと考えます。ユーザを識別することにより、ユーザがアクセスできるリソースを決定し、そのユーザ向けにコンテンツをパーソナライズする方法を決定できます。匿名のユーザを一意に追跡することは有用であり、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、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。認証時に、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。

ポリシー サーバでは、さまざまな認証方式がサポートされます。リスク低いネットワーク リソースでは、簡単な方式を使用し、重要なネットワーク リソースのセキュリティを高めるためには、複雑な方式を使用します。

ポリシー サーバは認証方式テンプレートを使用します。これにより、方式を処理するのに必要な情報がポリシー サーバに提供されます。管理 UI を使用して、認証方式テンプレートを設定します。

サポートされる認証方式とパスワード ポリシー

一部の認証方式タイプでは、パスワード ポリシーがサポートされます。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、[スキーム共通セットアップ] 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ます。この認証方式のチェック ボックスに対して有効にされたパスワード ポリシーを確認します。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イアウトされ、チェック ボックスは使用できません。

以下の表に、サポートされる認証方式タイプと、これらがパスワード ポリシーをサポートするかどうかを示します。

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
OAuth	いいえ
OpenID	いいえ

認証方式タイプ	パスワード ポリシーのサポート
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。ポリシー サーバは、認証プロセス中に、1 つのユーザ ストアにおいてユーザが無効になっていることを検出できます。ただし、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のユーザを検索し続けます。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。

このデフォルトの動作は設定可能です。ユーザストアにおいて無効になっているユーザを検索した後に、ポリシーサーバによるディレクトリの検索を停止します。

以下の手順に従います。

1. レジストリ キー `ReturnOnDisabledUser` を追加します。

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\Wow6432Node\Netegrity\SiteMinder\CurrentVersion  
PolicyServer
```

Solaris

以下の行を `sm.registry` ファイルに追加します。

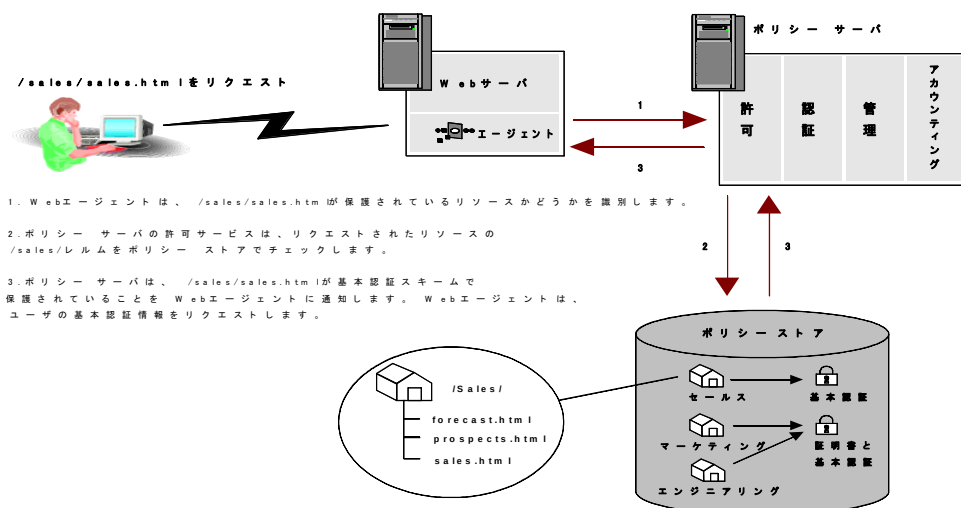
```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion  
PolicyServer  
ReturnOnDisabledUser=0x1; REG_DWORD
```

2. `ReturnOnDisabledUser` に値 `1` を割り当てます。

認証方式処理

ユーザの識別情報を特定するために、ポリシーサーバはリソースが含まれているレルムの認証方式を使用します。認証方式は、ユーザが提供する必要がある認証情報、およびポリシーサーバがユーザの識別情報を確認するために使用する方法を指定します。

認証方式を設定するには、管理 UI を使用し、レルムに方式を割り当てます。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/` レルムの保護されたリソース `sales.html` を要求しています。このレルムには、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、Web エージェントを使用してユーザの基本認証情報を要求します。エージェントは、ユーザにユーザ名とパスワードの入力を求めます。

詳細情報:

[レルムの設定](#) (P. 602)

永続認証コンテキスト データ

ポリシー サーバは、セッションストアの認証コンテキストデータを保持できます。セッションストアへのデータの格納は、一部の認証方式のオプション機能です。セッションストアは、ユーザデータを格納するためのセッションチケットとは別のリポジトリです。

ポリシー サーバはセッション変数を作成し、それらを、セッション変数にちなんで命名されたセッション チケット フィールドとして処理します。ポリシー サーバは、セッションストアのセッション変数にアクセスし、認証の判断に影響を与えることができます。

保持された認証データからのセッション コンテキスト属性を操作、格納、または送り返すためにレスポンスおよびポリシーを設定できます。情報はセッションストアから取得され、Web エージェントに送られます。Web エージェントは再度データを格納したり、評価のために認証エンジンにデータを提供したりできます。さらに、独自のセッション変数を設定でき、それらを認証に使用できます。

認証コンテキスト情報を保存するには、認証方式設定における [方式のセットアップ] セクション内の [認証セッション変数を保持する] チェックボックスをオンにします。このオプションは以下の方式に使用できます。

- カスタム方式
- SAML 認証方式
- OpenID 認証方式
- OAuth 認証方式
- X.509 証明書方式

重要: セッションストア内の永続する認証データにより、認証時間の悪化が生じます。このオプションは、後で認証決定のために、この変数を使用する場合にのみ選択してください。それ以外の場合は、パフォーマンスに重大な悪影響をもたらす可能性があります。

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。ユーザがある方式で認証されると、その認証方式と同等もしくは低い保護レベルのリソースにアクセスできます。ただし、ユーザにはリソースへのアクセスを取得するための認証が必要です。

注: 匿名認証方式の保護レベルは、常に 0 です。カスタム方式には、0 から 1000 までの保護レベルがあります。他のすべての認証方式には 1 から 1000 までの保護レベルがあります。

たとえば、リソースのセットがすべてのネットワーク ユーザで利用できるのであれば、基本（ユーザ名とパスワード）認証方式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、高い保護レベルが設定されている X.509 クライアント証明書方式を割り当てることができます。ユーザ名とパスワードで認証したユーザは、収益情報にアクセスするデジタル証明書で 2 回目を認証できます。

認証方式の事前定義済み保護レベルが不適切になる場合があります。たとえば、フェデレーションシナリオでは、アサーティングパーティは依存パーティに合わせた別の保護レベルが必要になる場合があります。そのような場合、管理者は認証方式ライブラリの保護値が管理 UI で指定された保護レベルより優先するように指定できます。ライブラリの値は、ユーザセッションチケットに書き込まれます。カスタムと SAML 認証方式用の「認証方式の作成」ダイアログボックスの「各方式共通セットアップ」セクション内の「保護のオーバーライドを許可する」チェックボックスをオンにします。

認証方式と認証情報要件

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。

認証情報要件					
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
NTLM または Windows	必須*	必須*			
RADIUS CHAP/PAP	はい	はい			
RADIUS サーバ	はい	はい			

認証情報要件					
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
SafeWord サーバ	はい	はい			
SafeWord およびフォー ム	はい	はい			任意
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 によるリソースへのアクセスの場合、ポリシーサーバは、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、ユーザの認証情報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。

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 フォームベースの方式を作成できます。1つの方式タイプの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ます。

基本認証方式の設定方法

ポリシー サーバのインストールは、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。基本認証方式は、ASCII 文字のみサポートしています。

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、**Web** 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。ユーザが名前とパスワードを入力した後に、エージェントは暗号化された接続で、認証情報をポリシー サーバへ渡します。ポリシー サーバは認証情報を処理し、正常に処理した場合は、アクセスを許可するように **Web** エージェントに指示します。認証されなかった場合、ユーザは認証情報の再入力を求められます。

注: デフォルトでは、この方式は、ブラウザから **Web** エージェントに渡される認証情報を暗号化しません。標準的な HTTP 基本プロトコルを使用してデータが渡されます。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暗号化される認証方式の場合は、SSL を介した基本認証方式を使用します。

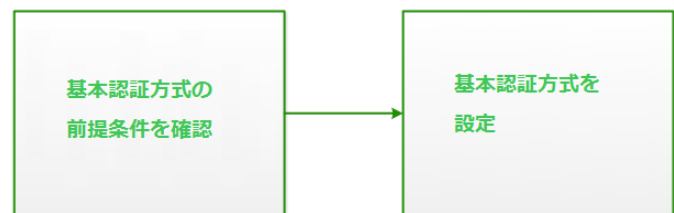
デフォルトでは、管理 UI で作成するアプリケーションおよびレلمは基本認証方式を使用します。認証方式は変更できます。

英語以外のレلم名を表示し、ユーザが基本認証ウィンドウでログイン認証情報として英語以外の文字を使用できるようにするには、以下の条件が満たされていることを確認します。

- Internet Explorer のみを、Windows Web サーバ上で、すべての IIS および Apache に対する Web ブラウザとして使用します。
- Web サーバマシンのロケールは、Web ブラウザのロケールと一致する必要があります。



ポリシー管理者



基本認証方式の設定方法

1. [基本認証方式の前提条件を確認します](#) (P. 316)。
2. [基本認証方式を設定します](#) (P. 317)。

基本認証方式の前提条件を確認する

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。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

基本認証方式の設定

管理 UI 内で基本認証方式を設定して、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のユーザ 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。

以下の手順に従います。

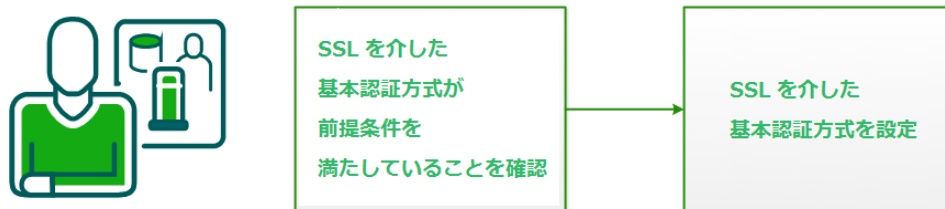
1. [インフラストラクチャ] - [認証] をクリックします。
2. [認証方式] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。
5. [認証方式のタイプ] リストから [基本テンプレート] を選択します。
6. [サブミット] をクリックします。

認証方式が保存され、これでレルムに割り当て可能になります。

SSL を介した基本認証方式の設定方法

SSL を介した基本認証方式は、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。このプロセスは基本認証と類似していますが、認証情報の送信は、暗号化された **Secure Sockets Layer** (SSL) 接続上で常に実行されます。保護されている URL が SSL を要求するように設定されていなくても、SSL 接続が使用されます。

注: SSL を介した基本認証方式は ASCII 文字のみサポートします。



ポリシー管理者

1. [SSL を介した基本認証方式の前提条件に適合していることを確認します](#) (P. 318)。
2. [SSL を介した基本認証方式を設定します](#) (P. 319)。

SSL を介した基本認証方式の前提条件の確認

SSL を介した基本認証方式を設定する前に、以下の前提条件が満たされていることを確認します。

- クライアントユーザ名とパスワード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、SSL が設定されていること。
- 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 に対してリソースがリダイレクトされる Web サーバに 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において必要な .scc MIME タイプを処理できるようになります。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

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。認証情報の送信は暗号化された SSL 接続を介して行われます。

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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前、および必要に応じて説明を入力します。
5. 保護レベルを選択します。
6. [認証方式のタイプ] リストから [SSL を介した基本テンプレート] を選択します。
7. 以下の方式に固有のフィールドに入力します。

Server Name

SSL 接続の確立を行う Web サーバの完全修飾ドメイン名を指定します。このサーバは、Web エージェントがインストールされているサーバと同じサーバである可能性もありますが、通常は別のサーバになります。

注: IP アドレスはサポートされていません。

このサーバは、ポリシー サーバが SSL 接続を介したユーザ認証情報のリダイレクトに使用する URL の起点となります。

ドメイン名には、ピリオドを 2 個以上含める必要があります。以下の形式を使用して、サーバ名を入力します。

`servername.domainname.com`

例: `server1.example.com`

ポート

SSL サーバがリスンするポートを指定します。この値は、デフォルト以外のポートを使用して通信する場合のみ必要です。

ターゲット

SSL 認証情報コレクタ (SCC) のパスおよび名前を指定します。

ターゲットの値は、エージェントに対して、SCC の呼び出しに何を使用するかを示します。ターゲットにより、ポリシー サーバが SSL 接続を介してユーザ認証情報をリダイレクトするときに使用する URL が完全なものになります。プロキシ サーバが SSL 認証を介した基本認証をサポートする特定の URL を必要とする場合は、必要に応じてターゲットをカスタマイズできます。

[ターゲット] のデフォルト値は次のとおりです。

`/siteminder/nocert/smgetcred.scc`

8. [サブミット] をクリックします。

認証方式が保存されます。これで、アプリケーションやレルムに方式を割り当てることができます。

HTML フォーム認証の設定方法

HTML フォーム認証方式では、カスタム HTML フォームで収集される認証情報を使用してユーザを認証します。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。
- ユーザ名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を使用)。ポリシー サーバシステム上の認証方式ライブラリにより、ユーザ データが DN にマップされます。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、ユーザ属性が含まれる認証情報に基づいた認証処理が可能です。属性の検証には、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 フォーム認証方式は、マルチバイト文字をサポートしています。

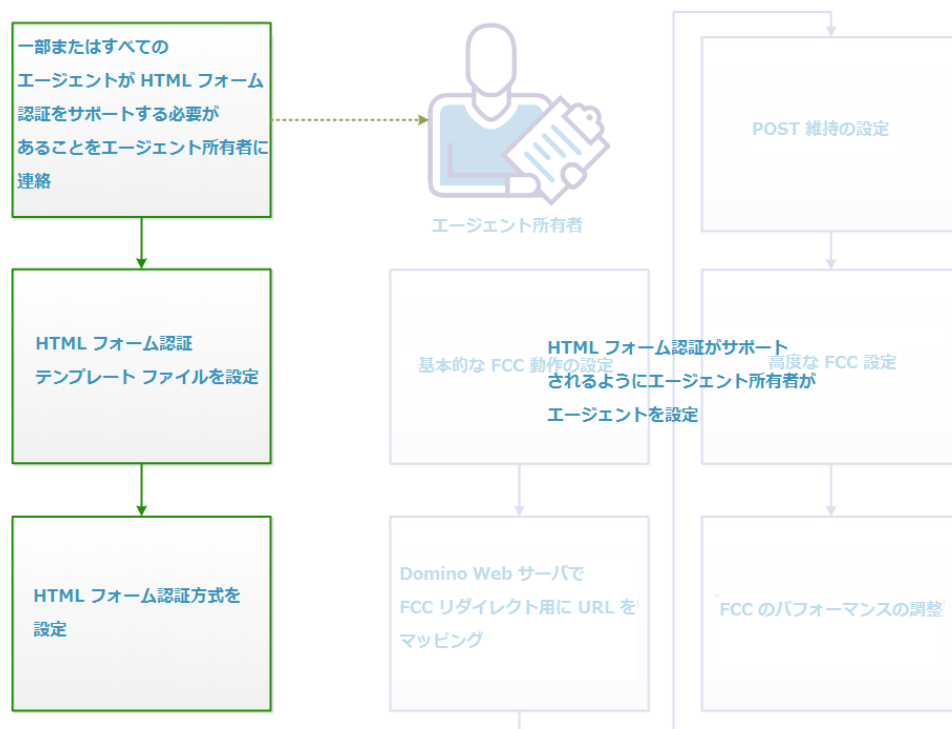
HTML フォーム認証は、SSL（Secure Sockets Layer）接続を介して実行するように設定できますが、これはオプションです。

HTML フォーム認証を使用するようにポリシー サーバを設定するには、ポリシー管理者とエージェント所有者の両方が設定プロセスを実行する必要があります。このシナリオでは、HTML フォーム認証を設定するためにポリシー管理者が従う必要があるプロセスを説明します。

注: HTML フォーム認証をサポートするための Web エージェントの設定方法の詳細については、関連シナリオの「Web エージェントの設定による HTML フォーム認証のサポート」を参照してください。



ポリシー管理者



1. [一部またはすべてのエージェントが HTML フォーム認証をサポートする必要があることをエージェント所有者に伝えます \(P. 323\)。](#)
2. [HTML フォーム認証テンプレート ファイルを設定します \(P. 323\)。](#)
3. [HTML フォーム認証方式を設定します \(P. 336\)。](#)

HTML フォーム認証が必要なことをエージェント所有者に伝える

このシナリオで概説されている手順に加えて、エージェント所有者は、HTML フォーム認証方式を使用して ID を検証するようにエージェントを設定する必要があります。

エージェント所有者、またはエージェントが HTML フォーム認証方式を使用するように設定するエージェントに対して責任を持つ所有者に伝えます。

HTML フォーム認証テンプレート ファイルの設定

Web エージェントには、HTML フォーム認証を処理する *FCC*（フォーム認証情報コレクタ）コンポーネントがあります。ユーザが HTML フォーム認証方式で保護されているリソースを要求すると、エージェントはそのユーザをフォーム認証テンプレート ファイルにリダイレクトします。

フォーム認証テンプレート ファイルは *FCC* を呼び出します。その後、*FCC* は、フォーム認証テンプレート ファイルの内容に基づいてブラウザ ページを生成します。テンプレート ファイルは Web サーバのネームスペースに存在し、ほかの HTML ファイルのようにアクセスできます。

フォーム認証テンプレート 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書かれています。

テンプレート *FCC* ファイルは、以下のディレクトリにあります。

- Windows : webagent¥samples¥forms
- UNIX : webagent/samples/forms

これらのサンプルのフォーム認証テンプレート ファイルを使用するには、[HTML フォーム認証方式を設定する](#) (P. 337) ときに、適切なターゲット ディレクトリを指定します。

エージェントは、他のいくつかの言語用にローカライズされたサンプル フォーム認証テンプレート ファイルを提供します。デフォルトでは、SiteMinder は英語のフォーム認証テンプレート ファイルを使用します。ローカライズされたフォーム認証テンプレート ファイルを設定する方法の詳細については、「*FCC (339P.)*の国際化を有効にする方法」を参照してください。

詳細情報:

[フォームベースの認証および基本的なパスワードサービスファイルのローカライズ \(P. 347\)](#)

安全な HTML フォーム認証テンプレートの使用

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。

カスタマイズ可能なデフォルトの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`

また、英語 (en-US) ロケール用の安全なフォームのセットは、以下のディレクトリにあります。

- Windows : webagent¥secureforms_en-US
- UNIX : webagent/secureforms_en-US

英語ロケール フォームの安全なバージョンを使用するには、secureforms_en-US 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_en-US
- UNIX : webagent/samples/forms_en-US

詳細情報:

[フォームベースの認証および基本的なパスワード サービス ファイルのローカライズ \(P. 347\)](#)

フォーム認証テンプレート ファイルに別のファイル拡張子を使用

フォーム認証テンプレート ファイルのデフォルト拡張子は .fcc です。そのため、フォーム認証テンプレート ファイルは、.fcc ファイルと呼ばれることもあります。ただし、その他の拡張子を使用することもできます。

以下の手順に従います。

- 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: 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。

- Domino または IIS Web サーバの場合は、Web エージェント設定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 エージェント設定ガイド**」を参照してください。

フォーム認証テンプレートのカスタマイズ

フォーム認証テンプレート (.fcc) ファイルは、標準の HTML タグ、および属性を確認し、カスタム機能を利用するために必要ないくつかの固有の表記を使用して書かれています。

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、そのファイルには UNIX システムと互換性のない改行文字が含まれていることがあります。UNIX システム上の非 UNIX 改行文字は、認証時に .fcc ファイルが失敗する原因となります。Windows の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、ファイルを調べて、追加された文字を削除してください。この状況を回避するには、UNIX 環境で使用する .fcc ファイルは UNIX システムで作成および編集してください。

FCC ファイルの最初の部分には、.fcc ファイルでの POST 操作中の制御文があります。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し、形式は @name=value にしてください。

name は変数の名前です。値は変数の値です。値には、%name1% の形式の文字列を含めることができます。FCC は、この文字列を name1 と関連付けられた変数の値に置換します。

.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、FCC が名前と関連付けられた値に置換する引用符 (") を含む、"\$\$name\$\$" 形式のテキストを含めることができます。この名前では大文字と小文字は区別されません。

以下のテーブルの非表示項目は、認証情報コレクタの状態を保持します。各値に引用符 (") を含めます。

名前	動的な値	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。
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。

.fcc ファイルは、少なくとも以下の項目を収集する必要があります。

- User name
- パスワード
- ターゲット

重要: ユーザが、以下のリストの認証方式のいずれかで保護されているリソースに **POST** リクエストを送信する場合は、**postpreservationdata** 入力を使用します。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

以下に、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%
@ sm retries= 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= PASSW ORD></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= sm authreason value= "$sm authreason$">
<tr><td><input type=submit value= LOGIN></td></tr>
</table></form></body>
</html>
```

前述のファイルは、エージェントのデフォルトのインストールに含まれている usermap.fcc サンプルファイルです。この .fcc ファイルでは、HTML フォームの [ユーザ名] フィールドおよび [組織] 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。

特殊な名前/値ペア

FCC では、数多くの特殊な名前/値ペア (@ ディレクティブ) を解釈して、特殊な処理を呼び出すことができます。次に、特殊な @ ディレクティブ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

ログインを実行するパスワード。

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レスポンス名のコロンの区切りのリスト。コロンの区切りのリストには、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、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 に以下の行を含めます。

```
@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、システムは、**.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られますが、**smretries** で指定されているのは、ユーザが有効な認証情報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: デフォルト値を使用してください。デフォルト値を使用している限り、**SafeWord** 認証方式は正常に動作します。

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

ユーザブラウザ上の永続的な **Cookie** にユーザの認証情報を保存するには、**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

ユーザが認証情報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が **FCCCompatMode=NO** の場合は、このディレクティブを使用して値を指定します。

smlogout

システムからユーザをログアウトさせます。**LogoffUri** パラメータと同じです。**.fcc** テンプレートに **@smlogout=true** と指定すると、FCC は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>**) と共に使用します。

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 テンプレート 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行われます。このテンプレートは、US-ASCII エンコーディングを使用します。

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/値ペアのプレフィックスである「sm」は、システム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、プレフィックス「sm」を使用しないでください。

FCC ファイルにおける名前/値ペアの生成方法

login.fcc テンプレートは、**\$\$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. @ ディレクティブの名前/値 (POST の場合のみ)
6. 「smheaders」の名前/値ペアで名づけられるレスポンス (POST の場合のみ)。

ローカライゼーション用の名前/値ペア

.fcc テンプレート 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 値は以下のような形式になります。

COUNTRY-LANGUAGE

たとえば、米語用の **smlocale** 値は以下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、以下の **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に加えて、電子メールアドレスや役職などの他の属性をユーザから収集できます。

以下の手順に従います。

1. HTML フォーム認証方式と関連する .fcc テンプレート ファイルを設定して属性を指定します。
2. [方式のセットアップ] ダイアログ ボックスの [追加属性リスト] フィールドに新しい属性を指定して、HTML フォーム認証方式を設定します。

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 テンプレート 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行われます。このテンプレートは、US-ASCII エンコーディングを使用します。

3. 追加属性を収集するように .fcc テンプレート ファイルを変更します。
ファイルの先頭に以下の行を追加します。

```
@password=PASSWORD=%PASSWORD%&newattr1=%newattr1%&newattr2=%newattr2%
```

追加属性に特殊文字が含まれる場合は、この行は以下のサンプルのようになります。

```
@password=PASSWORD=%urlencode(PASSWORD)%&newattr1=%urlencode(newattr1)%&newattr2=%urlencode(newattr2)%
```

```
newattr1=%newattr1%
```

最初の追加属性を表します。

```
newattr2=%newattr2%
```

2 つ目の追加属性を表します。

等号の前にある値は属性名で、パーセント記号 (%) の間にある値は属性値です。

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 ディレクティブに追加属性を追加してください。

属性をテンプレート ファイルに追加する場合は、以下の点に注意してください。

- 属性名は、**%attribute_name%** というフォーマットによって識別されます。この属性名は、ファイルに追加する **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)%
```

以下の文字列のような行を **.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、以下の文字列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、ユーザの認証を妨げる場合があります。
- HTML フォーム認証方式の属性の名前は、**.fcc** ファイルの追加属性の名前と一致している必要があります。

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] フィールドに以下の文字列を入力します。

```
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。

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するには、**smretries** ディレクティブ (**@smretries**) を設定します。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は表示されないことに注意してください。

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 **.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 ファイルとは異なるテンプレート ファイルです。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、非許可のページでは、無効な認証情報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表示されます。

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

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、「パスワード ポリシー」を参照してください。

HTML フォーム認証方式の設定

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。

同じポリシー ストアに複数の フォーム ベースの認証方式を設定できます。

フォーム認証方式をセットアップするには、以下の手順が必要です。

1. [HTML フォーム方式の前提条件を確認します](#) (P. 336)。
2. [HTML フォーム認証方式を設定します](#) (P. 337)。

HTML フォーム方式の前提条件の確認

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、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 が収集する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、さらに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。SmAuthHTML はポリシー サーバにバンドルされており、ポリシー サーバシステムに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を収集する場合は、FCC がそのデータをパックして、**user name** フィールドと **password** フィールドに格納する必要があります。これらのフィールドの長さは、511 文字未満である必要があります。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 サーバと同じシステムにインストールできます。

HTML フォーム認証方式の設定

HTML フォーム認証方式を使用して、カスタム HTML フォームでユーザを認証できます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。

5. [認証方式のタイプ] リストから [HTML フォーム テンプレート] を選択します。
6. Web サーバ名、ターゲットおよび追加属性リスト情報を入力します。
注: [ターゲット] フィールドに指定する .fcc ファイルが、このフィールドの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。
認証方式が保存され、レルムに割り当て可能になります。

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、基本認証情報（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、HTTP 認証ヘッダを使用して、最初のリクエストと一緒に基本認証情報を送信する必要があります。そうしないと、ポリシー サーバはユーザを認証しません。認証情報が HTTP 認証ヘッダを使用して送信されないと、ポリシー サーバは、ブラウザ以外のクライアントをサポートせずに、HTML フォーム方式にリダイレクトします。

以下の手順に従います。

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。

FCC の国際化を有効にする方法

SiteMinder は、FCC の国際化に対応するために以下の機能をサポートします。

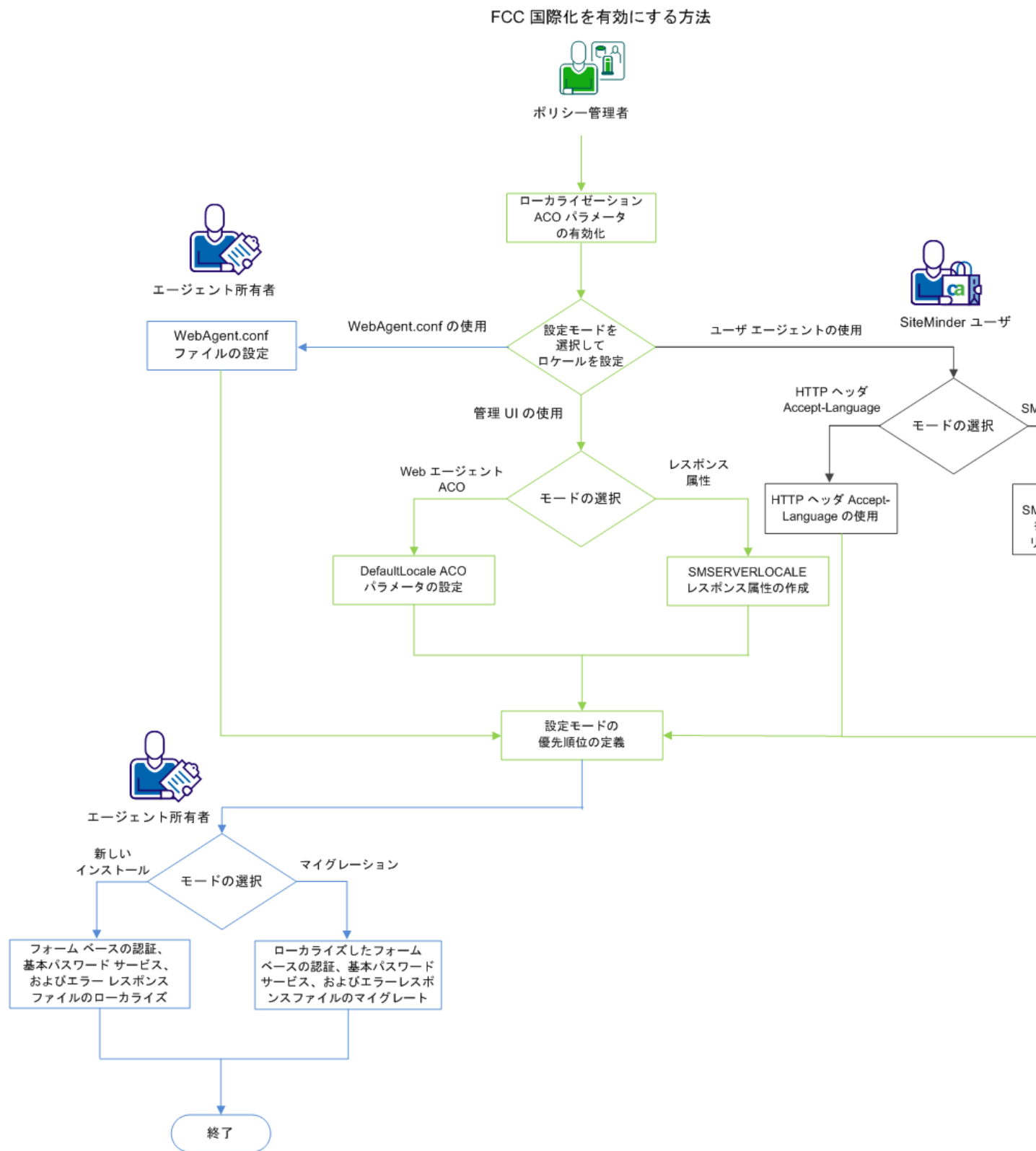
- 認証方式に対する FCC
- 基本的なパスワード サービス
- エラー レスポンス ページ

SiteMinder がログイン、基本的なパスワード サービスおよびエラー レスポンス用に HTML ページを表示する必要がある優先ロケールを設定できます。ユーザの組織にこのシナリオを実装するには、以下のユーザ ロールが必要です。

- ポリシー管理者
- エージェント所有者
- SiteMinder ユーザ

以下の図では、**FCC** の国際化において各ユーザが実行する必要がある手順について説明します。

注: 以下の図で、青色は、ポリシー管理者のタスクを示します。緑色は、エージェント所有者のタスクを示します。黒色は、**SiteMinder** ユーザのタスクを示します。



FCC の国際化を有効にするには、以下の手順を実行する必要があります。

1. [管理 UI を使用して、Localization ACO パラメータを有効にします](#) (P. 343)。ポリシー管理者は、このタスクを実行する必要があります。
2. [優先ロケールを設定するために、設定モードを選択します](#) (P. 343)。このタスクは、以下の方法で実行できます。
 - ユーザ エージェントの使用。SiteMinder ユーザは、ユーザ エージェントを使用して、以下の方法で優先ロケールを設定できます。
 - HTTP ヘッダ SMCLIENTLOCALE を使用します。
 - [HTTP ヘッダ Accept-Language](#) (P. 344) を使用します。
 - [管理 UI の使用](#) (P. 344)。ポリシー管理者は、管理 UI を使用して、以下の方法で優先ロケールを設定できます。
 - [SMSERVERLOCALE レスポンス属性を使用します](#) (P. 344)。
 - [DefaultLocale ACO パラメータを使用します](#) (P. 345)。
 - [the WebAgent.conf ファイルの使用](#) (P. 346)。エージェント所有者は、WebAgent.conf ファイル内の Locale パラメータを設定して、優先ロケールを設定できます。
3. [設定モードの優先順位を定義します](#) (P. 346)。ポリシー管理者は管理 UI を使用して、このタスクを実行する必要があります。
4. 以下の手順のいずれかを実行します。
 - HTML ページを初めてローカライズする場合、エージェント所有者は以下の手順を実行する必要があります。
 - a. [フォーム ベースの認証ファイルおよび基本的なパスワード サービス ファイルをローカライズします](#) (P. 347)。
 - b. [エラー レスポンス ファイルをローカライズします](#) (P. 349)。
 - ローカライズされたファイルを移行する場合、エージェント所有者は以下の手順を実行する必要があります。
 - a. [ローカライズされたフォーム ベースの認証ファイルおよび基本的なパスワード サービス ファイルを移行します](#) (P. 352)。
 - b. [ローカライズされたエラー レスポンス ファイルを移行します](#) (P. 354)。

SiteMinder は、FCC の国際化を有効にするように設定されます。

ローカライゼーション パラメータの有効化

ポリシー管理者は、Localization ACO パラメータを **true** に設定して、サポートされている機能をローカライズする必要があります。

以下の手順に従います。

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] をクリックします。
3. [エージェント設定オブジェクト] をクリックします。
4. **Web** エージェントのエージェント設定オブジェクトに移動します。
5. エージェント設定オブジェクトの名前をクリックします。
6. [変更] をクリックします。
7. **Localization** パラメータに移動し、パラメータの横にある右方向矢印をクリックします。
[パラメータの編集] ページが表示されます。
8. パラメータの値を **yes** に設定します。
9. 変更を保存します。

優先ロケールの設定

Accept-Language HTTP ヘッダ形式で、複数の優先ロケールを定義できます。

例: `es-ES,en-US;q=0.8,as-IN;q=0.6,fr-FR;q=0.4,en-IN;q=0.2`

ロケールは実行時に動的に決定されますが、リクエスト処理中に変更できます。たとえば、ロケールは認証成功後に変わる場合があります。

以下の設定モードで、優先ロケールを定義します。

- ユーザ エージェントの使用。
- 管理 管理 UI の使用。
- WebAgent.conf ファイルの使用

ユーザ エージェントを介した優先ロケールの設定

ユーザは、優先ロケールの **Web** ブラウザ設定を設定できます。ポリシー サーバは、優先ロケールを使用して、サポートされている機能に関連するリクエストを送信します。

ユーザは、以下の方法を使用してロケールを設定できます。

- HTTP ヘッダ **SMCLIENTLOCALE** の使用
- HTTP ヘッダ **Accept-Language** の使用

HTTP ヘッダ **SMCLIENTLOCALE** の使用

Web リクエストの HTTP ヘッダ **SMCLIENTLOCALE** を使用した優先ロケールの設定 **Web** アプリケーションが優先ロケールを設定するオプションを提供する場合、ユーザはロケールを選択できます。ポリシー サーバは **Web** リクエストで **SMCLIENTLOCALE** ヘッダを使用して、ロケールを受信します。ロケールが利用可能な場合、サポートされている機能を示すページが、リクエストされたロケールで表示されます。

HTTP ヘッダ **Accept-Language** の使用

ユーザはローカル **Web** ブラウザ設定を使用して、優先ロケールを設定できます。 **Web** リクエストの HTTP ヘッダ **Accept-Language** はロケールを指定します。

管理 UI による優先ロケールの設定

ポリシー管理者は 管理 UI を使用して、ポリシー サーバがサポートされている機能に関連するリクエストに対応する優先ロケールを設定できます。

SMSEVERLOCALE レスポンス ヘッダ変数の設定

レスポンスとして優先ロケールを設定するには、**SMSEVERLOCALE** レスポンス属性を使用します。

以下の手順に従います。

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] をクリックします。
3. [レスポンス] をクリックします。
4. 変更するレスポンスを編集します。

5. [レスポンスの定義] タブをクリックします。
6. 変更するレスポンス属性を編集します。
7. WebAgent-HTTP-Header 変数として [属性タイプ] を選択します。
8. [変数名] に「SMSERVERLOCALE」と入力します。
9. [変数値] に優先ロケールを入力します。
10. 変更を保存します。

DefaultLocale ACO プロパティの設定

DefaultLocale ACO パラメータで優先ロケール情報を設定します。

以下の手順に従います。

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] をクリックします。
3. [エージェント設定オブジェクト] をクリックします。
4. Web エージェントのエージェント設定オブジェクトに移動します。
5. エージェント設定オブジェクトの名前をクリックします。
6. [変更] をクリックします。

設定とコントロールがアクティブになります。

7. DefaultLocale パラメータに移動し、パラメータの横にある右方向矢印をクリックします。

[パラメータの編集] ページが表示されます。

8. 優先ロケールで、プロパティを更新します。
9. 変更を保存します。

WebAgent.conf ファイルを使用した優先ロケールの設定

エージェント所有者として **Locale** プロパティを設定するには、**WebAgent.conf** ファイルを設定します。このファイルには、インストール中に指定されるロケール情報が含まれています。

以下の手順に従います。

1. Web エージェントマシンにログオンします。
2. WebAgent.conf ファイルをテキスト エディタで開きます。
3. Locale プロパティに移動し、値を優先ロケールに設定します。
4. 変更を保存します。

設定モードの優先順序の定義

5 つの設定モードの優先順位を決定するには、ポリシー管理者は **ClientLocalePreferred ACO** パラメータを設定する必要があります。**ClientLocalePreferred** パラメータが **true** に設定されている場合、システムは設定モードを確認し、以下の順序でロケールを決定します。

1. HTTP ヘッダ **SMCLIENTLOCALE**
2. HTTP ヘッダ **Accept-Language**
3. レスポンス ヘッダ変数 **SMSERVERLOCALE**
4. **DefaultLocale ACO** パラメータ
5. WebAgent.conf ファイル内の **Locale** プロパティ

ClientLocalePreferred が **false** に設定されている場合、システムは設定モードを確認し、以下の順序でロケールを決定します。

1. レスポンス ヘッダ変数 **SMSERVERLOCALE**
2. HTTP ヘッダ **SMCLIENTLOCALE**
3. HTTP ヘッダ **Accept-Language**
4. **DefaultLocale ACO** パラメータ
5. WebAgent.conf ファイル内の **Locale** プロパティ

ポリシー サーバは、定義されたモードを使用して優先ロケールを処理し、優先ロケールのリストを作成します。このリストは、各優先ロケールの **q** ファクタをベースに並べ替えられます。**Web** リクエストが送信されると、このリストにより、サポートされている機能の **HTML** ページが表示される必要があるロケールが示されます。

ロケールの優先順序に合わせて **ClientLocalePreferred** パラメータを設定します。

以下の手順に従います。

1. [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. [エージェント設定オブジェクト] をクリックします。
3. **Web** エージェントのエージェント設定オブジェクトに移動します。
4. エージェント設定オブジェクトの名前をクリックします。
5. [変更] をクリックします。
6. 設定とコントロールがアクティブになります。
7. **ClientLocalePreferred** パラメータに移動します。
[パラメータの編集] ページが表示されます。
8. パラメータ値を変更します。
9. 変更を保存します。

フォーム ベースの認証および基本的なパスワード サービス ファイルのローカライズ

エージェント所有者はフォーム ベースの認証ファイルおよび基本的なパスワード サービス ファイルをローカライズできます。デフォルトでは、さまざまなロケールのサンプル フォームが用意されています。これらのサンプルは、**webagent_home/samples** ディレクトリにあります。

各ロケールには、ロケール固有のファイル用のフォルダがあります。各フォルダおよびファイルの末尾に、命名規則の RFC 3066 標準に準拠するロケール言語コードが、以下の形式で追加されます。

`forms_<ロケール言語コード>`

`forms_<ロケール言語コード>/filename_<ロケール言語コード>.fcc`

例：

日本語の場合、フォルダ名は `samples/forms_ja-JP`、`login.fcc` ファイル名は `login_ja-JP.fcc`、`WebAgent.properties` ファイル名は `WebAGent_ja-JP.properties` です。

サンプル フォーム フォルダ以外に、デフォルト フォーム フォルダを使用できます。このフォルダには、ロケール言語コードのないファイルが含まれています。**Web** エージェントがロケールのリクエストを受け取ると、ポリシー サーバは、優先ロケールのリストと照合して、リクエストされたロケールが利用可能かどうかを確認します。次に、システムは以下のいずれかのタスクを実行します。

- リクエストされたロケールがリスト内のロケールに一致する場合は、リクエストされたロケールでリクエストが処理されます。
- リクエストされたロケールがリスト内のロケールに一致しない場合は、デフォルト フォーム フォルダのロケールを使用してリクエストが処理されます。

新規ロケールをサポートするためのフォルダのカスタマイズ

デフォルト フォーム フォルダは `en-US` にあります。エージェント所有者は、デフォルトのフォルダおよびファイルをカスタマイズして、`en-US` 以外の新規ロケールをサポートする必要があります。

以下の手順に従います。

1. **Web** エージェント マシンにログインします。
2. 以下の場所に移動します。
`webagent_home/samples`
3. 以下の形式で、新規ロケール用のフォルダを作成します。

`forms_<ロケール言語コード>`

4. .fcc ファイルをカスタマイズします。
 - a. 関連する FCC のファイルを、デフォルトのフォーム フォルダから新規フォルダへ配置します。
 - b. 各 FCC ファイルの末尾に、ロケールの言語コードを以下の形式で追加します。
filename_<ロケール言語コード>.fcc
5. プロパティ ファイルをカスタマイズします。
 - a. 関連する .properties ファイルを、デフォルトのフォーム フォルダから新規ロケール フォルダへ配置します。
 - b. 各ファイルの末尾に、新規ロケールの言語コードを以下の形式で追加します。
filename_<ロケール言語コード>.properties
6. 英語のメッセージのユーザの新規ロケール言語への変更など、ロケールに対応するようにファイルを変更します。
7. 変更を保存します。

エラー レスpons ファイルのローカライズ

エージェント所有者は、エラー レスpons ファイルをローカライズできます。以下のディレクトリでは、さまざまなロケールのサンプルエラー レスpons ファイルが提供されます。

webagent_home/samples/error-responses

各ロケールには、ロケール固有のファイル用のフォルダがあります。各フォルダおよびファイルの末尾に、命名規則の RFC 3066 標準に準拠するロケール言語コードが、以下の形式で追加されます。

responses_<ロケール言語コード>

filename_<ロケール言語コード>.err

サンプルエラー レスpons フォルダに加えて、ロケール言語コードが含まれないデフォルトエラー レスpons フォルダおよびファイルがあります。Web エージェントがロケールでリクエストを受け取ると、ポリシーサーバは以下のタスクのいずれかを実行します。

- ロケールが利用可能で、エラーが発生する場合は、対応するロケールフォルダにあるエラー レスpons ファイルを使用して、エラー レスpons が表示されます。
- ロケールが利用可能でなく、エラーが発生する場合は、デフォルトエラー レスpons フォルダにあるファイルを使用して、エラー レスpons が表示されます。

新規ロケールをサポートするための SiteMinder のカスタマイズ

デフォルトのフォルダおよびファイルをカスタマイズして、デフォルトでは提供されない新規ロケールをサポートします。

以下のエラー レスpons ファイルは、ユーザの優先ロケールに適するようにカスタマイズできます。

- CSS エラーに対して生成される `csserror.err`
- サーバエラーに対して生成される `servererror.err`
- Cookie エラーに対して生成される `cookieerror.err`
- 認証および許可エラーに対して生成される `custom401error.err`

エラーが発生する場合、エラー ページは以下の場所から取得されます。

`ErrorResponseLocation_home/responses_<ロケール言語コード>`

例：

英語ロケールのエラー レスpons ファイルは、`ErrorResponseLocation_home/responses_en-US` フォルダに格納されます。Cookie エラーが発生する場合は、このフォルダから `cookieerror_en-US.err` エラー レスpons ファイルが取得されます。

エラー レスpons をカスタマイズするには、以下の手順を実行します。

1. Web エージェントを設定する。
2. Web エージェント ACO を設定する。

Web エージェントの設定

エージェント所有者は、Web エージェントを設定してエラー レスpons ファイルをローカライズする必要があります。

以下の手順に従います。

1. Web エージェント マシンにログオンします。
2. 以下の場所に移動します。

`webagent_home/samples/error-responses`

3. 以下の形式で、新規ロケール用のフォルダを作成します。

`responses_<ロケール言語コード>`

4. レスpons ファイルを、デフォルトのレスpons フォルダから新規フォルダへ配置します。
5. 各エラー レスpons ファイルの末尾に、新規ロケールの言語コードを以下の形式で追加します。

`filename_<ロケール言語コード>`

6. 新規ロケールに対応するようにファイルを変更します。たとえば、英語のメッセージを新規ロケールの言語に変更します。

Web エージェント ACO の設定

エラー レスpons ファイルをローカライズするには、ポリシー管理者はエージェントの ACO パラメータを設定する必要があります。

以下の手順に従います。

1. 管理 UI にログオンします。
2. [インフラストラクチャ] - [エージェント] をクリックします。
3. [エージェント設定オブジェクト] をクリックします。
4. Web エージェントのエージェント設定オブジェクトに移動します。
5. エージェント設定オブジェクトの名前をクリックします。

6. [変更] をクリックします。
7. 以下のパラメータの値が **no** に設定されていることを確認します。

- `servererrorfile`
- `csserrorfile`
- `reqcookieerrorfile`
- `custom401errorfile`

注: これらのプロパティの優先度は **ErrorResponseLocation** パラメータより上です。エラー レスポンス ファイルをローカライズするには、これらのプロパティが無効であることを確認します。

8. **ErrorResponseLocation** パラメータに移動し、パラメータの横にある右方向矢印をクリックします。

[パラメータの編集] ページが表示されます。

9. エラー レスポンス ファイルの新しいパスを入力します。

注: **ErrorResponseLocation** プロパティの値は **URL** にはできません。

10. 変更を保存します。

ローカライズされたファイルの移行

12.51 より前のリリースでローカライズされた **FCC** ファイル、基本的なパスワード サービス ファイル、およびエラー レスポンス ファイルが利用可能な場合、エージェント所有者は、それらを 12.51 以降のリリースに移行して、再利用できます。

フォーム ベースの認証ファイルおよび基本的なパスワード サービス ファイルの移行

エージェント所有者はフォーム ベースの認証ファイルおよび基本的なパスワード サービス ファイルを移行できます。

以下の手順に従います。

1. Web エージェント マシンにログオンします。
2. ローカライズされたファイルが存在する場所に移動します。

デフォルト パス : `webagent_home/samples`

3. フォルダを作成するか、または次の手順にスキップし、既存のローケルフォルダの名前を変更します。これらの手順のいずれかのみを実行します。

- a. 以下の形式でフォルダを作成します。

`forms_<ローケル言語コード>`

- b. 既存のローカライズされた FCC ファイルを新しいフォルダに配置します。

- c. 各ファイルの末尾に、ユーザのローケルの言語コードを以下の形式で追加します。

`filename_<ローケル言語コード>.fcc`

- d. 関連する `.properties` ファイルを、既存のフォルダから新規ローケルのフォームフォルダに配置します。

- e. 各ファイルの末尾に、ユーザのローケルの言語コードを以下の形式で追加します。

`filename_<ローケル言語コード>.properties`

4. (オプション) 既存のフォルダの名前を変更します (これは、フォルダ作成に代わる手順です)。

- a. 以下の形式で、既存フォルダの名前を変更します。

`forms_<ローケル言語コード>`

- b. ユーザのローケルの言語コードを使用して、各ファイルの名前を以下の形式で変更します。

`filename_<ローケル言語コード>.fcc`

- c. ユーザのローケルの言語コードを使用して、各ファイルの名前を以下の形式で変更します。

`filename_<ローケル言語コード>.properties`

5. FCC ファイルで、以下の手順に従います。
 - a. ハードコードされたエンコーディング パラメータを `$$$MENC$$` タグに置換します。
例：
 - `<meta http-equiv="Content-Type" content="text/html; charset=$$$MENC$$">`
 - `<INPUT TYPE=HIDDEN NAME="SMENC" VALUE="$$$MENC$$">`
 - b. 各ファイルの最初の行には、フォームのエンコードに使用される文字セットが含まれる必要があります。例：`<!-- SiteMinder Encoding=UTF-8; -->`。この例では、**UTF-8** がフォームのエンコードに使用されます。
6. 変更を保存します。

エラー レスpons ファイルの移行

エージェント所有者は、ローカライズされたエラー レスpons ファイルを移行できます。

以下の手順に従います。

1. Web エージェント マシンにログインします。
2. ローカライズされたファイルが存在する場所に移動します。
デフォルトパス：`webagent_home/samples/error-responses`
3. フォルダを作成するには、以下の手順に従います。必要に応じて、次の手順にスキップして既存のフォルダの名前を変更します。
 - a. 以下の形式でフォルダを作成します。
`responses_<ロケール言語コード>`
 - b. ローカライズされた既存のエラー レスpons ファイルを新しいフォルダに配置します。
 - c. 各ファイルの末尾に、ユーザのロケールの言語コードを以下の形式で追加します。

`filename_<ロケール言語コード>`

4. (オプション) 既存のフォルダの名前を変更する方法：

- a. 以下の形式で、既存ファイルの名前を変更します。

`responses_<ロケール言語コード>`

- b. ユーザのロケールの言語コードを使用して、各ファイルの名前を以下の形式で変更します。

`filename_<ロケール言語コード>`

これで、FCC の国際化が完了しました。

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。ポリシー サーバで使用可能な Windows 認証方式では、Microsoft 統合 Windows 認証インフラストラクチャが取得するユーザ認証情報を処理することにより、リソースを保護します。

本製品の旧バージョンでは、NTLM 認証方式を使用して Windows 認証をサポートしていました。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、Active Directory がネイティブ モードで実行されている展開においてアクセス制御を使用できます。この方式では、NTLM 認証をサポートするように設定された Active Directory もサポートされます。Windows 認証方式は、本製品の NTLM 認証方式に取って代わるものです。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方が適しています。

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します。この場合、ユーザは Internet Explorer Web ブラウザを使用してリソースにアクセスします。この方式では、ユーザの認証情報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。

詳細情報:

[Windows ユーザ セキュリティ コンテキストの要件 \(P. 109\)](#)

Windows 認証の前提条件が満たされていることを確認してください。

Windows 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。
 - リクエストされたリソースは、任意のタイプの Web サーバに置くことができること。ただし、それらのリソースを保護する認証サーバおよび Web エージェントは、Microsoft IIS Web サーバの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。
 - リクエストされたリソースは、任意のタイプの Web サーバに置くことができること。ただし、それらのリソースを保護する認証サーバおよび Web エージェントは、Microsoft IIS Web サーバの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、Windows 認証情報を送信することをサポートし、認証情報を自動的に送信するように正しく設定されているブラウザを使用する必要があります。

- Windows 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 オプションで、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。

Windows 認証方式の設定

Windows 認証方式を使用して、Windows 環境にあるユーザを認証します。

注: 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認証情報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。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。
5. [認証方式タイプ] リストから [Windows 認証テンプレート] を選択します。

6. サーバ名、ターゲットおよびユーザ **DN** 情報を入力します。NT チャレンジ/レスポンス認証を必要とする環境の場合は、エージェントの所有者から以下の値を取得します。

Server Name

IIS Web サーバの完全修飾ドメイン名。たとえば次のようになります。

server1.myorg.com

ターゲット

/siteminderagent/ntlm/smntlm.ntc

注: このディレクトリは、インストール時にすでに設定された仮想ディレクトリと一致している必要があります。ターゲットである **smntlm.ntc** は、存在していなくてもかまいません。また、**.ntc** で終わる名前や、デフォルトの代わりに使用するカスタム **MIME** タイプでもかまいません。

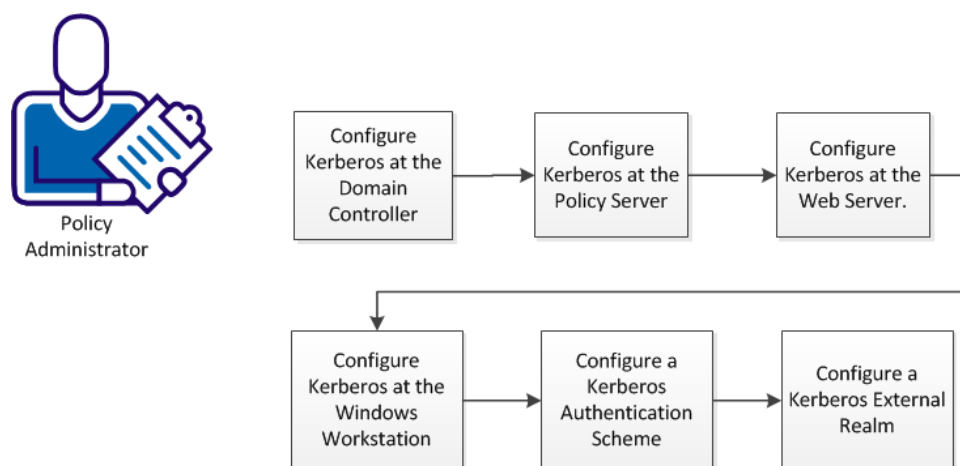
ライブラリ

smauthntlm

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。

Kerberos 認証の設定方法

Kerberos 認証は、クライアントとサーバのホスト環境に応じて、さまざまな設定シナリオをサポートします。各シナリオはわずかに異なりますが、SiteMinder 環境で Kerberos 認証を実装するには、ポリシー管理者が以下の図に表されたタスクを実行する必要があります。



Kerberos 認証を設定するために以下の手順を完了します。

1. [ドメインコントローラで Kerberos を設定します](#) (P. 359)。
2. [ポリシーサーバで Kerberos を設定します](#) (P. 366)。
3. [Web サーバで Kerberos を設定します](#) (P. 369)。
4. [Windows ワークステーションで Kerberos を設定します。](#) (P. 369)
5. [Kerberos 認証方式の設定](#) (P. 370)
6. Kerberos 外部レルムの設定

ドメインコントローラでの Kerberos 認証の設定

純粋な Windows 環境では、Kerberos レルムは Windows ドメインに相当します。ドメインコントローラホストは、ユーザ、サービスアカウント、認証情報、Kerberos チケットサービス、および Windows ドメインサービスのためのストレージを提供します。

keytab ファイルは Kerberos 認証に必要です。それによりユーザはパスワードを要求されることなく KDC により認証できます。**keytab** ファイルは **ktpass** ユーティリティで作成されます。**ktpass** コマンドツールユーティリティは Windows サポート ツールです。UNIX 上で相当するのが **ktadd** ユーティリティです。

ドメインコントローラ ホスト (Windows または UNIX) 上でポリシー サーバに対して KDC を設定するには、以下の一般的な順序に従います。

1. ユーザアカウントを作成します。このアカウントは、ワークステーションにログインするためのものです。
2. Web サーバホストにログインするための Web サーバ用のサービスアカウントを作成します。
3. ポリシーサーバホストにログインするためのポリシーサーバ用のサービスアカウントを作成します。
4. Web サーバアカウントを Web サーバプリンシパル名と関連付けます。
5. **keytab** ファイルを作成します。そのファイルは Web サーバホストに転送されます。
6. ポリシーサーバアカウントをポリシーサーバプリンシパル名と関連付けます。
7. 別の **keytab** ファイルを作成します。そのファイルはポリシーサーバホストに転送されます。
8. Web サーバおよびポリシーサーバアカウントが委任用のトラステッドアカウントであることを指定します。

重要: Kerberos プロトコルを使用するサービスについては、必ず標準形式 (すなわち **service/fqdn_host@REALM_NAME**) でサービスプリンシパル名 (SPN) を作成してください。

詳細情報:

[Windows での KDC 設定の例](#) (P. 361)

[UNIX 上での KDC 設定の例](#) (P. 364)

Windows での KDC 設定の例

以下の手順では、Kerberos 認証をサポートする Windows ドメイン コントローラ/KDC を設定する方法を示します。この例では、以下のサーバおよびアカウント名を使用します。

- サーバ
ドメイン コントローラ : kdc.example.com
ポリシー サーバ : smps.example.com
Web サーバ : www.example.com
Active Directory ドメイン : EXAMPLE
Kerberos レルム : EXAMPLE.COM
- アカウント
ユーザ アカウント : testkrb
Web エージェント サービス アカウント : krbsvc-smwa
ポリシー サーバ サービス アカウント : krbsvc-smps

以下の手順に従います。

1. Active Directory ドメイン サービス ロールをお使いの Windows サーバに追加します。

このアクションにより、お使いの Windows サーバがドメイン コントローラになります。Windows ドメイン コントローラは、Kerberos Key Distribution Center (KDC) として機能し、Kerberos V5 プロトコルを実装し、Kerberos 関連の認証機能を実行します。
2. コマンドラインから Windows dcpromo ユーティリティを実行するか、または Service Manager 上で dcpromo ユーティリティを起動するリンクをクリックします。
3. ドメイン機能レベルの確認
 - a. 管理ツールから [Active Directory ユーザとコンピュータ] ダイアログ ボックスを開きます。
 - b. ダイアログ ボックスの左側にある example.com ドロップダウンを右クリックします。
 - c. [ドメイン機能レベルの昇格] をクリックします。
 - d. Active Directory のドメインの機能レベルが少なくとも Windows Server 2003 であることを確認します。そうでない場合は、ドメインの機能レベルを上げます。

重要: ドメインの機能レベルを上げた場合は戻せません。

4. ユーザアカウントを作成します (例: testkrb)。このアカウントのパスワードを提供します。[ユーザは次回ログオン時にパスワード変更が必要] オプションをオフにします。Windows ワークステーションは、このアカウントを使用して example.com にログインします。
5. Web サーバのサービスアカウントを作成します (例: krbsvc-smwa)。このアカウントのパスワードを作成します。[ユーザは次回ログオン時にパスワード変更が必要] オプションをオフにします。
6. ポリシーサーバのサービスアカウントを作成します (krbsvc-smpps)。このアカウントのパスワードを提供します。[ユーザは次回ログオン時にパスワード変更が必要] オプションをオフにします。

7. Web サーバサービスアカウント (krbsvc-smwa) と Kerberos サービスプリンシパル名 (HTTP/www.example.com@EXAMPLE.COM) を関連付け、Ktpass ユーティリティを使用して keytab ファイルを作成します。サポート ツールのバージョンを常に確認してください。デフォルトの暗号化タイプは RC4-HMAC です。暗号化タイプは、コマンドプロンプトで ktpass /? を実行することにより確認できます。

```
ktpass -out c:\krbsvc-smwa.keytab -princ HTTP/www.example.com@EXAMPLE.COM
dc.example.com -ptype KRB5_NT_PRINCIPAL -mapuser EXAMPLE\krbsvc-smwa -pass "*"
Targeting domain controller: dc.example.com
Using legacy password setting method
Successfully mapped HTTP/www.example.com to krbsvc-smwa..
Key created.
Output keytab to c:\krbsvc-smwa.keytab:
Keytab version: 0x502
keysize 82 dc.example.com ptype 1 (KRB5_NT_PRIN
CIPAL) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xf3d3d2a17c7ea172ff22044806443
9a9)
```

パスワードは、Web サーバ用のサービス アカウントを作成する際に使用されるものと同じです。

8. ポリシー サーバ サービス アカウント (krbsvc-smps) を Kerberos サービス プリンシパル名 (smpls/smpl.example.com@EXAMPLE.COM) に関連付けて、ポリシー サーバ ホストに対して別の keytab ファイルを作成します。

```
Ktpass -out c:\krbsvc-smpls.keytab -princ smpls/smpl.example.com@EXAMPLE.COM  
-ptype KRB5_NT_PRINCIPAL -mapuser EXAMPLE\krbsvc-smpls -pass *  
Targeting domain controller: dc.example.com.com  
Using legacy password setting method  
Successfully mapped smpls/smpl.example.com to krbsvc-smpls.  
Key created.  
Output keytab to c:\krbsvc-smpls.keytab:  
Keytab version: 0x502  
keysize 72 smpls/smpl.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 2  
etype 0x17 (RC4-HMAC) keylength 16 (0xfd77a26f1f5d61d1fafd67a2d88784c7)
```

パスワードは、ポリシー サーバ 用のサービス アカウントを作成する際に使用されるものと同じです。

9. Web サーバ サービス アカウントが、以下のように **Trusted for Delegation** であることを指定します。
 1. サービス アカウント (krbsvc-smwa) プロパティを右クリックします。
 2. [委任] タブを選択します。
 3. 2 番目のオプション、[任意のサービスへの委任でこのユーザーを信頼する (Kerberos のみ)] を選択します。

ドメイン コントローラは Kerberos 認証に対応できるようになりました。

UNIX 上での KDC 設定の例

以下のプロセスでは、Kerberos 認証をサポートするために UNIX ホスト上で KDC Kerberos レルムを設定する方法を例証します。この例では、以下のサーバおよびアカウント名を使用します。

- サーバ
ドメイン コントローラ : dc.example.com
ポリシー サーバ : smps.example.com
Web サーバ : www.example.com
Active Directory ドメイン : EXAMPLE
Kerberos レルム : EXAMPLE.COM
- アカウント
ユーザ アカウント : testkrb
Web エージェント サービス アカウント : krbsvc-smwa
ポリシー サーバ サービス アカウント : krbsvc-smps

以下の手順に従います。

1. 必要に応じ、MIT Kerberos をインストールします。
2. `kdb5_util` コマンドを使用して Kerberos データベースおよびオプションの `stash` ファイルを作成します。`stash` ファイルは、ホスト自動ブートシーケンスの一部として `kadmind` と `krb5kdc` のデーモンを開始する前にそれ自体へ KDC を自動的に認証するために使用されます。

`stash` ファイルおよび `keytab` ファイルはどちらも、潜在的な侵入のエントリーポイントです。`stash` ファイルをインストールする場合、それはルートによってのみ読み取り可能でなければならず、バックアップされてはならないし、KDC ローカルディスクにのみ存在する必要があります。`stash` ファイルを望まない場合は、`-s` オプションなしで `kdb5_util` を実行します。

この例では、`kdc.conf` ファイルで指定されたディレクトリに以下の 5 つのデータベース ファイルを生成します。

- 2 つの Kerberos データベース ファイル： `principal.db` と `principal.ok`
- 1 つの Kerberos の管理データベース ファイル： `principal.kadm5`
- 1 つの管理データベース ロック ファイル： `principal.kadm5.lock`
- 1 つの stash ファイル： `.k5stash`

```
[root@rhasmit init.d]# kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/kerberos/krb5kdc/principal' for realm
'EXAMPLE.COM',
```

マスタ キー名 'K/M@EXAMPLE.COM'

データベース マスタ パスワードを求められます。

このパスワードは重要なので忘れないでください。

KDC データベース マスタ キーを入力します：

KDC データベース マスタ キーを再入力し、次を確認します：

3. ユーザ プリンシパル (`testkrb`) を作成します。
4. Web サーバ ホスト用のユーザ プリンシパル (たとえば `testwakrb`)、
ホスト プリンシパル (`HTTP/www.example.com@EXAMPLE.COM`)、および
サービス プリンシパル (`krbsvc-smwa`) を作成します。ホスト ア
カウントを作成するために使用されるパスワードは、Web サーバ ホ
スト上で `ksetup` ユーティリティを使用する際に指定されるパスワード
と同じである必要があります。
5. ポリシー サーバ ホスト用にユーザ プリンシパル (`testpskrb`)、ホスト
プリンシパル (`smps/smps.example.com`)、およびサービス プリンシ
パル (`krbsvc-smps`) を作成します。ホスト アカウントを作成するために
使用されるパスワードは、ポリシー サーバ ホスト上で `ksetup` ユー
ティリティを使用する際に指定されるパスワードと同じである必要が
あります。
6. Web サーバ サービス プリンシパル用の `keytab` ファイルを以下のよう
に作成します。

```
ktadd -k /tmp/krbsvc-smwa.keytab HTTP/www.example.com
```

7. ポリシー サーバ サービス プリンシパル用の `keytab` を以下のよう
に作成します。

```
ktadd -k /tmp/krbsvc-smps.keytab smps/smps.example.com
```

Kerberos レルムが UNIX ホスト上で設定されます。

ポリシー サーバでの Kerberos 認証の設定

Kerberos 認証方式では、エージェント設定オブジェクト（ACO）を以下のように変更する必要があります。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [エージェント設定] - [エージェント設定の変更] をクリックします。
3. 対象のエージェントのエージェント設定オブジェクトに対して編集ボタンをクリックします。
4. エージェント設定オブジェクトにこれら 3 つのパラメータを追加します。

パラメータ	値
KCCExt	.kcc
HttpServicePrincipal	Web サーバプリンシパル名を指定します。 例： HTTP/www.example.com@EXAMPLE.COM
SmppsServicePrincipal	ポリシー サーバプリンシパル名を指定します。 例： smpps.example.com@EXAMPLE.COM

ポリシー サーバで、以下の追加の手順を完了します。

- ポリシー サーバプリンシパル認証情報が含まれる KDC 上の keytab ファイルをポリシー サーバ上の安全な場所に展開します。
- Kerberos 設定ファイル (krb5.ini) を設定し、Windows 上のシステム ルートパスおよび UNIX 上の KRB5_CONFIG に配置します。

重要：ポリシー サーバが Windows 上にインストールされ、KDC が UNIX 上で展開する場合は、[Ksetup ユーティリティ \(P. 371\)](#)を使用して、ポリシー サーバ ホスト上で追加の設定を実行してください。

詳細情報：

[ポリシー サーバでの Kerberos 設定の例 \(P. 367\)](#)

ポリシー サーバでの Kerberos 設定の例

以下の手順では、Kerberos 認証方式をサポートする Windows 上のポリシー サーバの設定例について説明します。以下の相違点を除いて、Windows 上での設定と UNIX 上での設定は類似しています。

- Windows 上の Kerberos 設定ファイルは C:\windows\krb5.ini. です。UNIX では、KRB5_CONFIG 環境変数が完全修飾パスに設定されている限り、ファイルをどの場所にも置くことができます。
- keytab の場所はシステムによります。たとえば、Windows では C:\windows\krbsvc-smps.keytab、UNIX では /opt/CA/siteminder/krbsvc-smps.keytab です。Kerberos 設定ファイルの default_keytab_name 設定オプションは、ファイルへの完全修飾パスを指定します。
- UNIX の keytab は、サービス アカウントの keytab およびホストの keytab を 1 つの keytab ファイルにマージします。

注: ポリシー サーバが Windows 上にインストールされ、KDC が UNIX 上で展開される場合は、[Ksetup ユーティリティ \(P. 371\)](#)を使用して、ポリシー サーバ ホスト上で必要な他の設定を実行してください。

以下の手順に従います。

1. ポリシー サーバをインストールし設定します。
2. ポリシー ストア ディレクトリ サービスをインストールし設定します。
3. ポリシー サーバを参照するホスト設定オブジェクトを追加します。

- エージェント設定オブジェクトを作成し、これらの新しい3つのパラメータを追加します。

KCCExt

値: .kcc

HttpServicePrincipal

Web サーバプリンシパル名を指定します。

例: HTTP/www.example.com@EXAMPLE.COM

SmgsServicePrincipal

ポリシー サーバプリンシパル名を指定します。

例: smgs@smgs.example.com

- ユーザディレクトリを作成します。
- ユーザディレクトリで、ユーザを作成します (たとえば `testkrb`) 。
- 管理 UI を使用して、新しい認証方式を設定します。「[Kerberos 認証方式の設定](#) (P. 370)」を参照してください。
- ポリシー ドメインを設定します。
- 認証方式を使用して、リソースを保護するためのレلمを追加します。
- ユーザ (`testkrb`) にアクセスを許可するためのルールとポリシーを追加します。
- Kerberos 設定ファイル (`krb5.ini`) を設定し、Windows システムルートパスに置きます。ポリシー サーバプリンシパル認証情報が含まれる Windows 2003 KDC keytab ファイルを使用するよう `krb5.ini` を設定します。以下のサンプル `krb5.ini` を参照してください。

```
[libdefaults]
default_realm = EXAMPLE.COM
default_keytab_name = C:\WINDOWS\krbsvc-smgs.keytab
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
[realms]
EXAMPLE.COM = {
  kdc = dc.example.com:88
  default_domain = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
```

12. ポリシー サーバ プリンシパル認証情報が含まれる Windows KDC keytab ファイルをポリシー サーバ上の安全な場所に展開します。

Windows ホスト上のポリシー サーバが Kerberos 認証に対して設定されます。

Web サーバでの Kerberos 認証設定

Kerberos 認証をサポートする Windows または UNIX の Web サーバの設定は、これらの一般的な手順に従います。

1. Kerberos 認証方式サポートで Web エージェントをインストールします。
2. トラストドホストをポリシー サーバに登録し、Web エージェントを設定します。
3. Web サーバ認証情報が含まれる keytab ファイル（KDC 上で作成されたファイル）を Web サーバ上の安全な場所に展開します。
4. Kerberos 設定ファイル（krb5.ini）を設定します。
 - Kerberos レルム（ドメイン）に対する KDC を設定します。
 - Web サーバプリンシパルの認証情報が含まれる keytab ファイルを使用するために krb5.ini を設定します。
 - Windows 上のシステムルートパスおよび UNIX 上の KRB5_CONFIG に krb5.ini を配置します。

重要：Windows に Web サーバがインストールされ、KDC が UNIX 上で展開する場合は、[Ksetup ユーティリティ](#) (P. 371)を使用して、Web サーバ上で追加の設定を実行してください。

Windows ワークステーションでの Kerberos 認証の設定

Kerberos 認証をサポートするには、いくつかの Internet Explorer 設定が必要です。また、ワークステーションホストは KDC ドメインに追加されます。

重要：KDC が UNIX 上で展開する場合は、[Ksetup ユーティリティ](#) (P. 371)を使用して、ワークステーション上で追加の必要な設定を実行してください。

以下の手順に従います。

1. Windows ワークステーションのホストを KDC ドメインに追加します。
2. KDC で作成されたユーザアカウントを使用して、ホストにログインします。
3. 認証情報を自動的に渡すようブラウザを設定します。

Kerberos 認証方式の設定

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。この認証方式の使用は、動作する Kerberos が設定されていることを前提とします。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前を入力します。
5. 保護レベルを入力します。デフォルトは 5 です。
6. (オプション) この認証方式に対してパスワードポリシーを有効にします。
7. [認証方式のタイプ] リストから [Kerberos 認証テンプレート] を選択します。
8. サーバ名、ターゲット、プリンシパル名、およびユーザ DN 情報を入力します。これらのフィールドの説明については、ヘルプを参照してください。

9. Kerberos レalmと Windows ドメイン間のマッピングを指定します。複数の Windows ドメインに 1 つの Kerberos レalmをマップできます。

注: ユーザが [ユーザ DN の検索] フィールドで DOMAIN 変数を指定する場合、レalmとドメイン名が同じであっても、このマッピングは必要です。

10. [サブミット] をクリックします。

Kerberos 認証方式が保存されます。方式をレalmに割り当てます。

Windows ホスト上での Kerberos 外部レalmの設定

Windows ワークステーションが UNIX 上で展開した Kerberos KDC を使用するには、Kerberos KDC サーバおよびワークステーションの両方を設定する必要があります。

Kerberos レalmで、Windows ホストのホスト プリンシパルを作成します。次のコマンドを使用します。

```
kadmin.local: addprinc host/machine-name.dns-domain_name.
```

たとえば、Windows ワークステーション名が W2KW で、Kerberos レalm名が EXAMPLE.COM である場合、プリンシパル名は host/w2kw.example.com です。

Kerberos レalmが Windows ドメインではないので、KDC オペレーティング環境はワークグループのメンバとして設定される必要があります、このプロセスに従うことで自動的に発生します。

1. Windows ドメインからホストを削除します。
2. テストユーザ（たとえば testkrb）をローカルユーザデータベースに追加します。

3. Kerberos レalmを追加します。

```
ksetup /SetRealm EXAMPLE.COM
```

4. ホストを再起動します。

5. KDC を追加します。

```
ksetup /addkdc EXAMPLE.COM rhasmit
```

6. 新しいパスワードを設定します。

```
ksetup /setmachpassword password
```

注: ここで使用されるパスワードは、MIT KDC でホスト プリンシパル アカウントを作成する際に使用されるものと同じです。

7. ホストを再起動します。

注: 外部 KDC およびレルム設定に対する変更が行われた場合は常に、再起動が必要です。

8. レルム フラグを設定します。

```
ksetup /SetRealmFlags EXAMPLE.COM delegate
```

9. AddKpasswd の実行

```
ksetup /AddKpasswd EXAMPLE.COM rhasmit
```

10. Windows ホスト アカウント間のアカウント マッピングを Kerberos プリンシパルへ定義することにより、ローカル ワークステーション アカウントへのシングル サインオンを設定するために Ksetup を使用します。例:

```
ksetup /mapuser testkrb@EXAMPLE.COM testkrb  
ksetup /mapuser * *
```

2 番目のコマンドは、クライアントを同じ名前のローカルアカウントにマップします。引数のない Ksetup を使用して現在の設定を参照します。

SiteMinder Kerberos 認証のトラブルシューティング

Kerberos 認証と連動するときは以下の点に注意してください。

- ポリシー サーバのシステム クロックは、KDC システム クロックに同期する必要があります (2 分以内)。そうでないと、Kerberos 認証はクロック スキュー エラーのために失敗します。
- ホストはすべて、`/etc/hosts` ファイル内の IP アドレス、完全修飾ドメイン名 (FQDN) およびホスト名に対し適切なエントリを有している必要があります。これらのエントリの順序も重要になる場合があります。また、これらのエントリ間に少なくとも 1 つのスペースを含めます。

IP アドレス FQDN ホスト名

- **klist** (UNIX KDC クライアント) を使用して、認証情報キャッシュ内にあるチケットをリスト表示します。フラグ **-f** はチケット フラグに関する情報を提供します。これらのチケットは **Kerberos** 認証処理を完了するために転送されます。

たとえば、以下に示すサンプル **klist** 出力をご覧ください。

```
bash-2.05$ klist
チケット キャッシュ: /tmp/krb5cc_1002
デフォルト プリンシパル: HTTP/sol8sunone.test.com@TEST.COM

有効な開始日時          有効期限          サービス プリンシパル
2007 年 12 月 26 日水曜日 15:00:03  2007 年 12 月 26 日水曜日 21:40:03
krbtgt/TEST.COM@TEST.COM
```

- **MIT KDC** ホストおよび **UNIX KDC** クライアントでユーザおよびサービス プリンシパル向けの **kinit** プログラムを常に使用します。

ユーザ プリンシパル向けの **kinit** 構文は次のとおりです。

```
kinit principalname
```

このコマンドはユーザ プリンシパルを作成する際に使用されるパスワードの入力

を要求します。

サービス プリンシパル向けの **kinit** 構文 (ホストからホストに変わる場合があります) は次のとおりです。

```
kinit -k [-t keytab location containing service principal] principalname
```

kinit コマンドはパスワードの入力を求めません。それは、**keytab** ファイルを使用してサービス プリンシパルを認証するためです。

- 任意のネットワーク パケット トレース ユーティリティをワークステーションにインストールして、**Kerberos** トークンがブラウザと **Web** サーバ間で交換されたことを確認します。**TIR** で始まるトークンは **NTLM** トークンを示します。また、**YII** で始まるトークンは **Kerberos** トークンを示します。
- **KDC** で暗号化タイプの変更を行った後にワークステーション ホストからログオフします。また、**Kerberos** 設定または **keytab** ファイルに対し変更を行った後は、ポリシー サーバおよび **Web** サーバサービスを再起動します。
- **KDC** は、先日付チケットをサポートしません。
- **DES-CBC-MD5** および **DES-CBC-CRC** 暗号化タイプのみが **MIT** の相互運用に利用可能です。

- 認証エラー メッセージを記録するために、ポリシー サーバおよび Web エージェント ログは常に有効にします。
- 任意のホスト向けの **keytab** ファイルの名前および場所が、**krb5.conf** ファイルの名前および場所と一致することを確認してください。
- **SPN** は、常に大文字と小文字が区別されます。ポリシー サーバおよび Web サーバ ホスト向けの **krb5.keytab** ファイルには、ホストプリンシパルおよびサービス プリンシパルが含まれる必要があります。
- **ktpass** ユーティリティのバージョンを確認してください。 **ktpass** コマンド ツール ユーティリティは **Windows** サポート ツールに含まれており、**MSDN** ダウンロードからインストールできます。
- **keytab** ファイルを作成する際に使用される暗号化タイプを常に確認してください。デフォルトでは **Windows** は **RC4-HMAC** 暗号化をサポートします。
- **Kerberos** バージョン番号 (**kvno**) を確認します。サービス プリンシパルの **kvno** は、その **keytab** ファイルの **kvno** に一致する必要があります。作成されると、任意の **keytab** の **kvno** 番号が表示されます。

Windows Active Directory のサービス アカウントの **kvno** を決定するには、以下のように **ADSI Edit** を使用します。

1. コマンドプロンプトから **adsiedit.msc** を実行します。
2. 左のドロップダウン メイン (**fqdn_ADhost**) 下の **CN=Users, DC=domain, DC=com** 下で、サービス アカウントに移動します。
3. サービス アカウントを右クリックして、プロパティをクリックします。
4. **msDs-KeyVersionNumber** 属性の値が、サービス アカウント向けの **keytab** ファイルを作成する際に表示されたものと一致することを確認します。

任意のユーザ アカウント向けの **kvno** は、パスワードが変更されるごとに変わります。バージョン番号が一致しない場合は、アカウントと **keytab** を作成するか、またはパスワードを変更して **kvno** を **keytab** の **kvno** と一致させます。

UNIX MIT KDC のサービス アカウントの **kvno** を決定するには、**kvno** ユーティリティを使用します。このコマンドの構文は以下のとおりです。

Kvno principalname

- keytab ファイルの有効性を確認します。

Windows : ポリシー サーバおよび Web エージェントに Windows サポート ツールをインストールして、keytab ファイルが有効であるかどうかを確認します。以下のコマンドを実行します。

```
Kinit -k -t <keytab file location> <respective spn>
```

例 :

```
kinit -k -t C:\Windows\Webserver.keytab HTTP/websvriis6.test.com@KRISH.COM
```

keytab ファイルが有効なときは、このコマンドはエラーを返しません。

Solaris : 以下のコマンドを実行して keytab ファイル両方の有効性を確認します。

```
Kinit -k -t <keytab file> <SPN>
```

例 :

```
kinit -k -t host.keytab host/<fqdn>@DOMAIN.COM
```

```
kinit -k -t smps.keytab smps/<fqdn>@DOMAIN.COM
```

- エラーが発生しない場合は、keytab ファイルは問題なく、krb.conf ファイルは有効な値を持っています。
- エラーが発生する場合は、以下のコマンドを使用して SPN が KDC 内で有効かどうかを確認します。

```
Kinit host/<fqdn>@DOMAIN.COM
```

このコマンドは通常パスワードを要求します。有効なパスワードを提供すると、エラーメッセージは発生しません。このコマンドがパスワードを要求しない場合、SPN は識別されませんでした。そのオブジェクトのプロパティを確認します。SPN エントリは [アカウント] タブにあります (たとえば、*host/fqdn*)。SPN と同じ エントリ セットを持つ他のオブジェクトがないことを確認します。

Information Card 認証方式

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、運転免許に相当する情報カードには、写真、生年月日、氏名、免許番号といった重要な個人情報を含めることができます。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。

アイデンティティ セレクタは情報カードを表示するアプリケーションです。

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、依存パーティ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。依存パーティ (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、ユーザが依存パーティ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、不正確であったり、誤用による脆弱性があったりするユーザ識別情報のデータベースを保守する必要がなくなります。ユーザおよびデータベースの保守が不要になることにより、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、ユーザは各依存パーティにリリースする識別情報を制御できます。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えています。

Windows CardSpace

Windows CardSpace は、アイデンティティ セレクタの Microsoft による実装です。Windows CardSpace は、任意の依存パーティ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。ポリシー サーバは、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。

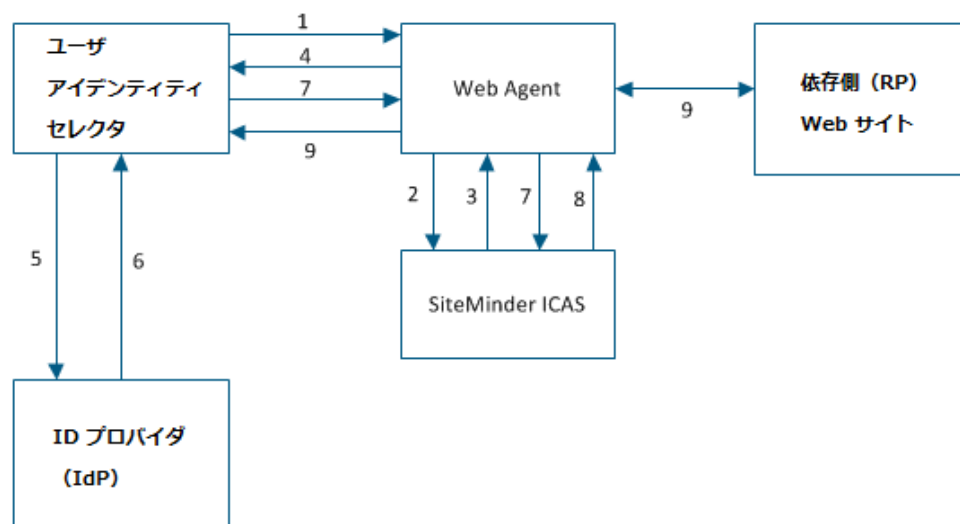
SiteMinder Information Card 認証方式 (ICAS)

Information Card 認証方式 (ICAS) は、Windows CardSpace をサポートするポリシー サーバ上で使用できます。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、他のカスタム認証方式と同様に実装します。

ICAS の概要

ICAS によるユーザ認証には、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
- 依存パーティ (RP)
- アイデンティティ プロバイダ (IdP)



1. ユーザが保護された **Web** サイト、または依存パーティ (**RP**) を訪問しようとしています。
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、ユーザのコンピュータ上でアイデンティティセクタを起動するようにブラウザに指示し、**RP** のポリシー要件を送信します。
5. アイデンティティセクタ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クタはユーザの認証情報を収集し、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クタは、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。

6. **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クタに送り返します。
7. アイデンティティセクタ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。
8. **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、ポリシーサーバ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。
9. 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、依存パーティ、アイデンティティ プロバイダで識別情報を共有す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。

依存パーティ (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依存パーティ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 プロバイダが使用する技術。セキュリティ トークン サービスでは以下を実行します。

- ユーザを認証します。
- 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、これらは依存パーティ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: SAML 1.0 および 1.1 仕様がサポートされています。

- 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。ユーザまたは第三者のアイデンティティ プロバイダはクレームを作成できます。

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、免許番号、状態、身長、性別といった重要な個人情報を含めることができます。

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、第三者は、このクレームを確認しません。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 へのポイントが含まれます。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、依存パーティ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 プロバイダや依存パーティに一貫したユーザ 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

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 ファイルが必要です。

filename.fcc

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

以下のいずれかの Web ブラウザを使用します。

- 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。
- 証明書データ ストアの設定。
- 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。

以下の手順に従います。

1. <http://www.oracle.com/technetwork/java/index.html> から Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction ポリシー ファイルをダウンロードします。
2. ダウンロード ファイルを `$NETE_JRE_ROOT¥lib¥security` ディレクトリ にインストールします。
3. `$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。

1. ICAS プロパティ ファイルを設定します。
2. 後からアクティブ レスポンスで使用するクレームを保存します。
3. ICAS 用の証明書データ ストアを設定します。
4. ICAS 用のユーザ ディレクトリを設定します。
5. ICAS のインスタンスを作成します。
6. 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。
7. ICAM ホワイト リストのサポートを設定します。

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 ファイルを使用できます。

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

情報カード 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_attribute

VPPID 属性値を指定します。ユーザ属性を更新するために、コマンド UpdateUserStoreWithVPPID には、この属性が必要です。

例：vppid_attribute=mail

vppid_claim

ユーザを判別するために必要な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

エイリアス

依存パーティの SSL 証明書を取得するために使用される証明書データストアでキーを指定します。

例：alias=rpssl

guestuser

ICAS が匿名のモードで設定される場合、この設定によりゲストユーザログインを指定します。

例：guestuser=guest

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

storeclaims_attribute

コマンド *StoreClaimsToUserStore* が実行されるとき、クレームが格納される場所にユーザ属性を指定します。

例：storeclaims_attribute=description

preprocessingchain

ユーザの判別中に実行するコマンドのチェーンを定義します。

例：preprocessingchain=+vppidchain

注：詳細については、「[ICAS コマンドチェーンの設定 \(P. 386\)](#)」を参照してください。

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。

例：

postprocessingchain=+vppidchain;com.ca.sm.icas.command.StoreClaimsToContext

注：詳細については、「[ICAS コマンドチェーンの設定 \(P. 386\)](#)」を参照してください。

errorprocessingchain

エラー処理中に実行するコマンドのチェーンを定義します。

注：詳細については、「[ICAS コマンドチェーンの設定 \(P. 386\)](#)」を参照してください。

whiteListLocation

(オプション) 連邦政府の ICAM (Identity, Credentialing, and Access Management) によって指定されたホワイトリストが含まれる XML ファイルの場所を定義します。

例：whiteListLocation=C:\Program Files\ca\siteminder\config\WhiteList.xml

注：ICAM ホワイトリストの詳細については、「[ICAM ホワイトリストのサポートの設定 \(P. 396\)](#)」を参照してください。

ICAS コマンド チェーンの設定

コマンドチェーンは **ICAS** プロパティ ファイルで定義されます。チェーンとは、通常の方法で実行されるコマンドのセットです。

コマンドの名前付きチェーンには次のものが含まれます。

- 前処理
- 後処理
- エラー処理

これらの名前付きチェーンは、デフォルトで存在します。

以下の点を考慮します。

- 特定の名前付きチェーンが **ICAS** プロパティ ファイルで見つかる場合、**ICAS** はこれらの名前付きチェーンを使用します。
- 独自のチェーンを定義し **ICAS** プロパティ ファイルにそれらを配置し、名前付きチェーンでそれらを参照できます。
- いずれかのカスタマイズされたチェーンがこれらの名前付きコマンドチェーンのどれでも参照されていない場合、そのチェーン実行が無視されます。

ユーザの要件に従って、以下のクラス ファイルを設定します。

- [ICAS を設定するキー クラス ファイル](#) (P. 387)
- [クレームの値を確認するクラス ファイル](#) (P. 388)
- [デバッグするクラス ファイル](#) (P. 389)

ICAS を設定するキー クラス ファイル

以下のキー クラス ファイルが ICAS を設定するために使用されます。

`com.ca.sm.icas.StoreClaimsToSessionStore`

ポリシー サーバのセッション ストアに抽出されたクレームを保存します。

`com.ca.sm.icas.command.ExecuteClaimChain`

クレームに関する一連のコマンドを実行します。ネームスペースの最後の部分を形成する各クレーム名については、ICAS はプロパティ ファイルで `chainID.claimID` として定義されたチェーンを探します。チェーン定義が見つかる場合、ICAS はそれを実行し、特定のクレームを持ったコンテキストを渡します。

例：`executeClaimCommand1=com.ca.sm.icas.command.ExecuteClaimChain` の場合、以下のように電子メールのクレームを定義します。

`executeClaimCommand1.emailaddress=com.ca.sm.icas.command.DumpClaimsCommand;`

トークンが `emailaddress` クレームで見つかる場合、`DumpClaimsCommand` を処理します。

`com.ca.sm.icas.command.HashVPPID`

VPPID クレーム属性のハッシュを生成します。

`com.ca.sm.icas.command.SetVPPIDAsAnonymous`

VPPID クレーム属性値を匿名に設定します。このクラス ファイルは、どんなユーザ アカウント（匿名アクセス）にもリンクされない情報カードでログインする場合に使用されます。

`com.ca.sm.icas.command.SetVPPIDFromToken`

VPPID クレーム属性値を情報カード トークンからコンテキスト オブジェクトに設定します。

`com.ca.sm.icas.command.StopChain`

現在実行中のチェーンの実行を停止する値 `true` を返します。

`com.ca.sm.icas.command.StoreClaimsToContext`

情報カード トークンからクレームを読み取り、アプリケーション固有のコンテキストにそれらを格納します。

`com.ca.sm.icas.command.StoreClaimsToUserStore`

情報カード トークンから読み取ったクレームをユーザ ディレクトリ
の指定されたユーザ属性 (ICAS プロパティ ファイル内) に格納します。

`com.ca.sm.icas.command.TransformScriptInClaim`

HTML エンティティを使用するクレームでスクリプト タグ (<、>) を
エスケープします。たとえば、<td> は <td;> になります。

`com.ca.sm.icas.command.TransformSqlInClaim`

SQL クエリにそれを渡すことができるようにクレーム値内の文字をエ
スケープします。

`com.ca.sm.icas.command.TranslateErrorCode`

コンテキスト内で見つかる例外文字列を変換します。この変換された
文字列は、FCC で利用可能な `context.userText` に戻して格納されます。

`com.ca.sm.icas.command.TranslateErrorCodeFromParms`

ICAS プロパティ ファイル内で見つかる例外文字列を変換します。この
変換された文字列は、FCC で利用可能な `context.userText` に戻して格納
されます。

`com.ca.sm.icas.command.UpdateUserStoreWithVPPID`

情報カード トークンから読み取ったクレームをユーザ ディレクトリ
から指定されたユーザ属性 (ICAS プロパティ ファイル内で使用可能)
に格納します。

クレームの値を確認するクラス ファイル

以下のクラス ファイルがクレームの値を確認するために使用されます。

`com.ca.sm.icas.StoreClaimsToSessionStore`

ポリシー サーバのセッション ストアに抽出されたクレームを保存し
ます。

`com.ca.sm.icas.command.ErrorIfEmptyClaim`

クレームが空のとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfMultiValueClaim`

指定されたクレームに複数の値があるとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfNotEmptyClaim`

指定されたクレームが空でないとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfNotMultiValueClaim`

指定されたクレームが複数値でないとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfNotNullClaim`

指定されたクレームが `NULL` でないとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfNotSingleValueClaim`

指定されたクレームが単一値でないとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfNullClaim`

指定されたクレームが `NULL` であるとき、例外を生成します。

`com.ca.sm.icas.command.ErrorIfSingleValueClaim`

指定されたクレームが単一値のとき、例外を生成します。

デバッグするクラス ファイル

以下のクラス ファイルがデバッグ目的に使用されます。

`com.ca.sm.icas.command.debug.DumpChainID`

コンソール上にチェーン ID を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpClaim`

コンソール上に VPPID クレーム詳細（名前と値）を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpClaims`

コンソール上のトークン内のすべてのクレームの詳細を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpContext`

コンソール上にコンテキスト オブジェクトの詳細を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpParameters`

コンソール上の ICAS プロパティ ファイルで指定されたパラメータの詳細を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpSystemVars`

コンソール上にシステム環境の詳細を表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.DumpThreadStack`

コンソール上に実行するスレッドのスタック トレースを表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

`com.ca.sm.icas.command.debug.GenerateClaim_claims_AsHTML`

既知のクレームを表す HTML テーブルを持つ新しいクレーム "`_claims_`" を生成します。アクティブなレスポンスでこのクレームを使用して、取得されたクレームをユーザに表示する Cookie またはヘッダを作成します。

`com.ca.sm.icas.command.debug.GenerateClaim_parameters_AsHTML`

ICAS プロパティ ファイルで指定されたパラメータを表す HTML テーブルを生成します。アクティブなレスポンスでこの HTML テーブルを使用して、取得されたクレームをユーザに表示する Cookie またはヘッダを作成します。

`com.ca.sm.icas.command.debug.LogClaim`

チェーン コンテキストでポリシー サーバ トレース ログにクレーム詳細（名前と値）を書き込みます。

`com.ca.sm.icas.command.debug.LogClaims`

トークンでポリシー サーバ トレース ログにすべてのクレームの詳細を書き込みます。

`com.ca.sm.icas.command.debug.LogDecryptedClaims`

復号された XML トークンをポリシー サーバ トレース ログに記録します。

com.ca.sm.icas.command.debug.LogEncryptedClaims

IDP から取得した暗号化されたトークンをポリシー サーバ トレース ログに記録します。

com.ca.sm.icas.command.debug.START

コンソール上に **START** メッセージを表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

com.ca.sm.icas.command.debug.STOP

コンソール上に **STOP** メッセージを表示します。

注: 出力を表示するには、ポリシー サーバをコンソール モードで起動します。

com.ca.sm.icas.command.debug.ThrowException

メッセージの一部としてクレーム値が含まれる例外オブジェクトを生成します。

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 用の証明書データストアの設定

以下の点を考慮します。

- 依存パーティは、SSL を使用して fcc ファイルを保護する必要があります。
- 依存パーティは、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。

- ポリシー サーバ管理者は、依存パーティの Web サイトから SSL 証明書をインポートすることにより、ICAS 用の証明書データ ストアを設定します。
- インポートされた証明書はエイリアスと関連付けられ、それは fcc ファイルに格納されます。証明書の秘密キーを使用してセキュリティ トークンを復号し、デジタル署名を検証します。

以下の手順に従います。

1. Web Server 証明書ウィザードを使用して、IIS Web サーバから pfx ファイルに SSL 証明書をエクスポートします。

注: 詳細については、Microsoft のドキュメントを参照してください。
pfx ファイルから SSL 証明書をインポートするために、ポリシー サーバは、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を使用します。

2. 管理 UI を使用して、SSL 証明書を証明書データ ストアにインポートします。

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してください。

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

先頭

(mail=

終端

)

SQL の例

SQL クエリ

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できます。

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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
4. [認証方式タイプの新しいオブジェクトの作成] を選択し、[OK] をクリックします。
5. 認証方式の名前および説明を入力します。
6. [認証方式のタイプ] 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 フィールドが表示されます。
7. [保護レベル] フィールドに値を入力します。
8. この認証方式のタイプに対して有効にされたパスワード ポリシーをクリアします。

注： ICAS はパスワード ポリシーをサポートしません。

9. [方式のセットアップ]に以下のフィールドに対する値を入力します。

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

10. (オプション) [方式のセットアップ] 内の [Auth スキーム コンテキストをセッションストアへ保存] オプションを選択します。このオプションは、認証コンテキストデータが存続することを指定します。
11. [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

認証の完了後にクレーム値を取得するアクティブ レスポンスを設定するには、カスタム クラスの com.ca.sm.icas.GetClaimValue を使用します。

注: クレーム値の格納と取得には、セッションストアが必要です。セッションストアの詳細については、「ポリシー サーバ管理ガイド」を参照してください。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [レスポンス] をクリックします。
3. [レスポンスの作成] をクリックします。
4. ドメインを選択し、[次へ] をクリックします。
5. レスポンスの名前および説明を入力します。
6. エージェントタイプとして **Web** エージェントを指定します。
7. [属性リスト] 内の [レスポンス属性の作成] をクリックします。
8. [属性タイプ] の属性リストから **WebAgent-HTTP-Header-Variable** または **WebAgent-HTTP-Cookie-Variable** を選択します。
9. [属性のセットアップ] 内の [属性の種類] で [アクティブ レスポンス] を選択します。
10. [属性フィールド] で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

[Library Name]

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

11. [OK] をクリックします。

クレーム値を取得するレスポンスが作成されます。

ICAM ホワイトリストのサポートの設定

ICAS 実装は、ホワイトリスト発行者および LOA 認証に基づく Federal Identity, Credentialing, and Access Management Identity Metasystem Interoperability 1.0 Profile (ICAM IMI 1.0 Profile) 仕様をサポートします。

注: ホワイトリストおよび LOA サポートの詳細については、「ICAM IMI 1.0 Profile リリース候補版」を参照してください。

ホワイトリスト処理を有効にするには、*Infocard.properties* ファイル内の以下のパラメータを更新します。

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。

例: `postprocessingchain=com.ca.sm.icas.whiteListChecker.whiteListChecker`

whitelistlocation

ICAM によって指定されたホワイトリストが含まれる XML ファイルの場所を定義します。

例: `whitelistlocation=C:\Program Files\ca\siteminder\config\WhiteList.xml`

ICAS 実装は、Level of Assurances (LOA) の 3 つの検証をサポートします。

- LOA1
- LOA2
- LOA3

ホワイトリスト処理は LOA1、LOA2 および LOA3 の確認に必須です。LOA 処理をサポートするには `postprocessingchain` パラメータに以下の詳細を追加します。

- LOA1 処理の場合

```
postprocessingchain=com.ca.sm.icas.whiteListChecker.whiteListChecker;com.ca.sm.icas.whiteListChecker.ErrorIfLOA1ClaimMissing
```

- LOA2 処理の場合

```
postprocessingchain=com.ca.sm.icas.whiteListChecker.whiteListChecker;com.ca.sm.icas.whiteListChecker.ErrorIfLOA2ClaimMissing
```

- LOA3 処理の場合

```
postprocessingchain=com.ca.sm.icas.whiteListChecker.whiteListChecker;com.ca.sm.icas.whiteListChecker.ErrorIfLOA3ClaimMissing
```

RADIUS CHAP/PAP 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ます。次に、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。暗号化された文字列の比較は、暗号化された転送を使用せずに実行されます。

以下のダイジェスト認証方法を使用できます。

- RADIUS CHAP
- RADIUS PAP

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。認証が確認されるか、または接続が終了するまで、ユーザのマシンは認証を行うサーバに `Id/パスワード` のペアを繰り返し送信します。

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**方式では、ユーザを識別するために以下の処理が実行されます。

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、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**識別子は頻繁に変更され、サーバはいつでも認証リクエストを行うことができるため、**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、ユーザパスワードのダイジェストの計算によりユーザを認証します。次に、ポリシーサーバはダイジェストを**RADIUS**パケット内の**CHAP**パスワードと比較します。ダイジェストはハッシュ化パスワードで構成されます。これは、ディレクトリ属性を使用して計算されます。この属性は、**RADIUS CHAP/PAP**認証方式の設定中に指定します。

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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。
5. [認証方式のタイプ] リストから [RADIUS CHAP/PAP テンプレート] を選択します。
6. [方式のセットアップ] セクション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、場合によってはレルムに割り当てられます。

RADIUS サーバ認証方式

RADIUS プロトコルは、RADIUS サーバとしてポリシー サーバを動作させることによりサポートされます。NAS クライアントは RADIUS クライアントとして動作します。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 サーバは保護されているネットワークに接続されます。

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。

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」を参照してください。

RADIUS サーバ認証方式の前提条件

RADIUS 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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [RADIUS サーバテンプレート] を選択します。

6. [方式のセットアップ] に、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。

認証方式が保存されます。この認証方式をレルムに割り当てることができます。

SafeWord サーバ認証方式

SafeWord 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 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。

- 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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。
5. [認証方式のタイプ] リストから [SafeWord テンプレート] を選択します。
6. 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。
認証方式が保存されます。これで、この認証方式をレルムに割り当てることができます。

SafeWord サーバ認証方式と HTML フォーム認証方式

SafeWord サーバ認証方式および HTML フォーム認証方式では、カスタム HTML フォームと組み合わせて SafeWord サーバに照合してユーザを認証します。この方式では、SafeWord ハードウェア トークンの使用時にログインするユーザも対象に含まれます。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 設定ファイルに定義されています。

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 ディレクトリの間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、フォームを使用した認証リクエストがある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

カスタム HTML フォームと組み合わせて SafeWord サーバに照合してユーザを認証する際には、SafeWord サーバ認証方式および HTML フォーム認証方式を使用します。この方式は、SafeWord 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。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前と保護のレベルを入力します。
5. [認証方式のタイプ] リストから [SafeWord HTML フォーム テンプレート] を選択します。
6. サーバ名、SafeWord 設定ファイルの場所、およびサーバ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。

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 ユーザストアに保存され、ACE/Server 管理者は、これらの認証情報を変更できます。SecureID トークンにより、1 回のみ使用可能な TOKENCODE が生成されます。

以下の SecurID ダイジェスト認証方法を使用できます。

SecurID 認証

この方式は、RSA SecureID ハードウェア トークンを使用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` にマッピングするには、LDAP ディレクトリ属性の「`uid`」を使用します。そうしないと、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

この方式には HTML フォーム サポートが含まれています。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 方式は、ACE PIN の変更を許可されないユーザ、または *NextTokenCode* モードにないユーザを認証します。ただし、PIN を変更するユーザや、*NextTokenCode* モードにあるユーザは、HTML フォームにより RSA ACE/SecurID 方式によって認証されます。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) 製品が必要になります。

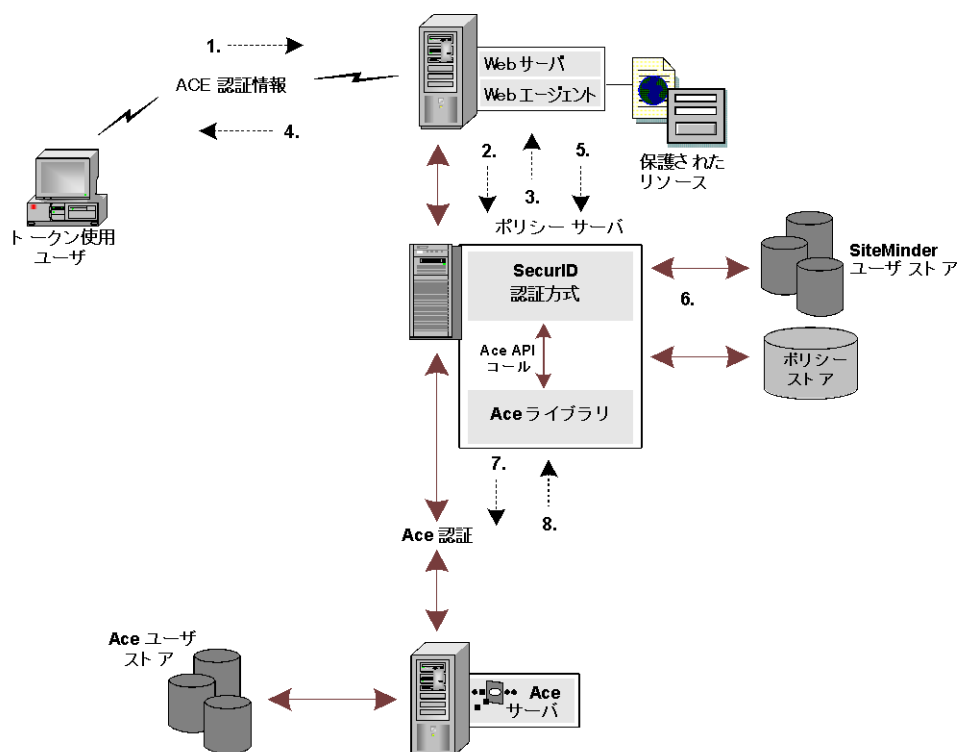
RSA ACE/Server は RSA SecurID トークンと連携し、有効な RSA Ace/Agent を通じてユーザの識別情報を認証します。また、RSA は Web エージェントとカスタムエージェントもサポートしています。SecurID 認証方式は、カスタム Ace/Agents として機能し、ACE/SDK とライブラリを使用します。

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、ユーザごとの複数の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 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、SecureID 認証方式では ACE サーバとユーザストア内の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、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。

HTML フォームに対応した RSA ACE/SecurID 方式の拡張は、「新しい PIN モード」に影響します。

下図は、RSA ACE/Server がどのようにポリシーサーバと対話するかを示します。



フローを以下に示します。

1. ユーザが Web ページなどのリソースを要求します。
2. 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 サーバがユーザの特定を行い、ユーザ名を **RSA/ACE** ユーザ名にマッピングします。ポリシー サーバはパスワード 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。サーバはユーザの認証情報も **ACE/Server** に送信します。
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** フォームを生成します。そのフォームはユーザに表示され、ユーザ名、以前の **PIN**、新しい **PIN** の入力、および新しい **PIN** の確認が要求されます。
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。

PIN は以下の場合に拒否される可能性があります。

- **PIN** が長すぎる。
- **PIN** が短すぎる。
- **PIN** にアルファベット文字が含まれている。

上記の図の認証では、**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。SecurID 認証方式では、新しい **PIN** を **ACE/Server** に送る前に **PN** ポリシー検証が実行されます。**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検証され、該当するエラー メッセージがユーザーに表示されます。検証は以下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。

これらの新しいエラー メッセージでは、ポリシーの関連する部分のみがユーザーに表示されます。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。

次に指定した PIN が「novel」である場合、メッセージは以下のとおりです。

新しい PIN に英数字が含まれています。

次に指定した PIN が「123412341234」である場合、メッセージは以下のとおりです。

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 認証方式を設定する方法

SecurID 認証方式を設定するには、以下の手順を実行します。

1. [SecurID 方式の前提条件を確認します](#) (P. 409)。
2. [SecurID 認証方式を設定します](#) (P. 409)。

SecurID 認証方式の前提条件

SecureID 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。

- 以下の条件に該当する場合は、ACE パスが SecurID ファイルの場所を指しているように設定します。
 - ACE 環境で ACE Client 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。
 - ACE 環境で、ポリシー サーバで保護していない別のアプリケーションを保護している。
 - ACE 環境で、別のサードパーティ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: SM_ACE_FAILOVER_ATTEMPTS 環境変数は削除されました。この変数は、ACE サーバへのフェールオーバーの試行を設定するために使用されていました。

SecurID 認証方式の設定

SecurID 認証方式を使用して、ACE 認証情報でログインするユーザを認証できます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。

5. [認証方式のタイプ] リストから [SecurID テンプレート] を選択します。
 6. 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。
- 認証方式が保存され、レルムに割り当て可能になります。

SecurID HTML フォーム認証方式を設定する方法

SecurID HTML フォーム認証方式を設定するには、以下の手順を実行します。

1. [SecurID HTML フォーム方式の前提条件を確認します](#) (P. 410)。
2. [SecurID HTML フォーム認証方式を設定します](#) (P. 411)。

SecurID HTML フォーム スキーム前提条件

SecureID 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。

- 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 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 HTML フォーム認証方式の設定

SecurID HTML フォームの認証方式を使用して、ACE 認証情報でログインするユーザを認証するカスタム HTML フォームを使用できます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
 3. [認証方式の作成] をクリックします。

[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。

[OK] ボタンをクリックします。
 4. 名前および保護レベルを入力します。
 5. [認証方式のタイプ] リストから [SecurID HTML フォーム テンプレート] を選択します。
 6. 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。
- 認証方式が保存され、レルムに割り当て可能になります。

SecurID ユーザを再アクティブ化し、確認するための SecurID HTML フォーム サポート

SecurID HTML フォーム スキームでレ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、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。

ポリシー サーバ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。

PWNextToken.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。

新しいユーザ アカウントをアクティブにするための SecurID HTML フォーム サポート

管理者がユーザ アカウントを作成し、そのユーザがログインするときに、ポリシー サーバは以下のフォームを使用します。 フォームを使用して、ユーザは PIN を作成するか、またはポリシー サーバにランダムな 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、PIN 番号の作成手順に置き換わります。
例：

PIN の長さは 4 ～ 8 文字で指定してください。

PWPINAccept.template

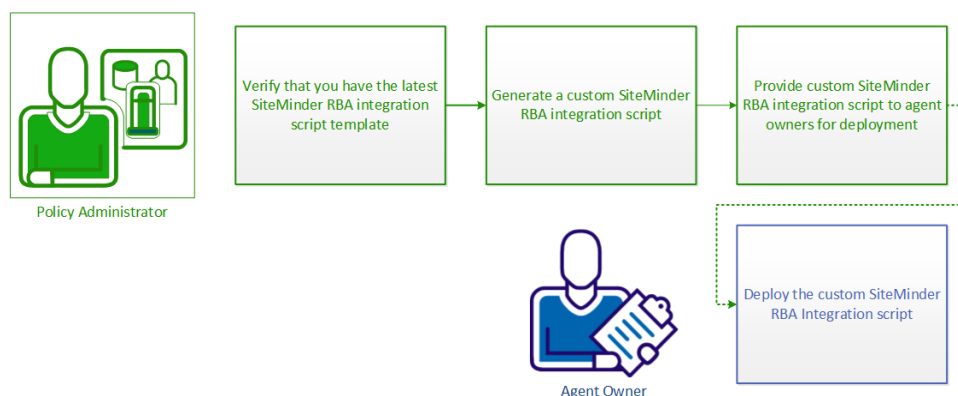
このテンプレートは、システム生成の新しい PIN が作成されたことを示します。このテンプレートでは、システムが生成した PIN によって \$USRMSG\$ 変数が置き換わります。

「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。

リスク ベース認証をサポートする SecurID HTML フォーム認証を設定する方法

SecurID 用の RSA リスク ベース認証 (RBA) は、ユーザ名/パスワードによるログイン操作を保持しながら、ユーザ ID のリスク ベース検証を提供します。

リスク ベース認証 (RBA) をサポートする SecurID HTML フォーム認証を設定するには、ポリシー管理者およびエージェント所有者の協力が必要です。このシナリオでは、両者が実行する必要があるすべての手順について説明します。



ポリシー管理者は、以下の手順を実行します。

1. [最新の SiteMinder RBA 統合スクリプトテンプレートがあることを確認します](#) (P. 414)。
2. [カスタム SiteMinder RBA 統合スクリプトを生成します](#) (P. 415)。
3. [展開を行うエージェント所有者に、カスタム SiteMinder RBA 統合スクリプトを提供します](#) (P. 415)。

エージェント所有者は、以下の手順を実行します。

1. [各 Web サーバにカスタム SiteMinder RBA 統合スクリプトを展開します](#) (P. 416)。

最新の SiteMinder RBA 統合スクリプトテンプレートがあるかどうかの確認

RBA 統合スクリプトは、RSA Authentication Manager に付属するテンプレートに基づいています。ただし、RSA はリリースとリリースの間にテンプレートを更新できるため、最新のテンプレートであることを確認します。

以下の手順に従います。

1. このリンク
<https://sftp.rsa.com/human.aspx?Username=partner&password=rsasecured&arg01=859293799&arg12=downloaddirect&transaction=signon&quiet=true>を使用して、SiteMinder RBA 統合スクリプトテンプレートをダウンロードします。
2. RSA Authentication Manager サーバに付属する統合スクリプトテンプレートを見つけます。
3. サーバに統合スクリプトテンプレートがない場合は、手順 1 でダウンロードしたテンプレートをインストールします。それ以外の方法では、テンプレートのヘッダを比較し、どちらか最新のテンプレートをインストールします。

カスタム SiteMinder RBA 統合スクリプトの生成

エージェントで展開するカスタム RBA 統合スクリプトを生成するには、RSA セキュリティ コンソールを使用します。

以下の手順に従います。

1. RSA セキュリティ コンソールにログインし、1 つ以上のエージェントに対して RBA を有効にします。
2. エージェントがユーザーを認証するために使用するプライマリ検証方法を選択します (RSA SecurID または修正されたパスコード)。
3. スクリプトを生成し一時ディレクトリにそれをダウンロードするには、CA SiteMinder テンプレートを選択します。

展開を行うエージェント所有者に、カスタム SiteMinder RBA 統合スクリプトを提供します。

RSA セキュリティ コンソールで生成したカスタム RBA 統合スクリプトは、RBA をサポートする Web サーバごとに展開する必要があります。

各エージェント所有者にカスタム RBA 統合スクリプトを提供し、[その展開方法](#) (P. 416)を通知します。

エージェント所有者は、各 Web サーバにカスタム SiteMinder RBA 統合スクリプトを展開する

RBA をサポートする各 Web サーバに、ポリシー管理者によって提供されたカスタム SiteMinder RBA 統合スクリプトを展開します。

以下の手順に従います。

1. エージェントホストにログインし、デフォルトの RSA SecurID ログインテンプレート (smpwsservices.fcc) を見つけます。テンプレートは、エージェントルートから相対の /siteminderagent/forms/ ディレクトリに置かれています。
2. テキストエディタで smpwsservices.fcc を開き、一番下の </body> タグの直前に以下の 2 行を追加し、そのファイルを保存します。

```
<script src="am_integration.js" type="text/javascript"></script>  
<script>window.onload=redirectToIdP;</script>
```

重要: 編集を始める前に smpwsservice.fcc のバックアップを作成し、必要に応じてそれを使用して変更を元に戻します。

3. カスタム SiteMinder RBA 統合スクリプト (am_integration.js) を /siteminderagent/forms/ ディレクトリにコピーし、Web サーバを再起動します。

X.509 クライアント証明書認証方式

X.509 V3 クライアント証明書を設定できます。証明書がクライアントにインストールされると、リソースを要求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ると、より高い安全性を提供することができます。

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。

X.509 クライアント証明書の認証方式により、証明書認証が実行されます。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設定されている必要があります。これらのタスクは、ポリシーサーバ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、X.509 認証方式を設定します。以下の設定タスクがあります。

- 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、「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。

X.509 クライアント証明書認証方式は以下の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。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 クライアント証明書認証方式の証明書マッピング \(P. 471\)](#)

証明書認証の証明書の抽出

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認証情報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。

SiteMinder でユーザの識別に証明書データを使用する方法

Web エージェントは証明書情報を収集後、そのデータを確認のためポリシー サーバに渡します。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きユーザを特定することです。

まず、ポリシー サーバは、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。ポリシー サーバは、証明書発行者 DN を使用してマッピングを特定します。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のみ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 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 クライアント証明書認証方式の証明書マッピング](#) (P. 471)

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、およびいずれのユーザ証明書でも証明書認証を受諾および実行できるよう設定します。
2. インストールされた Web エージェントが SSL 認証を処理できることを確認します。
3. 管理 UI で 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の情報に基づいたユーザを識別できるようにします。
5. (オプション) 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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] 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。

6. SSL 認証情報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. (オプション) [認証セッション変数を保持する] を選択します。このオプションは、認証コンテキスト データが、後で認証判断の際に使用できるようセッションストアで保存されることを指定します。
8. [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. 471)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。

ユーザが正常に認証されるためには、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。

AND

- 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。

X.509 クライアント認証の場合は、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 Web エージェントに指示が出されます。次に、ポリシー サーバはユーザが存在していることを確認し、ユーザの基本認証情報を確認し、証明書認証情報と基本認証情報が同じユーザを表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。

詳細情報:

[ユーザ ディレクトリ](#) (P. 157)

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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [X509 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。
6. SSL 認証情報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. (オプション) [方式のセットアップ] で [認証セッション変数を保持する] を選択します。このオプションは、認証コンテキストデータがセッションストアで保存されることを指定します。
8. [サブミット] をクリックします。
認証方式が保存され、レلمに割り当て可能になります。

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 2 つのイベントのいずれか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 ベーシック プロトコルを介してユーザに要求します。HTTP ベーシック認証により、エージェントはユーザ名とパスワードを取得することができます。

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: X509 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選択しない場合、ブラウザは自動的にこの決定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、必須証明書（X509 証明書、X509 証明書およびベーシック、または X509 証明書およびフォームなど）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。

認証を受ける証明書を送信するかどうかの選択をユーザに認めない X509 証明書またはベーシック認証方式およびその他の証明書ベースの認証方式を含む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配置において、有効な証明書を保有しているユーザに、その証明書をリソースにアクセスするために使用するよう促します。

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上に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。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [X509 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。

6. SSL 認証情報コレクタのサーバおよびターゲット情報を入力します。
7. (オプション) [方式のセットアップ] で [認証セッション変数を保持する] を選択します。このオプションは、認証コンテキストデータがセッションストアで保存されることを指定します。
8. [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 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。

ユーザが正常に認証されるためには、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。
- AND
- ユーザが HTML フォームで要求される認証情報を指定していること。

この方式の場合、認証処理では以下の手順に従います。

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 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. ポリシーサーバはユーザの HTML フォームの認証情報を確認し、証明書認証情報と HTML フォームの認証情報が同じユーザを表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、ポリシー サーバ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と一緒にポリシー サーバに渡されます。

状況	結果
証明書がない場合	ブラウザがエラー 500 を発行します。
証明書とフォームのクレデンシャルが拒否された場合	ブラウザがエラー 500 を発行します。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。

[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。

[OK] ボタンをクリックします。

4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [X509 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。
6. SSL 認証情報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. (オプション) [方式のセットアップ] で [認証セッション変数を保持する] を選択します。このオプションは、認証コンテキスト データがセッションストアで保存されることを指定します。
8. [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 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 HTML フォーム認証方式では、フォーム認証または X.509 クライアント証明書認証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 2 つのイベントのいずれかが発生する必要があります。

- 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。

ユーザが保護されたリソースをリクエストすると、Web エージェントがブラウザに証明書の提示を要求します。この方式では、以下の影響があります。

状況	結果
証明書が提示されます。	証明書が処理されます。
証明書が拒否された場合	ブラウザがエラー 500 を発行します。
証明書は提示されません。	ブラウザ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	ブラウザによってフォームへの入力再度求められます。

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フォーム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場合、ブラウザは自動的にこの決定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 証明書およびベーシック、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、「禁止されています (Forbidden)」という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。

認証を受ける証明書を送信するかどうかの選択をユーザに認めない X509 証明書またはフォーム認証方式およびその他の証明書ベースの認証方式を含む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配置において、有効な証明書を保有しているユーザに、その証明書をリソースにアクセスするために使用するよう促します。

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、ポリシー サーバ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。

詳細情報:

[ユーザ ディレクトリ](#) (P. 157)

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_AuthApi_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

X.509 証明書または HTML フォーム認証方式を使用して、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できます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [X509 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。

6. サーバとターゲット情報を入力します。
7. (オプション) [方式のセットアップ] で [認証方式データを保持する] を選択します。このオプションは、認証コンテキスト データがセッションストアで保存されることを指定します。
8. [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。

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、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。

詳細情報:

[レルム](#) (P. 595)

匿名認証方式の前提条件

匿名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。

- 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の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には、管理 UI の [グローバル設定] ペインでユーザ追跡を有効にします。

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。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

匿名認証方式の設定

匿名認証方式を使用して、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ことができます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] リストから [匿名テンプレート] を選択します。

6. 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。
- 認証方式が保存され、レルムに割り当て可能になります。

カスタム認証方式

使用したい認証方式がポリシー サーバにない場合は、CA の API を使用して、カスタムの認証方式を作成します。

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。カスタム認証方式を作成したら、その方式を[認証] ペインで設定します。

注: 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 の統合 \(P. 871\)](#)」を参照してください。

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

カスタムの認証方式を使用して、製品で提供されていない方式を指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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。
5. [認証方式のタイプ] 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。
6. (オプション) [各方式共通セットアップ] で [保護のオーバーライドを許可する] チェック ボックスをオンにします。このオプションは、ライブラリの保護レベルが 管理 UI で指定された保護レベルより優先されることを指定します。
7. 認証方式の認証情報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] に入力します。
8. (オプション) [方式のセットアップ] で [認証セッション変数を保持する] を選択します。このオプションを選択しない場合、セッション変数はセッションストアに保存されません。
9. [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。

OAuth 認証方式を設定する方法

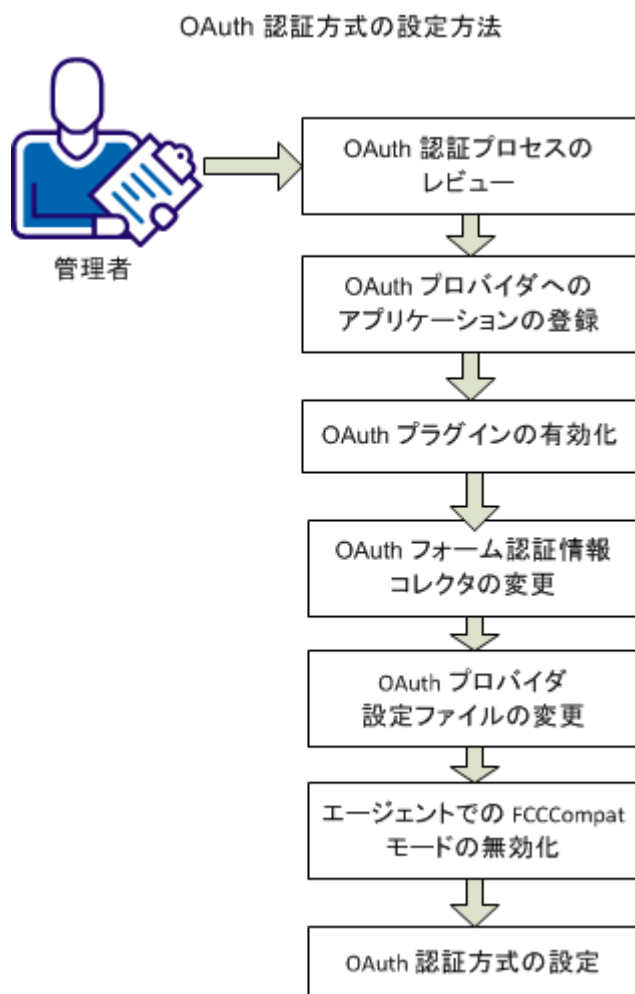
このシナリオでは、ポリシー サーバの管理者およびエージェント所有者が OAuth 認証方式を設定する方法について説明します。

OAuth 認証方式では、ユーザが OAuth プロバイダによって認証情報をサブミットできます。OAuth プロバイダはユーザを認証し、ユーザに関するクレームを含む認証レスポンスを送信します。ポリシー サーバは認証レスポンスを確認し、認証プロセスを完了して、リソースへのアクセスを認可します。

シナリオの目標は以下のとおりです。

- OAuth 認証プロセスがどのように動作するかを識別する。
- 必要なエージェント側コンポーネントを設定する。
- 必要なポリシー サーバ側コンポーネントを設定する。

以下の図は、OAuth 認証方式を設定するために必要なタスクを示しています。



OAuth 設定の以下の手順を実行します。

1. [OAuth 認証プロセスを確認します](#) (P. 439)。
2. [OAuth プロバイダへのアプリケーションの登録](#) (P. 440)
3. [OAuth プラグインを有効化します](#) (P. 443)。
4. [OAuth フォーム認証情報コレクタを変更します。](#) (P. 444)
5. [OAuth プロバイダ設定ファイルの変更](#) (P. 446)。
6. [エージェントで FCCCompatMode を無効にします](#) (P. 451)。
7. [OAuth 認証方式を設定します](#) (P. 451)。

OAuth 認証プロセス

以下のプロセスでは、OAuth 認証方式の動作の仕組みについて説明します。

1. ユーザはリソースをリクエストします。
2. エージェントはリクエストをインターセプトし、リソースが保護されているかどうかを確認するためにポリシー サーバにアクセスします。
3. ポリシー サーバは、リソースが OAuth 認証方式で保護されていることを確認します。ポリシー サーバは、エージェントがユーザを OAuth フォーム認証情報コレクタ (FCC) にリダイレクトするようリクエストします。
4. エージェントはユーザを FCC にリダイレクトします。
5. ユーザは OAuth プロバイダを選択し、プロバイダ サイトで認証情報をサブミットします。
6. FCC は OpenID ユーザ ID を作成して、それをポリシー サーバに渡します。
7. ポリシー サーバは認証リクエストを構築して、OAuth プロバイダにユーザをリダイレクトします。
8. ユーザはプロバイダ固有の認証情報を使用して、OAuth プロバイダでの認証を行います。
9. プロバイダは、正常な認証のレスポンスを FCC にリダイレクトします。
10. エージェントは、認証のレスポンスをポリシー サーバに渡します。
11. ポリシー サーバはプロバイダ認証レスポンスを確認し、それを使用して最初の必要なクレームの値を判定します。ポリシー サーバは、クレーム値に一致する属性を持ったユーザを見つけるためにポリシー ドメインのユーザ ディレクトリをすべて検索します。一致すると、ポリシー サーバはユーザを認証します。

注: この認証方式は、認証の匿名モードをサポートします。匿名モードでは、ポリシー サーバは、ユーザにマップするためにクレーム値を使用しません。ポリシー サーバは、認証方式で定義した匿名ユーザに一致するユーザを見つけるためにポリシー ドメインでユーザ ディレクトリをすべて検索します。

12. ポリシー サーバは、FCC に認証されたユーザのセッション詳細を返します。
13. FCC はセッション Cookie を作成して、それを Web ブラウザに渡します。ユーザはリクエストされたリソースにリダイレクトされます。また、ポリシー サーバは認可決定をすべて維持します。

OAuth プロバイダへのアプリケーションの登録

前提条件

SecureURLs パラメータが Web エージェントの設定およびその値（設定されている場合）に設定されるかどうかを判断します。

登録処理の一部として、リダイレクト URI 設定 (Google) またはサイト URL 設定 (Facebook) でエントリを指定します。このエントリの形式は、SecureURLs エージェントパラメータの値によって異なります。管理 UI から SecureURLs パラメータの値を確認します。このパラメータは、ホストエージェント用のエージェント設定オブジェクトにあります。

- SecureURLs パラメータが [はい] に設定されている場合は、以下の形式で URL を入力します。

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample`

- SecureURLs パラメータが [いいえ] に設定されている場合は、以下の形式で URL を入力します。

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc`

Google

以下の手順に従います。

1. Google App アカウントを作成します。
2. <https://code.google.com/apis/console> に移動し、ログインします。

3. [API Access] を選択し、OAuth クライアント ID を作成します。クライアント ID を作成するには、以下の情報を入力します。

ホーム ページ URL

`http(s)://homepage.com`

例 : `http://www.forwardinc.com`

アプリケーション タイプ

Web アプリケーション

ユーザのサイトまたはホスト名

`http://agent_host`

例 : `http://myagent.ca.com`

4. [Create clientID] をクリックします。
5. リダイレクト URI の設定を以下の URL に編集します。

リダイレクト URIs

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc`

(SecureURLs が設定されていないか、または [いいえ] に設定されている場合)

または

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample`

(SecureURLs が [はい] に設定されている場合)

6. [更新] をクリックします。

Google 登録処理が完了します。

Facebook**以下の手順に従います。**

1. Facebook アカウントを作成します。
2. `https://developers.facebook.com/apps` に移動します。
3. [AppName] を選択し、[Continue] をクリックします。

4. 表示される **captcha** テキストを入力し、[Continue]をクリックします。
5. Web サイトの [Facebook Login] セクションで、以下のフィールドに入力します。

Site URL

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc`

(SecureURLs が設定されていないか、または [いいえ] に設定されている場合)

または

`http(s)://agent_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample`

(SecureURLs が [はい] に設定されている場合)

6. [Save changes] をクリックします。
7. 以下の設定を完了します。
 - ドメインを **ca.com** に設定する
 - サンドボックス モードを無効にする
 - キャンバス URL を `https://apps.facebook.com/application_name/` に設定する
 - 保護されたキャンバス URL を `https://apps.facebook.com/application_name/` に設定する

登録処理によって、クライアント アプリケーション URL、クライアント ID、およびそれに関連付けられた秘密キーが生成されます。登録によって、OAuth 許可サーバエンドポイント URL も生成されます。ここから OAuth サービスは許可コードおよびアクセス トークンを取得します。この情報の一部は、OAuth 認証方式が正常に動作するために使用するファイルを設定する際に必要です。

OAuth プラグインの有効化

OAuth プラグインは Web エージェント設定ファイル (WebAgent.conf) で参照されます。プラグインにより、エージェントが OAuth プロバイダと通信し、ポリシー サーバに OAuth プロバイダ認証レスポンスを転送します。

エージェント所有者に連絡し、必要なプラグインを有効にするように指示します。

以下の手順に従います。

1. Web エージェント ホスト システムにログインします。
2. Web エージェント設定ファイルを開きます。デフォルトのファイルの場所は Web サーバ タイプによって異なります。

IIS

`agent_home¥bin¥IIS`

`agent_home`

エージェント インストールパスを指定します。

Oracle iPlanet (iPlanet/SunOne)

`web_server_home/https-hostname/config`

`web_server_home`

Web サーバのインストールパスを指定します。

Apache、IBM HTTP サーバ、および Oracle HTTP サーバ

`web_server_home/conf`

`web_server_home`

Web サーバのインストールパスを指定します。

Domino (Windows)

`C:¥lotus¥domino`

Domino (UNIX)

`$HOME/notesdata`

3. OAuth プラグインをロードする行のコメントを外します。

例 :

`#LoadPlugin="C:¥Program Files¥CA¥webagent¥bin¥OAuthPlugin.dll"`

4. ファイルを保存します。
5. Web サーバを再起動します。

OAuth フォーム認証情報コレクタの変更

サンプル OAuth FCC は Web エージェント インストールに含まれています。FCC では、ユーザが OAuth プロバイダ ユーザ名を入力することによって認証できます。

デフォルトでは、FCC は多くの OAuth プロバイダを表示します。エージェント所有者に連絡し、保護されているアプリケーションがサポートするプロバイダのみファイルに含まれるように、FCC を変更するように指示します。

以下の手順では、FCC を変更する方法について説明します。

以下の手順に従います。

1. Web エージェント ホスト システムにログインします。
2. 以下の場所に移動します。

`agent_home¥samples¥forms`

`agent_home`

Web エージェントのインストールパスを指定します。

3. `oauth.fcc` ファイルをテキスト エディタで開きます。
4. FCC を確認します。プロバイダを追加または削除するかどうかを判断します。
5. FCC でプロバイダのセクションを見つけます。テキスト文字列 **var oauth_providers** を検索します。
6. 不足しているプロバイダ エントリを `var oauth_providers` セクションに追加します。 **apps** 設定の横にアプリケーションの名前を指定します。

以下に、プロバイダ エントリの 2 つの例を示します。

```
var oauth_providers = {
  google : {
    name : 'Google',
    image : 'images/google.gif',
    apps : 'ForwardIncGoogleApp'
  },
  facebook : {
    name : 'Facebook',
    image : 'images/facebook.gif',
    apps : 'ForwardIncFacebookApp'
  }
}
```


7. テキスト文字列 **var oauth_applications** を検索して、FCC のアプリケーションセクションを見つけます。各アプリケーションで、以下の情報を指定します。

- アプリケーション名。
- ラベル（ユーザがプロバイダアイコンをクリックした後に表示されるテキスト）。
- イメージ。イメージ値は、FCC が表示するプロバイダアイコンの場所を表します。

以下に、アプリケーションエントリの 2 つの例を示します。1 つは ForwardIncGoogleApp、もう 1 つは ForwardIncFacebookApp という名前です。

重要: これらのアプリケーション名は、値の例を示すことのみが目的です。展開には、実際のアプリケーション名を入力します。また、この FCC ファイルに入力するアプリケーション名は、[oauthproviders.xml ファイル](#) (P. 446)に入力するアプリケーション名と一致する必要があります。

```
};  
var oauth_applications = {  
  
ForwardIncGoogleApp : {  
    name : 'ForwardIncGoogleApp',  
    label : 'Login using google application',  
    image : 'images/google.ico'  
}  
ForwardIncFacebookApp : {  
    name : 'ForwardIncFacebookApp',  
    label : 'Login using facebook application',  
    image : 'images/facebook.ico'  
},
```

8. プロバイダエントリをコメントアウトして、FCC から任意の不要なプロバイダを削除します。プロバイダ名の前に、コメント文字列 **/*** を入力します。エントリの最後に、コメント文字列を閉じる ***/** を入力します。以下に例を示します。

```
/* google : {  
    name : 'ForwardIncGoogleApp',  
    image : 'images/google.gif',  
    apps : 'ForwardIncGoogleApp'  
},*/
```

9. すべての変更が完了したら、スクリプトを保存します。
10. Web サーバを再起動します。

OAuth プロバイダ設定ファイルの変更

OAuth プロバイダ設定ファイル (oauthproviders.xml) が、ポリシー サーバにインストールされます。プロバイダ設定ファイルには、各プロバイダおよび保護されているアプリケーションの設定の詳細が含まれます。ファイルに正しい設定が含まれていないと、認証は失敗します。

ファイルに関する情報：

- デフォルトでは、ファイルには、OAuth FCC が利用可能にするすべてのプロバイダのサンプル設定が含まれます。サンプル設定を確認し、必要に応じてそれらを変更します。

重要: 値はサンプルのみです。認証方式を展開する前に、OAuth プロバイダでの設定をすべて確認することをお勧めします。

- 複数のアプリケーションでプロバイダ設定を再利用するため、プロバイダ設定は登録されているアプリケーション設定から分離されています。
- プロバイダ設定の詳細については、アプリケーション設定の詳細に従う必要があります。各アプリケーションは、事前定義済みプロバイダ設定のいずれかを使用します。
- 各アプリケーションには、登録で使用するため、いずれかのプロバイダに対する PROVIDERLINK が必要です。
- FCC ファイル内にプロバイダ設定を追加または変更する場合は、このファイル内のプロバイダの設定も追加または変更します。
- このファイルに入力するアプリケーション名は、oauth.fcc ファイルに入力するアプリケーション名と一致する必要があります。
- 複数の OAuth 認証方式を設定する場合、各方式で独自のプロバイダ設定ファイルを使用できます。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. 以下の場所に移動します。

`siteminder_home¥config¥properties`

`siteminder_home`

ポリシー サーバのインストールパスを指定します。

3. バックアップとして、デフォルトのプロバイダ設定ファイルのコピーを作成します。
4. `oauthproviders.xml` ファイルを開きます。
5. ファイルを確認し、必要な OAuth プロバイダ設定が利用可能かどうかを判断します。プロバイダを追加するには、以下の手順に従います。
 - a. 既存の OAuth プロバイダ ノードおよびその子ノードをすべてコピーします。すべてのプロバイダ ノードは、以下のルート ノード内に含まれています。

```
<OAuthProvider>  
</OAuthProvider>
```

- b. 新しい OAuth プロバイダ ノードおよびその子ノードをすべて以下のルート ノードに追加します。

```
<TrustedOAuthProviders>  
</TrustedOAuthProviders>
```

6. 各プロバイダの設定を行います。以下の設定の値を更新します。

OAuth providername

このノードの OAuth プロバイダを識別します。プロバイダの名前を入力します。

注: プロバイダ名の入力には小文字を使用します。

AuthorizationURL

このプロバイダの許可サーバエンドポイント URL を指定します。この URL は、ユーザの認証が成功した後に許可トークンを生成する必要があります。

Google の例 : <https://accounts.google.com/o/oauth2/auth>

Facebook の例 : <https://www.facebook.com/dialog/oauth>

AccessTokenURL

アクセス トークン エンドポイント URL を指定します。アプリケーション設定の詳細と共に許可コードを交換することにより、アクセス トークンに対するクエリを行うことができます。

Google の例 : <https://accounts.google.com/o/oauth2/token>

Facebook の例 : https://graph.facebook.com/oauth/access_token

7. 各登録済みアプリケーションの設定を行います。アプリケーションノードおよびすべての子ノードは、ルート ノードの下に存在します。

<Application>

</Application>

以下の設定の値を更新します。

Application appname

OAuth に登録済みのアプリケーションの設定およびユーザ認証設定を識別します。エンドユーザは、OAuth 認証の設定を使用するために、FCC ページでこの識別子を指定する必要があります。

例 : ForwardIncGoogleApp, ForwardIncFacebookApp

重要: これらのアプリケーション名は、値の例を示すことのみが目的です。展開には、実際のアプリケーション名を入力します。また、このファイルに入力するアプリケーション名は、[oauth.fcc ファイル](#) (P. 444)に入力するアプリケーション名と一致する必要があります。

ApplicationURL

登録済みのアプリケーション URL を指定します。アプリケーションの登録時に入力したものと同一リダイレクト URI (Google) またはサイト URL (Facebook) で、この設定の値を更新します。たとえば、Google Redirect URI 値が以下に設定されている場合：

URI をリダイレクトします：

`https://fedserver.bizpartnerinc.ca/siteminderagent/forms/oauthcb.fcc`

このファイル内の ApplicationURL の値は次のとおりです：

```
</OAuthProvider>
```

```
<Application appname="ForwardIncGoogleApp">
```

```
<ApplicationURL>
```

```
<!-- HOSTNAME : WebAgent/SPS host name -->
```

```
https://fedserver.bizpartnerinc.ca/siteminderagent/forms/oauthcb.fcc
```

```
</ApplicationURL>
```

URL の形式は、SecureURLs エージェント パラメータの値によって異なります。管理 UI から SecureURLs パラメータの値を確認します。このパラメータは、ホスト エージェント用のエージェント設定オブジェクトにあります。

- SecureURLs パラメータが [はい] に設定されている場合は、以下の形式で URL を入力します。

```
http(s)://agent_host/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample
```

例：

```
https://myagent.ca.com/siteminderagent/forms/oauthcb.fcc?SMQUERYDATA=Sample
```

- SecureURLs パラメータが [いいえ] に設定されている場合は、以下の形式で URL を入力します。

```
http(s)://agent_host/siteminderagent/forms/oauthcb.fcc
```

ClientID

OAuth サーバで登録されたクライアント アプリケーションの識別子が含まれています。生成されたクライアント ID で、この設定の値を更新します。アプリケーションが正常に登録されると、許可サーバがこの ID 値を提供します。

Secret

ClientID に関連付けられている秘密キーを示します。 ClientID に関連付けられている秘密キーで、この設定の値を更新します。 アプリケーションが正常に登録されると、許可サーバがこの値を提供します。

PROVIDERLINK

アプリケーションをプロバイダとリンクします。 定義済みのプロバイダの **providername** 値を指定します。このアプリケーションは、OAuth 認証の実行中にプロバイダ設定を使用します。

例： google、facebook

スコープ

アプリケーションがユーザに要求する必要な権限のタイプを指定します。 たとえば、スコープ値が

<https://www.googleapis.com/auth/userinfo.profile> の場合、アプリケーションは、基本的なユーザ プロファイル情報への読み取り専用アクセス権を取得できます。

このスコープ値は、許可トークン リクエストで渡されます。 クライアントは、コードを使用して、**UserInfoURL** 属性で指定されたリソース URL にアクセスできます。 管理者は、この属性に対して単一の値、またはスペースで区切られた複数の値を指定できます。

UserInfoURL

生成されたアクセス トークンを使用してユーザ情報を問い合わせることができる、単一の URL、またはスペースで区切られた複数の URL を指定します。 URL は、クライアントがアクセスを試行しているリソースを表します。

UserAttribute

ユーザ属性を指定します。 OAuth ユーザ情報からのユーザを識別するクレームで、この値を更新します。 この属性の値は、ユーザを特定するために使用されます。 たとえば、このユーザ属性は「email」にすることができます。

8. ファイルを保存して閉じます。

エージェントでの FCCCompatMode の無効化

cam エージェント設定で、FCCCompatMode 設定を無効にします。

以下の手順に従います。

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] をクリックします。
3. [エージェント設定オブジェクト] をクリックします。
4. [エージェント設定オブジェクト] ページが表示されます。
5. 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。

6. cam エージェントを選択して変更します。

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。

7. [変更] をクリックします。
8. FCCCompatMode を no に設定します。
9. 変更を保存します。

OAuth 認証方式の設定

管理 UI を使用して、OAuth 認証方式オブジェクトを設定します。

ネットワークに複数の Cookie ドメインが含まれ、それぞれの Cookie ドメインに認証方式が必要な場合は、Cookie ドメインごとに個別の認証方式オブジェクトを設定します。

注: Solaris プラットフォーム上の操作の場合には、**sun.security.provider.Sun** プロバイダが最初のプロバイダとして登録されるように、ディレクトリ `jre_root/lib/security` 内の `java.security` ファイルを変更します。

以下の手順に従います。

1. [インフラストラクチャ] - [認証] - [認証方式] をクリックします。
2. [認証方式の作成] をクリックします。
3. 新しいオブジェクト オプションを選択し、[OK] をクリックします。
[認証方式の作成] ページが表示されます。
4. 名前および保護レベルを入力します。
5. [認証方式タイプ] リストから OAuth テンプレートを選択します。
方式固有の設定が表示されます。
6. その他のパラメータを設定し、[サブミット] をクリックします。

重要: プロキシ設定のいずれかを変更する場合は、ポリシー サーバを再起動します。

OAuth 認証方式を設定するためのタスクを完了しました。これで、ポリシーまたはアプリケーション コンポーネント内の認証方式を使用して、リソースを保護することができます。

OpenID 認証方式

このシナリオでは、OpenID 認証方式を設定する方法について説明します。

OpenID 認証方式では、ユーザーが OpenID プロバイダによって認証情報をサブミットできます。OpenID プロバイダはユーザーを認証し、認証レスポンスをポリシー サーバに送信します。ポリシー サーバはレスポンスを確認し、認証プロセスを完了して、リソースへのアクセスを許可します。

シナリオの目的は対象読者が以下のことを実行できるようにすることです。

- OpenID 認証プロセスがどのように動作するかを識別する。
- 認証方式を有効にするように必要なエージェント側のコンポーネントを設定する。
- 認証方式を有効にするように必要なポリシー サーバ側のコンポーネントを設定する。

SiteMinder 12.52 SP1 は、OpenID バージョン 1.1 および 2.0、ならびに OpenID Attribute Exchange バージョン 1.0 をサポートしています。

SiteMinder での OpenID 認証の動作の仕組み

OpenID 認証プロセスのフローは、以下の手順のとおりです。

1. ユーザはリソースをリクエストします。
2. エージェントはリクエストをインターセプトし、リソースが保護されているかどうかを確認するためにポリシー サーバにアクセスします。
3. ポリシー サーバは、リソースが OpenID 認証方式で保護されていることを判定し、エージェントでユーザを OpenID フォーム認証情報コレクタ (FCC) にリダイレクトするようリクエストします。
4. エージェントはユーザを FCC にリダイレクトします。
5. ユーザは以下のいずれかのタスクを完了します。FCC をどのようにカスタマイズするかによって、この手順でのワークフローが決定されます。ユーザは以下のことを実行できます。
 - OpenID プロバイダを選択し、プロバイダ サイトで OpenID 認証情報をサブミットする。
 - OpenID プロバイダを選択し、OpenID ユーザ名をサブミットする。
 - OpenID を選択し、完全な OpenID ユーザ識別子をサブミットする。
6. FCC は OpenID ユーザ識別子を構築して、それをポリシー サーバに渡します。
7. ポリシー サーバは OpenID 認証リクエストを構築して、認証用の OpenID プロバイダにユーザをリダイレクトします。
8. ユーザはプロバイダ固有の認証情報を使用して、OpenID プロバイダでの認証を行います。

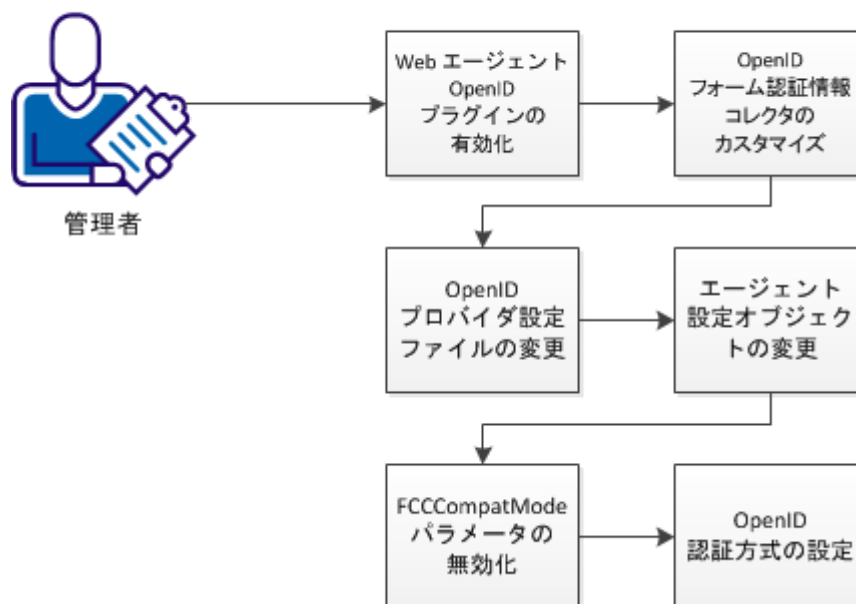
9. OpenID プロバイダは、正常な認証のレスポンスを FCC にリダイレクトします。
10. エージェントは、認証のレスポンスをポリシー サーバに渡します。
11. ポリシー サーバはプロバイダ認証レスポンスを確認し、それを使用して最初の必要なクレームの値を判定します。ポリシー サーバは、クレーム値に一致する属性を持ったユーザを見つけるためにポリシー ドメインのユーザ ディレクトリをすべて検索します。一致すると、ポリシー サーバはユーザを認証します。

注: この認証方式は、認証の匿名モードをサポートします。匿名モードでは、ポリシー サーバは、ユーザにマップするためにクレーム値を使用しません。ポリシー サーバは、認証方式で定義した匿名ユーザに一致するユーザを見つけるためにポリシー ドメインでユーザ ディレクトリをすべて検索します。
12. ポリシー サーバは、FCC に認証されたユーザのセッション詳細を返します。
13. FCC はセッション Cookie を作成して、それを Web ブラウザに渡します。ユーザはリクエストされたリソースにリダイレクトされます。また、ポリシー サーバは認可決定をすべて維持します。

OpenID 認証方式を設定する方法

このセクションでは、ポリシー サーバ管理者がどのように OpenID 認証方式を設定できるかについて説明します。以下の図で必要なタスクについて説明します。

OpenID 認証方式の設定方法



以下の手順に従います。

1. [Web エージェント OpenID プラグインを有効にします](#) (P. 456)。
2. [OpenID フォーム認証情報コレクタをカスタマイズします](#) (P. 457)。
3. [OpenID プロバイダ設定ファイルを変更します](#) (P. 460)。
4. [エージェント設定オブジェクトを変更します](#) (P. 463)。
5. [FCCCompatMode パラメータを無効にします](#) (P. 464)。
6. [OpenID 認証方式を設定します](#) (P. 465)。

Web エージェント OpenID プラグインの有効化

OpenID プラグインは Web エージェント設定ファイル (WebAgent.conf) で参照されます。プラグインは、エージェントが OpenID プロバイダと通信し、ポリシー サーバに OpenID プロバイダ認証レスポンスを伝えられるようにするために必要です。

エージェント所有者に連絡し、必要なプラグインを有効にするように指示します。

以下の手順に従います。

1. Web エージェント ホスト システムにログインします。
2. Web エージェント設定ファイルを開きます。

注: ファイルのデフォルトの場所は Web サーバ タイプによって異なります。

- IIS

`agent_home¥bin¥IIS`

`agent_home`

エージェント インストール パスを指定します。

- Oracle iPlanet (iPlanet/SunOne)

`web_server_home/https-hostname/config`

`web_server_home`

Web サーバのインストール パスを指定します。

- Apache、IBM HTTP サーバ、および Oracle HTTP サーバ

`web_server_home/conf`

`web_server_home`

Web サーバのインストール パスを指定します。

- (Windows) Domino

`C:¥lotus¥domino`

- (UNIX) Domino

`$HOME/notesdata`

3. OpenID プラグインをロードする行のコメントを外します。

例：

```
#LoadPlugin="C:¥Program Files¥CA¥webagent¥bin¥OpenIDPlugin.dll"
```

4. ファイルを保存します。
5. Web サーバを再起動します。

OpenID フォーム認証情報コレクタのカスタマイズ

サンプル OpenID FCC は Web エージェント インストールに含まれています。FCC はユーザが次の方法で認証できるようにするために必要です。

- OpenID プロバイダのユーザ名の入力。
- 完全な OpenID 識別子の入力。

デフォルトでは、FCC は多くの OpenID プロバイダを表示します。エージェント所有者に連絡し、FCC を変更して、保護されているアプリケーションがサポートするプロバイダのみを表示するように指示します。

以下の手順に従います。

1. Web エージェント ホスト システムにログインします。
2. 以下の場所に移動します。

```
agent_home¥samples¥forms
```

```
agent_home
```

Web エージェントのインストールパスを指定します。

3. テキストエディタで以下のファイルを開きます。

```
openid.fcc
```

4. FCCを確認します。必要とする OpenID プロバイダが利用可能かどうか、またはプロファイルを追加する必要があるかどうかを判定します。デフォルトのプロバイダは、以下のセクションにあります。

```
var providers_large  
var providers_small
```

5. コメントして、FCC から不要なプロバイダを削除します。プロバイダ名でコメントを開始します。プロファイルの終わりでコメントを終了します。

例 :

```
/*google : {  
    name : 'Google',  
    url : 'https://www.google.com/accounts/o8/id'  
},*/
```

6. プロバイダを追加する必要がある場合は、**large** または **small** プロバイダセクションでカスタム プロバイダ ID を探します。

例 :

```
*/*,  
myprovider : {  
    name : 'MyProvider',  
    label : 'Enter your provider username',  
    url : 'http://ca.com/{username}/',  
    image : 'images/image.png'  
}*/
```

注: 個別のプロバイダセクションは FCC が表示するプロバイダアイコンのサイズに対応します。

- **large** セクション内のアイコンのサポートされているサイズは、100 ピクセル x 60 ピクセルです。FCC は 5 つまでの大きいアイコンを表示できます。
 - **small** セクション内のアイコンのサポートされているサイズは、24 ピクセル x 24 ピクセルです。FCC は 11 個までの小さいアイコンを表示できます。
- a. 以下の文字を削除して、新しいプロバイダを追加します。

```
/*  
*/
```

- b. ラベルと名前の値を更新します。ラベル値は、ユーザがプロバイダ アイコンをクリックした後に表示されるテキストを決定します。

例：

```
myprovider : {  
  name : 'Foward Inc',  
  label : 'Enter your Forward Inc user name',  
  url : 'http://ca.com/{username}/',  
  image : 'images/image.png'  
}
```

注: Forward, Inc. は架空の会社名であり、厳密に説明のみを目的として使用されています。現存する会社を参照するものではありません。

- c. URL 値を更新します。URL 値は OpenID ユーザ識別子を表します。ポリシー サーバは、OpenID プロバイダにユーザの識別子を転送します。

例：

```
myprovider : {  
  name : 'Foward Inc',  
  label : 'Enter your Forward Inc user name',  
  url : 'http://{username}.forwardinc.com/'  
  image : 'images/image.png'  
}
```

- d. イメージ値を更新します。イメージ値は、FCC で表示するプロバイダ アイコンの場所を表します。

例：

```
myprovider : {  
  name : 'Foward Inc',  
  label : 'Enter your Forward Inc user name',  
  url : 'http://{username}.forwardinc.com/'  
  image : 'images/forwardinc.png'  
}
```

7. デフォルトでは、FCC は、プロバイダ ID が FCC で設定され有効にされる順序でプロバイダ アイコンを表示します。アイコンの順序を変更する場合は、それに応じてプロバイダ ID の順序を調整します。

重要: デフォルト プロバイダ ID には以下のイメージインデックス プロパティが含まれます。

imageidx

プロパティを削除したり変更しないでください。プロパティは、FCC が正しいプロバイダ アイコンを表示することを確認します。

8. スクリプトを保存します。
9. Web サーバを再起動します。

OpenID プロバイダ設定ファイルの変更

製品には、OpenID プロバイダ設定ファイルがあります。ファイルは、保護されているアプリケーションがサポートする各プロバイダの設定の詳細を参照する必要があります。ファイルに正しい設定が含まれていないと、認証は失敗します。

- デフォルトでは、ファイルには、OpenID FCC が利用可能にするすべてのプロバイダのサンプル設定が含まれます。サンプル設定を確認し、必要に応じてそれらを変更します。

重要: 値はサンプルのみです。認証方式を展開する前に、OpenID プロバイダでの設定をすべて確認することをお勧めします。

- FCC にプロバイダを追加した場合は、プロバイダの設定を追加します。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. 以下の場所に移動します。

```
siteminder_home¥config¥properties  
siteminder_home
```

ポリシー サーバのインストールパスを指定します。

3. 以下のいずれかを実行します。

- デフォルトのプロバイダ設定ファイルを開きます。

```
Openidproviders.xml
```

- デフォルト設定ファイルをコピーして、別のインスタンスを作成します。設定するそれぞれの OpenID 認証方式で、その固有のプロバイダ設定ファイルを使用できます。

例: 認証方式の 1 つのインスタンス用に連邦政府の Identity, Credential, and Access Management (ICAM) コンプライアンスを有効にでき、別のインスタンスでは ICAM コンプライアンスを無効にできます。

4. ファイルを確認し、必要とする OpenID プロバイダ設定が利用可能かどうか、または設定を追加する必要があるかどうかを判定します。
5. 設定を追加する必要がある場合は、以下の手順に従います。

- a. 既存の OpenID プロバイダ ノードおよびその子ノードをすべてコピーします。すべての必須ノードおよびオプションのノードは、以下のノード内に含まれています。

```
<OpenIDProvider>  
</OpenIDProvider>
```

- b. 新しい OpenID プロバイダ ノードおよびその子ノードをすべて以下のルート ノードに追加します。

```
<TrustedOpenIDProviders>  
</TrustedOpenIDProviders>
```

6. 以下のノードの説明を使用して、認証方式でサポートする各プロバイダの設定を設定します。

OpenIDProvider RequestType="value"

プロバイダの設定の始まりを示します。

RequestType

(オプション) プロバイダがサポートする方式のタイプを指定します。

有効な値： ax または sreg。

デフォルト： ax。

ProviderName

サービスをホストする OpenID プロバイダの URL を指定します。値には、プロバイダ URL のカンマ区切りリストを含めることができます。

Required Claims

OpenID プロバイダが認証リクエストの一部として返すクレームを指定します。プロバイダが必要なすべてのクレームを提供できない場合、認証は失敗します。このノードは少なくとも 1 つのクレーム ノードを必要とします。

クレーム

個々の必要なクレームを定義します。

URI

OpenID プロバイダ クレームの URI フォームを指定します。ポリシー サーバはこの値を使用して、認証リクエストを構築します。

重要: ユーザ ディレクトリ内のユーザ属性に必要な最初のクレームの値がマップされていることを確認してください。ポリシー サーバはプロバイダ認証レスポンスに基づく最初の必要なクレームの値を判定します。ポリシー サーバは、クレーム値に一致するユーザを見つけるためにポリシー ドメインのユーザ ディレクトリをすべて検索します。ポリシー サーバでクレーム値をユーザ属性にマップできないと、認証に失敗します。

値: 値はプロバイダがサポートするスキーマのタイプに従う必要があります。

エイリアス

(オプション) URI ノード値のわかりやすい名前を定義し、URI が格納または参照されないようにします。システムは、クレームを識別するためにエイリアスを使用します。

値: 任意の文字列。

例: セッション ストアでユーザの最初の名前を返す URI を格納する代わりに、システムはフルネームとしてクレーム名を参照できます。

注: システムは、セッション ストアに格納されるエイリアスに以下のプレフィックスを追加します。

smopenidclaim

Optional Claims

(オプション) OpenID プロバイダが認証リクエストの一部として返すオプションのクレームを指定します。プロバイダがオプションのクレームを提供できない場合、認証は失敗しません。このノードは少なくとも 1 つのクレーム ノードを必要とします。

Pape

(オプション) ICAM コンプライアンスに必要なプロパティを定義します。ICAM コンプライアンス用の認証方式を設定している場合、このノードおよびすべての子ノードが必要です。

max_auth_age

(オプション) OpenID プロバイダ ユーザセッションが有効な期間を指定します。ユーザセッションが有効な場合、OpenID プロバイダはプロバイダ固有の Cookie を使用して、保護されているリソースに対してユーザを認証します。セッションが期限切れの場合、ユーザは再認証するように指示されます。

測定単位： 秒

デフォルト： 0

デフォルト値を残す場合、ユーザは有効なセッションにかかわらず OpenID プロバイダに対して認証する必要があります。

値： 値は正の整数である必要があります。

ポリシー

(オプション) OpenID プロバイダが従う必要がある ICAM ポリシーのカンマ区切りリストを指定します。プロバイダがコンプライアンス レベルに従っていないと、認証は失敗します。

7. ファイルを保存して閉じます。

エージェント設定オブジェクトの変更

ignore 拡張 ACO パラメータのデフォルト設定では、OpenID FCC は表示されません。管理 UI を使用して、デフォルト設定を変更します。

以下の手順に従います。

1. [インフラストラクチャ]-[エージェント設定オブジェクト]をクリックします。
2. FCC にユーザ リクエストをリダイレクトするエージェント用のエージェント設定オブジェクトを特定します。
3. 編集アイコンをクリックして、オブジェクトを開きます。
4. 以下のパラメータを探します。

IgnoreExt

5. 編集アイコンをクリックし、以下の値を追加します。

- .css

- .js

注: 値はカンマで区切ります。

例：

.class, .gif, .jpg, .jpeg, .png, .fcc, .scc, .sfcc, .ccc, .ntc,
.css, .js

6. [OK] をクリックします。
7. [サブミット] をクリックします。

エージェント設定オブジェクトは除外されたリソース拡張で更新されます。

FCCCompatMode パラメータの無効化

エージェントは、旧バージョンの製品との下位互換性のため、**FCCCompatMode** 設定パラメータを使用します。新しいバージョンの製品（12.52 SP1 など）については、特定の機能を使用する場合、セキュリティを向上させるためにこのパラメータを無効にする必要があります。

注：この手順では、ポリシー サーバでエージェント設定オブジェクトを使用して、中央でエージェントを設定することを想定しています。代わりにローカル エージェント設定方式を使用する場合は、**LocalConfig.conf** ファイル内の **FCCCompatMode** パラメータを無効にします。

以下の手順に従います。

1. 管理 UI から、[インフラストラクチャ] - [エージェント設定オブジェクト] をクリックします。
2. 目的のエージェント設定オブジェクトの隣の編集アイコンをクリックします。

3. 以下のパラメータを探します。

FCCCompatMode

4.x の Web エージェントまたはサードパーティのアプリケーションによって保護されているリソースに対してフォームを提供するよう FCC/NTC を有効にします。

注: SMUSRMSG は、FccCompatMode が yes に設定されている場合のみ、カスタム認証方式でサポートされます。

制限: yes、no

デフォルト: (従来のエージェント) Yes

デフォルト: (フレームワーク エージェント) No

重要: このパラメータを no に設定すると、Netscape ブラウザのバージョン 4.x のサポートが削除されます。

4. 値が no に設定されていることを確認します。値が yes の場合は、以下の手順に進みます。
 - a. パラメータの左側の [編集] アイコンをクリックします。
[パラメータの編集] ダイアログ ボックスが表示されます。
 - b. [値] フィールドを強調表示し、「no」と入力します。
 - c. [OK] をクリックします。
[パラメータの編集] ダイアログ ボックスが閉じます。
 - d. [サブミット] をクリックします。

FCCCompatMode パラメータが無効になり、確認メッセージが表示されます。

OpenID 認証方式の設定

管理 UI を使用して、OpenID 認証方式オブジェクトを設定します。

ネットワークに複数の Cookie ドメインが含まれ、それぞれの Cookie ドメインに認証方式が必要な場合は、Cookie ドメインごとに個別の認証方式オブジェクトを設定します。

注: Solaris プラットフォーム上での操作の場合には、**sun.security.provider.Sun** プロバイダが最初のプロバイダとして登録されるように、ディレクトリ *jre_root/lib/security* にある *java.security* ファイルを変更します。

以下の手順に従います。

1. [インフラストラクチャ] - [認証] - [認証方式] をクリックします。
2. [認証方式の作成] をクリックします。
3. 新しいオブジェクト オプションを選択し、[OK] をクリックします。
[認証方式の作成] ページが表示されます。
4. 名前および保護レベルを入力します。
5. [認証方式タイプ] リストから OpenID テンプレートを選択します。
方式固有の設定が表示されます。
6. その他のパラメータを設定し、[サブミット] をクリックします。

重要: プロキシ設定または後処理チェーン設定のいずれかを変更する場合は、ポリシー サーバを再起動します。

OpenID 認証方式を設定するためのタスクを完了しました。認証方式をポリシー レルムおよびエンタープライズ ポリシー管理 (EPM) アプリケーション コンポーネントにバインドできます。

レガシー フェデレーション認証方式

以下のフェデレーション認証方式が利用可能です。

- SAML Artifact 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関する詳細については、「レガシー フェデレーションガイド」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](#)」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 \(P. 748\)](#)」を参照してください。

- ポリシー サーバと、インパーソネーションの実行ユーザと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。

- 以下のデフォルト HTML フォーム ライブラリ（認証処理を取り扱う）がポリシー サーバにインストールされます。
 - Windows では smauthimpersonate.dll。
 - Solaris では smauthimpersonate.so。

これらのライブラリでは、認証処理を取り扱います。これらのファイルは、ポリシー サーバをインストールすると、自動的にインストールされます。

注: ディレクトリ マッピングはインパーソネーション（偽装）をサポートしません。 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。

詳細情報:

[ユーザディレクトリ](#) (P. 157)

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が作成されていることを前提とします。 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。

以下の手順に従います。

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] をクリックします。
3. [認証方式の作成] をクリックします。

[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。

[OK] ボタンをクリックします。
4. 名前および保護レベルを入力します。

5. [認証方式のタイプ] 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。

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. 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。

認証方式が保存され、レルムに割り当て可能になります。

第 9 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 クライアント証明書認証方式の証明書マッピング \(P. 471\)](#)

[X.509 クライアント証明書認証方式における証明書の有効性チェック \(P. 480\)](#)

X.509 クライアント証明書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「サブジェクト名」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 \(P. 157\)](#)

証明書マッピングの設定

ユーザの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する方法を決定するには、証明書マッピングを設定します。

以下の手順に従います。

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] をクリックします。
3. [証明書マッピングの作成] をクリックします。

4. 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: ポリシーストアとしてリレーショナルデータベースを使用する場合、発行者 DN は 255 文字以下にする必要があります。ポリシーストアとして LDAP ディレクトリを使用する場合は、その特定のディレクトリに対する文字制限を確認します。

5. 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] オプションを使用すると、この確認が必須になります。

注: 証明書が格納されるユーザレコードの属性は「**usercertificate**」と命名されます。

6. 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、以下のいずれかのプロパティ使用しマッピングを適用することにより、正しいユーザエントリを特定できます。
 - 単一属性
 - カスタムマッピング式
 - ユーザ証明書からの対象名全体
7. リストから属性名を選択します。
8. 証明書マッピングをテストするために [テスト] をクリックします。

9. (オプション) [CRL チェックの実行] を選択し、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。

10. [サブミット] をクリックします。

証明書は選択したユーザディレクトリとマップされます。

詳細情報:

[X.509 クライアント証明書認証方式における証明書の有効性チェック](#) (P. 480)

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。

以下の手順に従います。

1. 証明書マッピングを開きます。
2. [マッピング] セクションで [テスト] をクリックします。
[証明書マップテスト] ペインが開きます。
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。

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。カスタム マッピング式を使用すると、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: 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 マッピング式が役立ちます。

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。この構文は、証明書から取得する情報と、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}
```

詳細情報:

[証明書ベースの認証テスト](#) (P. 847)

ユーザ DN 検索の不要な先頭文字列および終端文字列をカスタム証明書マッピングから省略する

LDAP ユーザディレクトリにカスタム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、LDAP ユーザ DN 検索の先頭文字列および終端文字列が誤って含まれます。（「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のみが必要です。）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。

LDAP 構文検索フィルタにおける論理演算子のサポートの設定

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 キーを有効にします。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. Navigate to
HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥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用できるようになります。

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¥Wow6432Node¥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用できるようになります。

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

前の例では、CN に複数の属性が含まれています。この構文は、「CN.CN」または「CN.UID」を指定することにより、どのコンポーネントの CN を取得して、UserDN の CN に適用するかを示しています。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。

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

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（たとえば、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 のこれらの属性は、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ような表記規則を追加します。

- 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

たとえば、証明書のサブジェクト DN に以下の属性が含まれている場合、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。

`CN=user,ou=dev,sn=1234,sn=2345,sn=3456,o=company,c=us`

したがって、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 番目の番号にマップできます。その場合は、次の表記法を使用します。

`%{attribute_name2/attribute_name}`

前の例では、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 がない場合は CN を使用できることを示す場合は、「`%{SN/CN}`」と入力できます。

X.509 クライアント証明書認証方式における証明書の有効性チェック

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 サーバは、単一のユーザに関するリクエストを OCSP レスポンドに送信します。OCSP レスポンドは、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 \(P. 503\)](#)

有効性チェックを実行するための前提条件

ユーザ証明書を検証するには、**X.509 クライアント証明書認証方式**を設定して、保護されているリソースをユーザが要求したときにユーザを認証します。

[X.509 クライアント証明書認証方式](#) (P. 416)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 を取得します。

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 を取得します。

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 を無効と見なします。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 サーバは CRL をキャッシュします。ポリシー サーバのデフォルトのキャッシュ サイズは最大 2 MB です。CRL がデフォルトのキャッシュ サイズを超える場合は、最高 1 GB までキャッシュ サイズを増やしてください。キャッシュ サイズを増やすには、MaxCRLBufferMB レジストリ キーを追加します。

以下の手順に従います。

1. ポリシー サーバにアクセスし、ご使用のオペレーティング プラットフォームの手順に従ってください。

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¥Wow6432Node¥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。

測定単位 : 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. 以下のいずれかの操作を実行します。

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トリから 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。

CDP 拡張は、CRL を特定するために複数のソースを指定できます。ソースにはそれぞれ、CRL を特定するための情報がすべて含まれます。取得に関するそれぞれのオプションは、ポリシー サーバが CRL を確実に取得するのに役立ちます。CDP 拡張内のソースには次のものが含まれます。

- 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。この安全な接続を行うために、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。有効な証明書がない場合、HTTPS サーバへの接続は失敗します。

HTTPS 接続のための 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。証明書が PEM 形式にない場合は、OpenSSL コマンドラインユーティリティを使用して、それを変換します。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 で最初に取得された 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、2 番目の CRL、またその次というように取得を試行します。ポリシー サーバは、このように取得が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** の間のフェー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** にフェールオーバーします。

注: **OCSP** の設定ファイルで [フェールオーバー](#) (P. 503) 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、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** という名前の属性で設定します。この属性は、多値属性です。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。この確認により、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できます。

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、そのディレクトリへの接続を設定します。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。

以下の手順に従います。

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] をクリックします。
[証明書マッピング] ページが表示されます。
3. 発行者 DN 名をクリックし、証明書マッピングを選択します。
[証明書マッピングの表示] ページが表示されます。
4. [変更] をクリックします。
設定とコントロールがアクティブになります。
5. [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。
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。

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] 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] 内の DN にエントリ ポイントの値を入力します。

[CRL ディレクトリ] 内の DN に指定された値は、ポリシー サーバが CRL を検索するために CRL ディレクトリを調べる DN です。この値は、LDAP ディレクトリが CRL ディレクトリとして選択されている場合にのみ有効です。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] 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 つといえるのが、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。ポリシー サーバは OCSP チェックのみを実行し、ポリシー サーバで発行者 DN が見つかった場合は、証明書が有効であると見なします。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、OCSP の信頼されたレスポンド証明書を格納します。この証明書により、SiteMinder に返された OCSP レスポンスの署名が検証されます。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。証明書のコレクションを格納している場合は、すべての証明書を格納するために、ディレクトリ エントリには必ず複数値のバイナリ属性を使用してください。

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、証明書の場所またはその集合を `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нда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。このキー/証明書のペアを証明書データ ストアに格納します。

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。このファイルは、1 つ以上の `OCSPResponder` レコードを含む ASCII ファイルです。

`SMocsp.conf` ファイルは `siteminder_home/config` ディレクトリに存在する必要があります。設定を容易にするため、サンプル ファイル (`SMocsp.Sample.conf`) がポリシー サーバと共に `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、ファイル名の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示されています。すべての設定が必須とは限りません。

```
[
OCSPResponder
IssuerDN C=US,ST=Massachusetts,L=Boston,O=,OU=QA,CN=Issuer,E=user@domain.com
AlternateIssuerDN C=US,ST=New York,L=Islandia,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
ResponderCertAlias cert1
ResponderGracePeriod 0
]
```


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。
- ファイル内の設定が空白のままの場合、ポリシー サーバはエラー メッセージを送信します。メッセージは、エントリが無効であることを示します。ポリシー サーバは設定を無視します。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。

AlternatelIssuerDN

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 設定で指定される) 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」に設定します。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 設定または ResponderLocation 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、ポリシー サーバは検証用に証明書の AIA 拡張を使用します。この拡張は、証明書内に存在する必要があります。
- AIAExtension が YES に設定され、ResponderLocation 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」に設定される場合、ポリシー サーバは ResponderLocation 設定を使用します。AIA 拡張が存在しても、ポリシー サーバはそれを無視します。

「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認証情報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

エイリアス

任意です。OCSP リクエストに署名するキー/証明書ペアのエイリアスを指定します。OCSP リクエストは、OCSP レスポンダに送信されます。このキー/証明書ペアは、SiteMinder 証明書データ ストア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ールオーバーするように、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

ResponderCertAlias

(フェデレーションに対してのみ必須) OCSP レスポンスの署名を検証する証明書のエイリアスを指定します。ポリシー サーバによってレスポンスの署名の検証を実行するには、この設定のエイリアスを指定してください。指定しない場合、CA 発行者には利用できる OCSP 設定がありません。

注: ポリシー サーバは、X.509 証明書の認証に対してこの設定を使用しません。

エイリアスを指定する文字列を入力します。

SMocsp.conf ファイルがロードされると、それぞれの発行者に OCSP 設定があるかどうかを確認できます。以下のメッセージはステータス メッセージの一例です。

SMocsp.conf ファイルがロードされました。

以下の発行者エイリアスに対して OCSP 設定が追加されました:

ocspcert
ocspcert1
ocspcert2

ステータス メッセージ内の発行者エイリアスは、CA 証明書をデータストアに追加するときに、管理 UI で指定したエイリアスを参照します。発行者エイリアスがリストにない場合は、SMocsp.conf および cds.log ファイルを確認します。ログファイルは `siteminder_home¥log` にあります。

RevocationGracePeriod

(オプション)失効した後の証明書の無効化を遅らせるための期間(日数)を指定します。OCSP 猶予期間によって、設定が突然機能しなくならないように証明書を更新する時間が与えられます。値 0 は、証明書が破棄されると直ちに無効になることを示します。

このフィールドに値を指定しない場合、ポリシー サーバは 管理 UI で設定するデフォルトの廃棄猶予期間を使用します。デフォルトの設定は、[インフラストラクチャ] - [X509 証明書管理] - [証明書管理] を選択して確認できます。

デフォルト : 0

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. 416)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нда レコードを設定してください。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 \(P. 503\)](#)
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) OSCP リクエストに署名するようポリシー 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 OSCP レスポнда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 OSCP レスポнда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 OSCP 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。

以下の手順に従います。

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 リクエストを署名するための前提条件の設定(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。

以下の手順に従います。

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. 489) を設定します。
2. 証明書データストアに、リクエストを署名するキー/証明書ペアを追加します。このタスクは、管理 UI を使用して実行します。

キー/証明書のペアを追加する際に、エイリアスを指定します。ポリシー サーバはエイリアスを使用して、証明書データストア内の証明書エントリを識別します。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。別のエイリアスで同じ証明書を格納しようとした場合、失敗します。同じ署名証明書を使用する場合は、複数のレスポンダに対して同じエイリアスを使用します。
- 指定するエイリアス値は、`SMocsp.conf` ファイルで設定するエイリアスの値に一致させる必要があります。

- 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。
- エイリアスには以下の文字を使用する必要があります。
 - すべて小文字の USA ASCII にする。
 - 英数文字を使用する。

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。

以下の手順に従います。

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нда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ндаレコードにない場合、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

エイリアス

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、いずれか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 レスポンダ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 レスポンダが「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 機能を見捨て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。
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。

以下の手順に従います。

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無効にするには、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 レスポンス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 レスポンス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 OCSPP および CRL のメッセージに加えて、ポリシー サーバでは、フェイルオーバー イベントに関する情報もログに記録されます。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、OCSPP 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 OCSPP 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。

OCSPP 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
```

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 10 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トアの概要 \(P. 509\)](#)

[クレデンシャルセレクトアのユースケース \(P. 510\)](#)

[このユースケースのクレデンシャルセレクトアソリューション \(P. 513\)](#)

[フロントエンド認証方式の確立 \(P. 513\)](#)

[失敗した認証の管理 \(P. 523\)](#)

[バックエンド処理のセットアップ \(P. 524\)](#)

[サンプルアプリケーションの保護 \(P. 532\)](#)

[クレデンシャルセレクトアソリューションのテスト \(P. 539\)](#)

クレデンシャルセレクトアの概要

認証情報セレクトアは、強力な認証ソリューションです。ユーザは認証情報セレクトアによって、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。

認証情報セレクトア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で使用できます。

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認証情報を選択できます。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。

Password And/Or Certificate Username: Password: ☐ Use an X.509 client certificate Login	SecurID Authentication Username: SecurID PIN: Login
Windows Authentication Login	SafeWord Authentication Username: Login

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、ユーザが提供する認証情報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。

詳細情報

[パスワード/証明書の認証によるアクセスのリクエスト](#) (P. 511)

[SafeWord 認証によるアクセスのリクエスト](#) (P. 512)

[SecurID 認証によるアクセスのリクエスト](#) (P. 512)

[Windows 認証によるアクセスのリクエスト](#) (P. 512)

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 ボックスのみをオンにした場合、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 サーバが証明書を認識すると、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。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、提供される認証情報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認証情報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、ユーザによって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

保護されたリソースを要求したときに、ユーザが **Windows** ドメインにログイン中の場合、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 the Windows domain authentication
```

ユーザが **Windows** ドメインにログインしていない場合、**Windows** ドメイン認証情報を要求されます。

SecurID 認証によるアクセスのリクエスト

保護されたリソースを要求したときに、**SecurID** 認証において有効な **Username** と **SecurID PIN** をユーザが提供した場合、以下のメッセージが表示されます。

```
Greetings, SampleUser!  
Your authentication level is 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認証情報を決定します。**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、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。
 - 認証情報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認証情報のタイプごとに 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、認証情報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、FCC (フォーム認証情報コレクタ)、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。

注: この `selectlogin.fcc` は認証情報セレクタ用のサンプルです。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。

フォーム認証情報コレクタ(FCC)の使用

FCC 形式は、Web エージェントが使用する独自の形式であり、認証情報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 % を含めることができ、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。FCC 本文は、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。

本文には置換文字を含めることができ、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。

`$$smagentname$$`

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。

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。フロントエンド認証方式では `selectlogin.fcc` ファイルを使用して、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。認証方式のために定義された FCC ファイルは、`%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。例：

```
<!-- 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: `@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 が提供される認証情報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`

ユーザが選択する認証情報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は、選択された認証方式によって FCC が収集する認証情報が処理されます。選択された認証方式により、ユーザの認証レベルを含むユーザセッションが確立されます。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、ユーザ認証コンテキストを確認でき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。以下のリストに、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、エージェントは GET リクエストが取得するものと同じ `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。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、ユーザが異なる認証情報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、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 認証情報を直接 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できます。ただし、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、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します。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。

General	
Name: AuthChannel	Description: Auth scheme for
Scheme Common Setup	
Authentication Type Style: HTML Form Template	
Protection Level: 1	[1-1,000,higher i
☐ Password Policies enabled for this Authentication Scheme	
Scheme Setup	
Web Server Name auth.sample.com	
☐ Use SSL Connection	
Target /siteminderagent/forms/selectlogin.fcc	
☐ Allow this Scheme to save credentials	

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ため、AuthChannel 認証方式の保護レベルは 1 に設定します。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ユーザが選択する認証方式によって、そのユーザの実際の保護レベルが決まります。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

サンプル アプリケーションが置かれている Web サーバを指定します。

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、その認証方式に対して指定されるターゲット パラメータで指定された URL にリダイレクトされます。

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認証情報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。この動作を設定するには、各認証方式のターゲット パラメータとして `selectlogin.fcc` を指定します（該当する場合）。

認証情報の特定の選択で `selectlogin.fcc` ファイル以外の FCC のファイルに認証情報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 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";
```

ターゲット パラメータのない方式（基本認証方式など）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

クレデンシャルセレクトアを使用するには、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。

- ログイン認証情報の選択肢ごとに 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許可するポリシー。

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認証情報ごとに 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します。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、使用する認証情報のタイプを選択できるようになります。

認証情報のタイプごとに、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

認証情報セレクトアのバックエンド処理を表すには、ポリシードメインを設定します。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を 1 つずつ定義します。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
```

以下のいずれかの値を指定できます。

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。



レルムを保護する Web エージェントは同一にできますが、同一にする必要はありません。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。

認証情報セクタ機能の一部としてレルムを保護する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。

- 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、`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る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。エージェントが URL クエリ パラメータを無視するのを防ぐために、`IgnoreQueryData` パラメータを `No` に設定してください。

バックエンド ポリシー ドメインのルールの作成

`BackendAuth` ポリシー ドメインとして設定する各レルムには、以下の 2 つのルールを設定します。

- `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] フィールドの値としてアスタリスク (*) を設定します。

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、その値は、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 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。

Attribute WebAgent-OnAuthAccept-Session-AuthContext				
Value Type Text				
Response Form Response				
Attribute Setup				
<table border="1"> <tr> <th>Attribute Kind</th> <th>Attribute Fields</th> </tr> <tr> <td> ☒ Static ☐ User Attribute ☐ DN Attribute ☐ Active Response Allow Nested Groups ☐ </td> <td> Variable Name Variable Value username/password HTTP Variable Name Response Variable Name </td> </tr> </table>	Attribute Kind	Attribute Fields	☒ Static ☐ User Attribute ☐ DN Attribute ☐ Active Response Allow Nested Groups ☐	Variable Name Variable Value username/password HTTP Variable Name Response Variable Name
Attribute Kind	Attribute Fields			
☒ Static ☐ User Attribute ☐ DN Attribute ☐ Active Response Allow Nested Groups ☐	Variable Name Variable Value username/password HTTP Variable Name Response Variable Name			
Advanced				
Script username/password				

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。レガシー フェデレーション 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、ハッシュ値、または両方の SAML アサーション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。

Domain BackendAuth

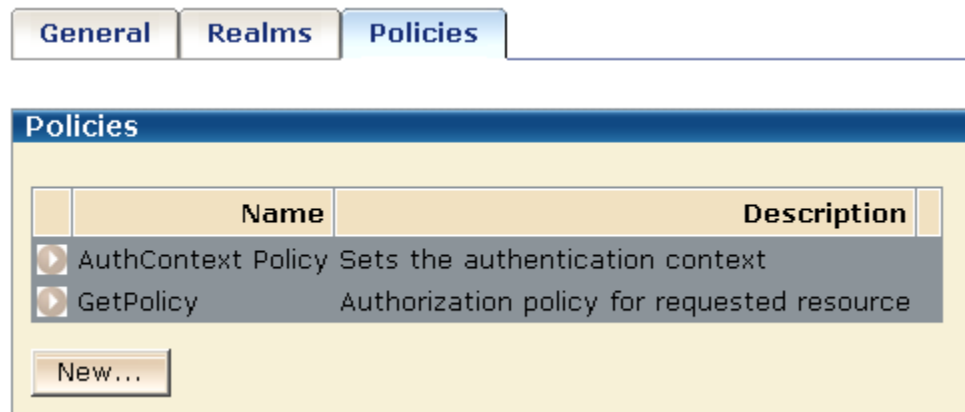
Attribute List		
	Agent Type Attribute Name	Value
	WebAgent-OnAuthAccept-Session-AuthContext	username/password

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 つのポリシーを設定します。

- ユーザ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。



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できます。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、認証情報セ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ません。したがって、ユーザが選択する認証情報が記憶されます。

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 ポリシーで記述することにより、認証コンテキストを上書きから保護します。

レガシー フェデレーション が使用されている場合は、AuthContext と呼ばれるユーザ コンテキスト変数を作成できます。この変数を AuthContext ポリシーで使用すれば、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。

General	
●Name:	AuthContext
Description:	

Attributes	
Variable Type	UserContext
Variable Information	
Return Type	String

Context Item	
Item	User Property
●Property	SM-AUTHENTICATIONCONTEXT
●DN	
●Buffer Size	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。

General	Users	Rules	Expression
Expression			
Expression (AuthContext=="")			
Edit			

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。

この保護ポリシーを設定して、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

認証情報セレクタを使用するには、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 **SiteMinder** アプリケーションのレルムを設定します。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、異なるレルムに含まれる 4 種類のリソースがあります。それぞれのレルム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。

Modify Domain: *SampleApp*

General Policies **Realms**

Realms

	Name	Description	
	SampleApp		
	Limited	Limited to users authenticated with SafeWord	
	Protected	Auth level must be 5 or higher	
	Public	Unprotected resources	

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。

General	
●Name:	Protected

Resource	
●Agent	sampleagent
Resource Filter	/protected/
Effective Resource	sampleagent/protected/
Default Resource Protection	☒ Protected ☐ Unprotected
Authentication Scheme	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。

認証情報セクタ機能の一部としてレルムを保護する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。

- 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、`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る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。エージェントが URL クエリ パラメータを無視するのを防ぐために、`IgnoreQueryData` パラメータを `No` に設定してください。

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。

レガシー フェデレーション が使用されている場合は、AuthLevel という名前のユーザ コンテキスト変数を作成できます。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。



Create Policy: *GetProtected*

Modify Domain: *BackendAuth* > Create Policy: *GetProtected*

General Users Rules **Expression**

Expression

Expression (AuthLevel >= 5)

Edit

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。

General

Name: SampleResponse

Description: Username, Auth level, Auth Cont

Attributes

Agent Type

☐

Radius

☒SiteMinder

Agent Type Web Agent

Domain SampleApp

Attribute List

	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、認証情報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。

以下の手順に従います。

1. リンクをクリックするなどの方法で、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 つのタイプの認証情報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 11 章: アプリケーションオブジェクトでの Web アプリケーションのセキュリティポリシーの定義

このセクションには、以下のトピックが含まれています。

[アプリケーションオブジェクトを使用してリソースを保護する利点 \(P. 541\)](#)

[アプリケーションオブジェクトでの Web アプリケーションのセキュリティポリシーの定義方法 \(P. 543\)](#)

[アプリケーションオブジェクトを使用して、アプリケーションセキュリティポリシーを定義するためのユースケース \(P. 556\)](#)

アプリケーションオブジェクトを使用してリソースを保護する利点

アプリケーションオブジェクトは、SiteMinder に固有の概念やコンポーネントに関する詳しい知識がなくても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を提供します。このモデルは、EPM（エンタープライズポリシー管理）とも呼ばれます。

アプリケーションオブジェクトは、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、アプリケーションオブジェクトを作成し、デフォルトが指定されていない環境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。そのため、アプリケーションオブジェクトはより簡単にポリシー設定ができます。アプリケーションの保護をさらにきめ細かく定義できるその他の設定を操作できますが、必須ではありません。

ドメインベース ポリシー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基本ポリシー オブジェクト間の関係も考慮できます。この関係は 管理 UI に反映されており、以下のテーブルで表示されます。

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 ドメインの定義
コンポーネント	レルムの定義
リソース	ルールの指定
アプリケーション ロール	ポリシー ユーザの定義

アプリケーション ロールは、アプリケーション オブジェクトで定義するリソースまたはリソースのグループへのアクセスを持つユーザのセットを定義します。ロールには、以下のプロパティを使用できます。

- ロールには、設定されたユーザ ディレクトリ内のすべてのユーザを含めることができます。
- ロールは、選択したグループ、組織、および一致するユーザ属性を持ったユーザに制限できます。
- 名前付き式または名前のない式を使用して指定できます。

アプリケーション オブジェクト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

アプリケーション オブジェクトのセキュリティ エンフォースメント モデルは、よりドメイン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。ただし、ドメイン固有のコンポーネントは設定には表示されません。

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。

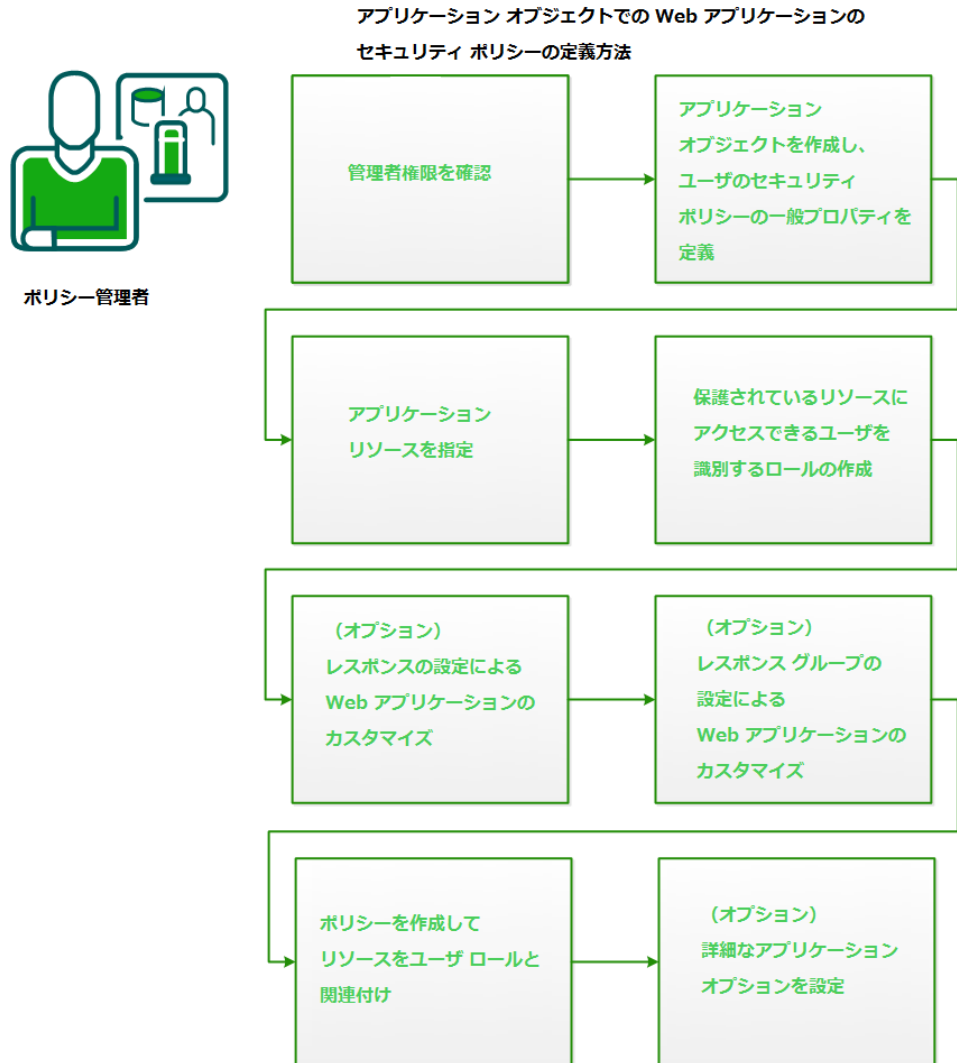
強化された委任機能

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。この機能によって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 オブジェクトでの Web アプリケーションのセキュリティポリシーの定義方法

アプリケーション オブジェクトは、Web アプリケーション（または Web サイト）のための完全なセキュリティ ポリシーを定義する直観的な方法を提供します。アプリケーション オブジェクトは、どのユーザがどのリソースにアクセスできるか決める資格ポリシーを指定するために、リソースとユーザ ロールを関連付けます。

注: アプリケーション オブジェクトはポリシー情報を定義します。この情報はポリシー ドメインおよびそのサブオブジェクトでも設定できます。これには、レルム、ルール、ルール グループ、レスポンス、およびポリシーが含まれます。



アプリケーション オブジェクトで Web アプリケーションのセキュリティポリシーを定義するには、以下の手順に従います。

1. [管理者権限を確認します](#) (P. 545)。
2. [アプリケーション オブジェクトを作成し、セキュリティ ポリシーの一般プロパティを定義します](#) (P. 546)。
3. [アプリケーション リソースを指定します](#) (P. 547)。

4. [保護されているリソースにアクセスできるユーザを識別するロールを作成します \(P. 548\)](#)。
5. [\(オプション\) レスポンスを設定して Web アプリケーションをカスタマイズします \(P. 549\)](#)。
6. [\(オプション\) レスポンス グループを設定して Web アプリケーションをカスタマイズします \(P. 550\)](#)。
7. [ポリシーを作成してリソースとユーザ ロールを関連付けます \(P. 551\)](#)。
8. [\(オプション\) 詳細なアプリケーション オプションを設定します。 \(P. 552\)](#)

管理者権限の確認

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。

必要な権限がない場合は、SiteMinder のスーパーユーザに問い合わせてください。

アプリケーション オブジェクトの作成およびセキュリティポリシーの一般プロパティの定義

アプリケーション オブジェクトを作成し、それが定義するセキュリティポリシーの以下の一般プロパティを設定します。

- 1つ以上のコンポーネント：同様のセキュリティ要件を持った関連するアプリケーション リソースのグループ。通常は、共通の場所に存在するリソース。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などです。
- リソースの使用を許可されているユーザの 1 つまたは複数のディレクトリ。

以下の手順に従います。

1. 管理 UI にログインします。
2. [ポリシー] - [アプリケーション] をクリックします。
3. [アプリケーション] をクリックします。
4. [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] ページが表示されます。
5. アプリケーションの名前と説明を入力します。
6. [コンポーネント] セクションで、同様のセキュリティ要件を持った関連するリソースのグループを 1 つ以上定義します。コンポーネントごとに、以下の手順に従います。
 - a. [コンポーネントの作成] をクリックします。
 - b. コンポーネントの名前を入力します。
 - c. [エージェント/エージェント グループの検索] をクリックします。
 - d. 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。
 - e. [リソース フィルタ] フィールドに保護するリソースのルート URL を入力します。
 - f. リソースをデフォルトで保護するかどうかを指定します。
 - g. リソースをリクエストするユーザの ID を検証するために使用する認証方式を指定します。

7. [ユーザディレクトリ] セクションで、以下の手順を実行して、アプリケーション リソースの使用を許可されているユーザの 1 つまたは複数のディレクトリを選択します。

- a. [追加/削除] をクリックします。

- b. [使用可能なメンバー] 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。

- c. [OK] をクリックします。

選択したユーザディレクトリは [アプリケーションの作成] ページの [ユーザディレクトリ] の下にリスト表示されます。

8. [OK] をクリックします。

9. [サブミット] をクリックします。

アプリケーション オブジェクトが作成されます。

アプリケーション リソースの指定

保護するアプリケーション コンポーネントを定義した後に、保護する各コンポーネント内の特定のリソースを指定します。

以下の手順に従います。

1. [アプリケーションの作成] ページで、[リソース] タブをクリックします。
2. 複数のコンポーネントを作成した場合、[コンテキスト ルートの選択] ドロップダウンリストから保護するリソースのルート URL（[一般] タブの [リソース フィルタ] で指定）を選択します。
3. [作成] をクリックします。
4. リソースの名前を入力します。

5. 保護するリソースを[リソース]フィールドに入力します。特定のファイルを指定するか、または正規表現を使用して、リソース照合の柔軟性を高めます。
[有効なリソース] が更新され、このリソースが含まれます。
6. [リソース] フィールドで正規表現を使用した場合は、[正規表現] オプションを設定します。
7. [アクション] セクションで、ポリシー サーバにリクエストを処理させるために、指定されたリソースに発生する必要があるアクションのタイプを選択します。
[アクション] リストに、選択したアクション タイプに適したアクションが入力されます。
8. 1 つ以上のアクションを選択します。
9. [OK] ボタンをクリックします。
リソースが作成されます。
10. Web アプリケーション内の各リソースに対して手順 2 ～ 9 を繰り返します。

これで、Web アプリケーション リソースが定義されました。

保護されているリソースにアクセスできるユーザを識別するロールの作成

Web アプリケーション コンポーネントおよびリソースを定義した後に、特定のリソースへのアクセス権がある一連のユーザを定義するロールを指定します。

以下の手順に従います。

1. [アプリケーションの作成] ページで、[ロール] タブをクリックします。
2. [ロールの作成] をクリックします。
3. [タイプ「ロール」の新規オブジェクトの作成] (Create new object of type Role) オプションが選択されていることを確認し、[OK] をクリックします。
4. 名前を入力し、必要に応じて、ロールの説明を入力します。

5. そのロールが、設定されたユーザディレクトリ内の「全ユーザ」または「選択されたユーザ」に適用されるかどうかを指定します。

注: 「全ユーザ」オプションを設定すると、「ユーザセットアップ」および「詳細」セクションは適用されず、表示されなくなります。

6. 「ユーザセットアップ」グループボックスで選択することで、ロールのメンバを定義するグループ、組織、およびユーザ属性式を定義します。
7. 「OK」をクリックします。
8. 必要な追加ロールごとに手順 2 ～ 7 を繰り返します。

(オプション)レスポンスの設定による Web アプリケーションのカスタマイズ

レスポンスを設定して、テキスト、ユーザ属性、DN 属性、アクティブレスポンス、または定義された変数のランタイム値をポリシーサーバからエージェントに渡します。Web アプリケーションは、権限を決定するため、またはきめの細かいアクセス制御のために、レスポンスデータを使用してカスタマイズされたコンテンツを表示できます。レスポンスデータは、SiteMinder 設定を変更するため、またはユーザを別のリソースにリダイレクトするためにも使用できます。

以下の手順に従います。

1. 「アプリケーションの作成」ページで、「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。
3. レスポンスの名前を入力します。
4. 1 つ以上のレスポンス属性を作成します。レスポンス属性ごとに、以下の手順に従います。
 - a. 「レスポンス属性の作成」をクリックします。
 - b. 設定する属性タイプを選択します。たとえば、WebAgent-HTTP-Header-Variable 属性タイプなどです。
 - c. 属性の種類を選択します。

「属性フィールド」の詳細は、指定した属性の種類と一致するように更新されます。
 - d. 「属性フィールド」の詳細を入力します。

- e. (オプション) [スクリプト] フィールドで属性を編集します。

注: [詳細] セクションで属性を編集すると、[属性のセットアップ] セクションが閉じます。

- f. [キャッシュ値] (デフォルト) または [値の再計算間隔 (秒)] (Recalculate value every ... seconds) を指定します。

注: 入力できる最大時間制限は 3600 秒です。

- g. [OK] をクリックします。

レスポンス属性が [属性リスト] に追加されます。

- 5. [OK] をクリックします。

レスポンスが作成されます。

(オプション)レスポンス グループの設定による Web アプリケーションのカスタマイズ

1 つのオブジェクトに複数のレスポンスを結合するには、レスポンス グループを設定します。アプリケーション ポリシーを作成すると、そのポリシー内の 1 つのリソース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。

以下の手順に従います。

1. [アプリケーションの作成] ページで、[レスポンス] タブをクリックします。
2. [レスポンス グループの作成] をクリックします。
3. レスポンス グループの名前を入力します。
4. [追加/削除] をクリックします。

[レスポンス グループのメンバ] ページが表示されます。

注: [使用可能なメンバ] 列には、アプリケーション オブジェクトで定義されているすべてのレスポンスとレスポンス グループがリスト表示されます。

5. [使用可能なメンバ] リストから 1 つ以上のレスポンスまたはレスポンス グループを選択し、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。

6. [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。

7. [OK] をクリックします。

レスポンス グループが作成されます。

ユーザ ロールとリソースを関連付けるポリシーの設定

アプリケーション ポリシーでユーザ ロールとリソースを関連付け、各リソースへのアクセスを許可するユーザを定義します。リソースがアクセスされるときに許可するエージェントにデータを返す場合は、ポリシーでレスポンスとリソースも関連付けます。

以下の手順に従います。

1. [アプリケーションの作成] ページで、[ポリシー] タブをクリックします。

[ポリシー] タブには 2 つのテーブルが表示されます。1 つにはリソースとロールがリスト表示され、もう 1 つにはリソースとレスポンスがリスト表示されます。

2. リソーステーブルで、各リソースと関連付けるロールを選択します。選択されたロールのユーザのみ、それらのリソースへのアクセスが許可されます。

- レスポンス テーブルで、各リソースと関連付けるレスポンスとレスポンス グループを選択します。関連するリソースがアクセスされると、選択したレスポンスで定義されているデータが返されます。

注: レスポンスは、ロール/リソース テーブル エントリを選択しないと、レスポンス テーブルにリスト表示されません。

- [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。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。

(オプション) 詳細なアプリケーション オプションの設定

アプリケーション セキュリティ ポリシーの以下の詳細なオプションも設定できます。

- [カスタム属性の設定によるメタデータの追加](#) (P. 552)
- [信頼レベルの設定](#) (P. 553)
- [CA DataMinder コンテンツ分類の設定](#) (P. 554)
- [拡張ポリシー コンポーネントの設定](#) (P. 555)

(オプション) カスタム属性の設定によるアプリケーションに関するメタデータの追加

カスタム属性を定義して、アプリケーションに関する一意な識別メタデータを追加できます。メタデータは、アプリケーションを作成した人の名前やアプリケーションの目的などの情報を追加することで、アプリケーションを説明します。

以下の手順に従います。

- [アプリケーションの作成] ページで、[カスタム属性] タブをクリックします。

[カスタム属性] タブには、既存のメタデータの名前と値を含むテーブルが表示されます。

- [作成] をクリックします。

空白のエントリが [カスタム属性] テーブルに追加されます。

- 追加するメタデータの名前と値を入力します。

4. [作成] をクリックします。

カスタム属性がテーブルに追加され、新しい空白のエントリが追加されます。

5. 追加するカスタム属性ごとに手順 3 と 4 を繰り返します。

アプリケーション内の信頼レベルの設定

SiteMinder がサポートされたリスク分析エンジンに統合される場合、信頼レベルはアプリケーション オブジェクトで利用可能です。信頼レベルは、アプリケーションを拡張して、ユーザ認証の一部として完了するリスク評価の結果を含めます。許可決定を下すとき、ポリシー サーバはこれらの結果を使用できます。

以下のオブジェクトに信頼レベルを適用できます。

- アプリケーション コンポーネント。

アプリケーション コンポーネントで設定する信頼レベルは、コンポーネントと関連付けられるすべてのリソースに適用されます。信頼レベルはデフォルト アプリケーション設定が提供するより高いレベルの精度を表します。アプリケーション コンポーネントの高度な設定を使用して、信頼レベルを適用します。

注: アプリケーション コンポーネントに信頼レベルを適用する場合、ユーザが信頼レベルサポートを有効にする必要があります。詳細については、「SiteMinder 実装ガイド」を参照してください。

- アプリケーション ロール。

アプリケーション ロールの一部として設定する信頼レベルによって、より詳細な許可決定が可能になります。信頼レベルは、リソースにアクセスできるユーザ グループまたはグループをさらに定義するために使用できるアクティブなコンポーネントを表します。信頼レベルは、デフォルトのロール メンバシップが提供するより高いレベルの精度を表します。信頼レベルをロールに追加するには、**SM_USER_CONFIDENCE_LEVEL** SiteMinder が生成する属性を参照する名前付きの式を使用します。

注: アプリケーション ロールへの信頼レベルの適用は、以前のリリースからサポートされており、デフォルトで有効になります。

アプリケーションでの CA DataMinder コンテンツ分類の設定

SiteMinder が CA DataMinder と統合されている場合、コンテンツ分類はアプリケーション オブジェクトと併用できます。コンテンツ分類は、ユーザが要求するコンテンツの種類を含めるためにアプリケーションを拡張します。ポリシー サーバは、CA DataMinder コンテンツ分析の結果を使用して許可の判断を行うことができます。

注: コンテンツ分類をアプリケーション コンポーネントに適用するには、CA DataMinder 統合に対して環境を有効にする必要があります。詳細については、「SiteMinder 実装ガイド」を参照してください。

CA DataMinder コンテンツ分類を以下のオブジェクトに適用できます。

- アプリケーション コンポーネントについては以下のとおりです。
 - アプリケーション コンポーネントに適用するコンテンツ分類は、コンポーネントに関連付けられているすべてのリソースに適用されます。
 - デフォルトでは、CA DataMinder Content Classification Service は、すべてのコンテンツ分類をアプリケーション コンポーネントに対して使用できるようにします。許可の判断を行うときに、ポリシー サーバに 1 つ以上の分類を無視させる場合は、アプリケーション からそれらを削除します。
 - SiteMinder 管理者はコンテンツ分類を管理できません。変更が必要な場合は、CA DataMinder 管理者と変更を調整します。
- アプリケーション ロールについては以下のとおりです。
 - アプリケーション ロールに適用するコンテンツ分類によって、より詳細な許可の判断を行うことができます。コンテンツ分類は、リソースにアクセスできる 1 つまたは複数のユーザ グループを詳細に定義するために使用できるアクティブ コンポーネントを表します。コンテンツ分類は、デフォルト ロール メンバシップが提供する、より高い詳細レベルを表します。
 - ロールとコンテンツ分類の関係は累積的です。ユーザが同じアプリケーション上の複数のロールのメンバである場合は、ロールに関連付けられているすべてのドキュメントにアクセスできます。

アプリケーションの詳細なポリシー コンポーネントの設定

アプリケーション オブジェクト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。
- アプリケーション オブジェクトを使用して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断する必要がある監査者またはその他の担当者。

以下の手順に従います。

1. [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。
3. [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、[詳細設定] をクリックします。
[コンポーネント変更] ページが表示されます。[コンポーネント変更] ページには、ポリシー レルムのセッションおよび詳細機能が含まれます。たとえば、信頼レベル サポートが有効な場合、コンポーネントに最小限の信頼レベルを追加できます。
4. 以下のいずれかを実行します。
 - コンポーネントに対する 1 つ以上の詳細な設定を設定します。
 - レガシー認可ディレクトリ マッピングを選択します。
 - サブ コンポーネントを作成するには、[サブ コンポーネントの作成] をクリックし、[コンポーネントの作成] 画面を開きます。サブ コンポーネントの作成は、サブ レルムまたはネストされたレルムの作成と同等です。
5. 完了後、[OK] をクリックして変更を保存し、アプリケーションの残りの部分の設定を引き続き行ってください。

アプリケーション オブジェクトを使用して、アプリケーション セキュリティ ポリシーを定義するためのユース ケース

以下のユース ケースでは、アプリケーション オブジェクトを使用してアプリケーション セキュリティ ポリシーを定義する方法について説明します。

- [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](#) (P. 556)
- [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. 562)
- [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. 570)
- [CA DataMinder コンテンツ分類に基づくアプリケーション セキュリティ ポリシー](#) (P. 580)

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が作成されていることを前提とします。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。

以下の手順に従います。

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] をクリックします。
[アプリケーション] ページが表示されます。
3. [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] ページが表示されます。
4. アプリケーションの名前と説明を入力します。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特徴のある値を入力します。

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。

5. [コンポーネント] で、コンポーネントの名前を入力し、保護するリソースが含まれ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。

コンポーネント名

ダウンロード

エージェント 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

注: メイン コンポーネントを保存した後でのみ、サブコンポーネントが作成されます。

6. その他の設定はデフォルトのままにします。
7. [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] ページが表示されます。
8. 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。
9. [OK] をクリックします。
[一般] タブに戻ります。
10. [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。

以下の手順に従います。

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

*

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 オプション 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 リソースが指定された後、Web ポータルの登録済みのカスタマのためのロールを作成します。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が作成されていることを前提とします。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。

以下の手順に従います。

1. [ロール] タブをクリックします。
2. [ロールの作成] をクリックします。
3. [タイプ「ロール」の新規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。

ロールは以下に適用されます。

すべてのユーザ

注: このオプションが設定されると、[ユーザ セットアップ] および [詳細設定] グループ ボックスは適用されず、表示されなくなります。

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。

以下の手順に従います。

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でき、実績管理の部分へのアクセスは管理者にのみ許可されます。

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、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 という名前の 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. アプリケーションを作成します。
2. 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。
3. 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。
4. 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。
5. 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します。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します。

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] をクリックします。
[アプリケーション] ページが表示されます。
3. [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] ページが表示されます。
4. [一般] タブをクリックします。
5. [一般] 内の以下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。

6. [コンポーネント] 内の以下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。

コンポーネント名

利点

エージェント 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。

注: メイン コンポーネントを保存した後でのみ、サブコンポーネントが作成されます。

7. 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。

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、以下の値を入力します。

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、以下の値を入力します。

リソース

managebenefits.jsp

5. 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後、特定のリソースへのアクセス権を持つユーザのセットを定義するロールを指定します。すべての従業員のためのロールを作成します。

以下の手順に従います。

1. [ロール] タブをクリックします。
2. [ロールの作成] をクリックします。
[ロールの作成] ペインが表示されます。

3. [作成] オプションが選択されていることを確認し、[OK] 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、以下の値を入力します。

名前

従業員

説明

Acme Financial Services のすべての従業員

ロールは以下に適用されます。

すべてのユーザ

注: このオプションが設定されると、[ユーザ セットアップ] および [詳細設定] グループ ボックスは適用されず、表示されなくなります。

5. [OK] をクリックします。

マネージャ 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後、特定のリソースへのアクセス権を持つユーザのセットを定義するロールを指定します。マネージャ用のロールを作成します。

以下の手順に従います。

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] ペインが表示されます。
3. [作成] オプションが選択されていることを確認し、[OK] 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、以下の値を入力します。

名前

マネージャ

説明

Acme Financial Services のマネージャ

ロールは以下に適用されます。

選択されたユーザ

5. [選択されたユーザへのロールの適用] オプションが選択されており、[ユーザ セットアップ] および [詳細設定] グループ ボックスが表示されていることを確認します。
6. [ユーザ セットアップ] グループ ボックスで選択することにより、マネージャ ロールでユーザのセットを定義します。このユース ケースについては、[メンバー グループ] テーブル内の以下のエントリを選択します。

cn=managers、ou=Groups、o=acme-financial.com

このエントリは、企業ユーザ ディレクトリ内で管理者のグループを指定します。

7. [OK] 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 ボックスで [レスポンス] 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

従業員 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

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 レスポンスが作成されます。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。

ロールに基づいたポリシーの確立

リソースとロールを定義したら、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。

以下の手順に従います。

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ロ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、どのロ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. 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するには [Benefits Management] 行で [Employees] ロールを選択します。
 - b. 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するには、[Performance Appraisals] 行で [Managers] ロールを選択します。
3. [サブミット] をクリックします。

ロールに基づいて、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: リソースまたはロールを編集するには、[ポリシー] ペインではなく、各タブで変更を行ってください。

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が必要としています。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、アプリケーションとアプリケーションが完了した日付です。

以下の手順に従います。

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、以下の値を入力します。

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、さらに以下の値を入力します。

名前

目的

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、企業のクレジット ポリシーに適合するように、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ということは、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. 拡張式内の属性マッピングを使用して、ロールを確立します。
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. 266)」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. 監査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の小売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 \(P. 247\)](#)」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、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] をクリックします。
[アプリケーション] ページが表示されます。
3. [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] ページが表示されます。
4. [一般] タブをクリックします。

5. 以下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。

6. [コンポーネント] 内の以下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。

コンポーネント名

カタログ

エージェント 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。

注: メイン コンポーネントを保存した後でのみ、サブコンポーネントが作成されます。

7. 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] 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 [選択されたメンバ] 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、**Online Catalog** という名前の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、以前に作成した **Online Catalog** アプリケーションにリソースを追加します。

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] をクリックします。
2. [アプリケーション] をクリックします。

[アプリケーション] ページが表示されます。
3. 検索条件を指定し、[検索] をクリックします。

条件に一致するアプリケーションが表示されます。
4. 変更するアプリケーションの名前をクリックします。このユース ケースでは、[オンラインカタログ] をクリックします。

[アプリケーションの表示] ページが表示されます。
5. ページをスクロールダウンして [変更] をクリックします。

設定とコントロールがアクティブになります。
6. [リソース] タブを選択します。
7. [作成] をクリックします。

[リソースの作成] ページが表示されます。

8. 以下のフィールドに値を入力します。このユース ケースでは、以下の値を入力します。

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。

9. [属性] 内の以下フィールドに値を入力します。このユース ケースでは、以下の値を入力します。

リソース

total_charges.jsp

10. [アクション] で [Web エージェント アクション] を選択し、アクション Get および Post を選択します。
11. [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

Web ポータル リソースが定義された後、クレジット限度を超えない限り顧客が Web ベースの購入ができ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が作成されていることを前提とします。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. [ロールの作成] 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] オプショ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、以下の値を入力します。

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。

ロールは以下に適用されます。

選択されたユーザ

5. [拡張式] 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。

ユーザの式

@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。

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。

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警告します。

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

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。

以下の手順に従います。

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、以下の値を入力します。

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、さらに以下の値を入力します。

名前

Admin_email

値

jdoe@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。

CA DataMinder コンテンツ分類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 ケースでは、金融サービス会社の **Forward** 社は **Microsoft SharePoint** を展開しています。会社が行いたいこと

- 全社員の **SharePoint** へのアクセス
- 社員の補償情報が含まれるドキュメントへのアクセス制限 人事の社員のみが、補償情報が含まれるドキュメントにアクセスできます。

注: **Forward, Inc.** は架空の会社名であり、厳密に説明のみを目的として使用されています。現存する会社を参照するものではありません。

以下では、社員の補償情報を保護するセキュリティ ポリシーを作成する **CA DataMinder** コンテンツ分類を含むアプリケーションを設定する方法について説明します。

例:

- **SiteMinder** 環境には 1 つの **LDAP** ユーザ ディレクトリが含まれます。ディレクトリの名前は **ForwardLDAP** です。
- ユーザ ディレクトリは、人事で働く社員を含む、すべての社員を識別します。ユーザ ディレクトリは以下のように人事社員を識別します。
cn= 人事、o=forwardinc.com
- 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。

- 一つの SharePoint サイトが展開されます。
 - SharePoint サイトのルート URL は次のとおりです。
`usecase.forwardinc.com/SitePages/Home.aspx`
 - 共有ドキュメント ディレクトリの URL は次のとおりです。
`usecase.forwardinc.com/Shared Documents`

解決方法:

1. 共有ドキュメント ディレクトリの保護
2. すべてのドキュメント リソースの保護
3. 人事の役割の作成
4. 人事の役割に基づいたポリシーの確立

共有ドキュメント ディレクトリの保護

このユース ケースでは、保護する SharePoint 環境での最高ポイントは共有ドキュメント ディレクトリです。全社員がアクセスできるかどうかを確認するために保護されていない SharePoint サイト (`usecase.forwardinc.com/SitePages/Home.aspx`) を離れます。

以下の手順に従います。

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] をクリックします。
3. [アプリケーションの作成] をクリックします。
4. [DLP サーバを使用] オプションを選択します。
5. [一般] セクションに必要な値を入力します。このユース ケースでは、以下のデータを指定します。

名前

SharePoint サイト

6. [コンポーネント] セクションに必要な値を入力します。このユース ケースでは、以下のデータを指定します。

コンポーネント名

共有ドキュメント

エージェント

SharePoint

リソース フィルタ

/usecase.forwardinc.com/Shared Documents

7. [ユーザ ディレクトリ] セクションで、アプリケーション ポリシーに利用可能なユーザ ディレクトリをすべてリスト表示するために [追加/削除] をクリックします。
8. ユーザ ディレクトリを選択し、[OK] をクリックします。このユース ケースでは、以下のユーザ ディレクトリを指定します。

ForwardLDAP

9. [DLP 分類] セクションで、コンポーネントに追加しないコンテンツ分類を削除します。このユース ケースでは、分類の削除は行いません。
10. [サブミット] をクリックしてアプリケーションを保存します。

すべてのドキュメント リソースの保護

このユース ケースでは、共有ドキュメント ディレクトリ内のすべてのドキュメントを保護します。 リソースをアプリケーションに追加することによりすべてのドキュメント リソースを保護します。

以下の手順に従います。

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] をクリックします。
3. 検索条件を指定し、[検索] をクリックします。
4. 変更するアプリケーションを確認し、[修正] アイコンをクリックします。このユース ケースでは、以下のアプリケーションを修正します。

SharePoint サイト

5. [リソース] タブを選択し、[作成] をクリックします。

6. 以下のフィールドに値を入力します。このユース ケースでは、以下の値を入力します。

名前

すべてのドキュメント

リソース

/*

7. ルールが起動するのに必要なイベントやアクションのタイプを選択します。このユース ケースでは、以下のイベント タイプを選択します。
 - Web エージェント アクション
 - Connect
 - Delete
 - Get
 - Head
 - Options
 - Post
 - Put
 - Trace
8. [OK] をクリックしてリソースを保存します。
9. [サブミット] をクリックしてコンポーネントにリソースを追加します。

人事の役割の作成

このユース ケースでは、人事の従業員のみが補償情報が含まれるドキュメントにアクセスするロールを作成するとします。

以下の手順に従います。

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] をクリックします。
3. 検索条件を指定し、[検索] をクリックします。
4. 変更するアプリケーションを確認し、[修正] アイコンをクリックします。このユース ケースでは、以下のアプリケーションを修正します。

SharePoint サイト

5. [ロール] タブを選択し、[ロールの作成] をクリックします。
6. 以下のオプションを選択します。
タイプ ロールのオブジェクトを作成します。
7. [OK] をクリックします。
8. [一般] セクションに値を入力します。このユース ケースでは、以下の値を入力します。

名前

人事

説明

従業員の補償情報が含まれるドキュメントへのアクセスがある
ロールです。

9. ロールを利用可能なディレクトリのすべてまたは特定のユーザに適用するかどうかを指定します。このユース ケースでは以下のオプションを選択します。

選択されたユーザ

10. [ユーザ セットアップ] セクションでロール メンバを定義します。このユース ケースでは、[メンバー グループ] セクションの以下の組織を選択します。

cn= 人事、o=forwardinc.com

11. [DLP 分類] セクションでロールへのコンテンツ分類を追加します。このユース ケースでは、以下の値を選択します。

- 米国従業員補償情報
- 英国従業員補償情報

12. [OK] をクリックしてロールを保存します。
13. [サブミット] をクリックしてアプリケーションへロールを追加します。

人事の役割に基づいたポリシーの確立

このユース ケースでは、社員の補償情報が含まれるドキュメントを保護するポリシーを作成します。

以下の手順に従います。

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] をクリックします。
3. 検索条件を指定し、[検索] をクリックします。
4. 変更するアプリケーションを確認し、[修正] アイコンをクリックします。このユース ケースでは、以下のアプリケーションを修正します。

SharePoint サイト

5. [ポリシー] タブを選択します。
6. すべてのドキュメント リソースに対して人事の役割を選択します。
7. [サブミット] をクリックします。

人事社員にのみ社員の補償情報が含まれる SharePoint ドキュメントへのアクセスを許すポリシーが作成されます。

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. 587)

[ドメインとユーザ メンバシップ](#) (P. 589)

[ポリシー ドメインの設定方法](#) (P. 589)

[ドメインへの Identity Manager 環境の追加](#) (P. 593)

[ドメインのグローバル ポリシー処理の無効化](#) (P. 594)

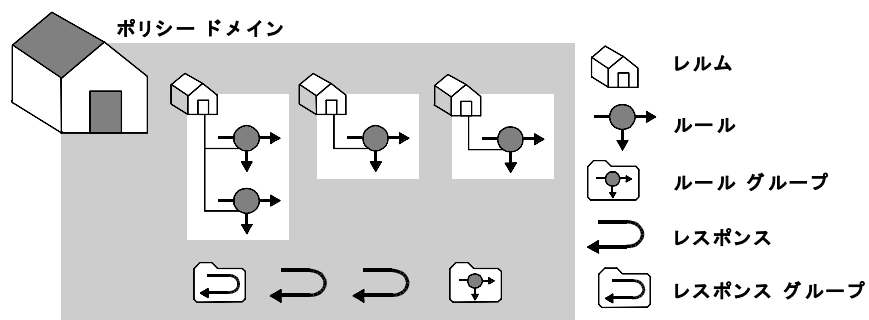
[ドメインの変更](#) (P. 594)

[ドメインの削除](#) (P. 594)

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。

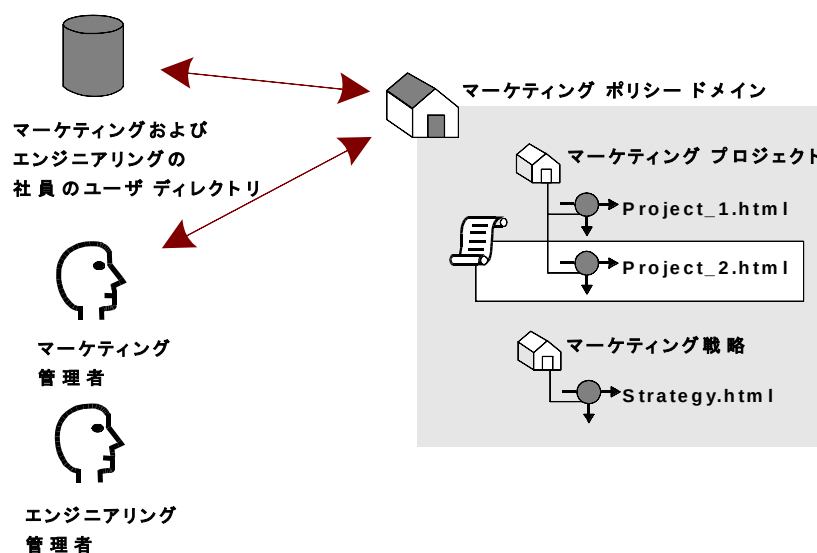
ポリシードメイン内のリソースは、1 つ以上のレルムにグループ化することができます。レルムとは、共通のセキュリティ（認証）要件を持つリソースのセットです。リソースを含むレルムに関連するルールにより、リソースへのアクセスが制御されます。次の図は、レルムとそれに関連したルール、およびルール グループ、レスポンス グループ、レスポンスのペアを含む小規模なポリシー ドメインを示しています。



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。



詳細情報:

[ポリシー](#) (P. 667)

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 リソースがあるレルム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、レルムが属するポリシー 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。
- レルムを 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、レルム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](#) (P. 590)
2. [ユーザ ディレクトリの割り当て](#) (P. 590)
3. [レルムの作成](#) (P. 591)

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します。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ドメイン] をクリックします。
2. [ドメインの作成] をクリックします。
3. ポリシーの名前および説明を入力します。
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。

ポリシードメインを定義しました。

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認証情報にユーザが入力する認証情報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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ドメイン] をクリックします。
2. 検索条件を指定し、[検索] をクリックします。

検索条件に一致するドメインのリストが表示されます。

3. 変更するドメインの名前をクリックします。
4. [変更] をクリックします。

設定とコントロールがアクティブになります。

5. [一般] タブで、[追加/削除] をクリックします。

[ユーザディレクトリの選択] ページが表示されます。

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。

7. [OK] をクリックします。

選択したユーザディレクトリは [ユーザディレクトリ] 下にリスト表示されます。

注: 新規ユーザディレクトリを作成しドメインにそれを追加するには、[作成] をクリックします。

8. [サブミット] をクリックします。

選択したユーザディレクトリがポリシードメインに追加されます。

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 **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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. [レルムの作成] をクリックします。
3. ドメインを選択し、[次へ] をクリックします。
4. レルムの名前および説明を入力します。
5. 省略記号ボタンをクリックし、エージェントを選択します。

6. **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. 残りのリソース プロパティを指定します。
8. 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、または既存のサブレルムを削除します。
10. セッション プロパティを指定します。
11. 認可ディレクトリ マッピングおよびレルムが処理する必要があるイベントのタイプを指定します。
12. **[完了]** をクリックします。
レルムが作成されます。

詳細情報:

[ポリシー サーバ オブジェクトのコピー \(P. 10\)](#)

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。

以下の手順に従います。

1. **[ポリシー]** - **[ドメイン]** - **[レルム]** をクリックします。
2. **[レルムの作成]** をクリックします。
3. ドメインを選択し、**[次へ]** をクリックします。
4. レルムの名前および説明を入力します。
5. 省略記号ボタンをクリックし、エージェントを選択します。
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. 残りのリソース プロパティを指定します。
 8. 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 プロパティを指定します。
 10. [完了] をクリックします。
- ルールが作成され、選択された RADIUS エージェントまたはエージェントグループと関連付けられます。

詳細情報:

[ポリシー サーバ オブジェクトのコピー \(P. 10\)](#)

ドメインへの Identity Manager 環境の追加

ドメインに Identity Manager 環境および関連するユーザディレクトリを追加すると、ポリシーで利用できる Identity Manager ロールが作成されます。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ドメイン] をクリックします。
2. 検索条件を指定し、[検索] をクリックして対象のドメインを見つけます。
3. 環境を追加するドメインの名前をクリックします。
4. [変更] をクリックします。
5. 環境と関連付けられるユーザがドメインにバインドされない場合は、それぞれのユーザディレクトリを追加します。
6. [IDM 環境] セクションで [追加/削除] をクリックします。
7. 必要な [IDM 環境] を選択し、[OK] をクリックします。
8. [サブミット] をクリックします。

Identity Manager 環境がドメインに追加されます。環境と関連付けられたロールは、ドメインで作成されたすべてのポリシーに利用可能です。

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。

以下の手順に従います。

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。

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。

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、管理者接続、およびルール、レルム、レスポンス、ポリシーまたはアフィリエイトなどの、ルールとしてドメインにバインドされたオブジェク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、少々時間がかかる場合があります。

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 \(P. 595\)](#)

[信頼レベルの導入 \(P. 596\)](#)

[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 \(P. 597\)](#)

[ネストされたレルム \(P. 600\)](#)

[リクエスト処理におけるレルム \(P. 602\)](#)

[レルムの設定 \(P. 602\)](#)

[レルムの変更 \(P. 605\)](#)

[レルムの削除 \(P. 605\)](#)

[ネストされたレルムの設定 \(P. 606\)](#)

[リソース キャッシュからのレルムのクリア \(P. 607\)](#)

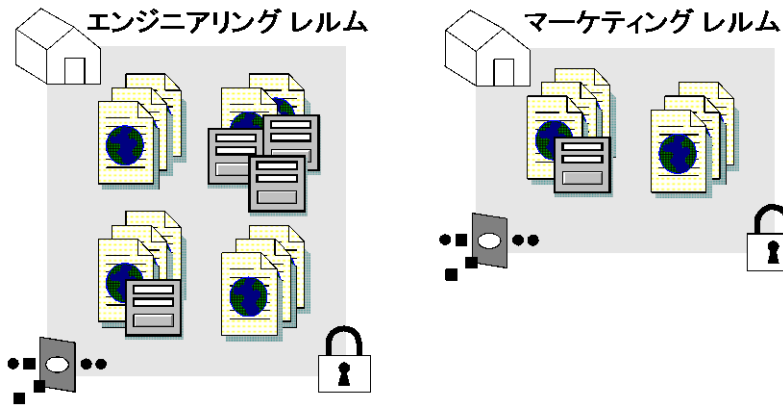
レルムの概要

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。

信頼レベルの導入

SiteMinder がサポートされたリスク分析エンジンに統合される場合、信頼レベルを持つレルムを設定できます。信頼レベルは認証情報保証を表します。これは保護されているリソースをリクエストするユーザが正当である可能性を表します。

以下の点を考慮します。

- 信頼レベルは数値スケール（0 ～ 1000）に基づきます。信頼レベルが高いほど、認証情報保証も高くなります。
- 両方のアクセス管理モデルを使用して、許可決定に信頼レベルを適用できます。詳細については、「[ポリシーの信頼レベル](#) (P. 671)」と「[アプリケーションの信頼レベル](#) (P. 553)」を参照してください。
- 信頼レベルを生成する認証方式がユーザを認証するとき、信頼レベルはユーザのセッションチケットに挿入されます。ポリシーまたはアプリケーションがユーザに適用される限り、ユーザはこの値以上の信頼レベルを必要とするどんなリソースにもアクセスできます。

- 信頼レベルが計算される要因は、認証情報保証を決定するためにリスク分析エンジンが使用するデータに基づきます。

注: SiteMinder は、CA Arcot WebFort および CA Arcot Risk のオンプレミス実装による統合および CA Arcot A-OK のホスト実装をサポートします。

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ムに対して相対的なリソースを指定します。たとえば、レムリソース 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。 例：

```
/Managers/*
```

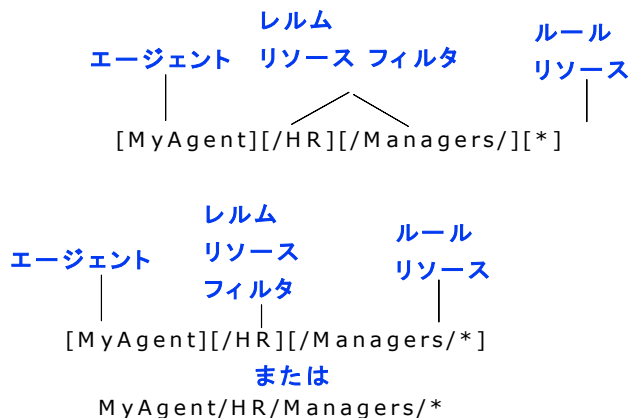
この3つの要素を使用して、次のような例を考えてみましょう。

- 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。
- このレールムに次のような 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あります。

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せん。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 \(P. 433\)](#)」を参照してください。

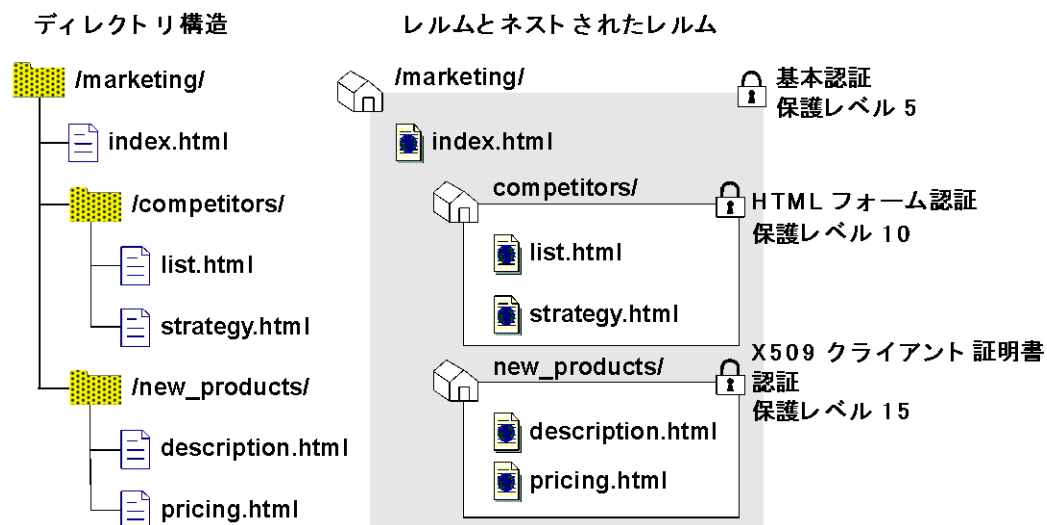
ネストされたレルム

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方式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できます。

注: 以下の手順では、新しいオブジェクトが作成されていることを前提とします。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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. [レルムの作成] をクリックします。
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。
4. レルムの名前および説明を入力します。
5. 省略記号ボタンをクリックし、エージェントを選択します。
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. 残りのリソース プロパティを指定します。
8. 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、または既存のサブレルムを削除します。
10. セッション プロパティを指定します。
11. ディレクトリ マッピングおよびレルムが処理する必要があるイベントのタイプを指定します。
12. レルム内のすべてのリソースに適用するために、最小限の信頼レベルを指定します。

注: レルムに信頼レベルを適用するには、**SiteMinder** をサポートされたリスク分析エンジンと統合し信頼レベルサポートを有効にする必要があります。詳細については、「**実装ガイド**」を参照してください。

13. [完了] をクリックします。

選択された **SiteMinder Web** エージェントまたはエージェント グループと関連付けられたレルムが作成されます。

詳細情報:

[ポリシー サーバ オブジェクトのコピー \(P. 10\)](#)

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できます。

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が作成されていることを前提とします。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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. [レルムの作成] をクリックします。
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。
4. レルムの名前および説明を入力します。
5. 省略記号ボタンをクリックし、エージェントを選択します。
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. 残りのリソース プロパティを指定します。
8. 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 プロパティを指定します。
10. [完了] をクリックします。

選択された RADIUS エージェントまたはエージェント グループと関連付けられたレルムが作成されます。

詳細情報:

[ポリシー サーバ オブジェクトのコピー](#) (P. 10)

レールの変更

既存のレールを変更する場合、以下の項目は変更できません。

- エージェント

既存のレール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、レールを削除し、新しいエージェントでレールを作成し直してください。

- リソース フィルタ

既存のレールのリソース 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、レールを削除し、新しいリソース フィルタでレール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、「ポリシー サーバ オブジェクトの管理」を参照してください。

レールの削除

レールを削除すると、そのレールに関連するネストされたレールもすべて削除されます。さらに、削除されたレールとそのネストされたレールに関連付けられているルールもすべて削除されます。関連するネストされたレールをすべて削除するには、親レールを削除するのみにします。削除する個々のネストされたレールを選択しないでください。

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、「ポリシー サーバ オブジェクトの管理」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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. 検索条件を指定して、[検索] をクリックします。
3. 変更するレルムの名前をクリックします。
4. [変更] をクリックします。
設定とコントロールがアクティブになります。
5. [サブレルム] で、[サブレルムの作成] をクリックします。
6. レルムの名前および説明を入力します。
7. 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。

8. セッション プロパティを指定します。
9. 以下の詳細な設定を指定します。
 - 認可ディレクトリ マッピング
 - イベントの処理
10. [OK] をクリックします。
11. [サブミット] をクリックします。

新しく作成されたサブレルムは、親レルムに追加されます。

新しく作成されたサブレルムと関連付けられたレルムが作成されます。

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。ただし、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必要があります。

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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レルム] をクリックします。
2. 検索条件を指定して、[検索] をクリックします。
3. 変更するレルムの名前をクリックします。
4. [変更] をクリックします。
設定とコントロールがアクティブになります。
5. [レルム内のリソースのクリア] 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. 609)

[Web エージェントアクションのルールの設定](#) (P. 618)

[認証イベントアクション用のルールの設定](#) (P. 619)

[許可イベントアクションのルールの設定](#) (P. 620)

[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. 622)

[リソース照合と正規表現](#) (P. 623)

[ルールの有効化および無効化](#) (P. 625)

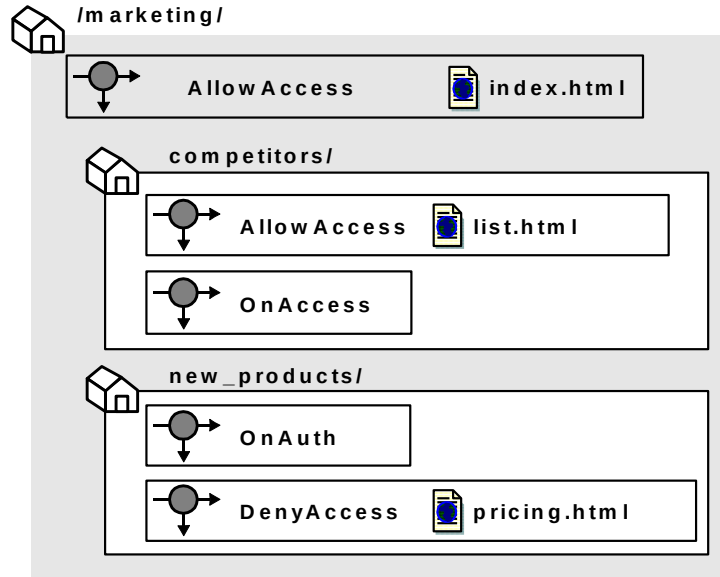
[高度なルール オプション](#) (P. 625)

[ルールの削除](#) (P. 627)

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時、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ムを検索します。ポリシー サーバは、そのレム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/`

以下のレムには、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。

レム名	レムの説明	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>
パフォーマンス管理	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ムは、**Performance Management** です。許可イベントに応じて、ポリシー サーバは、**Performance Management** レムに関連付けられたすべてのルールを起動します。

レム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。

ネストされたレルム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ルム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ルム

ネストされたレルムとは、既存のレルム内に作成されたレルムです。ネストされたレルムは親レルムか最上位レルムに属し、親レルムの子と見なされます。ネストされたレルムを作成すると、子レルム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、他のレルム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。

詳細情報:

[ネストされたレルム \(P. 600\)](#)

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とする、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。

詳細情報:

[ポリシーの概要 \(P. 667\)](#)

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すると、SiteMinder は、認証の成功時にユーザが指定リソースにアクセスできるようにします。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により、さらにセキュリティを強化できます。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。

- HEAD
- GET
- POST
- PUT
- DELETE
- TRACE
- OPTIONS

WSS エージェント アクション

2 つの追加の **Web** エージェント ルール アクションが **WSS** エージェントで使用できます。

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。

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。レルムの一部に適用される **On-Auth** ルールを作成することはできません。

以下のリストは、発生する可能性がある認証イベントです。

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できます。

OnAuthAcceptCredentials

ログインの段階でのみ発生します。ユーザ認証情報が示され、新しいセッションの作成が行われます。

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)。

Web エージェントが **EnforceRealmTimeouts** オプションをサポートし、そのオプションが有効である場合を除き、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ユーザが最後に認証したレルムの値のままとなります。その値は、ユーザが認証情報を再入力する必要がある場合にのみ変更されます。

リダイレクト

リダイレクトが許可されるのは、認証時に無限のリダイレクトループの可能性を防ぐことのみです。

ユーザ 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: **OnAuth** イベントの結果はレルム単位になります。たとえば、ユーザがレルム **A** からレルム **B** に移動し、そのユーザがレルム **A** 内に **OnAuthAccept** ヘッダを設定していても、レルム **B** ではその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 がこのユーザを識別しないため、ユーザが拒否される場合に発生します。たとえば、未登録のユーザは登録のためまずリダイレクトされます。

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のみ使用します。アクティブ レスポンス以外のレスポンスをトリガするためにはこのイベントを使用しません。

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。ユーザが認証されると（または拒否されると）、ポリシー サーバは該当する **On-Auth** ルールに関連するレスポンスを要求してきたエージェントに戻します。

注: **SiteMinder** 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できます。パフォーマンスを最適化するには、**OnAuthAccept** 認証イベントに基づいたルールをセットアップし静的情報を戻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ます。静的レスポンスは、正常に認証されたユーザに単に戻されます。

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。

詳細情報:

[ルールへの時間制限の追加](#) (P. 626)

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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルール] をクリックします。
2. [ルールの作成] をクリックします。
3. リストからドメインを選択し、[次へ] をクリックします。
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。

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. ルールの名前および説明を入力します。

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. ルールが保護されているリソースへのアクセスを許可するか拒否するべきかどうかを指定します。
8. **Web** エージェント アクションを選択します。
[アクション リスト] に HTTP アクションが読み込まれます。
9. 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] 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。

Web エージェント アクション用のルールが作成されます。

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できます。

ルールを作成するレلمは、認証イベントを処理可能である必要があります。[認証イベントの処理] オプションが選択されていることを確認します。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルール] をクリックします。
2. [ルールの作成] をクリックします。
3. リストからドメインを選択し、[次へ] をクリックします。
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。

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. ルールの名前および説明を入力します。

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. 認証イベントを選択します。

[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لم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. 1 つ以上の認証イベントを選択します。
8. (オプション) [詳細] で、時間制限および (または) アクティブなルール設定を設定します。
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لم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の設定

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لمは、許可イベントを処理可能である必要があります。[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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルール] をクリックします。
2. [ルールの作成] をクリックします。
3. リストからドメインを選択し、[次へ] をクリックします。
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。

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. ルールの名前および説明を入力します。

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. ルールが保護するリソースを入力します。

[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. 許可イベントを選択します。

[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. 1 つ以上の認可イベントを選択します。

9. (オプション) [詳細] で、時間制限、アクティブなルールまたは両方を設定します。

10. [サブミット] をクリックします。

ルールは保存され、指定されたレム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ありません。

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、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

リソースにアクセスす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できます。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルール] をクリックします。
2. [ルールの作成] をクリックします。
3. リストからドメインを選択し、[次へ] をクリックします。
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。

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. ルールの名前および説明を入力します。

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. ルールが保護するリソースを入力します。

[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. [インパーソネーション イベント] 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. 1 つ以上のインパーソネーション イベントを選択します。

9. (オプション) [詳細] で、時間制限、アクティブなルールまたは両方を設定します。

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。

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、レلم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	使用数
*	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。*.html という文字列を使用すると、index.html や out.html など、拡張子が .html 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。lmn?p という文字列を使用すると、lmnop や lmnep 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
使用され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 abc... 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{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。

以下の手順に従います。

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」にあります。

以下の手順に従います。

1. [時間制限] グループ ボックスの「設定」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。

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。

以下の手順に従います。

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。
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、「ポリシー サーバ オブジェクトの管理」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. 629)

[ルール グループの作成](#) (P. 630)

[ルール グループへのルールの追加](#) (P. 631)

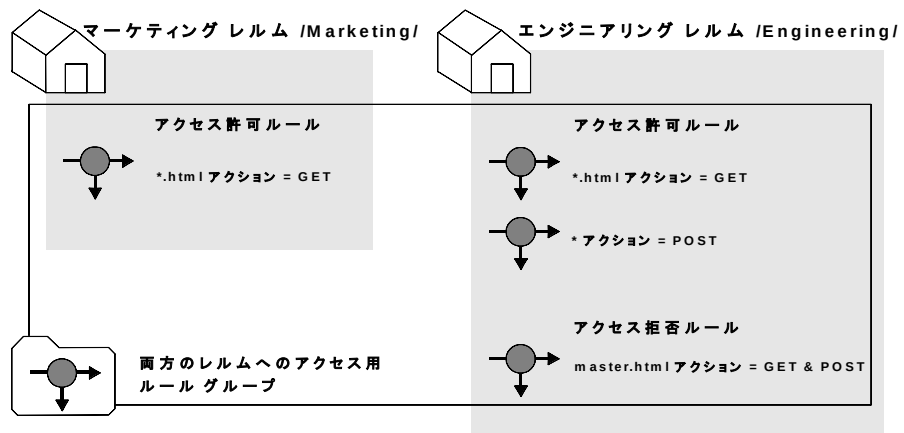
[ルール グループの変更](#) (P. 632)

[ルール グループの削除](#) (P. 632)

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 レルムとエンジニアリング レルム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。

詳細情報:

[ポリシーの概要](#) (P. 667)

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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルール グループ] をクリックします。
2. [ルール グループの作成] をクリックします。
3. [OK] をクリックします。
4. ドメインを選択し、[次へ] をクリックします。
5. ルール グループの名前および説明を入力します。
6. 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] で、[追加/削除] をクリックします。

[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ル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。

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。
10. [完了] をクリックします。
[ルールグループ] が作成されます。

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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [ルールグループ] をクリックします。
2. 検索条件を指定して、[検索] をクリックします。
3. 変更するルールグループの名前をクリックします。
4. [変更] をクリックします。

設定とコントロールがアクティブになります。

5. [グループメンバ] で、[追加/削除] をクリックします。

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。

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。
8. [サブミット] 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。

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、「ポリシー サーバ オブジェクトの管理」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、「ポリシー サーバ オブジェクトの管理」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. 633)

[レスポンスの設定](#) (P. 636)

[オープン形式の Cookie を生成する Web エージェント レスポンスの作成方法](#) (P. 646)

[SiteMinder が生成するユーザ属性](#) (P. 655)

[レスポンス グループ](#) (P. 662)

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、ステック 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。

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。

使用可能なレスポンスのタイプは、以下のとおりです。

- Web 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。

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。

詳細情報:

[RADIUS エージェント レスポンス属性](#) (P. 636)

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。

eTrust SOA Security Manager Web エージェントレスポンス属性

SiteMinder Web エージェント レスポンス属性以外に、eTrust SOA Security Manager では、WSS エージェントと共に使用する場合にのみ適用可能な、以下の Web エージェント レスポンス属性を使用できます。

WebAgent-SAML-Session-Ticket-Variable

[set AGENT value for your book] が SAML アサーションの生成に使用するポリシー サーバデータを指定します。データは、（関連するレスポンス属性による指定に従って）XML メッセージ HTTP または SOAP エンベロープ ヘッダ、または Cookie に挿入されます。

SAML セッション チケット レスポンスを設定するとき、ポリシー サーバはレスポンス データを生成します。このデータは、[set AGENT value for your book] に対してアサーションを作成する方法を指示します。[set AGENT value for your book] は、セッション チケット（および必要に応じて Web サービス コンシューマの公開キー）およびレスポンス データを暗号化します。その後、エージェントはアサーションを生成します。エージェントは Web サービスにアサーションを送信します。トークンは、[set AGENT value for your book] がそのエージェント キーを使用する場合のみ、暗号化および復号化できます。

WebAgent-WS-Security-Token

[set AGENT value for your book] が（関連するレスポンス属性の指定に従って）WS-Security Username、X509v3、または SAML トークンの生成に使用するポリシー サーバデータを指定します。これらのトークンは SOAP メッセージ ヘッダに追加されます。

WS-Security レスポンスを設定するとき、ポリシー サーバはレスポンス データを生成します。このデータは、[set AGENT value for your book] に対してトークンを作成する方法を指示します。その後、エージェントは、トークンを生成して SOAP リクエストに追加し、Web サービスに送信します。

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、アプリケーション オブジェクト コンポーネントまたはレルムに別個の許可ユーザ 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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レスポンス] 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。
3. ドメインを選択し、[次へ] をクリックします。
4. レスポンスの名前および説明を入力します。
5. 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。
7. [完了] をクリックします。
レスポンスが作成されます。

レスポンス属性の設定

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 smetssocookie 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」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は、SiteMinder がどのように属性に対して適切なコンテンツを提供するか決定します。

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エントリからプロフィール情報を返します。

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。

ユーザ DN の一部であるユーザ グループおよび組織単位 (OU) は、DN 属性として扱うことのできるディレクトリ オブジェクト属性の例です。

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) に準拠します。

注: 戻り値が有効であることを確認します。 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。

セッション変数

セッション変数の値を返します。

レスポンスが認証リクエストの一部であるとき、SiteMinder はセッションストア、またはメモリから値を取得します。

式

管理者は式を提供することができます。

たとえば、証明書発行者 DN 属性から特定の文字列を抽出し、新しいセッション変数として格納するために管理者は [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。

以下の手順に従います。

1. [レスポンス属性の作成] をクリックします。
2. レスポンス属性を選択します。
3. 属性タイプを選択します。

[属性フィールド] の詳細は、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] の詳細を入力します。

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 \(P. 655\)](#)」を参照してください。

5. (オプション) [スクリプト] フィールドで属性を編集します。

注: [詳細] セクションで属性を編集すると、[属性のセットアップ] セクションが閉じます。

6. [キャッシュ値] または [値の再計算間隔 (秒)] (Recalculate value every ... seconds) を指定します。

注: 入力できる最大時間制限は 3600 秒です。

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] ページの [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。

以下の手順に従います。

1. [レスポンス属性の作成] をクリックします。
2. レスポンス属性を選択します。
3. 属性タイプを選択します。

[属性フィールド] の詳細は、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] の詳細を入力します。

注: レスポンスで使用できる、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 \(P. 655\)](#)」を参照してください。

5. (オプション) [スクリプト] フィールドで属性を編集します。

注: [詳細] セクションで属性を編集すると、[属性のセットアップ] セクションが閉じます。

6. [キャッシュ値] または [値の再計算間隔 (秒)] (Recalculate value every ... seconds) を指定します。

注: 入力できる最大時間制限は 3600 秒です。

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] ページの [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の使用

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。

以下の手順に従います。

1. 「レスポンスの設定」の説明に従って、レスポンスを作成します。
2. [属性] セクションで [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] セクションの [レスポンス属性の作成] をクリックします。
4. [属性タイプ] セクション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] セクション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] セクションの [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] グループセクション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

ライブラリの名前、ライブラリ関数の名前を指定します。オプションで、[ライブラリ名]、[関数名] および [パラメータ] フィールドで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。

セッション変数

管理者が値を取得できるセッション変数の名前を指定します。

式

属性から値を抽出し、新しいセッション変数としてそれを格納する式を指定します。

注: SiteMinder は、[属性フィールド] セクション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。

以下の手順に従います。

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。
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。
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。
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。

以下の手順に従います。

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。
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化します。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。

以下の手順に従います。

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。
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。
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、「ポリシー サーバ オブジェクトの管理」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。

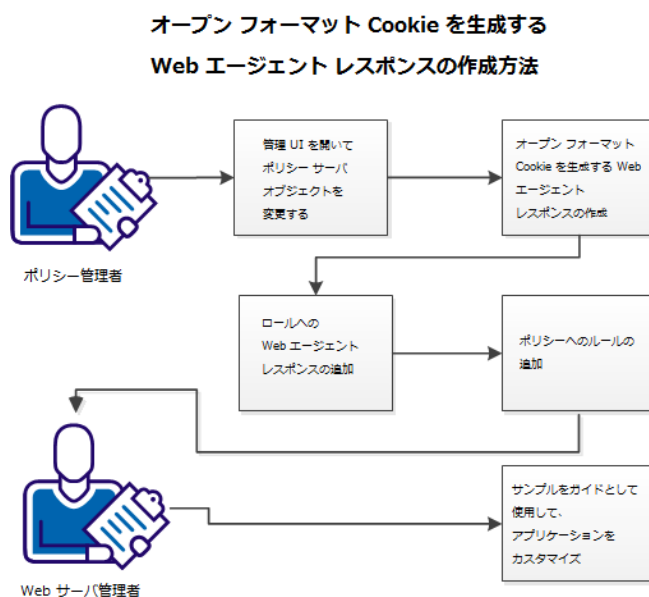
注: ポリシー サーバ オブジェクトの変更および削除の詳細については、「ポリシー サーバ オブジェクトの管理」を参照してください。

オープン形式の Cookie を生成する Web エージェント レスポンスの作成方法

組織がオープン形式の Cookie を使用している場合、ポリシー サーバ上にエージェント レスポンスを設定して、オープン形式の Cookie を生成できます。

重要: SiteMinder はオープン フォーマット Cookie のみを作成します。SiteMinder は、アプリケーションがオープン フォーマット Cookie に加える変更を認識しません。SiteMinder は、処理中に発生するイベントのシーケンスに応じてオープン フォーマット Cookie を上書きすることができます。SiteMinder は、ユーザのログアウト時にオープン フォーマット Cookie を自動的に削除できません。

オープン形式の Cookie を生成するように SiteMinder を設定した後に、Web サーバ管理者に、オープン形式の Cookie を使用するように Web アプリケーションをカスタマイズしてもらいます。提供されているサンプルをガイドとして使用します。



以下の手順に従います。

1. [管理 UI を開いてポリシー サーバオブジェクトを変更します](#) (P. 90)。
2. [オープン形式の Cookie を生成する Web エージェント レスポンスを作成します](#) (P. 648)。
3. [ルールに Web エージェント レスポンスを追加します](#) (P. 653)。
4. [ポリシーにルールを追加します](#) (P. 654)。
5. [サンプルをガイドとして使用して、アプリケーションをカスタマイズします](#) (P. 655)。

ポリシー サーバオブジェクトの変更

管理 UI を開いて、ポリシー サーバ上のオブジェクトを変更します。

以下の手順に従います。

1. ブラウザで以下の URL を入力します。

`https://host_name:8443/iam/siteminder/adminui`

`host_name`

管理 UI ホスト システムの完全修飾名を指定します。

2. [ユーザ名] フィールドに **SiteMinder** スーパーユーザ名を入力します。
3. [パスワード] フィールドに **SiteMinder** スーパーユーザ アカウントのパスワードを入力します。

注: スーパーユーザ アカウントのパスワードにドル記号 (\$) 文字が含まれている場合は、ドル記号文字の各インスタンスを **\$DOLLAR\$** に置換します。たとえば、**SiteMinder** スーパーユーザ アカウント パスワードが **\$password** の場合は、パスワード フィールドに「**\$DOLLAR\$password**」と入力します。

4. 適切なサーバ名または IP アドレスが [サーバ] 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。
5. [ログイン] を選択します。

オープン形式 Cookie を生成する Web エージェント レスポンスの作成

管理 UI を使用して、オープン形式 Cookie を生成する Web エージェント レスポンスを作成します。

以下の手順に従います。

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。
2. オープン形式 Cookie を発行するドメインのオプション ボタンをクリックします。
3. [次へ] をクリックします。
4. 以下の手順に従い、オープン形式 Cookie を生成するレスポンスを作成します。
 - a. レスポンスの名前および説明（オプション）を入力します。
 - b. 以下の設定を確認します。
 - [SiteMinder] オプション 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。
 - c. [レスポンス属性の作成] をクリックします。
 - d. [属性] ドロップダウン リストをクリックし、以下の値を選択します。

WebAgent-HTTP-Open-Format-Cookie

- e. 以下のフィールドに値を入力します。

Cookie 名

HTTP ヘッダ変数としてオープンフォーマット **Cookie** で返される名前を指定します。

制限: 任意の印刷可能な ASCII 文字 (! から ~ まで。ただし、ハ、=、スペースを除く)。

Cookie ドメイン範囲

オープンフォーマット **Cookie** で設定されたセクション (ピリオドで区切られた文字) の数を指定します。

値を 0 に設定すると、オープンフォーマット **Cookie** は最も具体的な **Cookie** ドメインを使用します。これは、**Cookie** ドメイン `myserver.example.com` に対応するドメインが `example.com` であり、`myserver.metals.example.org` に対応するドメインが `.metals.example.org` であることを意味します。逆に、[**Cookie ドメイン範囲**] の値が 2 に設定されている場合、**Cookie** ドメインは、それぞれ `.example.com` と `.example.org` になります。

この設定はオープンフォーマット **Cookie** に特有です。
SiteMinder エージェントが使用する **CookieDomainScope** パラメータとは異なります。

制限: 0、2、または 3 以上。

例: オープンフォーマット **Cookie** ドメインが `division.example.com` であると仮定します。
`server.division.example.com` の **Cookie** ドメインの範囲を設定するには、[**Cookie ドメイン範囲**] フィールドの値を 3 に設定します。

デフォルト: 2

生存時間(TTL)

オープンフォーマット **Cookie** が有効な秒数を指定します。この値が HTTP ヘッダ変数として返されます。

制限: 0 より大きい正の数。

Cookie パス

オープン フォーマット **Cookie** で使用するパスを指定します。
この設定はオープン フォーマット **Cookie** に特有です。
SiteMinder エージェントが使用する **CookiePath** パラメータとは異なります。

Cookie パス スコープ

オープン フォーマット **Cookie** で使用されるパスの範囲を定義する数を指定します。

この設定はオープン フォーマット **Cookie** に特有です。
SiteMinder エージェントが使用する **CookiePathScope** パラメータとは異なります。

制限：0 以上

暗号化アルゴリズム

オープン フォーマット **Cookie** の暗号化に使用する暗号化アルゴリズムを指定します。3DES 暗号化キーのキー長は 192 ビット（24 バイト）までに制限されています。

制限：AES128/CBC/PKCS5Padding、AES192/CBC/PKCS5Padding、AES256/CBC/PKCS5Padding、3DES_EDE/CBC/PKCS5Padding

デフォルト：3DES_EDE/CBC/PKCS5Padding

注: 128-ビットより長いキーを使用するには、Sun Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy ファイルをインストールします。 <http://java.sun.com/javase/downloads/index.jsp> からこれらのファイルをダウンロードできます。

暗号化パスワード

オープンフォーマット Cookie の暗号化に使用するパスワードを指定します。

- f. キーを生成する場合は、[暗号化キーの生成] (Generate Encryption Key) をクリックします。独自の既存のキーを入力することもできます。

暗号化キーは以下のフィールドに表示されます。

暗号化キー

オープンフォーマット Cookie で使用される暗号化キーを識別します。自分のキーを入力するか、または [暗号化パスワード] フィールドの値に基づくキーを生成します。また、アプリケーションは、このキーを持つオープンフォーマット Cookie を復号化します。ここで設定または変更される任意のキーを、アプリケーションにコピーします。

制限: 16 進数のみを使用します。

- g. オープン形式 Cookie を追加する属性に対応するチェック ボックスをオンにします。以下のリストから選択します。

注: 認証プロセス中に利用可能な値の一部は、許可プロセス中には使用されません。この状況では、認証プロセスに使用されるオープン形式 Cookie の情報は、その後の許可中に上書きされます。

SM_USERNAME

userdn (ユーザ識別名) 値をオープンフォーマット Cookie に追加します。

SM_USERLOGINNAME

ユーザ名 (ユーザがログインに使用した) の値をオープン形式の Cookie に追加します。

SM_USERIPADDRESS

ユーザの IP アドレス (認証または認可時に存在した) をオープン形式の Cookie に追加します。

SM_USERSESSIONIP

オープン形式の Cookie へのセッションをユーザが最初に認証し、確立した際に使用された IP アドレスを追加します。

SM_USERGROUPS

ユーザ グループをオープン フォーマット Cookie に追加します。

SM_USER_CONFIDENCE_LEVEL

リスク スコアまたは（レルムからの）信頼レベルをオープン フォーマット Cookie に追加します。

SM_AUTHENTICATIONNAME

認証方式の名前をオープン形式 Cookie に追加します。

制限： 認証フェーズ中にのみ使用可能です。

SM_AUTHENTICATIONTYPE

認証方式タイプをオープン フォーマット Cookie に追加します。

制限： 認証フェーズ中にのみ使用可能です。

SM_AUTHENTICATIONLEVEL

認証方式で定義された保護レベルをオープン フォーマット Cookie に追加します。

- h. （オプション）オープン形式 Cookie に含める値があるユーザ ディレクトリからその他の属性を追加します。
- i. 以下の属性キャッシング オプションのいずれかを選択します。

キャッシュ値

エージェントで属性値が一度計算されると、ポリシー サーバ用に値がキャッシュされるように指定します。属性が長期間にわたって変化しない場合、このオプション ボタンを使用します。

値の再計算間隔

レスポンス値が再計算されるまでの秒数を指定します。値が「0」の場合は、[キャッシュ値] と同等です。

- 5. [OK] をクリックします。
- 6. [完了] をクリックします。

オープン形式 Cookie を生成する Web エージェント レスポンスが作成されます。

ルールへの Web エージェントレスポンスの追加

オープン形式 Cookie を生成する Web エージェント レスポンスは、以下のルール タイプのいずれかに追加する必要があります。

- OnAuthAccept
- OnAccessAccept

前述の両タイプを使用する場合は、個別のルールを作成します。ただし、各ルールには 1 つのタイプだけ指定できます。

以下の手順に従います。

1. 管理 UI から、[ポリシー] - [ドメイン] - [ルール] - [ルールの作成] をクリックします。
2. オープン形式 Cookie を発行するドメインのオプション ボタンをクリックします。
3. [次へ] をクリックします。
4. オープン形式 Cookie を発行するレルムを選択します。
5. [次へ] をクリックします。
6. [ルールの作成] をクリックします。
7. 以下の手順に従い、ルールを作成します。
 - a. わかりやすい名前と説明（オプション）を入力します。
 - b. 以下の項目を指定します。
 - レルムとリソース
 - 許可/拒否と有効/無効
 - c. 以下のいずれかのオプション ボタンをクリックします。
 - 認証イベント
 - 許可イベント
 - d. 以下のリストから適切な値が表示されることを確認します。
 - 認証イベントの場合は、OnAuthAccept が [アクション] ドロップダウンリストに表示される必要があります。
 - 許可イベントの場合は、OnAccessAccept が [アクション] ドロップダウンリストに表示される必要があります。
 - e. （オプション）必要な時間制限をルールに追加します。

8. [完了] をクリックします。
Web エージェント レスポンスがルールに追加されます。
9. (オプション) 手順 6 ～ 8 を繰り返して、追加のルールを作成します。

ポリシーへのルールの追加

ルールを既存のポリシーに追加します。ルールは、オープン形式 Cookie を生成するレスポンスと関連付けられます。

以下の手順に従います。

1. 管理 UI から、[ポリシー] - [ドメイン] - [ドメイン ポリシー] をクリックします。
2. ルールを追加するポリシーの[編集] アイコンをクリックします。
[一般] タブが選択された状態で[ポリシーの変更] 画面が表示されます。
3. [ルール] タブをクリックします。
4. [ルールの追加] をクリックします。
使用可能なルールのリストが表示されます。
5. ポリシーに追加するルールのチェック ボックスをオンにします。
6. [OK] をクリックします。
7. [サブミット] をクリックします。
ルールがポリシーに追加され、確認メッセージが表示されます。

サンプルをガイドとして使用したアプリケーションのカスタマイズ

アプリケーションをカスタマイズするためのサンプルアプリケーションと Readme ファイルを使用できます。Web サーバ管理者は、オープン形式 Cookie を消費するアプリケーションを変更するときに、これらのサンプルをガイドとして使用できます。

以下のリストから、アプリケーションのタイプに応じて適切なディレクトリを選択します。

- `installation_directory¥SiteMinder¥samples¥SmOpenFormatCookie¥Java`
(Java)
- `installation_directory¥SiteMinder¥samples¥SmOpenFormatCookie¥dotnet`
(.NET)

`installation_directory`

ポリシー サーバがインストールされているファイル システム上の場所を指定します。

注: ユーザがログアウトする場合、SiteMinder はオープン形式 Cookie を自動的に削除できませんが、Cookie を削除するサンプル スクリプトが別のシナリオで提供されています。

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 レスポンス用のレスポンス属性として指定でき、名前付き式に利用可能です。

`%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。以下のいずれかの条件に当てはまる場合、Web エージェントは、`SM_USER` ヘッダ変数の値を設定しません。

- ユーザが証明書ベースの認証のようにユーザ名を提供しない場合。
- ユーザ名が不明の場合。

%SM_USER_CONFIDENCE_LEVEL

ユーザが認証方式で認証され、認証方式が信頼レベルを生成する場合、この属性は整数（0 ～ 1000）を保持します。認証方式は、ユーザのセッションチケットに整数を挿入します。信頼レベルが高いほど、認証情報保証も高くなります。信頼レベルがゼロの場合、認証情報保証がないことを示します。認証情報保証がないと、SiteMinder はリクエストされたリソースへのアクセスを拒否します。

注: 詳細については、「[信頼レベルの導入](#) (P. 596)」（Confidence Levels Introduced）を参照してください。

%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、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、この属性は、ユーザがログイン試行時に指定するユーザ ID を保持します。

%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

この属性は、ユーザがログイン試行時に指定するユーザ ID を保持します。

%SM_USERIPADDRESS

この属性には、認証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試行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。

%SM_USERSESSIONID

この属性には、すで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 **SiteMinder** によって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。次の 2 つの条件が満たされる場合のみ、この属性は値を持ちます。

- 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、直前のログインの前に正常にログインした時刻を 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。パスワード サービスが有効になっている場合のみ、この属性は値を持ちます。

%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。

例：

ユーザが Accounting グループに含まれる Accounts Payable グループに属する場合、SM_USERGROUPS 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_USERGROUPS[] を使用してください。

例：

ユーザが 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_USERGROUPS 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て、ユーザにアクセス許可を付与するポリシーが存在する場合、この属性にはそのポリシーの名前が格納されます。

例： 商品を購入するには、Buyer ポリシーに関連付けられたユーザである必要があります。ポリシー サーバが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、すべてのポリシー ドメイン内で、そのユーザに適用される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。

例：

「Equipment Purchasing」と呼ばれるレルムが存在するとします。このレルムには、「CheckCredit」と呼ばれるルールがあります。ルールは、次のようなレスポンス属性として購入者のクレジット上限を返すレスポンスと関連付けられます。

limit = \$15000

購入者が 5000 ドル相当の装置を購入しようとする、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。

%SM_USER_APPLICATION_ROLES

CA Identity Manager を購入済みであれば、この属性をレスポンスに使用できます。この属性には、ユーザに割り当てられたか、ユーザに委任されたすべてのロールの一覧が含まれています。アプリケーション名を指定すると、そのアプリケーションに関連付けられたロールのみがレスポンス属性で返されます。

レスポンス属性名は、[レスポンス属性]ウィンドウの[変数名]フィールドに入力します。レスポンス属性名の構文は以下のとおりです。

`SM_USER_APPLICATION_ROLES[:application_name]`

application_name には、Identity Manager で定義されたアプリケーションの任意の名前を指定します。

application_name に指定した値は、ポリシー サーバ管理者に知らせておく必要があります。アプリケーション名が、自動的に 管理 UI に渡されることはありません。

注: Identity Manager ロールの詳細については、「CA Identity Manager 操作ガイド」を参照してください。

%SM_USER_APPLICATION_TASKS

CA Identity Manager を購入済みであれば、この属性をレスポンスに使用できます。この属性には、ユーザに割り当てられているか、ユーザに委任されているすべてのタスクの一覧が含まれています。アプリケーション名を指定すると、そのアプリケーションに関連付けられたタスクのみがレスポンス属性で返されます。

レスポンス属性名は、[レスポンス属性]ウィンドウの[変数名]フィールドに入力します。レスポンス属性名の構文は以下のとおりです。

SM_USER_APPLICATION_TASKS[:*application_name*]

application_name には、Identity Manager で定義されたアプリケーションの任意の名前を指定します。

application_name に指定した値は、ポリシー サーバ管理者に知らせておく必要があります。アプリケーション名が、自動的に 管理 UI に渡されることはありません。

注: Identity Manager タスクの詳細については、「CA Identity Manager 操作ガイド」を参照してください。

詳細情報:

[Web エージェント レスポンス属性の設定 \(P. 639\)](#)

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 承諾	On Auth 拒否	On Access 承諾	On Access 拒否	
SM_USER_CONFIDENCE_LEVEL	はい	はい	はい	はい	はい	いいえ
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

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 承諾	On Auth 拒否	On Access 承諾	On Access 拒否	
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_USERGROUPS[	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
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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] - [レスポンス グループ] をクリックします。
2. [レスポンス グループの作成] をクリックします。
3. [OK] をクリックします。
4. ドメインを選択し、[次へ] をクリックします。
5. レスポンス グループの名前および説明を入力します。

6. 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] で、[追加/削除] をクリックします。

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. [完了] をクリックします。
レスポンス グループが作成されます。

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。

以下の手順に従います。

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、「ポリシー サーバ オブジェクトの管理」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、「ポリシー サーバ オブジェクトの管理」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. 667)

[ポリシーの設定方法](#) (P. 671)

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には、以下のプロパティを識別するオブジェクトをまとめて結び付け（バインド）ます。

- ユーザ
- リソース
- リソースに関連付けられているアクション

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。

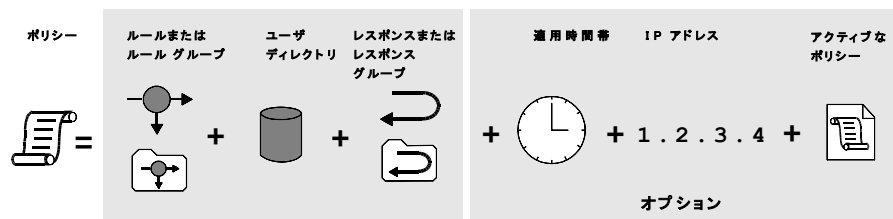
ポリシー サーバはルールによってリソースを識別します。ポリシーを作成する際は、ルールを選択して、ポリシーに含めるリソースを指定できます。

ポリシー内のユーザとリソースを識別した後は、ユーザが指定のリソースにアクセスした場合に発生するアクションを指定できます。アクションは、レスポンスの形式で実行されます。ポリシーには、以下の項目を実行するレスポンスを含めることができます。

- リソースへのアクセスを許可するまたは拒否する
- ユーザセッション時間をカスタマイズする
- ユーザを他のリソースにリダイレクトする
- ユーザディレクトリに含まれる属性に基づいて、ユーザが受信するコンテンツをカスタマイズします。

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。

ポリシーを構成するすべての要素を次の図に示します。



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、[ユーザディレクトリ] 画面で設定します。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ほかのアプリケーションが使用できるようにしたり、コンテンツ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 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。

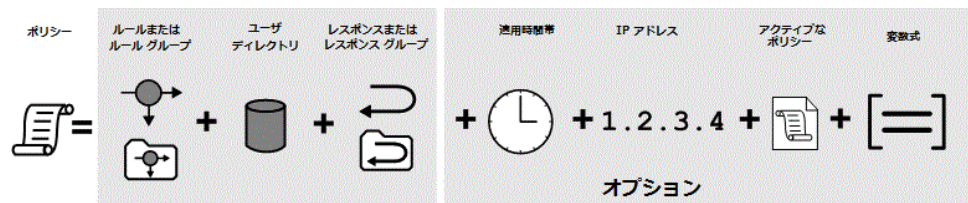
ポリシーにおける式

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」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。

詳細情報:

[変数](#) (P. 675)

ポリシーの信頼レベル

SiteMinder がサポートされたリスク分析エンジンに統合される場合、信頼レベルはポリシーで利用可能です。信頼レベルは、ポリシーを拡張して、ユーザ認証の一部として完了するリスク評価の結果を含めます。許可決定を下すとき、ポリシー サーバはこれらの結果を使用できます。

以下のオブジェクトに信頼レベルを適用できます。

- ポリシー レルム。

レルムで設定する信頼レベルは、レルムと関連付けられるすべてのリソース（ルール）に適用されます。ポリシー レルムに信頼レベルを適用する場合、ユーザが信頼レベル サポートを有効にする必要があります。

- アクティブなポリシー式。

アクティブなポリシー式として設定する信頼レベルは、ポリシーにバインドされるリソース（ルール）にのみ適用されます。アクティブなポリシー式ではより詳細な許可決定が可能です。信頼レベルを適用するアクティブなポリシー式の使用は、以前のリリースからサポートされており、デフォルトで有効になります。

注: SiteMinder は、CA Arcot WebFort および CA Arcot Risk のオンプレミス実装による統合および CA Arcot A-OK のホスト実装をサポートします。

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。

以下の手順に従います。

1. [ポリシーを作成](#) (P. 672) します。
2. [ポリシーにユーザを追加](#) (P. 673) します。
3. [ポリシーに 1 つ以上のルールを追加](#) (P. 674) 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ます。
5. (オプション) グローバル レスポンスにルールを関連付けます。
6. (オプション) 。 ポリシーの詳細オプションを設定します。

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、ポリシーを作成します。ポリシーでは、ユーザとリソースの関係を定義します。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [ドメイン] をクリックします。
3. 変更するドメインの名前をクリックします。
4. [変更] をクリックします。
5. [ポリシー] タブをクリックします。
6. [作成] をクリックします。
7. ポリシーの名前および説明を入力します。
8. (オプション) 必ずユーザの再認証を行いたいリソースをポリシーで保護する場合は、[アイデンティティの検証] チェック ボックスを選択します。たとえば、ある銀行口座から別の銀行口座にお金を振り込む前に、かならずユーザの再認証を行いたい場合は、アイデンティティの検証チェック ボックスをオンにします。振り込みを行う前に再認証が必要になります。SiteMinder セッションがまだ有効な間にマシンを離れても、ユーザはこの設定により保護されます。現在の SiteMinder セッションは影響を受けません。

注: この設定を行うには、ポリシー サーバとエージェントで追加の設定が必要です。詳細については、「*Scenario: Require Re-Authentication for Sensitive Resources*」というタイトルのナレッジ ベース ドキュメントを参照してください。

9. [ユーザ] タブをクリックします。
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。
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。

以下の手順に従います。

1. [ユーザ] ページに移動します。
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] ページが再度開き、新しいユーザが表示されます。

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 グループを追加してください。

以下の手順に従います。

1. [ポリシー] - [ルール] に移動します。
[ルール] ページが開きます。
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] セクションに、追加されたルールおよびグループが表示されます。
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。

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] セクションでルールの右側のマイナス記号 (-) をクリックします。ルールを作成するには、[使用可能なルール] ウィンドウで [New Rule] をクリックします。

第 18 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. 675)

[変数の概要](#) (P. 680)

[Web サービス変数](#) (P. 683)

[変数の作成](#) (P. 687)

eTelligent ルール

eTelligent ルールを使用して、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します。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 Web サイト上の保護されたリソースへのアクセスを試みた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、Web サービスが提供す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。

eTelligent ルールのメリット

eTelligent ルールには、以下のメリットがあります。

- 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。

SiteMinder 管理者は、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、ポリシー式で許可アクセスを定義します。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、ポリシー式を定義します。

- 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 ルールの設定方法

eTelligent ルールを設定するには、以下の手順に従います。

1. 以下の eTelligent ルールのプロパティ ファイルを設定します。
 - JVMOptions.txt
 - LoggerConfig.properties
2. 変数を設定します。
3. eTelligent ルールの変数を使用するポリシー式を設定します。

eTelligent ルールのプロパティ ファイル

eTelligent ルールにおいて必要なプロパティ ファイルは、以下のとおりです。

■ JVMOptions.txt

このファイルは、eTelligent ルールの JVM の設定に必要です。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

詳細情報:

[JVMOptions.txt ファイル](#) (P. 677)

[LoggerConfig.properties ファイルの変更](#) (P. 678)

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` 命令文を変更できます。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。デバッグ目的としてより多くの出力を取得するには、このファイルを変更します。

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。
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、以下のいずれかの値を指定できます。

LOG_LEVEL_NONE

メッセージはログに記録されません。

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には、このパラメータを **N** に設定します。

AlwaysWriteToSystemStreams

ロガー ストリーム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。ロガー ストリームが 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 サーバは、リクエストの許可段階に組み込むことができる値に変数オブジェクトを解決します。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. 680)
- [要求コンテキスト変数](#) (P. 680)
- [ユーザ コンテキスト変数](#) (P. 681)
- [フォーム POST 変数](#) (P. 681)
- [Web サービス変数](#) (P. 682)

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实践できることです。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

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、以下のプロパティを特定します。

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。

リクエスト コンテキスト変数を使用すると、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 POST 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` に入力される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 HTML ログイン フォームで指定される購入要求量が必要な場合は、フォーム ポスト変数オブジェクトを作成して、ユーザが指定する購入要求の値を収集します。この変数をポリシーに使用できます。

重要: EJB およびサーブレット エージェントは、フォーム ポスト変数をサポートしません。従って、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 ポスト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、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** に入力される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 **HTML** ログインフォームで指定される購入要求量が必要な場合は、フォーム ポスト変数オブジェクトを作成して、ユーザが指定する購入要求の値を収集します。この変数をポリシーに使用できます。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** ストリームから抽出し、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。

詳細情報:

[Web サービス変数](#) (P. 683)

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、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ための、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リ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。この例は 2 つの変数を必要とします。これは式で組み合わせる必要があります。

詳細情報:

[ポリシーにおける式 \(P. 670\)](#)

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処理した時点で作成されます。

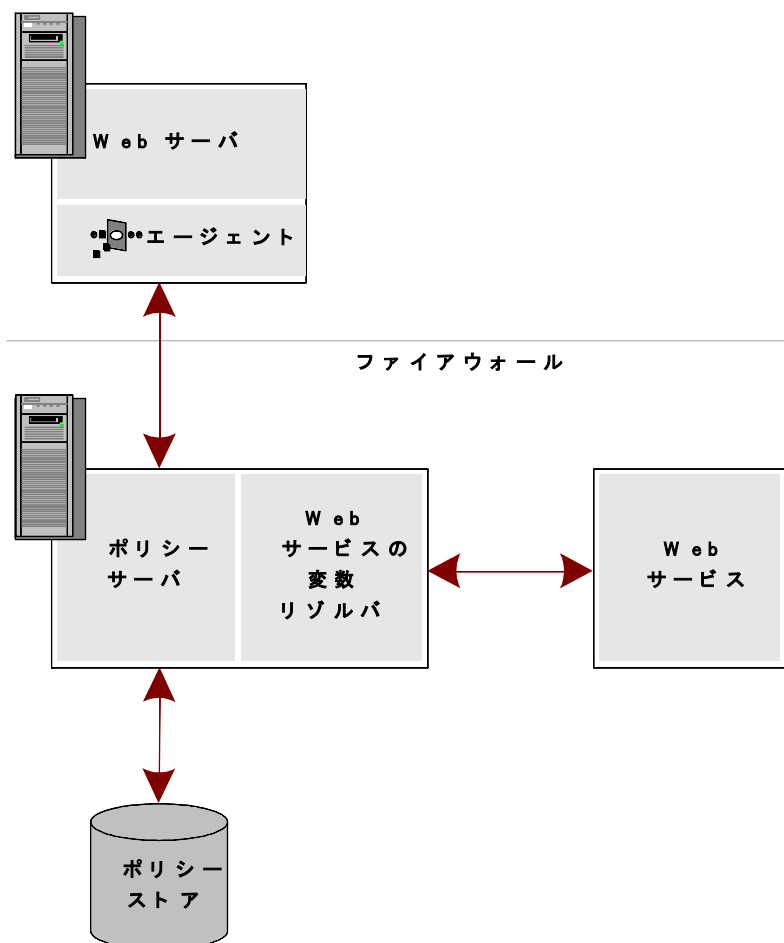
Web サービス変数

Web サービスの変数を使用すれば、Web サービスからの動的データを SiteMinder 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処理されます。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 展開のイントラネット内で処理される仕組みを示します。Web サービスは、ポリシー サーバと同じファイアウォール側にあります。



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処理します。Web サービスの変数リゾルバは同じ処理空間で実行されます。

Web サービスの変数を定義するとき、ユーザは、SOAP ドキュメントを指定して Web サービス、認証情報および他のパラメータに送ります。

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。**SSL** 接続は、**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された証明機関を使用します。

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 ストアが必要です。

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。セッション ストアをアップグレードする詳細については、「*SiteMinder アップグレードガイド*」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 サービスの変数进行处理する場合、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。モデルは、ユーザディレクトリの属性にアクセスするために、SiteMinder が使用するモデルに類似しています。
- Web サービスの認証情報を含む 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。これは、ポリシー サーバが信頼された CA のリストを使用して設定される必要があることを意味します。

注: SSL 接続については、Web サービスのサーバ側の証明書を設定します。ポリシー サーバ上で信頼された CA のリストを設定します。信頼された CA を設定するには、「認証機関と Web サービス変数」で説明されている証明書データストアを使用します。

Web サービスの変数リゾルバの設定

ポリシー サーバで Web サービスの変数を処理するには、Web サービスに接続するように Web サービスの変数リゾルバを設定します。Web サービス接続には、以下の 2 つのカテゴリがあります。

- 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。これらのプロパティは変更できます。

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 サービスの変数进行处理する際に SSL 接続を使用するには、信頼された認証機関 (CA) のリストを設定します。ポリシー サーバは「Web サービス」への接続を確立するとき CA を使用します。

証明書データ ストア内のリストを設定するには、管理 UI を使用します。

変数の作成

ポリシーまたはレスポンスででき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。

詳細情報:

[ドメイン](#) (P. 587)

スタティック変数の作成

ポリシーまたはレスポンスでできるように、スタティック変数を作成できます。

注: 解決された変数の値が 1 K を超えない値である必要があります。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [変数] をクリックします。
3. [変数の作成] をクリックします。
[変数タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
5. リストからドメインを選択し、[次へ] をクリックします。
6. [名前] フィールドに変数名を入力します。
7. [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。

8. [変数情報] で、変数のデータ型および値を指定します。
9. [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できます。

注: 解決された変数の値が 1 K を超えない値である必要があります。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [変数] をクリックします。
3. [変数の作成] をクリックします。
[変数タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
5. リストからドメインを選択し、[次へ] をクリックします。
6. [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

7. [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。
8. [プロパティ] リストから変数値を選択します。
9. [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 を超えない値である必要があります。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [変数] をクリックします。
3. [変数の作成] をクリックします。
[変数タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
5. リストからドメインを選択し、[次へ] をクリックします。
6. [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

7. [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。
8. [項目] 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。
9. (セッション変数の場合は必須) [戻り型] フィールドの変数 (ブール値、数値、文字列または日付) で表されるデータのタイプを指定します。
他の項目リストの選択については、戻り型の値は、文字列またはブール値で適宜事前に設定されており、ユーザは設定できません。
10. (ユーザ プロパティ、ディレクトリ エントリおよびセッション変数の場合は必須) [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。
11. (ユーザ プロパティ、ディレクトリ エントリおよびセッション変数の場合は必須) [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。

12. (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。
13. [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。

ユーザ コンテキスト変数进行处理する場合の複数値のユーザ属性結果のサポートの設定

デフォルトでは、ディレクトリの指定されたユーザ プロパティの値を取得するように設定されたユーザ コンテキスト変数は、単一値のユーザ属性のみをサポートします。

ユーザ コンテキスト変数进行处理の場合に、複数値のユーザ属性結果がサポートされるようにポリシー サーバを設定できます。

以下の手順に従います。

1. ポリシー サーバを停止します。
2. テキスト エディタで次のファイルを開きます。
`ps_install_dir¥config¥properties¥scriptActiveExpConfig.properties`
3. 以下のエントリを含む行を、ファイルの任意の場所に追加します。
`ALLOW_MULTIVALUSERATTR=1`
4. ファイルを保存し、テキスト エディタを閉じます。
5. ポリシー サーバを起動します

フォーム ポスト変数の作成

ポリシーで使用できるように、フォーム ポスト変数を作成できます。

注: 解決された変数の値が 1 K を超えない値である必要があります。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [変数] をクリックします。
3. [変数の作成] をクリックします。
[変数タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
5. リストからドメインを選択し、[次へ] をクリックします。
6. [名前] フィールドに変数名を入力します。
7. [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。
8. [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。
9. [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用でき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** を超えない値である必要があります。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [変数] をクリックします。
3. [変数の作成] をクリックします。
[変数タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
4. [OK] をクリックします。
5. リストからドメインを選択し、[次へ] をクリックします。
6. [名前] フィールドに変数名を入力します。
7. [変数のタイプ] リストから [**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。
8. [戻り型] リストからデータ タイプを選択します。
9. URL フィールドに、**Web** サービスの URL を入力します。
10. [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。
11. (オプション) [**Web** サービス認証情報] の [認証情報が必要] オプションを選択し、**Web** サービス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。
12. (オプション) [**SOAP** ドキュメント] セクション内の以下のボタンをクリックして、既存の変数を SOAP メッセージに追加します。

変数

13. (オプション) [HTTP ヘッダ] 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。

14. [完了] 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 19 章：キーおよび証明書管理

証明書および秘密キーの使用

以下のタスクでは、秘密キーと証明書が必要です。

- フェデレーション コンポーネントは、アサーション全体、または特定のアサーションの内容の署名、検証、暗号化および復号に対して秘密キー/証明書ペアを使用します。
- フェデレーション コンポーネントは、**Artifact** シングル サインオン用のバックチャネル認証に対してクライアントの証明書を使用します。
- SSL 接続の確立（SSL サーバ証明書）。

フェデレーション機能用の秘密キー/証明書ペアおよび単一の証明書は、証明書データストア（CDS）に格納されます。証明書データストアは、ポリシーストアと連結されます。同じポリシーストアに共通のビューを共有するポリシーサーバはすべて、同じキー、証明書、CDS 設定の証明書廃棄リスト（CRL）、および OCSP レスポンドにアクセスできます。

SSL サーバ証明書は、それらがインストールされている Web サーバ上に格納されます。SSL サーバ証明書は、証明書データストアには格納されません。

証明書データストア内のキー/証明書ペア、クライアント証明書、および信頼された証明書にはそれぞれ、一意のエイリアスが必要です。エイリアスを使用すると、証明書ストア内のあらゆる秘密キー/証明書ペアまたは単一の証明書を一意に参照できます。証明書データストアには複数のキー/証明書ペアおよび単一の証明書を格納できます。フェデレーション環境では、複数のパートナーを持てます。複数のパートナーに対して、各パートナー用に別のペアを使用できます。

署名するエイリアスがアサーションの署名に対して設定される場合、アサーション ジェネレータでは、そのエイリアスと関連付けられるキーを使用してアサーションを署名します。署名するエイリアスが設定されない場合、アサーション ジェネレータでは、**defaultenterpriseprivatekey** エイリアスによるキーを使用してアサーションを署名します。アサーション ジェネレータではデフォルトのエンタープライズ秘密キーが見つからない場合、アサーションを署名するために検索した最初の秘密キーを使用します。

重要: 複数のキーを格納する場合は、後のキーを追加する前に、**defaultenterpriseprivatekey** エイリアスで追加する最初のキーを定義します。

指定されたポリシー サーバは、レスポンドを署名したり、署名および検証することができます。署名および検証に使用されるキーと証明書を、同じ証明書データ ストアに追加できます。

管理 UI を使用して、証明書データ ストアの内容を管理します。

以下のタイプの秘密キー/証明書ペアおよび単一の証明書が、証明書データストアに格納されます。

機能	秘密キー/証明書ペア	証明書 (公開キー)	CA 証明書	クライアント証明書
アサーション、認証リクエスト、SLO リクエスト、レスポンスを署名する	X			
アサーション、認証リクエスト、SLO リクエスト/レスポンスを検証する		X		
アサーション、名前 ID、属性を暗号化する (SAML 2.0 のみ)		X		
アサーション、名前 ID、属性を復号化する (SAML 2.0)	X			
Artifact バック チャネルのクライアント証明書認証用の認証情報として機能する				X
他の証明書および証明書廃棄リストを検証する			X	
SSL 接続を使用して、Web サービス変数进行处理する			X	

詳細情報:

[署名および検証操作](#) (P. 698)

[SSL 接続用の証明書](#) (P. 699)

署名および検証操作

ポリシー サーバは、署名および検証タスクに秘密キー/証明書ペアを使用します。秘密キー/証明書ペアはアサーション、アサーション レスポンス、または認証リクエストを署名します。署名される特定のメッセージは、トランザクションが発生する場所および使用中のフェデレーション プロファイルによって異なります。

トランザクションの署名前に、アサーションを署名するパートナーは、パートナーに対して秘密キー/証明書ペアと関連付けられている証明書（公開キー）を送信します。この通信は、帯域外の通信として行われます。パートナーは証明書を使用して、署名を検証します。

トランザクションが発生すると、アサーティング パーティにはデフォルトでアサーションの証明書が含まれます。ただし検証プロセス中に、パートナーはそのサイトに格納される証明書を使用して署名を検証します。

SAML 2.0 の単一のログアウトについては、ログアウトを開始する側はリクエストに署名し、リクエストを受信する側は署名を検証します。反対に、受信する側は SLO レスポンスを署名し、ログアウト開始側はレスポンスを検証します。

暗号化/復号化の操作

SAML 2.0 の場合は、アサーション全体、NameID、またはその他の属性を暗号化するようにポリシー サーバを設定できます。暗号化を有効にすると、アサーティング パーティでは、依存パーティがデータを暗号化するために送信する証明書（公開キー）を使用します。トランザクションの前に、依存パーティは、帯域外の通信でアサーティング パーティに証明書を送信します。依存パーティは秘密キー/証明書ペアを使用して、データを復号します。

注: SAML 1.1 および WS フェデレーションは、アサーション データの暗号化をサポートしていません。

SSL 接続用の証明書

ポリシー サーバは以下の方法で SSL 接続を使用します。

- フェデレーション通信の場合

SAML HTTP Artifact バック チャネル、または一般的なフェデレーション通信に対して SSL を有効にできます。依存パーティが署名済みの SSL サーバ証明書に CA 証明書を関連付けるには、SSL 接続が必要です。

SSL サーバ証明書は、SSL 接続をセキュリティで保護します。CA 証明書は、SSL サーバ証明書の信頼性を検証します。

- ICAS の場合

依存パーティのフォーム認証情報コレクタ ファイルを保護するために、SSL 接続を有効にできます。依存パーティの Web サイトから証明書データストアに、SSL サーバ証明書をインポートします。サーバ証明書は、SSL 接続をセキュリティで保護します。

- Web サービス変数の場合

Web サービス変数を解決するために SSL 接続を有効にできます。

注: SSL サーバ証明書は、それらがインストールされている Web サーバ上に格納されます。SSL サーバ証明書は、証明書データストアには格納されません。

Artifact バック チャネルをセキュリティ保護する証明書

Artifact のバインドを使用して、シングルサインオンを実装するには、依存パーティがアサーションに対するリクエストをアサーティングパーティの SiteMinder に送信します。アサーション リクエストはアサーション検索サービス (SAML 1.1) または Artifact 解決サービス (SAML 2.0) に送られます。検索サービスは、依存パーティによって提供された Artifact を取得し、それを使用してアサーションを検索します。SiteMinder ではバックチャネルを介して依存パーティにレスポンスを送信します。バックチャネルはアサーティングパーティと依存パーティとの間の安全性が確保されている接続です。一方、Web ブラウザ通信はフロントチャネル上で発生します。

以下のいずれかの認証方式を使用して、不正なアクセスからバック チャネルと検索サービスを保護します。

- 基本
- SSL を介した基本
- X.509 クライアント証明書

認証方式として **X.509** クライアント証明書を使用する場合、依存パーティはその認証情報としてクライアント証明書を提供する必要があります。この認証情報を使用して、依存パーティはアサーションを検索するアサーティングパーティのサービスにアクセスが可能になります。

認証方式を選択する際には、以下の点を考慮します。

- バック チャネルに対して **SSL** 接続の使用を考慮します。信頼された **CA** によって署名された **SSL** サーバ証明書で、**SSL** 接続の安全性を確保します。

共通ルートおよび中間 **CA** 証明書のデフォルトのセットは、証明書データストアに付属しています。**CA** によって署名された別のサーバ証明書を使用するには、信頼済みの **CA** 証明書として **CA** 証明書をストアにインポートします。

バック チャネル リクエストを処理する場合、フェデレーションは **SSL** クライアントを使用します。アサーティングパーティの **Web** サーバを設定して、以下の暗号によって、**SSL** バージョン **TLSV1_1** および **TSLV1_2** を使用できます。

- **RSA_With_AES_128_CBC_SHA256**
- **RSA_With_AES_256_CBC_SHA256**

これらの暗号は **FIPS** および非 **FIPS** モードの両方でサポートされています。**SHA256** の使用の可否については、**SP** サーバ側で決定されます。フェデレーションには、アルゴリズムを選択するための設定がありません。管理者は、アサーティングパーティのサーバが適切に設定されていることを確認する必要があります。

- **X.509** クライアント証明書が、接続の確立に必須である場合、依存パーティにはキー/証明書ペアが必要であり、これがない場合にはクライアント証明書の認証が失敗します。クライアント証明書がアサーティングパーティの証明書データストアに存在することを確認します。依存パーティがアサーションのリクエストを送信するとき、クライアント証明書が依存パーティの認証情報として機能して検索サービスにアクセスします。

証明書データストアでサポートされている形式

証明書データストアは以下の形式をサポートします。

- 秘密キーは以下の形式でサポートされており、DER または PEM エンコード形式になります。
 - PKCS1
 - PKCS5
 - PKCS8
 - PKCS12
- RSA キーのみがサポートされています。
- 公開証明書は、以下の X.509 証明書形式でサポートされています。
 - V1
 - V2
 - V3
- 公開証明書は、以下のエンコード形式でサポートされています。
 - DER
 - Base64
 - PEM

信頼された証明書およびキー/証明書ペアのインポート

キー/証明書ペアおよび信頼された証明書は、多くのフェデレーション SSO やその他の機能に影響を与えます。キーと証明書を使用するタスクを実行するには、これらの項目が証明書データストアにある必要があります。

証明書データストアにキー/証明書のペアがない場合、処理として以下の2つの選択肢があります。

- 既存のファイル（.p12 または .pfx）からキー/証明書のペアをインポートする。
- キー/証明書のペアを生成する。

新規のキー/証明書ペアを生成するには、信頼された認証機関からの証明書をリクエストし、機関から戻される署名された証明書レスポンスをインポートします。

詳細情報:

[証明書および秘密キーの使用 \(P. 695\)](#)

[既存のファイルからキー/証明書のペアをインポートする \(P. 702\)](#)

[キー/証明書ペアの作成方法 \(P. 703\)](#)

既存のファイルからキー/証明書のペアをインポートする

証明書データストアにキー/証明書のペアがない場合、既存の .p12 または .pfx ファイルからそれをインポートします。

ポリシー サーバはインポートされた証明書を信頼された証明書として処理します。例外は自己署名証明書で、以下のガイドラインに従って処理されます。

- ポリシー サーバは V3 自己署名証明書を CA 証明書として識別します。この場合、ポリシー サーバは CA 証明書として処理します。この処理は、[証明書/秘密キー] ダイアログ ボックスからインポートを開始した場合も同様です。
- ポリシー サーバは以下の場合、証明書を信頼された証明書として処理します。
 - ポリシー サーバが V3 自己署名証明書を CA として識別しない場合。
 - 証明書が V1 自己署名証明書の場合。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書管理] - [信頼された証明書および秘密キー] を選択します。
3. [新規インポート] をクリックし、ウィザードに従います。

4. ウィザードを実行する際に、以下の点に注意します。
 - 中に含むキーと証明書、または個別のキーと証明書ファイルを使用して、単一のファイルをインポートできます。使用しているファイルに該当するオプション ボタンを選択します。
 - 認証機関証明書として自己署名証明書をインポートするには、
[CA として使用] オプション ボタンを [はい] に設定します。証明書は CA 証明書としてインポートされ、パートナーシップを設定する際には使用できません（たとえば、署名または暗号化）。
それ以外の場合、デフォルトの [いいえ] 設定を受け入れ、パートナーシップを設定する場合に使用可能な信頼された証明書として証明書をインポートします。
 - DER（バイナリ）形式の信頼済み証明書ファイルについては、ファイルに 1 つ以上の証明書エントリを含めることが可能です。PEM（base 64）形式の信頼済み証明書ファイルについては、1 ファイルにつき 1 つの証明書が必要です。
 - .p12 ファイルを使用している場合、パスワードの入力が必要です。
 - 証明書データ ストアに追加する予定の各エントリに対しては、そのエントリと関連付けるエイリアスを入力します。複数のエントリを選択する場合、各エントリごとに一意のエイリアスが必要です。
5. 「確認」の手順では、情報を確認し、[完了] をクリックします。

キー/証明書のペアが証明書データ ストアにインポートされます。

キー/証明書ペアの作成方法

証明書データ ストアにキー/証明書のペアがない場合、新規のキー/証明書のペアを作成できます。

以下の手順に従います。

1. 証明書リクエストを生成し、信頼された認証機関にリクエストを送信します。
2. 機関から署名済み証明書レスポンスをインポートします。

証明書リクエストの生成

証明書データストアにキー/証明書のペアがない場合、信頼された認証機関からそれをリクエストします。CA が署名済みの証明書レスポンスを返すとき、それを証明書データストアにインポートします。

証明書リクエストを生成すると、ポリシーサーバが秘密キーと自己署名証明書のペアを生成します。ポリシーサーバは証明書データストアにこのペアを格納します。生成されたリクエストを使用して、認証局に問い合わせ、CA 証明書リクエストフォームに入力します。生成されたリクエストの内容をフォームに貼り付けます。

CA は通常、署名された証明書レスポンスを PKCS #7 形式で発行します。署名済み証明書レスポンスを証明書データストアにインポートできます。署名された証明書レスポンスがインポートされると、同じエイリアスの既存の自己署名証明書のエントリが置き換えられます。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書管理] - [信頼された証明書および秘密キー] を選択します。
3. [証明書のリクエスト] をクリックします。
4. 必要なフィールドを入力します。
5. [保存] をクリックします。

PKCS #10 の仕様に一致するファイルが生成されます。

証明書リクエストが含まれるファイルを保存または開くように、ブラウザのメッセージが表示されます。このファイルを保存しない場合（または開いてテキストを抽出する場合）でも、ポリシーサーバは秘密キーと自己署名証明書のペアを生成します。秘密キーの新規リクエストファイルを取得するには、CSR の生成機能を使用して、新規の証明書署名リクエストを生成します。

署名済み証明書レスポンスのインポート

証明書リクエストを入力し、認証機関にそれを送信すると、認証機関は署名された証明書レスポンスを発行します。

署名された証明書を証明書データストアにインポートして、同じエイリアスの既存の自己署名証明書エントリを置き換えます。

以下の手順に従います。

1. [インフラストラクチャ] - [X509 証明書管理] - [証明書管理] - [信頼された証明書および秘密キー] を選択します。
2. リストから、更新する自己署名証明書を見つけます。
3. [アクション]、自己署名済みエントリの横の [証明書の更新] を選択します。
4. 必要なファイルを参照します。以下を使用できます。
 - 署名された証明書および対応する証明書チェーンが含まれる .p7 または .p7b ファイル。
 - 証明書チェーンのない署名済み証明書を含む .cer または .crt ファイル (base64 PEM ファイル)。
5. 該当のエントリを選択します。
6. 証明書情報を確認し、[完了] をクリックします。

署名された証明書が、証明書データストアにインポートされ、自己署名証明書が置き換えられます。

新規の証明書署名リクエストの生成

証明書署名リクエスト (CSR) は、認証機関にデジタル ID 証明書を申請するメッセージです。秘密キーを作成した後、CSR を生成できます。CSR には公開キーが含まれます。

自己署名または、CA 署名された秘密キー/証明書のペアに対して、新規の CSR を生成できます。秘密キーは既存の秘密キーを変更せずに、常に同一の CSR を生成します。以下の理由で、既存の秘密キーに対して新規リクエストを生成します。

- 秘密キー/自己署名証明書ペア用に生成された元のリクエストがもうない。
- 期限切れになるため新規の証明書が必要であり、それは認証機関にサブミットするために CSR の新規コピーを必要とする。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書管理] - [信頼された証明書および秘密キー] を選択します。
3. 必要としている新規 CSR の秘密キーのエントリに対して、[アクション]、[CSR の作成] を選択します。

PKCS #10 の仕様に一致するファイルが生成されます。

4. プロンプトが表示されたら CSR を保存します。
5. (オプション) 認証局で署名された証明書が必要な場合は、認証局に問い合わせます。認証局で要求される手順に従ってリクエストをサブミットします。リクエストについて前の手順で保存した PKCS#10 ファイルを使用します。

証明書リクエストのプロセスを完了した後、認証機関が署名済みの証明書レスポンスを発行します。このレスポンスを証明書データストアにインポートします。ポリシーサーバが、同じエイリアスの既存の証明書エントリを新しくインポートされた証明書に置き換えます。

証明書データストアの証明書の更新

以下の方法で、キー/証明書ペアおよびスタンドアロンの証明書を更新できます。

- 期限切れになる信頼済み証明書の更新は、既存の証明書を削除して信頼される新規の証明書をインポートすることにより対応します。新規証明書は、証明書データストア内の期限切れになる証明書に一致する必要があります。
- 信頼される署名済み証明書または PKCS7 署名済みレスポンスをインポートすることにより、証明書を更新します。新規証明書は、証明書データストア内の期限切れになる証明書に一致する必要があります。
- PKCS#12 ファイルからの証明書で証明書を更新します。新規の秘密キーと証明書のペアは、証明書データストア内の期限切れになる秘密キー/証明書ペアに一致する必要があります。

新規の証明書は、ポリシー サーバが期限切れになる証明書を更新するために使用する前に、有効になっている必要があります。証明書はインポートされると直ちに、更新され利用可能になります。新規の証明書が有効でない場合（その有効期間で判断）、ポリシー サーバは新規の証明書を使用できません。

信頼された証明書のみインポートできるように、PEM または DER エンコーディングの証明書を含むファイルを使用します。これらのタイプのファイルの標準的な拡張子は *.crt または *.cer です。ファイルが .p12 または .pfx で終わる場合、それはキー/証明書ペアを含む証明書データ ストア ファイルとして処理されます。また、ファイルが .p7 または .p7b で終わる場合は、署名されたレスポンス ファイルとして処理されます。それ以外のファイルはすべて証明書ファイルとして扱われ、SiteMinder では、そこから証明書をロードしようとします。

注: フェデレーション環境の証明書を更新する場合、期限切れの証明書を使用するフェデレーション オブジェクトを更新する必要はありません。

証明書とキー データのエクスポート

秘密キー/証明書ペアをエクスポートし、フェデレーション パートナーにそれを送信できます。パートナーは証明書を使用して、関連付けられる証明書の秘密キーで作成されたデジタル署名を検証できます。

重要: バックアップの一部として秘密キーをエクスポートする場合は、他の誰ともそれを共有しないでください。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書管理] - [信頼された証明書および秘密キー] を選択します。
3. エクスポートする必要があるエントリに対して、[アクション]、[エクスポート] を選択します。
4. エクスポートされたデータから作成するファイルの形式を選択します。

5. ファイル形式を選択します。
6. [Export] をクリックします。

ローカル システムでファイルを開くか保存するかを問われます。

キーまたは証明書を示す、エンコードされたファイル コンテンツが作成されます。

認証機関(CA)証明書の使用

フェデレーション システムは認証機関の証明書を使用して、以下の項目を検証します。

- SSL 接続の SSL サーバ証明書が SAML HTTP Artifact バック チャンネルが信頼できることを保証するものであるかどうか。
- HTTP Artifact シングル サインオンの場合、SSL 接続を使用してバック チャンネルをセキュリティで保護します。フェデレーション システムに組み込まれた Web サーバで、認証機関の証明書を検証することによって信頼される証明書によって SSL 接続がセキュリティ保護されていることを確認できます。この証明書は、証明書データ ストアに格納されている必要があります。
- 証明書廃棄リストが有効かどうか。

CRL は認証機関から取得されます。対応する CA の証明書は、信頼されるまで CRL を検証する必要があります。CRL はデータ ストアに格納され、ランタイムで使用されます。

共通ルートおよび中間 CA 証明書のデフォルトのセットは、その目的のために製品に付属しています。

CA 証明書のインポート

共通ルートおよび中間 CA のセットが、製品に含まれています。証明書データストアにない CA 証明書を使用するには、それらをインポートします。

インポートする証明書は CA 証明書として扱われます。例外は、以下のようない自己署名証明書です。

- システムが V3 自己署名証明書を非 CA 証明書として識別する場合、証明書は信頼された証明書として処理されます。この処理は、[CA 証明書のインポート] ダイアログ ボックスからインポートを開始した場合も同様です。
- システムが V1 自己署名証明書を識別する場合、証明書は CA 証明書として扱われます。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [認証機関] を選択します。
3. [新規インポート] をクリックします。
4. ウィザードに従って、新規エントリをインポートします。
5. 証明書情報を確認し、[完了] をクリックします。

CA 証明書が証明書データストアにインポートされます。変更は、インポートの終了直後に有効になります。

重要: CA 証明書がシステムで使用中の他の証明書の信頼チェーンの一部である場合、それを削除することはできません。使用中の CA 証明書を削除しようとする、証明書を削除できないことを知らせるエラーメッセージが表示されます。

CRL による証明書の有効性チェック

証明書失効リスト（CRL）は、証明機関によってそのサブスクライバへ発行されます。リストには、無効のまたは取り消された証明書のシリアル番号が含まれます。サーバにアクセスする要求が受信されると、サーバは CRL に基づいてアクセスを許可または拒否します。

フェデレーションは、その証明書機能の CRL を利用できます。SiteMinder が CRL を使用するには、証明書データストアが現在の CRL を参照する必要があります。SiteMinder が取り消されたパートナー証明書を使用しようとすると、エラーメッセージが表示されます。レガシーフェデレーションについては、エラーメッセージは SAML アサーション内に表示されます。メッセージは、認証が失敗したことを示します。

注: フェデレーション機能は、X.509 認証方式とは異なった方法で CRL を使用します。認証方式は、CRL を格納する独立した LDAP ディレクトリを使用します。認証方式は証明書データストアを使用しません。

SiteMinder は、以下の CRL 機能をサポートします。

- ファイルベースの CRL または LDAP CRL

SiteMinder は、証明書データストアに CRL を格納します。ファイルベースの CRL は、Base64 またはバイナリエンコーディングにある必要があります。LDAP CRL はバイナリエンコーディングにある必要があります。さらに、LDAP CRL には以下のいずれかの属性に CRL データが含まれる必要があります。

- certificateRevocationList;binary
- authorityRevocationList;binary

証明機関が LDAP CRL を発行する場合、RFC4522 および RFC4523 に従ってバイナリフォーマットで CRL データを返す必要があります。そうしないと、SiteMinder は CRL を使用できません。

- ファイル CRL 用の PEM および DER のエンコード形式
- LDAP CRL 用の DER エンコード形式

SiteMinder は、CRL に対する SSL サーバ証明書を検証しません。SiteMinder Web エージェントがインストールされている Web サーバは、SSL サーバ証明書を管理します。

システム内の各ルート CA の CRL を保持している必要はありません。ルート CA の CRL がない場合、その CA によって署名されたすべての証明書が信頼された証明書であると仮定されます。

証明書管理用の CRL の追加

CRL の使用によって有効な証明書のみが PKI 機能に使用されることを確認できます。CRL に対して証明書の有効性を検証します。

重要: SiteMinder は、バイナリ エンコーディングでの LDAP CRL を明示的に要求します。さらに、CRL データは *certificateRevocationList;binary* または *authorityRevocationList;binary* という名前の LDAP 属性に保存される必要があります。認証機関 (CA) は LDAP CRL を発行するときに、RFC4522 および RFC4523 に従って、バイナリ フォーマットで CRL データを返す必要があります。そうしないと、SiteMinder は CRL を使用できません。

CRL を使用するには、CRL の場所が必要です。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書の有効期間] を選択します。
3. [追加] をクリックします。
[取り消しリストの設定] ダイアログ ボックスが表示されます。
4. CRL のエイリアスおよび、証明書廃棄リストの場所 (URL) を指定します。
場所は、CRL ファイルのファイルパスおよび LDAP CRL の LDAP 検索パスである必要があります。
5. [保存] をクリックします。

CRL が証明書データ ストアに追加されました。

CRL の更新

CRL を更新して、使用中の証明書データが最新であることを確認します。

以下の手順に従います。

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [証明書の有効期間] を選択します。
3. リストから CRL を削除します。
4. 以下のいずれかの手順を実行して、CRL を追加します。
 - [追加] をクリックし、新規 CRL ファイルまたは LDAP の場所の設定を入力します。[保存] をクリックします。
 - [証明書管理] ダイアログボックスに戻ります。デフォルトの [CRL 更新期間] の値を指定します。次回 CRL 更新の際に、証明書データストアは設定されたファイルまたは LDAP の場所に自動的に移動し、CRL を再ロードします。

OCSP による証明書の有効性チェック

一部の機能では、証明書データストア内の証明書に対する証明書検証が必要になります。12.52 SP1 では、フェデレーション機能は証明書データストアを使用します。これらの機能には HTTP-Artifact バックチャネルの保護、SAML メッセージの確認、および SAML メッセージの暗号化が含まれます。

証明書の有効性を確認するために、証明書データストアは OCSP サービスを使用できます。OCSP は、証明書検証を要求に応じて提供するために証明機関 (CA) が提供する HTTP サービスを使用します。

注: フェデレーション機能は、X.509 認証方式とは異なった方法で OCSP を使用します。認証方式は、独立した LDAP ディレクトリを使用して OCSP レスポンド証明書を格納します。認証方式は証明書データストアを使用しません。

デフォルトでは、証明書データストア内の証明書の取り消しステータスは確認されません。OCSP レスポンダによって取り消しステータスを確認するには、OCSP アップデータ ユーティリティ (OCSPUpdater) を使用します。有効にすると、OCSPUpdater は設定された OCSP レスポンダの取り消しステータスを 5 分おきに確認します。このデフォルトの頻度は設定可能です。

OCSPUpdater の設定は以下のコンポーネントに依存します。

- SMocsp.conf ファイル

OCSPUpdater は、OCSP レスポンダ設定に関する SMocsp.conf ファイルを使用します。証明書を発行する各証明機関 (CA) には、それぞれ独自の OCSP レスポンダがあります。SMocsp.conf ファイルで、証明書データストアに各 CA 証明書に対するすべての OCSP レスポンダを含めます。

SMocsp.conf ファイルは OCSPUpdater を使用するために必要となります。

注: SMocsp.conf ファイルは、X.509 証明書認証方式がそれ独自の OCSP 実装を設定するために使用するファイルと同じです。

- XPSConfig ユーティリティ

XPSConfig によって、OCSPUpdater の動作をカスタマイズできます。たとえば、それを有効にしたり、更新の頻度を設定することができます。そのカスタマイズは OCSPUpdater を実行するポリシー サーバに固有のもので、展開において 1 つのポリシー サーバ上のみで OCSPUpdater を有効にします。

OCSP と CRL チェックとの間のフェールオーバー

証明書データストアは OCSP から CRL 検証へのフェールオーバーをサポートします。CRL と OCSP チェックを設定した場合、両者の間のフェールオーバーを有効にすることが可能です。

フェデレーション機能は、拡張が証明書内にあっても、フェールオーバーを設定して証明書の配布ポイントの拡張をサポートしません。

詳細情報:

[X.509 クライアント証明書認証方式における証明書の有効性チェック](#) (P. 480)

第 20 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 \(P. 715\)](#)

[グローバル ポリシーを設定する方法 \(P. 720\)](#)

[グローバル ポリシーの有効と無効 \(P. 733\)](#)

[グローバル アクティブ ポリシーの設定 \(P. 733\)](#)

[グローバル ポリシーで許可される IP アドレス \(P. 734\)](#)

[グローバル ポリシーの時間制限の追加と削除 \(P. 737\)](#)

グローバル ポリシー

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。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 ポリシーに追加されるのは、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わた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。このエンティティは、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 オブジェクトとの類似点および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。

標準レスポンスと比較したグローバル 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。
- これらのレスポンスでは、変数ベースの属性を使用できない。
- 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。
- これらのレスポンスは、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 グループは禁止されている。

類似点

- これらのレスポンスでは、アクティブな式を使用できる。
- レスポンスは特定のポリシーに指定されている場合にのみ返される。

標準ルールと比較したグローバルルール

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、特定のレルム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す。正規表現を使用できる。
- 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の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。
- これらのルールは、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は存在しません。
- これらのルールは、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。

標準ポリシーと比較したグローバルポリシー

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。
- これらのポリシーでは、変数ベースの属性や変数式を使用できない。
- これらのポリシーには、許可/拒否アクセスルールを含めることができない。グローバルポリシーでは、イベントルールのみを使用できます。

- これらのポリシーでは、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。これらのグローバルポリシーは、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

- これらのポリシーでは、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。
- ルールに対して指定されている同じエージェント/エージェントグループにより、リソースが保護される。
- 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。

重要: 以下の条件が満たされる場合、標準ポリシーはグローバルポリシーよりも優先されます。

- グローバルポリシー処理がドメインに対して有効である。
- 標準ルールとグローバルルールの両方が、同じエージェントまたはグループにバインドされている。

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. 594)

SiteMinder グローバル ポリシーの概念

本製品では、ポリシーベースのアクセス制御モデルを使用します。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。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。ポリシーは、ユーザ、ルール、およびレスポンスを結び付けることで、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。

標準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。ルールにより、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が決まります。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。

ポリシー 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）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、本製品の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の役割は、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。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

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フィルタは、（レルムからではなく）ルールから取得されます。
- 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。
- 同じエージェントまたはエージェントグループが、要求されたリソースを保護しており、これがルールに対して指定されている。

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、または許可イベントのグローバルルールを作成します
2. グローバルレスポンスの設定
3. [グローバル Web エージェント レスポンス属性の設定 \(P. 730\)](#)
4. [グローバルポリシーを設定します \(P. 731\)](#)

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ル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。これらのルールは、ドメイン固有ルールとほぼ同じです。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 ルールは使用できません。

認証イベント用のグローバル ルール

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の動作（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できます。

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できます。

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** が読み込まれたとき）に起動しますが、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。

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 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。

On-Auth-Attempt

本製品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。

詳細情報:

[グローバル レスポンス オブジェクト](#) (P. 728)

認証イベント用のグローバル ルールの作成

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のグローバル ルールを作成できます。

以下の手順に従います。

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] をクリックします。
3. [グローバルルールの作成] をクリックします。
4. グローバルルール名を入力します。
5. [レルムとリソース] 内で、エージェントとリソース設定を指定します。

注: 同じリソースに関連するドメイン固有ルールと共にエージェントグループを指定した場合は、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では、グローバルルールによって生成されたレスポンスが複製されることを考慮してください。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。

6. 認証イベントを選択します。
7. [アクション] リストから **OnAuth** イベントを選択します。
8. [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、**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

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のグローバル ルールを作成します。

以下の手順に従います。

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] をクリックします。
3. [グローバルルールの作成] をクリックします。
4. グローバルルール名を入力します。
5. [レルムとリソース] 内で、エージェントとリソース設定を指定します。

注: 同じリソースに関連するドメイン固有ルールと共にエージェントグループを指定した場合は、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では、グローバルルールによって生成されたレスポンスが複製されることを考慮してください。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。

6. 許可イベントを選択します。
7. [アクション] リストから **OnAccess** イベントを選択します。
8. [サブミット] をクリックします。

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すると、必ずルールが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たときに、ルールは起動しなくなります。

以下の手順に従います。

1. グローバル ルールを開きます。
2. 以下のタスクのいずれかを実行します。
 - [有効] チェック ボックスをオンにして、ルールを有効にします。
 - [有効] チェック ボックスをオフにして、ルールを無効にします。
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 ポリシーを特定の時間に適用するには、グローバル ポリシーに時間制限を追加します。指定した時間外にユーザがリソースへのアクセスを試みると、ポリシーは適用されません。

以下の手順に従います。

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。
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 ボックスをオンにすると、その時間中にルールが起動します。このルールは、指定したリソースに適用されます。チェック ボックスをオフにすることは、その時間中にルールは起動しないことを意味します。このルールは、指定したリソースに適用されません。

5. [OK] をクリックします。

詳細情報:

[ルールへの時間制限の追加](#) (P. 626)

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。

以下の手順に従います。

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。

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、そのルールを使用する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。

すべてのグローバル 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。

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 ポリシーの一部であり、ユーザに返される属性を定義します。これらの属性は、グローバル ルールで指定されている認証イベントまたは許可イベントをユーザがトリガした後に返されます。

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

グローバル レスポンスを設定して、関連するグローバル ルールで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できます。

以下の手順に従います。

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] をクリックします。
3. [グローバル レスポンスの作成] をクリックします。
4. グローバル レスポンスの名前を入力します。
5. [エージェント タイプ] リストからエージェント タイプを選択します。
6. [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。

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。

以下の手順に従います。

1. [レスポンス属性の作成] をクリックします。
2. レスポンス属性を選択します。
3. 属性タイプを選択します。

[属性フィールド] の詳細は、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] の詳細を入力します。

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 \(P. 655\)](#)」を参照してください。

5. (オプション) [スクリプト] フィールドで属性を編集します。

注: [詳細] セクションで属性を編集すると、[属性のセットアップ] セクションが閉じます。

6. [キャッシュ値] または [値の再計算間隔 (秒)] (Recalculate value every ... seconds) を指定します。

注: 入力できる最大時間制限は 3600 秒です。

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] ページの [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
2. [グローバル ポリシーへのグローバル ルールの追加](#) (P. 732)
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. 732)

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できます。

以下の手順に従います。

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] をクリックします。
3. [グローバル ポリシーの作成] をクリックします。
4. グローバル ポリシー名を入力します。
5. 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。
6. [サブミット] をクリックします。

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。

以下の手順に従います。

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。この方法で作成したルールは、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。

以下の手順に従います。

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。
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。

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。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。

以下の手順に従います。

1. 管理 UI で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。

以下の手順に従います。

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 ポリシーで許容される IP アドレス

ユーザが以下のいずれかの項目から保護されているリソースにアクセスすると、グローバル ポリシーが起動するように設定できます。

- 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、IP アドレスの範囲を指定すると、これらの IP アドレスからログインするユーザのみが、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

単一の IP アドレスを指定すると、指定した IP アドレスからリソースにアクセスしたユーザに対してのみ、グローバル ポリシーが起動します。

以下の手順に従います。

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。

3. [単一ホスト] オプション 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 アドレスが不明な場合は、[DNS の検索] をクリックし、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保護された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。

以下の手順に従います。

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。
3. [ホスト名] オプション 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保護された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。

以下の手順に従います。

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] オプション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 アドレスが不明な場合は、[DNS の検索] をクリックし、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すると、その範囲内の IP アドレスから保護されたリソースにアクセスしたユーザに対してのみ、ポリシーが起動します。

以下の手順に従います。

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] オプション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 アドレスが不明な場合は、[DNS の検索] をクリックし、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。

以下の手順に従います。

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 21 章: インパーソネーション(偽装)

インパーソネーションを使用すると、権限を持つユーザが以下を行うことができます。

- 権限を持つユーザ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。
- 別のユーザの識別情報で一時的に動作する。

インパーソネーションでは、1 人のユーザが別のユーザを偽装するためにパスワードを開示する必要はありません。

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

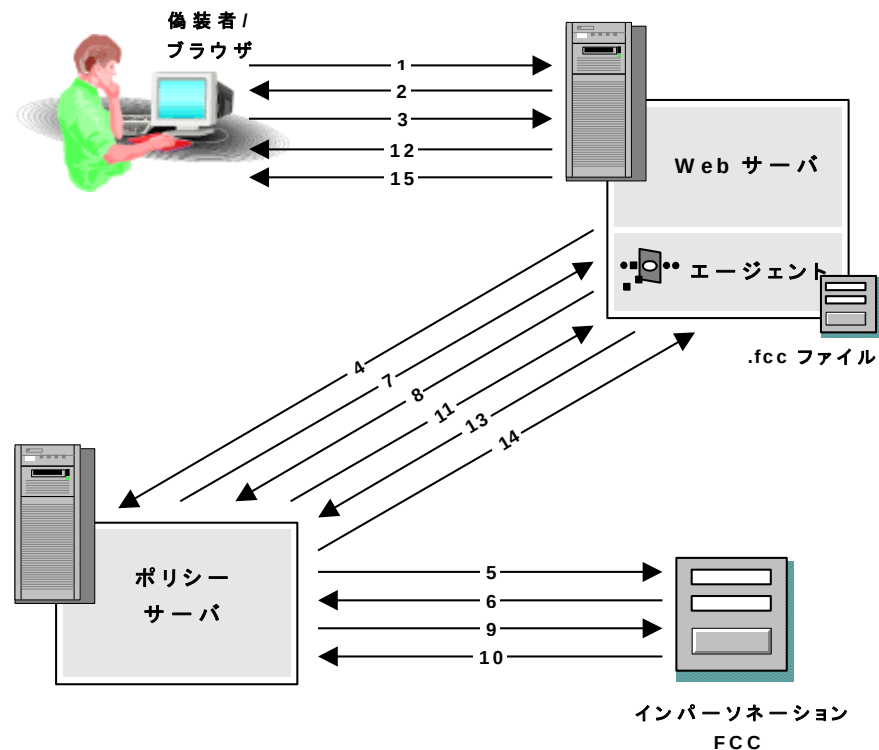
セッション仕様はセッション チケットまたはセッション スペックとも呼ばれます。セッション仕様は、ユーザと現在のセッションの特性を説明する情報であり、この情報は、ポリシー サーバ上に独自のフォーマットで格納されます。

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。

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لم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。

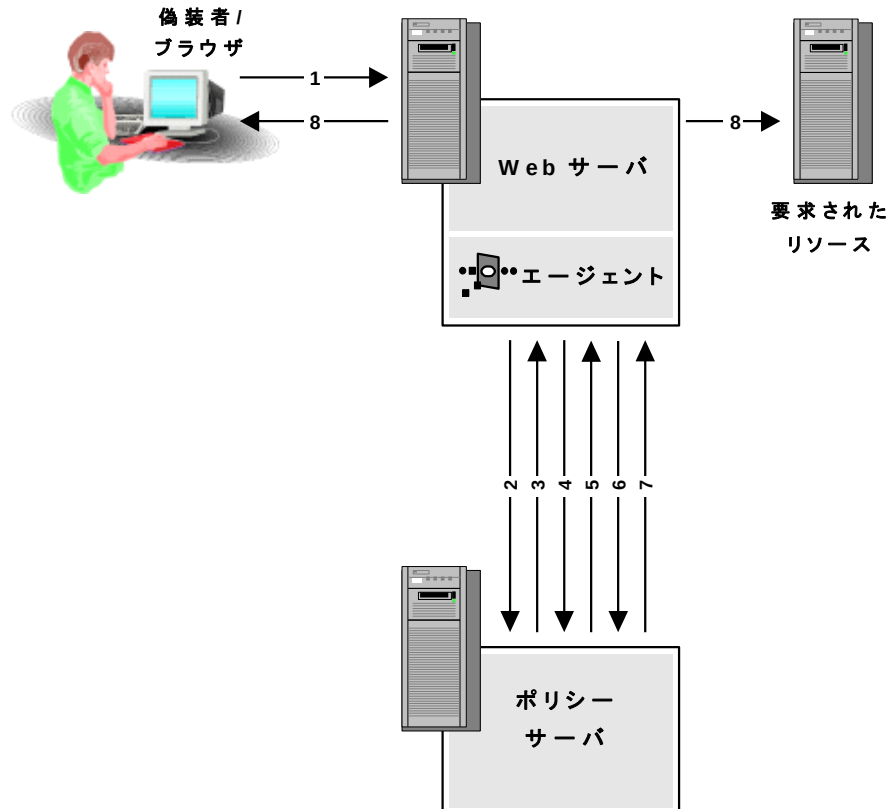
注: この段階では、偽装者がリソースにアクセスしていないため、インパーソネーションは実行されていません。

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。

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

偽装者のセッション仕様は、すべての顧客のセッション仕様と同様に処理されます。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。追加のフィールドにより、以下が可能になります。

- 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になります。
- 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、監査することができます。

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した場合、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。このアクションにより、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

偽装者が最初に認証された保護レベルは、インパーソネーション中に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ている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とされます。ただし、偽装者は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とされることはありません。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中に偽装しているユーザに固有の認証情報はありません。その結果、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、通常のように保護レベルの確認が適用されます。

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中には、偽装者の本来のセッションがアイドル状態になることがあります。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。偽装者が本来のアイドルタイムアウト値を超えている場合、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。

インパーソネーションレルムおよびイベント

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設定されたレルムで開始されます。偽装者がインパーソネーションレルム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と求められます。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、パスワードを偽装者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 サーバは偽装認証方式を使用して、偽装者の現在のセッションが有効であることを確認します。次に、ポリシー サーバは、インパーソネーションレルムに関連付けられたポリシー ドメインに含まれるディレクトリで、偽装対象のユーザを検索しようと試みます。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場合、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します。これらの権限は、ポリシー、ルール、および 2 つのインパーソネーションイベントを使用して設定および適用されます。

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、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ルムで許可または拒否することができます。レルムは、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の設定

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 **Web** エージェントをインストールする必要があります。

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

インパーソネーションには、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** つのレルムには偽装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、リソースを保護するポリシーに加えて、インパーソネーション レルムのリソースへのアクセスを許可する追加のポリシーが必要です。これらのポリシーは、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。

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ます、**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。

詳細情報:

[偽装認証方式](#) (P. 467)

.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、Web エージェントと共にインストールされます。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。

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 **SMSERVERSESSIONSPEC**（または **SM_SERVERSESSIONSPEC**、「**SMSERVERSESSIONSPEC** 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 [lpzCertBinary] 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 ディレクティブ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ます。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は、以下が必要です。

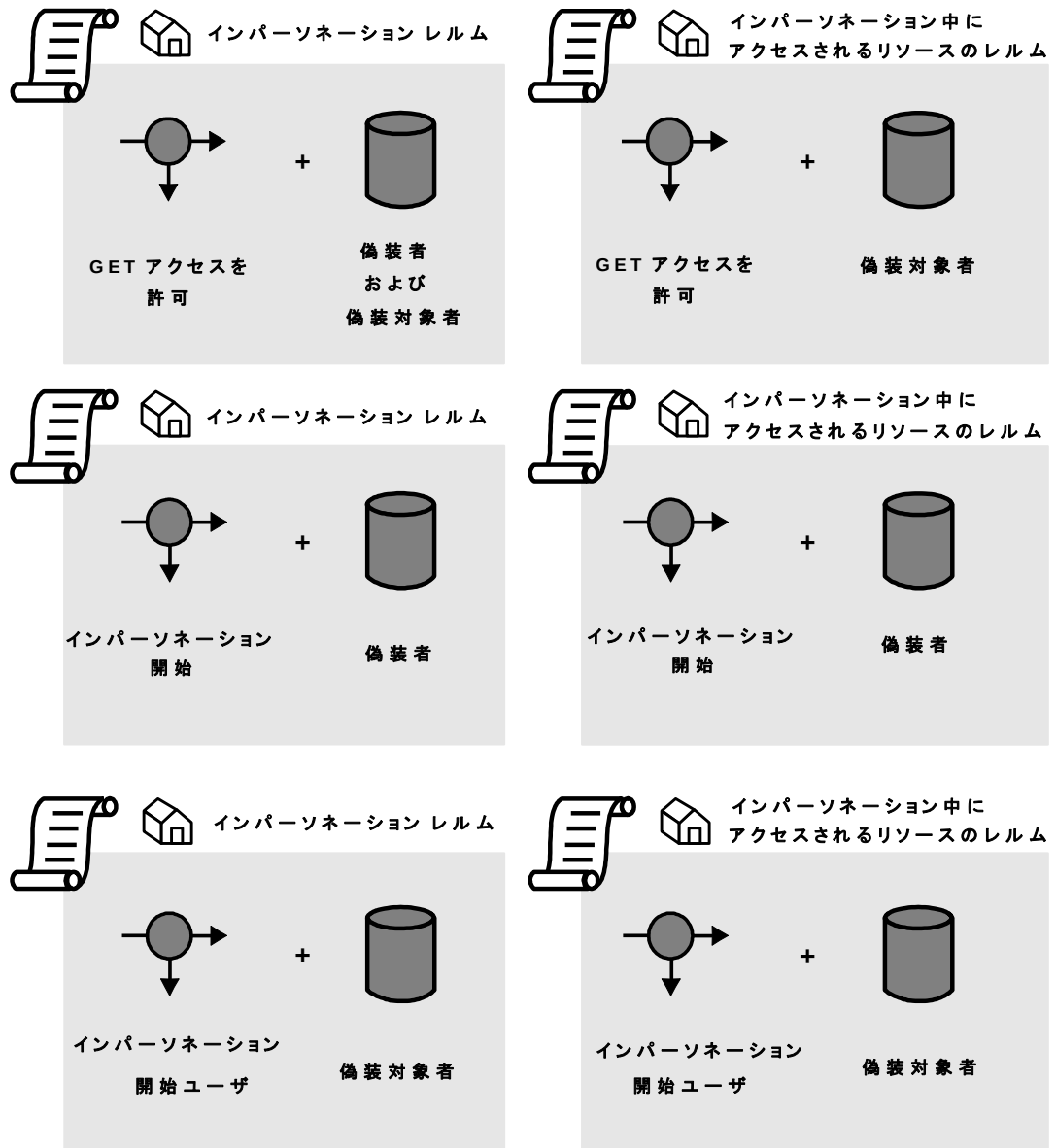
- アクセス制御ポリシー。偽装者と偽装対象の両方について設定します。たとえば、偽装者と偽装対象の両方について、**GET** アクセスルールが含まれるポリシーを設定します。
- **ImpersonationStart** ルールと偽装者を含むインパーソネーションポリシー。
- **ImpersonationStartUser** ルールと偽装対象を含むインパーソネーションポリシー。

偽装者がリソースにアクセスするためには、以下が必要です。

- 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: 偽装者がアクセスできる各リソースには、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偽装者に必要な最低限のポリシーを示します。



詳細情報:

[ポリシー](#) (P. 667)

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（おそらくは **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

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、偽証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、保護されたネットワークにログインします。
2. 管理者は認証情報を入力し、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、監査ログ機能を使用して追跡管理できます。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プロセスを開始するためのフォーム

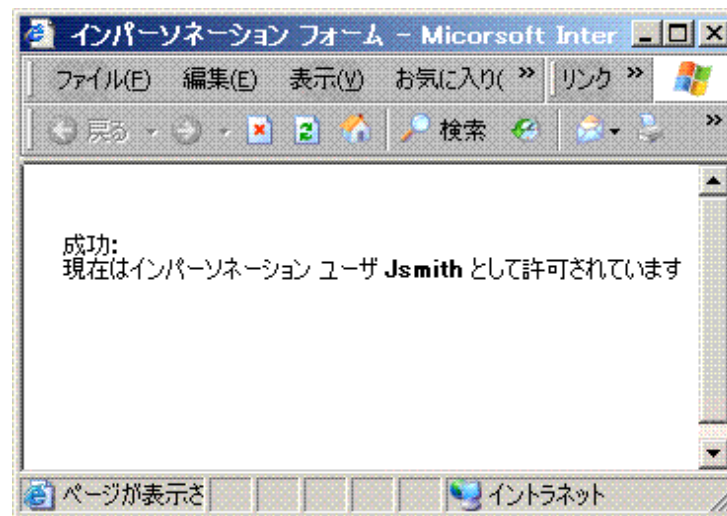
インパーソネーションにはいくつかの **Web** ベースのフォームが必要になります。これらのフォームは、それぞれの環境で使用するインタフェースに合わせてカスタマイズする必要があります。フォームには、インパーソネーション処理を呼び出す **.fcc** ファイルで表示する必要がある基本要素が含まれています。

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 フォーム - Micorsoft Inter

ファイル(E) 編集(E) 表示(V) お気に入り(>>) リンク(>>)

戻る 進む 印刷 検索

エラー:
JSmith はインパーソネーション ユーザとして許可されていません

インパーソネーション ユーザ名

サブミット

ページが表示さ

イントラネット

インパーソネーション フォーム - Micorsoft Inter

ファイル(E) 編集(E) 表示(V) お気に入り(>>) リンク(>>)

戻る 進む 印刷 検索

エラー:
JSmith はインパーソネーション ユーザとして許可されていません

インパーソネーション ユーザ名

ヒント

インパーソネーションを許可

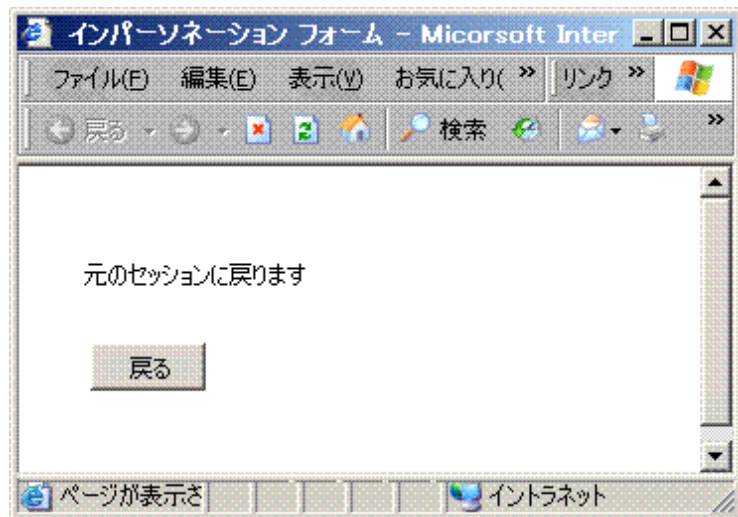
ページが表示さ

イントラネット

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 22 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 \(P. 761\)](#)

[パスワード ポリシーを設定する方法 \(P. 762\)](#)

[CA Directory パスワード ポリシー コントロール \(P. 773\)](#)

[ユーザが開始するパスワード変更 \(P. 774\)](#)

[パスワード ポリシーのトラブルシューティング \(P. 776\)](#)

パスワード サービスの概要

パスワード サービスは、保護されているリソースにセキュリティ レイヤを追加します。パスワード サービスは、以下のユーザ パスワードを管理するためにフォーム認証情報コレクタ (FCC) およびカスタマイズ可能な HTML フォームを使用します。

- ユーザ ディレクトリ
- ユーザ ディレクトリの一部

注: パスワード サービスがサポートするストアの詳細については、「[12.52 SP1 SiteMinder Platform Support Matrix](#)」を参照してください。エージェントとパスワード 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。

ユーザ パスワードを管理するためにパスワード ポリシーを使用します。オプションで、ユーザが主導するパスワード変更を有効にします。

- パスワード ポリシーは、以下を管理するルールおよび制限を定義します。
 - パスワードの有効期限
 - 構成
 - 使用方法

注: ユーザ ディレクトリ一部または全部に複数のパスワード ポリシーを適用できます。SiteMinder は、それぞれに指定する優先度に従ってポリシーを適用します。

- ユーザが保護されているリソースへのアクセスを試みたときに、**SiteMinder** はパスワードポリシーを呼び出して、ユーザのクレデンシャルを評価します。ポリシーによりパスワードが期限切れであると判定された場合、**SiteMinder** は次のことができます。
 - 不正アクセスを防ぐためにユーザアカウントを無効にします。無効な場合、**SiteMinder** 管理者は再度アカウントをアクティブにする必要があります。
 - パスワードを変更することをユーザに強制します。

パスワードポリシーを使用して、管理者の介入なしにパスワード構成ルールを適用できます。パスワードサービスは以下のことができます。

- 指定したときにパスワードを変更することをユーザに強制します。
- ユーザが主導するパスワード変更を有効にします。

パスワードポリシーを設定する方法

パスワードポリシーを設定して、保護されているリソースにセキュリティレイヤを追加します。

パスワードポリシーを設定するには、以下の手順を完了します。

1. [パスワードポリシーを作成します](#) (P. 764)。
2. (オプション) パスワードの有効期限を設定します。
3. (オプション) パスワードの構成を設定します。
4. (オプション) パスワードの正規表現を設定します。
5. (オプション) パスワードの制限を設定します。
6. (オプション) 詳細なパスワードオプションを設定します。

パスワードポリシーの考慮事項

企業内にパスワードポリシーを実装する予定の場合は、以下を考慮します。

- **SiteMinder** には、パスワードおよびパスワード関連情報を格納するディレクトリ内にあるいくつかの属性の独占的使用を含む、ユーザディレクトリへの読み取り/書き込みアクセス権限が必要です。
- パスワードポリシーは、パスワードを検証するために、ユーザディレクトリの追加検索が必要になるため、**SiteMinder** のパフォーマンスに影響を及ぼす可能性があります。ディレクトリ全体ではなく、ユーザディレクトリの一部のみを検索するように設定されたパスワードポリシーも、パフォーマンスに影響を及ぼす可能性があります。
- ユーザディレクトリにネイティブパスワードポリシーがある場合、このポリシーは以下の条件を満たす必要があります。
 - **SiteMinder** パスワードポリシーよりも制限が少ないか、または
 - 無効

上記を満たさない場合、ネイティブパスワードポリシーは、**SiteMinder** に通知することなくパスワードを許可または拒否します。その結果、**SiteMinder** はこれらのパスワードを管理できなくなります。

- デフォルトでは、パスワードを変更するときにユーザが間違った情報を入力した場合、**SiteMinder** は一般的な失敗メッセージを返します。このメッセージは失敗理由を明示しません。デフォルト動作を変更し、パスワードの変更に失敗した理由をユーザに明示的に伝えるには、**DisallowForceLogin** レジストリキーを作成し、有効にします。
- 複数のポリシーサーバ上でパスワードポリシーを使用する場合は、すべてのサーバのシステム時刻を同期します。時刻を同期することにより、アカウントやパスワードの強制変更が途中で無効になるのを防ぐことができます。

注: 詳細については、「**SiteMinder** ポリシーサーバ設定ガイド」を参照してください。

パスワード ポリシーの作成

パスワード ポリシーを作成して、保護されたリソースにセキュリティ レイヤを追加することができます。

以下の手順に従います。

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] をクリックします。
3. [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。

4. ポリシー名を入力します。
5. [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。
6. 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。

7. (オプション) 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] をクリックして適用する部分を指定します。
8. (オプション) [リダイレクト URL] フィールドでパスワードポリシーで無効と考えられるパスワードを入力した場合に、ユーザがリダイレクトされる FCC の場所を指定します。

エージェントと同じサーバ上のリダイレクト FCC をホストするには、デフォルト設定をそのまま使用します。

```
/siteminderagent/forms/smpwservices.fcc
```

特定のサーバ上のすべてのホスト用にリダイレクト FCC をホストする方法

```
http://server_name:port/siteminderagent/forms/smpwservices.fcc
```

SSL で特定のサーバ上のすべてのホスト用にリダイレクト FCC をホストする方法

```
https://server_name:port/siteminderagent/forms/smpwservices.fcc
```

9. 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。

注: 完全修飾ドメイン名 (FQDN) を使用して特定のサーバにエージェントをリダイレクトする設定、およびリダイレクト FCC のカスタマイズまたはローカライズの詳細については、「**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。
5. [サブミット] をクリックします。

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
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
- パスワードの以前使用したパスワードとの必要な相違の程度

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 ボックスで、新しいパスワードの以前のパスワードとの違いの程度を指定します。
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。

パスワードの詳細オプションの設定

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。

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、ユーザが入力したログイン ID（通常はユーザ名）がリクエスト URL に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¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥PolicyServer¥DisallowUsernameInURL`
2. DWORD の値を以下のいずれかの値に設定します。
 - 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¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥PolicyServer¥DisallowUsernameInURL`
4. DWORD の値を以下のいずれかの値に設定します。
 - 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。

CA Directory パスワード ポリシー コントロール

CA Directory パスワード ポリシーを適用するように、ポリシー サーバを設定できます。ポリシー サーバは、正しく設定された Web エージェントと一緒に、ディレクトリ パスワード ポリシーに基づく、設定された警告および通知をエンド ユーザに送信できます。

以下の CA Directory パスワード ポリシーがサポートされています。

- ネイティブ パスワードの有効期限。
- この機能を使用するには、以下のいずれかのオプションを実行します。
 - パスワードの有効期限に対してポリシー サーバのパスワード ポリシーを設定しない。
 - CA ディレクトリ パスワード ポリシー内の設定よりも制限が緩和されるように、有効期限および警告用のポリシー設定を設定します。
- ユーザ アカウントのロックアウト設定。

CA Directory 設定を使用する場合は、ポリシー サーバアカウントのロックアウト機能を無効にします。

- パスワード有効期限の警告。
- クライアントが猶予ログイン メッセージ用のパスワード メッセージコードを認識できる場合は、残りの猶予ログインの通知。

ポリシー サーバには、残りの猶予ログインという概念がありません。

CA Directory パスワード ポリシーの使用を許可する方法

1. ユーザが存在する CA Directory を参照する SiteMinder パスワード ポリシーを定義します。
2. XPSConfig ツールを使用し、設定パラメータ **CA.SM::\$LdapEnablePwdCtrlSupport** を true に設定します。
ツールを使用するには、XPSConfig 手順を参照してください。

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。

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。

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。

パスワード変更失敗メッセージの有効化

デフォルトでは、パスワードを変更するときにユーザが間違っただけの場合、**SiteMinder** は一般的な失敗メッセージを返します。このメッセージは失敗理由を明示しません。

デフォルト動作を変更して、変更失敗した理由をユーザに明示的に伝えることができます。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにアクセスし、以下のいずれかを実行します。
 - a. (Windows) レジストリ エディタを開き、`HKEY_LOCAL_MACHINE\Software\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\PolicyServer` に移動します。
 - b. (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. 以下の設定を使用して、`DisallowForceLogin` を作成します。

キータイプ: `REG_DWORD`

値: 0 または 1

0

(デフォルト) **SiteMinder** は一般的な失敗メッセージを返します。この動作はデフォルトの **SiteMinder** 動作と同じです。

1

SiteMinder は失敗した理由を明示的に返します。

注: 1 または 0 以外の値はサポートされていません。

3. 以下のいずれかを実行します。
 - (Windows) レジストリ エディタを終了します。
 - (UNIX) レジストリ 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。

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

問題の状況:

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

問題の状況:

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過し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

問題の状況:

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

問題の状況:

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

問題の状況:

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する。

- 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、および無効フラグ ユーザプロファイル属性の設定の詳細については、「[ディレクトリ属性の概要](#) (P. 160)」を参照してください。

Active Directory ユーザがパスワードを変更できない

問題の状況:

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する。

- 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、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に設定されている。

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

問題の状況:

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\Wow6432Node\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。

キータイプ： REG_DWORD

値： 0（無効） または 1（有効）

デフォルト： 0（無効）

注： レジストリ キーが有効な場合、0 または 1 以外の値はサポートされず、未定義の動作を起こします。

第 23 章: DeviceDNA™ による拡張セッション保証

このセクションには、以下のトピックが含まれています。

[DeviceDNA™ による拡張セッション保証を設定する方法 \(P. 781\)](#)

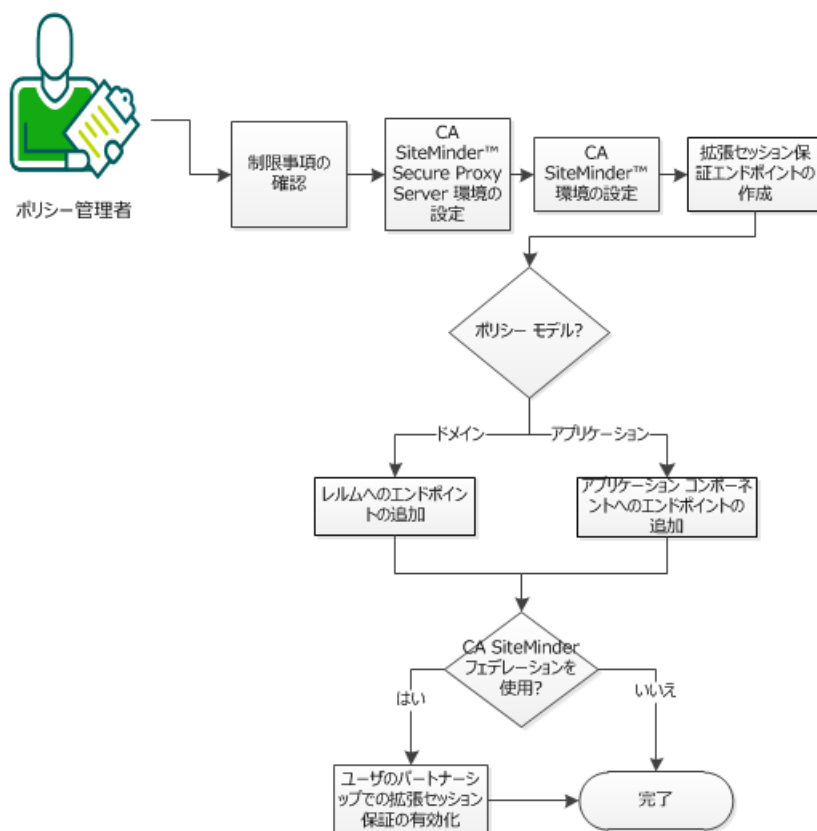
[アップグレード中に DeviceDNA™ による拡張セッション保証を設定する方法 \(P. 794\)](#)

DeviceDNA™ による拡張セッション保証を設定する方法

DeviceDNA™ による拡張セッション保証は、許可されていないユーザーが盗んだ Cookie で正当なセッションをハイジャックするのを防ぐのに役立ちます。セッションクライアントは、製品がユーザーのシステムから収集する一意の DeviceDNA™ を使用して認証されます。この検証によって、セッションを開始したクライアントがアクセスをリクエストしているのと同じクライアントであることが保証されます。有効な DeviceDNA™ のないユーザーは、保護されているリソースへのアクセスを拒否されます。

以下の図では、DeviceDNA™ による拡張セッション保証を設定する方法について説明しています。

DeviceDNA™ による拡張セッション保証を設定する方法



以下の手順に従います。

1. [制限を確認します](#) (P. 784)。
2. [SiteMinder 環境を設定します](#) (P. 786)。
3. [SiteMinder 環境を設定します](#) (P. 787)。
4. [セッション保証エンドポイントを作成します](#) (P. 788)。
5. ポリシー モデルに該当する適切な手順を以下から選択します。
 - ポリシー ドメインの場合、[エンドポイントをレルムに追加します](#) (P. 790)。
 - アプリケーション (EPM) の場合、[エンドポイントをアプリケーション コンポーネントに追加します](#) (P. 791)。
6. CA SiteMinder® Federation に対して、[パートナーシップにおいて拡張セッション保証を有効にします](#) (P. 792)。

DeviceDNA™ による拡張セッション保証の制限

DeviceDNA™ による拡張セッション保証では、以下のアイテムはサポートされません。

Web 2.0 クライアント

Web 2.0 アプリケーションは、CA SiteMinder® for Secure Proxy Server にリ-ダイレクトできない Web リクエストを作成する AJAX のようなテクノロジーをベースに作成されています。Web 2.0 クライアントには、非-ブラウザ-ベースのクライアント（モバイルデバイスにおける Flickr クライアント）が含まれています。両方のケースで、認証フロー アプリケーションをホストする CA SiteMinder® for Secure Proxy Server インスタンスにリ-ダイレクトできないリクエストがいくつか発生する可能性があります。こうした状況では、DeviceDNA™ による拡張セッション保証は Web 2.0 クライアントをサポートできません。Web 2.0 アプリケーションのログインページは保護できます。ただし、すべてのリクエスト（AJAX に関するリクエストなど）を DeviceDNA™ による拡張セッション保証で保護することはできません。

カスタム エージェント

SiteMinder SDK で作成されるエージェントは、DeviceDNA™ による拡張セッション保証をサポートしません。

JavaScript および Cookie をサポートしないクライアント

CA SiteMinder® for Secure Proxy Server 上の DeviceDNA™ スクリプトは、Web クライアントに固有の情報を抽出する Javascript です。このクライアント情報は、セッションをデバイスまたはクライアントに関連付けます。JavaScript のサポートなしでは、DeviceDNA™ は収集できません。したがって、DeviceDNA™ による拡張セッション保証はセッションをデバイスまたはクライアントに関連付けることはできません。

DeviceDNA™ による拡張セッション保証は Telnet や Lynx などのクライアントではサポートされていません。

POST 保存

DeviceDNA™ による拡張セッション保証は、CA SiteMinder® for Secure Proxy Server 上の認証フロー アプリケーションへのリ-ダイレクトを使用して、動作します。このアプリケーションには、DeviceDNA™ スクリプトが含まれています。認証フロー アプリケーションは、現在 POST データをサポートしていません。そのため、POST リクエストは認証フロー アプリケーションにリダイレクトされません。

共有ワークステーション

どの共有ワークステーションにも、すべてのユーザで同じ DeviceDNA™ 署名があります。たとえば、1 人のユーザが共有ワークステーションの別のユーザの有効な **SMSESSION Cookie** をハイジャックすると仮定します。ハイジャッカーが盗んだ **SMSESSION Cookie** を同じ共有ワークステーションから再生する場合、本製品は違いを検出できません。DeviceDNA™ による拡張セッション保証により保護が行われるのは、ハイジャッカーが盗んだ **SMSESSION Cookie** を別のデバイスから実行する場合です。

認証および許可 Web サービス

Web サービス クライアント アプリケーションは認証および認可 Web サービス（エージェント API コールから受信するあらゆるリダイレクトまたはレスポンスを元に返す）を処理します。ただし、呼び出し元のクライアントは DeviceDNA™ フローによる拡張セッション保証に関わるリダイレクトを処理できません。

CA SiteMinder® Federation

以下の設定は、DeviceDNA™ による拡張セッション保証をサポートしません。

- SAML 2.0 パートナリーシップの SP 側。
- パートナリーシップの IdP 側の SAML 2.0 HTTP-POST バインディング。

エージェント設定パラメータ

以下のエージェント設定パラメータは、DeviceDNA™ による拡張セッション保証をサポートしません。

- SecureURL
- TargetAsRelativeURI

注: DeviceDNA™ による拡張セッション保証を使用する場合は、これらのパラメータを設定しないでください。これらのパラメータに通常含まれるターゲット URL は、拡張セッション保証が DeviceDNA™ で使用するトークン リクエストにすでに含まれています。

レスポンス属性

以下のレスポンス属性は、DeviceDNA™ による拡張セッション保証をサポートしません。

- OnAuthAccept

DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定

DeviceDNA™ による拡張セッション保証では、CA SiteMinder® for Secure Proxy Server が動作している必要があります。

1. 以下のタスクのいずれかを実行します。

- CA SiteMinder® for Secure Proxy Server がない場合は、バージョン 12.52 SP1 をインストールします。
- CA SiteMinder® for Secure Proxy Server がある場合は、12.52 SP1 にアップグレードします。

注: CA SiteMinder® for Secure Proxy Server のインストールおよびアップグレードの詳細については、「[CA SiteMinder® for Secure Proxy Server 管理ガイド](#)」を参照してください。[インストールビデオ](#)もオンラインで提供されています。

2. SSL 接続を使用するように CA SiteMinder® for Secure Proxy Server を設定します。
3. CA SiteMinder® for Secure Proxy Server とポリシー サーバの間でポート番号 7658 を開きます。
4. CA SiteMinder® for Secure Proxy Server のインストールまたはアップグレードからの Advanced Authentication Server 暗号化キーを記録します。ユーザの環境内の SiteMinder ポリシー サーバは、後でこのキーが必要になります。
5. 複数の Cookie ドメインを使用するシングルサインオン環境については、して、Cookie プロバイダ ドメインの完全修飾ドメイン名 (FQDN) が必要です。設定ウィザードを実行する場合、この FQDN を ServerName 設定に使用します。たとえば、ユーザの Cookie ドメインが sso.example.com である場合、CA SiteMinder® for Secure Proxy Server 設定ウィザードで ServerName の値を sso.example.com に設定します。
6. CA SiteMinder® for Secure Proxy Server の [IgnoreExt] 設定パラメータに以下の値を追加します。

.sac

7. 以下の手順で、JSafeJCE セキュリティ プロバイダで JVM を設定して、暗号化を有効にします。
 - a. CA SiteMinder® for Secure Proxy Server コンピュータにログオンします。
 - b. お使いの Java バージョンの Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files パッケージを、Oracle Web サイトからダウンロードします。
 - c. 以下の場所に移動します。

Windows

`JVM_HOME\lib\security`

UNIX

`JVM_HOME/lib/security`

`JVM_HOME`

インストールの JDK に Java Runtime Environment (JRE) がインストールされる場所を定義します。

- d. JCE Unlimited Strength Jurisdiction Policy Files パッケージからファイルで、以下のファイルにパッチを適用します。
 - `local_policy.jar`
 - `US_export_policy.jar`

これで JVM の設定が完了しました。

DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定

DeviceDNA™ による拡張セッション保証は、SiteMinder 環境における特定のコンポーネントおよび設定を必要とします。

以下の手順に従います。

1. ポリシー サーバをインストールするかバージョン **12.52 SP1** にアップグレードします。

注: 旧バージョンからアップグレードした場合は、環境内の各ポリシー サーバでポリシー サーバ設定ウィザードを実行してください。

2. セッション ストアがない場合は、1 つ設定します。
3. ポリシー サーバと CA SiteMinder® for Secure Proxy Server の間でポート番号 **7658** を開きます。

DeviceDNA™ エンドポイントによる拡張セッション保証の作成

DeviceDNA™ による拡張セッションは、ユーザを DeviceDNA™ 情報を収集するために CA SiteMinder® for Secure Proxy Server 上でホストされている認証フローアプリケーションエンドポイントにリダイレクトします。この DeviceDNA™ 情報によって、それらのセッションが検証されます。

パフォーマンス上の理由で、組織内の各地域別区分につき 1 つのエンドポイントを作成することを推奨します。たとえば、ニューヨークおよびシカゴのオフィスがある場合は、各オフィスのエンドポイントを作成します。

DeviceDNA™ による拡張セッション保証をポリシーまたはアプリケーションに追加する前に、CA SiteMinder® for Secure Proxy Server 上でこれらのエンドポイントを設定します。

以下の手順に従います。

1. 管理 UI から、[ポリシー] - [グローバル] - [セッション保証エンドポイント] をクリックします。
2. [STS エンドポイントの作成] をクリックします。
3. わかりやすい名前と説明（オプション）を入力します。

4. 以下のフィールドに値を入力します。

Web サーバ名

ユーザを認証するために DeviceDNA™ を収集する CA SiteMinder® for Secure Proxy Server サーバの名前を指定します。

ポート

CA SiteMinder® for Secure Proxy Server がリダイレクト用にリスニングしているポート番号を指定します。（SSL を使用して）このポートを安全な接続に設定します。

デフォルト：443

ターゲット

ユーザが暗黙的にリダイレクトされる CA SiteMinder® for Secure Proxy Server の URL を指定します。このサーバは、ユーザの DeviceDNA™ を収集します。DeviceDNA™ を使用して、ユーザと関連付けられたセッションが検証されます。

DeviceDNA™ リフレッシュ間隔

ユーザと関連付けられた DeviceDNA™ が有効になる秒数を指定します。有効な DeviceDNA™ のないユーザは、サーバがユーザの現在の DeviceDNA™ を取得する拡張セッション保証エンドポイントにリダイレクトされます。

DeviceDNA™ リフレッシュ間隔は、DeviceDNA™ の収集を管理します。DeviceDNA™ リフレッシュ間隔の期限が切れた後に発生するリクエストは、DeviceDNA™ を再収集するために Authentication Flow アプリケーションにリダイレクトされます。

DeviceBinder は、このセッションに関連付けられたユーザを識別するセッションプロパティです。DeviceBinder とクライアント側のデバイス ID は認証処理中にリンクされています。このプロパティは、一意の DeviceHash と有効期限から構成されます。

デフォルト：300 秒（5 分）。

5. [サブミット] をクリックします。

レルムへのエンドポイントの追加

DeviceDNA™ による拡張セッション保証を使用してレルムのリソースを保護するには、1つのセッション保証エンドポイントをレルムに追加します。

注: DeviceDNA™ による拡張セッション保証が機能するには、レルムが永続的である必要はありません。

以下の手順に従います。

1. 管理 UI から、[ポリシー]、[ドメイン]、[レルム] をクリックします。
2. 目的のレルムの編集アイコンをクリックします。
3. [セッション] の下で、[拡張されたセッション保証] の横の [有効] チェックボックスをオンにします。
4. [エンドポイントの検索] をクリックします。
5. 目的のエンドポイントを選択します。
6. [OK] をクリックします。
7. [サブミット] をクリックします。
8. セッション保証機能で保護するリソースを持ったレルムが他にもあれば、手順 2 ～ 6 を繰り返します。

アプリケーション コンポーネントへのエンドポイントの追加

DeviceDNA™ による拡張セッション保証を使用して、アプリケーション内のリソースを保護するには、1つの拡張セッション保証エンドポイントに関連するアプリケーションのコンポーネントに追加します。

以下の手順に従います。

1. 管理 UI から、[ポリシー]、[アプリケーション]、[アプリケーション] の順にクリックします。
アプリケーションのリストが表示されます。
2. 目的のアプリケーションの編集アイコンをクリックします。
[アプリケーションの変更] ダイアログ ボックスが表示されます。
3. 以下のいずれかを実行します。
 - アプリケーションに 1 つのみコンポーネントがある場合、[詳細設定] をクリックします。
 - アプリケーションに複数のコンポーネントがある場合は、目的のコンポーネントの編集アイコンをクリックします。 [詳細設定] を選択します。
4. [セッション] の下で、[拡張されたセッション保証] の横の [有効] チェックボックスをオンにします。
5. [エンドポイントの検索] をクリックします。
エンドポイントのリストが表示されます。
6. 目的のエンドポイントを選択します。
7. [OK] をクリックします。
8. [サブミット] をクリックします。
[アプリケーションの変更] ダイアログ ボックスが閉じ、確認メッセージが表示されます。
9. セッション保証機能で保護するリソースを持ったアプリケーションが他にあれば、手順 2 ～ 7 を繰り返します。

パートナーシップにおける DeviceDNA™ による拡張セッション保証の有効化

CA SiteMinder® Federation を使用する場合、以下のパートナーシップにおいて、DeviceDNA™ による拡張セッション保証を有効することもできます。

- SP から IdP へのパートナーシップの IdP 側 (HTTP リダイレクトバインディングのみ)。
- コンシューマからプロデューサへのパートナーシップのプロデューサ側
- RP から AP へのパートナーシップの AP 側

以下の手順に従います。

1. 管理 UI から、[フェデレーション] - [パートナーシップ フェデレーション] - [パートナーシップ] をクリックします。
2. 目的のパートナーシップの左側の[アクション]ボタンをクリックし、[非アクティブ化]を選択します。
3. 同じ[アクション]ボタンを再度クリックし、[変更]を選択します。
4. [SSO と SLO] タブをクリックします。
5. 以下のチェックボックスをオンにします。

セッション保証の有効化

(ポリシー ドメイン モデルの) レルムまたは (アプリケーション モデルの) コンポーネントで指定されたリソースを保護します。また、特定の CA SiteMinder® Federation パートナーシップの認証リクエストを保護できます。セッション保証エンドポイントはユーザから DeviceDNA™ を収集し、セッションを検証します。

制限： この機能にはセッション保証エンドポイントが必要です。

エンドポイントのリストが表示されます。

6. 目的のエンドポイントのチェックボックスをオンにします。
7. [保存] をクリックします。
8. 目的のパートナーシップがほかにあれば、手順 2 ～ 7 を繰り返します。
9. (ローカル認証モードのみに対して) 認証 URL (Redirect.jsp) に関連付けられるレルム上で DeviceDNA™ による拡張セッション保証を有効にします。

SM--関連するログ ファイル

DeviceDNA™ による拡張セッション保証に関するトランザクションは、以下のログ ファイルに記録されます。

ポリシー サーバ

xps-*.audit -- 機能の設定への変更。

smaccesslog4 -- 機能に関連する認証アクティビティおよび認可アクティビティ。

Advanced Authentication Server の起動ログ

caauthminderstartup

cariskminderstartup

cariskmindercasemgmtserverstartup

Advanced Authentication Server のランタイム ログ

caauthminder

cariskminder

cariskmindercasemgmtserver

CA SiteMinder® for Secure Proxy Server サーバ (Advanced Authentication フロー アプリケーション)

arcotuds.log -- UDS サービスのログ ファイル

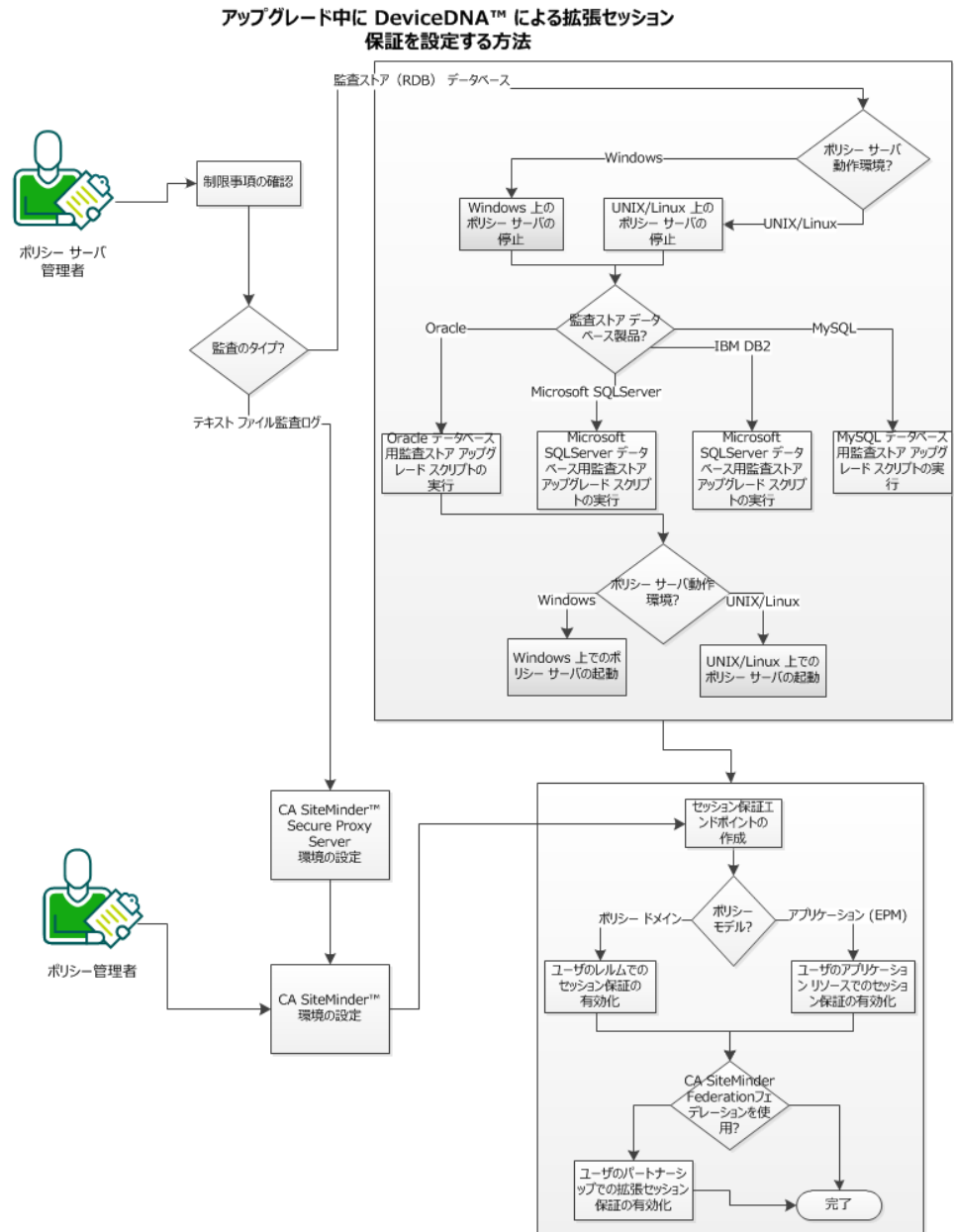
CAWebFlowLog.txt -- 認証フロー アプリケーションのログ ファイル

UIApplog.txt -- UIApp (認証フロー アプリケーションの実行プロセスの一部) のログ

アップグレード中に DeviceDNA™ による拡張セッション保証を設定する方法

DeviceDNA™ による拡張セッション保証は、許可されていないユーザが盗んだ Cookie で正当なセッションをハイジャックするのを防ぐのに役立ちます。セッションクライアントは、製品がユーザのシステムから収集する一意の DeviceDNA™ を使用して認証されます。この検証によって、セッションを開始したクライアントがアクセスをリクエストしているのと同じクライアントであることが保証されます。有効な DeviceDNA™ のないユーザは、保護されているリソースへのアクセスを拒否されます。

以下の図は、アップグレード中に DeviceDNA™ による拡張セッション保証を設定する方法を示しています：



以下の手順に従います。

1. [制限を確認します](#) (P. 784)。
2. 以下のいずれかを実行します。
 - 監査ログをリレーショナル データベースに格納する場合は、手順 3 に移動します。
 - 監査ログをテキスト ファイルとして格納する場合は、手順 4 に移動します。
3. 以下の手順で監査ストア (データベース) をアップグレードします。
 - a. 以下のいずれかの手順を使用して、ポリシー サーバを停止します。
 - [Windows オペレーティング環境でポリシー サーバを停止します](#) (P. 800)。
 - [UNIX/Linux オペレーティング環境のポリシー サーバを停止します](#) (P. 801)。
 - b. 以下のいずれかの手順で、[データベース用の適切なアップグレードスクリプト](#) (P. 801)を実行します。
 - [Oracle データベースの監査-ストアのアップグレードスクリプトを実行します](#) (P. 802)。
 - [SQL Server データベースの監査ストアのアップグレードスクリプトを実行します](#) (P. 803)。 -
 - [IBM DB2 データベースの監査-ストアのアップグレードスクリプトを実行します](#) (P. 805)。
 - [MySQL データベースの監査ストアのアップグレードスクリプトを実行します](#) (P. 807)。 -
 - c. 以下のいずれかの手順を使用して、ポリシー サーバを開始します。
 - [Windows オペレーティング環境でポリシー サーバを起動します](#) (P. 808)。
 - [UNIX/Linux オペレーティング環境でポリシー サーバを起動します](#) (P. 809)。
4. [SiteMinder 環境を設定します](#) (P. 786)。
5. [SiteMinder 環境を設定します](#) (P. 787)。
6. [セッション保証エンドポイントを作成します](#) (P. 788)。
7. ポリシー モデルに該当する適切な手順を以下から選択します。

- ポリシー ドメインの場合、[エンドポイントをレルムに追加します](#) (P. 790)。
 - アプリケーション (EPM) の場合、[エンドポイントをアプリケーション コンポーネントに追加します](#) (P. 791)。
8. CA SiteMinder® Federation に対して、[パートナーシップにおいて拡張セッション保証を有効にします](#) (P. 792)。

DeviceDNA™ による拡張セッション保証の制限

DeviceDNA™ による拡張セッション保証では、以下のアイテムはサポートされません。

Web 2.0 クライアント

Web 2.0 アプリケーションは、CA SiteMinder® for Secure Proxy Server にリ-ダイレクトできない Web リクエストを作成する AJAX のようなテクノロジーをベースに作成されています。Web 2.0 クライアントには、非-ブラウザ-ベースのクライアント（モバイルデバイスにおける Flickr クライアント）が含まれています。両方のケースで、認証フロー アプリケーションをホストする CA SiteMinder® for Secure Proxy Server インスタンスにリ-ダイレクトできないリクエストがいくつか発生する可能性があります。こうした状況では、DeviceDNA™ による拡張セッション保証は Web 2.0 クライアントをサポートできません。Web 2.0 アプリケーションのログインページは保護できます。ただし、すべてのリクエスト（AJAX に関するリクエストなど）を DeviceDNA™ による拡張セッション保証で保護することはできません。

カスタム エージェント

SiteMinder SDK で作成されるエージェントは、DeviceDNA™ による拡張セッション保証をサポートしません。

JavaScript および Cookie をサポートしないクライアント

CA SiteMinder® for Secure Proxy Server 上の DeviceDNA™ スクリプトは、Web クライアントに固有の情報を抽出する Javascript です。このクライアント情報は、セッションをデバイスまたはクライアントに関連付けます。JavaScript のサポートなしでは、DeviceDNA™ は収集できません。したがって、DeviceDNA™ による拡張セッション保証はセッションをデバイスまたはクライアントに関連付けることはできません。

DeviceDNA™ による拡張セッション保証は Telnet や Lynx などのクライアントではサポートされていません。

POST 保存

DeviceDNA™ による拡張セッション保証は、CA SiteMinder® for Secure Proxy Server 上の認証フロー アプリケーションへのリ-ダイレクトを使用して、動作します。このアプリケーションには、DeviceDNA™ スクリプトが含まれています。認証フロー アプリケーションは、現在 POST データをサポートしていません。そのため、POST リクエストは認証フロー アプリケーションにリダイレクトされません。

共有ワークステーション

どの共有ワークステーションにも、すべてのユーザで同じ DeviceDNA™ 署名があります。たとえば、1 人のユーザが共有ワークステーションの別のユーザの有効な **SMSESSION Cookie** をハイジャックすると仮定します。ハイジャッカーが盗んだ **SMSESSION Cookie** を同じ共有ワークステーションから再生する場合、本製品は違いを検出できません。DeviceDNA™ による拡張セッション保証により保護が行われるのは、ハイジャッカーが盗んだ **SMSESSION Cookie** を別のデバイスから実行する場合です。

認証および許可 Web サービス

Web サービス クライアント アプリケーションは認証および認可 Web サービス（エージェント API コールから受信するあらゆるリダイレクトまたはレスポンスを元に返す）を処理します。ただし、呼び出し元のクライアントは DeviceDNA™ フローによる拡張セッション保証に関わるリダイレクトを処理できません。

CA SiteMinder® Federation

以下の設定は、DeviceDNA™ による拡張セッション保証をサポートしません。

- SAML 2.0 パートナリーシップの SP 側。
- パートナリーシップの IdP 側の SAML 2.0 HTTP-POST バインディング。

エージェント設定パラメータ

以下のエージェント設定パラメータは、DeviceDNA™ による拡張セッション保証をサポートしません。

- SecureURL
- TargetAsRelativeURI

注: DeviceDNA™ による拡張セッション保証を使用する場合は、これらのパラメータを設定しないでください。これらのパラメータに通常含まれるターゲット URL は、拡張セッション保証が DeviceDNA™ で使用するトークン リクエストにすでに含まれています。

レスポンス属性

以下のレスポンス属性は、DeviceDNA™ による拡張セッション保証をサポートしません。

- OnAuthAccept

Windows ポリシー サーバの停止

続行前にポリシー サーバを停止します。ポリシー サーバを停止すると、以下の結果になります。

- ポリシー サーバが、環境から一時的に削除されます。
- 許可または認証の判断が必要なエージェントは、停止されているポリシー サーバにコンタクトできません。それらのエージェントは、引き続き利用可能なほかのポリシー サーバに接続できます。
- すべてのログ記録アクティビティは停止します。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。

注: 管理者権限を持ったアカウントを使用します。

2. [スタート] - [プログラム] - [SiteMinder] - [SiteMinder ポリシー サーバ管理コンソール] をクリックします。
3. [停止] ボタンをクリックします。
4. [OK] をクリックします。

ポリシー サーバは停止し、コンソールは閉じます。

UNIX ポリシー サーバの停止

ポリシー サーバを停止すると、以下の結果になります。

- ポリシー サーバが、環境から一時的に削除されます。
- 許可または認証の判断が必要なエージェントは、停止されているポリシー サーバにコンタクトできません。それらのエージェントは、引き続き利用可能なほかのポリシー サーバに接続できます。
- すべてのログ記録アクティビティは停止します。

以下の手順に従います。

1. ポリシー サーバをホストしているシステムに、最初にポリシー サーバをインストールしたのと同じユーザ アカウントでログインします。
2. 以下のいずれかのアクションを実行して、すべてのポリシー サーバプロセスを停止します。
 - 管理コンソールを開き、[ステータス] タブをクリックし、[停止] ボタンをクリックします。
 - 以下のスクリプトを使用します。 バプロセスが自動的に再起動されないように、このスクリプトによって **UNIX** エグゼクティブも停止されます。

```
installation_path/siteminder/stop-all
```

ポリシー サーバは、**UNIX** エグゼクティブの全アクティビティのログを `installation_directory/log/smexec.log` ファイルに記録します。ログのエントリは、常に既存のログファイルに追加されます。

監査ストア アップグレード スクリプト

ポリシー サーバを停止した後、使用しているデータベース ベンダー用の適切な監査ストア アップグレードスクリプトを実行します。- 以下のリストからスクリプトを選択します。

- [Oracle データベース](#) (P. 802)。
- [Microsoft SQL Server データベース](#) (P. 803)。
- [IBM DB2 データベース](#) (P. 805)。
- [MySQL データベース](#) (P. 807)。

Oracle データベースの監査ストアのアップグレード スクリプトの実行

ポリシー サーバを停止した後、監査ログ スキーマをアップグレードします。このアップグレードは、スクリプトを使用して実行されます。

以下の手順に従います。

1. sqlplus または他の Oracle ユーティリティを使用して、ポリシー サーバ データベース情報を管理するユーザとして、Oracle データベースにログインします。

注: SYS または SYSTEM ユーザの場合は、CA SiteMinder スキーマをアップグレードしないことをお勧めします。必要な場合、SMOWNER などの Oracle データベース ユーザを作成し、そのユーザとしてスキーマをアップグレードします。

2. 12.5 からアップグレードする場合は、以下のスクリプトをインポートします。

```
$NETE_PS_ROOT/db/SQL/sm_oracle_logs_upgrade_12.5_to_12.51.sql
```

3. 以下のスクリプトをインポートします。

```
$NETE_PS_ROOT/db/SQL/sm_oracle_logs_upgrade_12.51_to_12.52.sql
```

注: 一部の環境変数は、Oracle が提供する SQL ユーティリティで機能しない場合があります。ユーティリティを使用して、スクリプトをインポートしようとする問題が発生する場合は、明示的なパスを指定します。

監査ストア スキーマがアップグレードされます。

4. 以下のいずれかの手順を使用して、ポリシー サーバを起動します。
 - [Windows オペレーティング環境でポリシー サーバを起動します](#) (P. 808)。
 - [UNIX/Linux オペレーティング環境でポリシー サーバを起動します](#) (P. 809)。

Microsoft SQL Server データベースの監査ストアのアップグレード スクリプトの実行

ポリシー サーバを停止した後、監査ログ スキーマをアップグレードします。このアップグレードは、スクリプトを使用して実行されます。

以下の手順に従います。

1. 以下のタスクのいずれかを実行します。
 - バージョン 12.5 からアップグレードする場合は、手順 2 に移動します。
 - バージョン 12.51 からアップグレードする場合は、手順 10 に移動します。
2. テキスト エディタで次のファイルを開きます。
`installation_directory\%db%\SQL\sm_mssql_logs_upgrade_12.5_to_12.51.sql`
3. `installation_directory`
4. ポリシー サーバがインストールされているファイル システム上の場所を指定します。
5. 全ファイルの内容をコピーします。
6. クエリ アナライザを起動し、ポリシー サーバ データベースを管理するユーザとしてログインします。
7. [データベース] リストからデータベース インスタンスを選択します。
8. スキーマを `sm_mssql_logs_upgrade_12.5_to_12.51.sql` ファイルからクエリへ貼り付けます。
9. クエリを実行します。
10. テキストエディタで次のファイルを開きます。
`installation_directory\%db%\SQL\sm_mssql_logs_upgrade_12.51_to_12.52.sql`
`installation_directory`
ポリシー サーバがインストールされているファイル システム上の場所を指定します。
11. 全ファイルの内容をコピーします。
12. クエリ アナライザを起動し、ポリシー サーバ データベースを管理するユーザとしてログインします。
13. [データベース] リストからデータベース インスタンスを選択します。
14. スキーマを `sm_mssql_logs_upgrade_12.51_to_12.52.sql` ファイルからクエリへ貼り付けます。

15. クエリを実行します。

監査ストア スキーマがアップグレードされます。

16. 以下のいずれかの手順を使用して、ポリシー サーバを起動します。

- [Windows オペレーティング環境でポリシー サーバを起動します \(P. 808\)](#)。
- [UNIX/Linux オペレーティング環境でポリシー サーバを起動します \(P. 809\)](#)。

IBM DB2 データベースの監査ストアのアップグレード スクリプトの実行

SiteMinder 監査ログ スキーマをアップグレードします。このアップグレードは、スクリプトを使用して実行されます。

以下の手順に従います。

1. ポリシー サーバをホストしているシステムにログインします。
2. 以下のディレクトリに移動します

```
installation_directory¥db¥tier2¥DB2¥
```

```
installation_directory
```

ポリシー サーバがインストールされているファイル システム上の場所を指定します。

3. 以下のタスクのいずれかを実行します。
 - バージョン 12.5 からアップグレードする場合は、手順 4 に移動します。
 - バージョン 12.51 からアップグレードする場合は、手順 6 に移動します。

4. 以下のファイルを開き、内容をテキスト エディタにコピーします。

```
sm_db2_log_upgrade_12.5_to_12.51.sql
```

5. クエリに内容を貼り付けて、クエリを実行します。

注: クエリ実行の詳細については、IBM のドキュメントを参照してください。

監査ストア スキーマがアップグレードされます。

6. 以下のファイルを開き、内容をテキスト エディタにコピーします。

```
sm_db2_log_upgrade_12.51_to_12.52.sql
```

7. クエリに内容を貼り付けて、クエリを実行します。

注: クエリ実行の詳細については、IBM のドキュメントを参照してください。

監査ストア スキーマがアップグレードされます。

8. 以下のいずれかの手順を使用して、ポリシー サーバを起動します。

- [Windows オペレーティング環境でポリシー サーバを起動します](#) (P. 808)。

- [UNIX/Linux オペレーティング環境でポリシー サーバを起動します \(P. 809\)](#)。

MySQL データベースの監査ストア アップグレード スクリプトの実行

SiteMinder 監査ログ スキーマをアップグレードします。このアップグレードは、スクリプトを使用して実行されます。

以下の手順に従います。

1. ポリシー サーバ ホスト システムにログインします。
2. 以下の場所に移動します。

`installation_directory¥db¥tier2¥MySQL。`

`installation_directory`

ポリシー サーバがインストールされているファイル システム上の場所を指定します。

3. 以下のタスクのいずれかを実行します。
 - バージョン 12.5 からアップグレードする場合は、手順 4 に移動します。
 - バージョン 12.51 からアップグレードする場合は、手順 9 に移動します。

4. テキスト エディタで次のファイルを開きます。

`sm_mysql_logs_upgrade_12.5_to_12.51.sql`

5. 以下の行を確認します。

```
DROP FUNCTION IF EXISTS `dbName`.`getdate` $$
CREATE FUNCTION `dbName`.`getdate` () RETURNS DATE
```

6. 'dbName' の各インスタンスを監査ストアとして機能するデータベースの名前に置換します。

7. 全ファイルの内容をコピーします。

8. クエリにファイルの内容を貼り付けて、クエリを実行します。

9. テキストエディタで次のファイルを開きます。

`sm_mysql_logs_upgrade_12.51_to_12.52.sql`

10. 以下の行を確認します。

```
DROP FUNCTION IF EXISTS `dbName`.`getdate` $$
CREATE FUNCTION `dbName`.`getdate` () RETURNS DATE
```

11. 'dbName' の各インスタンスを監査ストアとして機能するデータベースの名前に置換します。

12. 全ファイルの内容をコピーします。

13. クエリにファイルの内容を貼り付けて、クエリを実行します。
監査ストア スキーマがアップグレードされます。
14. 以下のいずれかの手順を使用して、ポリシー サーバを起動します。
 - [Windows オペレーティング環境でポリシー サーバを起動します \(P. 808\)](#)。
 - [UNIX/Linux オペレーティング環境でポリシー サーバを起動します \(P. 809\)](#)。

Windows ポリシー サーバの起動

ポリシー サーバを起動します。ポリシー サーバを起動すると、以下の結果になります。

- エージェントは、許可または認証のためにポリシー サーバに接続します。
- ログ記録が開始されます。

以下の手順に従います。

1. [スタート] - [すべてのプログラム] - [SiteMinder] - [SiteMinder ポリシー サーバ管理コンソール] をクリックします。
[ステータス] タブが選択された状態でコンソールが開きます。
2. [開始] ボタンをクリックします。
3. [OK] をクリックします。
ポリシー サーバが起動します。

UNIX ポリシー サーバの起動

ポリシー サーバを開始すると、以下のような結果になります。

- エージェントはポリシー サーバに許可または認証の決定を問い合わせます。
- ログ記録が開始されます。

以下のいずれかのアクションを実行して、すべてのポリシー サーバプロセスを開始します。

- 管理コンソールを開き、[ステータス] タブをクリックし、[開始] ボタンをクリックします。
- 以下のスクリプトを使用します。このスクリプトは、UNIX エグゼクティブも開始します。

```
installation_path/siteminder/start-all
```

ポリシー サーバは、UNIX エグゼクティブの全アクティビティのログを `installation_directory/log/smexec.log` ファイルに記録します。ログのエントリは、常に既存のログファイルに追加されます。

DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定

DeviceDNA™ による拡張セッション保証では、CA SiteMinder® for Secure Proxy Server が動作している必要があります。

1. 以下のタスクのいずれかを実行します。
 - CA SiteMinder® for Secure Proxy Server がいない場合は、バージョン 12.52 SP1 をインストールします。
 - CA SiteMinder® for Secure Proxy Server がある場合は、12.52 SP1 にアップグレードします。

注: CA SiteMinder® for Secure Proxy Server のインストールおよびアップグレードの詳細については、「[CA SiteMinder® for Secure Proxy Server 管理ガイド](#)」を参照してください。[インストールビデオ](#)もオンラインで提供されています。
2. SSL 接続を使用するように CA SiteMinder® for Secure Proxy Server を設定します。
3. CA SiteMinder® for Secure Proxy Server とポリシー サーバの間でポート番号 7658 を開きます。
4. CA SiteMinder® for Secure Proxy Server のインストールまたはアップグレードからの Advanced Authentication Server 暗号化キーを記録します。ユーザの環境内の SiteMinder ポリシー サーバは、後でこのキーが必要になります。
5. 複数の Cookie ドメインを使用するシングルサインオン環境については、して、Cookie プロバイダ ドメインの完全修飾ドメイン名 (FQDN) が必要です。設定ウィザードを実行する場合、この FQDN を ServerName 設定に使用します。たとえば、ユーザの Cookie ドメインが sso.example.com である場合、CA SiteMinder® for Secure Proxy Server 設定ウィザードで ServerName の値を sso.example.com に設定します。
6. CA SiteMinder® for Secure Proxy Server の [IgnoreExt] 設定パラメータに以下の値を追加します。
.sac

7. 以下の手順で、JSafeJCE セキュリティ プロバイダで JVM を設定して、暗号化を有効にします。
 - a. CA SiteMinder® for Secure Proxy Server コンピュータにログオンします。
 - b. お使いの Java バージョンの Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files パッケージを、Oracle Web サイトからダウンロードします。
 - c. 以下の場所に移動します。

Windows

`JVM_HOME\lib\security`

UNIX

`JVM_HOME/lib/security`

`JVM_HOME`

インストールの JDK に Java Runtime Environment (JRE) がインストールされる場所を定義します。

- d. JCE Unlimited Strength Jurisdiction Policy Files パッケージからファイルで、以下のファイルにパッチを適用します。
 - local_policy.jar
 - US_export_policy.jar

これで JVM の設定が完了しました。

DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定

DeviceDNA™ による拡張セッション保証は、SiteMinder 環境における特定のコンポーネントおよび設定を必要とします。

以下の手順に従います。

1. ポリシー サーバをインストールするかバージョン 12.52 SP1 にアップグレードします。

注: 旧バージョンからアップグレードした場合は、環境内の各ポリシー サーバでポリシー サーバ設定ウィザードを実行してください。

2. セッション ストアがない場合は、1 つ設定します。
3. ポリシー サーバと CA SiteMinder® for Secure Proxy Server の間でポート番号 7658 を開きます。

DeviceDNA™ エンドポイントによる拡張セッション保証の作成

DeviceDNA™ による拡張セッションは、ユーザを DeviceDNA™ 情報を収集するために CA SiteMinder® for Secure Proxy Server 上でホストされている認証フローアプリケーションエンドポイントにリダイレクトします。この DeviceDNA™ 情報によって、それらのセッションが検証されます。

パフォーマンス上の理由で、組織内の各地域別区分につき 1 つのエンドポイントを作成することを推奨します。たとえば、ニューヨークおよびシカゴのオフィスがある場合は、各オフィスのエンドポイントを作成します。

DeviceDNA™ による拡張セッション保証をポリシーまたはアプリケーションに追加する前に、CA SiteMinder® for Secure Proxy Server 上でこれらのエンドポイントを設定します。

以下の手順に従います。

1. 管理 UI から、[ポリシー] - [グローバル] - [セッション保証エンドポイント] をクリックします。
2. [STS エンドポイントの作成] をクリックします。
3. わかりやすい名前と説明（オプション）を入力します。

4. 以下のフィールドに値を入力します。

Web サーバ名

ユーザを認証するために DeviceDNA™ を収集する CA SiteMinder® for Secure Proxy Server サーバの名前を指定します。

ポート

CA SiteMinder® for Secure Proxy Server がリダイレクト用にリスニングしているポート番号を指定します。（SSL を使用して）このポートを安全な接続に設定します。

デフォルト：443

ターゲット

ユーザが暗黙的にリダイレクトされる CA SiteMinder® for Secure Proxy Server の URL を指定します。このサーバは、ユーザの DeviceDNA™ を収集します。DeviceDNA™ を使用して、ユーザと関連付けられたセッションが検証されます。

DeviceDNA™ リフレッシュ間隔

ユーザと関連付けられた DeviceDNA™ が有効になる秒数を指定します。有効な DeviceDNA™ のないユーザは、サーバがユーザの現在の DeviceDNA™ を取得する拡張セッション保証エンドポイントにリダイレクトされます。

DeviceDNA™ リフレッシュ間隔は、DeviceDNA™ の収集を管理します。DeviceDNA™ リフレッシュ間隔の期限が切れた後に発生するリクエストは、DeviceDNA™ を再収集するために Authentication Flow アプリケーションにリダイレクトされます。

DeviceBinder は、このセッションに関連付けられたユーザを識別するセッションプロパティです。DeviceBinder とクライアント側のデバイス ID は認証処理中にリンクされています。このプロパティは、一意の DeviceHash と有効期限から構成されます。

デフォルト：300 秒（5 分）。

5. [サブミット] をクリックします。

レルムへのエンドポイントの追加

DeviceDNA™ による拡張セッション保証を使用してレルムのリソースを保護するには、1つのセッション保証エンドポイントをレルムに追加します。

注: DeviceDNA™ による拡張セッション保証が機能するには、レルムが永続的である必要はありません。

以下の手順に従います。

1. 管理 UI から、[ポリシー]、[ドメイン]、[レルム] をクリックします。
2. 目的のレルムの編集アイコンをクリックします。
3. [セッション] の下で、[拡張されたセッション保証] の横の [有効] チェックボックスをオンにします。
4. [エンドポイントの検索] をクリックします。
5. 目的のエンドポイントを選択します。
6. [OK] をクリックします。
7. [サブミット] をクリックします。
8. セッション保証機能で保護するリソースを持ったレルムが他にもあれば、手順 2 ～ 6 を繰り返します。

アプリケーション コンポーネントへのエンドポイントの追加

DeviceDNA™ による拡張セッション保証を使用して、アプリケーション内のリソースを保護するには、1つの拡張セッション保証エンドポイントに関連するアプリケーションのコンポーネントに追加します。

以下の手順に従います。

1. 管理 UI から、[ポリシー]、[アプリケーション]、[アプリケーション] の順にクリックします。
アプリケーションのリストが表示されます。
2. 目的のアプリケーションの編集アイコンをクリックします。
[アプリケーションの変更] ダイアログ ボックスが表示されます。
3. 以下のいずれかを実行します。
 - アプリケーションに 1 つのみコンポーネントがある場合、[詳細設定] をクリックします。
 - アプリケーションに複数のコンポーネントがある場合は、目的のコンポーネントの編集アイコンをクリックします。 [詳細設定] を選択します。
4. [セッション] の下で、[拡張されたセッション保証] の横の [有効] チェックボックスをオンにします。
5. [エンドポイントの検索] をクリックします。
エンドポイントのリストが表示されます。
6. 目的のエンドポイントを選択します。
7. [OK] をクリックします。
8. [サブミット] をクリックします。
[アプリケーションの変更] ダイアログ ボックスが閉じ、確認メッセージが表示されます。
9. セッション保証機能で保護するリソースを持ったアプリケーションが他にあれば、手順 2 ～ 7 を繰り返します。

パートナーシップにおける DeviceDNA™ による拡張セッション保証の有効化

CA SiteMinder® Federation を使用する場合、以下のパートナーシップにおいて、DeviceDNA™ による拡張セッション保証を有効することもできます。

- SP から IdP へのパートナーシップの IdP 側 (HTTP リダイレクトバインディングのみ)。
- コンシューマからプロデューサへのパートナーシップのプロデューサ側
- RP から AP へのパートナーシップの AP 側

以下の手順に従います。

1. 管理 UI から、[フェデレーション] - [パートナーシップ フェデレーション] - [パートナーシップ] をクリックします。
2. 目的のパートナーシップの左側の[アクション]ボタンをクリックし、[非アクティブ化]を選択します。
3. 同じ[アクション]ボタンを再度クリックし、[変更]を選択します。
4. [SSO と SLO] タブをクリックします。
5. 以下のチェックボックスをオンにします。

セッション保証の有効化

(ポリシー ドメイン モデルの) レルムまたは (アプリケーション モデルの) コンポーネントで指定されたリソースを保護します。また、特定の CA SiteMinder® Federation パートナーシップの認証リクエストを保護できます。セッション保証エンドポイントはユーザーから DeviceDNA™ を収集し、セッションを検証します。

制限： この機能にはセッション保証エンドポイントが必要です。

エンドポイントのリストが表示されます。

6. 目的のエンドポイントのチェックボックスをオンにします。
7. [保存] をクリックします。
8. 目的のパートナーシップがほかにあれば、手順 2 ～ 7 を繰り返します。
9. (ローカル認証モードのみに対して) 認証 URL (Redirect.jsp) に関連付けられるレルム上で DeviceDNA™ による拡張セッション保証を有効にします。

SM--関連するログ ファイル

DeviceDNA™ による拡張セッション保証に関するトランザクションは、以下のログ ファイルに記録されます。

ポリシー サーバ

xps-*.audit -- 機能の設定への変更。

smaccesslog4 -- 機能に関連する認証アクティビティおよび認可アクティビティ。

Advanced Authentication Server の起動ログ

caauthminderstartup

cariskminderstartup

cariskmindercasemgmtserverstartup

Advanced Authentication Server のランタイム ログ

caauthminder

cariskminder

cariskmindercasemgmtserver

CA SiteMinder® for Secure Proxy Server サーバ (Advanced Authentication フロー アプリケーション)

arcotuds.log -- UDS サービスのログ ファイル

CAWebFlowLog.txt -- 認証フロー アプリケーションのログ ファイル

UIApplog.txt -- UIApp (認証フロー アプリケーションの実行プロセスの一部) のログ

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. 819)

[テスト環境エージェントの設定](#) (P. 820)

[機能テストの実行](#) (P. 833)

[コマンドスクリプトファイルに記録されたテストの再生によるリグレーションテストの実行](#) (P. 838)

[ストレステストの実行](#) (P. 839)

[証明書ベースの認証テスト](#) (P. 847)

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 smregghost.exe アプリケーションを実行する必要があります。smregghost.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。

Windows でのテスト ツールの起動

ポリシー サーバ機能をテストするためにテスト ツールを起動します。

重要: Windows Server 2008 上のテスト ツールにアクセスしている場合は、管理者権限でショートカットを開きます。管理者としてシステムにログインしている場合でも、管理者権限を使用します。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。

以下の手順に従います。

以下のいずれかのメソッドを使用します。

- SiteMinder プログラム グループ内の [SiteMinder テスト ツール] アイコンを選択します。
- コマンド ウィンドウで、*policy_server_home\bin* に移動し、以下のコマンドを入力します。

```
smtest
```

UNIX でのテスト ツールの起動

ポリシー サーバ機能をテストするためにテスト ツールを起動します。

以下の手順に従います。

1. コマンド ウィンドウを開き、*policy_server_home/bin* に移動します。
2. 以下のコマンドを入力します。

```
./smtest
```

注: UNIX 上でテスト ツールを実行するには、X ディスプレイ サーバが実行されている必要があります。必要に応じて、コマンド ウィンドウに以下を入力して表示を有効にします。

```
export DISPLAY=n.n.n.n:0.0
```

```
n.n.n.n
```

ポリシー サーバのホスト システムの IP アドレスを指定します。

FIPS 専用環境でテスト ツールを使用する方法

FIPS 移行モードまたは FIPS 専用モードで設定されているポリシー サーバは、Advanced Encryption Standard (AES) アルゴリズムを使用して機密データを暗号化します。[対話型](#) (P. 829) テストを実行する場合、テスト ツールは、必要に応じて FIPS 準拠のアルゴリズムを使用して FIPS 専用モードのポリシー サーバと通信します。

ただし、デフォルトでは、テスト ツールは、[レコード](#) (P. 829) モードでコマンド スクリプト ファイルを作成するときに、機密データの暗号化に FIPS 準拠のアルゴリズムを使用しません。したがって、非 FIPS アルゴリズムで暗号化されたデータが含まれるコマンド スクリプトは、FIPS 専用モードのポリシー サーバをテストするために再生できません。

FIPS 移行モードまたは FIPS 専用モードのポリシー サーバに対するテストを記録および再生するには、以下の手順のいずれかを実行します。

- [コマンドライン オプションを使用した特定の FIPS モードでのテスト ツールの起動](#) (P. 823)
- [CA_SM_PS_FIPS140 環境変数の設定によるデフォルトの FIPS モードの定義](#) (P. 825)

注: コマンドライン オプションを使用してテスト ツールが起動されない場合、CA_SM_PS_FIPS140 環境変数で定義された FIPS モードを使用します。CA_SM_PS_FIPS140 が設定されていない場合、テスト ツールは、デフォルトで FIPS 互換モードになります。

コマンドライン オプションを使用した特定の FIPS モードでのテスト ツールの起動

記録または再生のために特定の FIPS モードでテスト ツールを開くには、**-cf** コマンドライン オプションを使用してテスト ツールを起動します。

以下の手順に従います。

1. コマンド ウィンドウを開いて、以下のいずれかの場所に移動します。

- `policy_server_home\bin` (Windows)
- `policy_server_home/bin` (UNIX)

2. 以下のコマンドを入力します。

```
smtest -cf FIPSmode [command_script]
```

FIPSmode

以下のいずれかの **FIPS** モード（ポリシー サーバの **FIPS** モードに一致）を指定します。

- **COMPAT** (デフォルト)
- **MIGRATE**
- **ONLY**

注: *FIPSmode* の値は大文字と小文字を区別しません。

command_script

(オプション) 再生するコマンド スクリプト ファイルのパス名を指定します。

各 *FIPSmode* 設定のテスト ツールの動作を以下に示します。

COMPAT

以下の特徴で動作するようにテスト ツールを設定します。

- コマンド スクリプト ファイルを記録するときに、非 **FIPS** アルゴリズムを使用して機密データを暗号化します。
- テストを再生するときに、非 **FIPS** または **FIPS** アルゴリズムを使用して、コマンド スクリプト ファイルに書き込まれる機密データを復号化します。

MIGRATE

以下の特徴で動作するようにテスト ツールを設定します。

- コマンド スクリプト ファイルを記録するときに、**FIPS** アルゴリズムを使用して機密データを暗号化します。
- テストを再生するときに、非 **FIPS** または **FIPS** アルゴリズムを使用して、コマンド スクリプト ファイルに書き込まれた機密データを復号化します。

ONLY

以下の特徴で動作するようにテスト ツールを設定します。

- コマンドスクリプト ファイルを記録するときに、FIPS アルゴリズムを使用して機密データを暗号化します。
- [基本的な再生] モードまたは [高度な再生] モードでテストを再生するときに、FIPS アルゴリズムのみを使用して、コマンドスクリプト ファイルに書き込まれた機密データを復号化します。

CA_SM_PS_FIPS140 環境変数の設定によるデフォルトの FIPS モードの定義

CA_SM_PS_FIPS140 環境変数を定義することにより、テスト ツール（およびその他のローカル SiteMinder コンポーネント）のデフォルトの FIPS モードを設定します。

注: -cf コマンドライン オプションを使用してテスト ツールを起動すると、CA_SM_PS_FIPS140 環境変数が無視されます。

以下の手順に従います。

1. 以下のいずれかの操作を実行します。
 - (Windows) 管理者ユーザとして、管理 UI ホスト システムにログインします。
 - (UNIX) 管理 UI をインストールしたユーザとして管理 UI ホスト システムにログインします。
2. 以下の環境変数を設定します。

CA_SM_PS_FIPS140=FIPSmode

FIPSmode

以下のいずれかの FIPS モード（ポリシー サーバの FIPS モードに一致）を指定します。

- COMPAT （デフォルト）
- MIGRATE
- ONLY

注: 環境変数の設定の詳細については、お使いの OS のドキュメントを参照してください。

3. 管理 UI を正しく実行する Windows または UNIX シェルが
CA_SM_PS_FIPS140 変数を認識することを確認してください。

各 *FIPSmode* 環境設定のテスト ツールの動作を以下に示します。

COMPAT

以下の特徴で動作するようにテスト ツールを設定します。

- コマンドスクリプト ファイルを記録するときに、非 FIPS アルゴリズムを使用して機密データを暗号化します。
- テストを再生するときに、非 FIPS または FIPS アルゴリズムを使用して、コマンドスクリプト ファイルに書き込まれる機密データを復号化します。

MIGRATE

以下の特徴で動作するようにテスト ツールを設定します。

- コマンドスクリプト ファイルを記録するときに、FIPS アルゴリズムを使用して機密データを暗号化します。
- テストを再生するときに、非 FIPS または FIPS アルゴリズムを使用して、コマンドスクリプト ファイルに書き込まれる機密データを復号化します。

ONLY

以下の特徴で動作するようにテスト ツールを設定します。

- コマンドスクリプト ファイルを記録するときに、FIPS アルゴリズムを使用して機密データを暗号化します。
- テストを再生するときに、非 FIPS または FIPS アルゴリズムを使用して、コマンドスクリプト ファイルに書き込まれる機密データを復号化します。

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、監査ポート

許可、認証および監査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。

フェールオーバー

フェールオーバーを有効にします。フェー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 \(P. 827\)](#)」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。

以下のリストの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決定します。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。

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 [サーバレスポンス] セクションに表示できます。

[レコード](#) (P. 835)

対話型操作と、テスト結果をプレーンテキストのコマンド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。

[基本的な再生](#) (P. 838)

[レコード] モードで作成されたコマンドスクリプトファイルを使用して、連続テストを自動化します。リグレーションテストに理想的です。

[詳細プレーバック](#) (P. 842)

複雑なテストを自動化するために、手動で設定されたスレッド制御ファイルを使用します。ストレステストに理想的です。

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。

ユーザ認証情報の指定

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。

ユーザ認証情報を指定するには、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。

エンコーディング指定の設定

[エンコーディング指定] 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。

エンコーディング指定を設定するには、エンコーディング指定の値を、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。

テスト ツール設定ファイルへのテスト設定の保存とロード

テスト ツール設定ファイルに設定を保存しておけば、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。その後、いつでもそれらの値を再ロードできます。

テスト ツールで指定されている現在値を保存する方法

1. [設定を保存] ボタンをクリックします。
 2. [別名で保存] ダイアログ ボックスでテスト ツール設定の場所と名前を入力し、[上書き保存] をクリックします。
- ファイルが .ini ファイル拡張子付きで保存されます。

テスト ツール設定ファイルから保存された値を取得するには

1. [設定のロード] ボタンをクリックします。
2. [オープン] ダイアログ ボックスでテスト ツール設定ファイルの場所と名前を入力し、[オープン] をクリックします。

注: コマンド スクリプトからもテスト ツール設定ファイルをロードできます。

(オプション)ポリシー サーバへのテスト ツール接続の制限

テスト ツール設定 (.ini) ファイルを編集し、以下のパラメータを追加して、テスト ツールがポリシー サーバに接続する方法を制限します。

MaxConnections:

テスト ツールがポリシー サーバに対して確立する接続の最大数を指定します。

MinConnections:

テスト ツールがポリシー サーバに対して確立する接続の最小数を指定します。

ConnectionsStep:

新しい接続を作成する必要がある場合、テスト ツールを一度に開くことができる新しいソケット数を指定します (**MaxConnections:** で指定された値まで)。

以下の手順に従います。

1. コマンドスクリプトファイル内の「**SM Agent or Radius:**」パラメータの値を確認し、以下のいずれかの手順を実行します。
 - 「**SM Agent or Radius**」値が **SM Agent v4** に設定されている場合は、手順 3 に進みます。
 - 「**SM Agent or Radius**」値が **SM Agent v5** に設定されている場合は、値 **1** の「**Override Bootstrap:**」パラメータをコマンドスクリプトファイルに追加します。
2. コマンドスクリプトファイル内の「**Agent Name**」パラメータの値がテスト ツール設定ファイル内の「**Agent name**」パラメータと同じであることを確認します。それ以外の場合、新しいパラメータは無視され、次のデフォルト値が使用されます: **MaxConnections=20**、**ConnectionStep=2**。
3. テキストエディタでテスト ツール設定ファイルを開きます。
4. ファイル内のほかのパラメータと同じ形式を使用して、必要なパラメータを 1 行に 1 つずつ追加します。つまり、行の列 24 からパラメータ値を入力します。
5. テスト ツール設定ファイルを保存して閉じます。

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。

(オプション)リグレッション テストおよびストレス テスト用のコマンドスクリプトファイルへのテストの記録

レコード モードでテストを実行する場合、テスト ツールは、プレーン テキストのコマンドスクリプト ファイルにテスト コマンドとテスト結果を書き込みます。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。

同じコマンドスクリプト ファイルに複数のテストを記録できます。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 ファイルをリグレッションテストに使用できます。

以下の手順に従います。

1. [レコード] テスト モードを選択します。
2. テスト結果を保存するコマンドスクリプト ファイルのパスとファイル名を[出力スクリプト] フィールドに入力します。
3. 必要に応じて、記録されたテストを実行する回数を[繰り返し回数] フィールドに入力します。
4. 必要に応じて、コマンドスクリプト ファイルに追加するコメントを[コメント] フィールドに入力します。
5. 1 つ以上のテストを実行します。
6. 記録を停止するには、新しいテスト モードを指定します。

詳細情報:

[FIPS 専用環境でテスト ツールを使用する方法 \(P. 823\)](#)

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 [メッセージ] フィールドに [保護] 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 認証情報 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 [メッセージ] フィールドに [エラー] または [保護されていません] と表示されます。 [エラー] 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 [メッセージ] フィールドに [認証されました] 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。例：

```
ID          Length  ASCII Format  Hexadecimal format
4> id 215   len 005   'LDAP:' -    '4c 44 41 50 3a'
```

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。

理由

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 [メッセージ] フィールドに「認証されていません」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 [メッセージ] フィールドに「許可」と表示され、SiteMinder によって割り当てられたセッション ID が [セッション 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。

IsAuthorized の状況	結果
失敗	テストツールの [メッセージ] フィールドに [未許可] 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、[コマンド] グループボックスの [経過時間] 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. [繰り返し回数] 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。

コマンド スクリプト ファイルに記録されたテストの再生によるリグレッション 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. [\[レコード\] モードで、1つ以上の機能テストを実行します](#) (P. 835)。
2. [モード] グループ ボックスで [基本的な再生] を選択します。

3. [入力スクリプト] フィールドにコマンド スクリプト ファイルの名前を入力します。

このファイル名は、[レコード] モードで [出力スクリプト] フィールドに入力した値と同じ値でなければなりません。

4. [出力スクリプト] フィールドに、出力ファイルの名前を指定します。
5. [コマンド] グループボックスで [スクリプトの実行] をクリックします。

コマンド スクリプト ファイルが実行され、出力スクリプトファイルが作成されます。

詳細情報:

[FIPS 専用環境でテスト ツールを使用する方法 \(P. 823\)](#)

ストレス テストの実行

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、[詳細プレーバック] 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の設定

ストレス テストに対する複雑なテストを自動化するため、スレッド制御 ファイルを設定して、実行するコマンド スクリプト ファイル、繰り返しの数、スレッドなどを定義します。スレッド制御ファイルには、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。

基本的な命令の形式は、以下のとおりです。

command_script_file_name, *repetition_count*, *thread_count*, [*max_time_in_seconds*]

command_script_file_name

以前に記録されたコマンド スクリプト ファイルのパス名を指定します。

repetition_count

テスト ツールがコマンド スクリプトを実行する回数を指定します。

thread_count

コマンド スクリプトを実行するために、テスト ツールが開始する同時スレッド数を指定します。

max_time_in_seconds

(オプション) テストの期間の制限 (秒単位) を指定します。テストの経過時間がこの制限を超えた場合、設定された繰り返し回数が完了したかどうかに関係なく、再生が停止します。

例 :

```
# c:¥temp¥test_data.txt, 8, 6, 120
```

この行は以下の内容を指定します。

- コマンド 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
- 同時に **6 つ**のスレッドがスクリプトを実行
- テストは、**8 回**の繰り返しが完了しているかどうかにかかわらず **120 秒**後に終了

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
.report	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。
.reportspread	レスポンス時間スプレッド レポートが含まれるファイルを生成します。ファイル名は <code>command_script_stats_spread</code> （例： <code>isprotected-record.txt_stats_spread</code> ）です。
.output	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。
.viewstats	最終的な統計情報をテキストエディタに表示します。
.verbose	各サーバ要求の詳細を含む出力ファイルを生成します。
.brief	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。
.sleep	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。
.userselectionmode	ユーザ名が順次使用されるか、またはランダムに選択されるかを決定します。有効な値は <code>0</code> （順次）または <code>1</code> （ランダム）です。詳細については、「 (オプション) テストでのユーザ名の処理方法の設定 (P. 843) 」を参照してください。
.randomseed	各テストに対して、名前の疑似ランダム順を繰り返すことが (スレッド順序指定まで) 可能になります。詳細については、「 (オプション) テストでのユーザ名の処理方法の設定 (P. 843) 」を参照してください。

コマンド	説明
<code>.connect settings_file</code>	テスト ツール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。

スレッド制御ファイルの例

```
.output
.connect c:%test%smtest.ini
.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
```

高度な再生モードでスレッド制御ファイルを実行することによるストレス テストの実行

高度な再生テスト モードを使用して、スレッド制御ファイルで定義されたストレス テストを実行します。

以下の手順に従います。

1. [モード] グループボックスで [高度な再生] を選択します。
2. [制御ファイル] フィールドに、スレッド制御ファイルの名前を入力します。

3. [コマンド] グループボックスで [スクリプトの実行] をクリックします。

テスト ツールはスレッド制御ファイルを実行し、出力スクリプト ファイルを作成します。

(オプション)テストでのユーザ名の処理方法の設定

以下の 3 つのパラメータは、テストでユーザ名をどのように処理するかを設定します。

34 UserCount:

(コマンド スクリプト ファイルに定義) テストで認証と許可に使用されるユーザ数を **AAAAAA**、**BAAAAA** から **ZZZZZZ** までの形式で指定します。**UserCount:** パラメータが設定されている場合、**UserName:** パラメータは無視されます。

注: **Username:** パラメータは無視されますが、引き続き必要です。削除しないでください。

コマンド スクリプト ファイル内のほかのパラメータと同じ形式を使用して、**UserCount:** パラメータを追加します。つまり、行の列 24 から完全なパラメータ名 (「34」プレフィックスを含む) およびパラメータ値を入力します。

例:

```
34 UserCount:          1000
```

.userselectionmode

(スレッド制御ファイルに定義) ユーザ名が順次選択されるか、またはランダムに選択されているかを決定します。有効な値は 1 および 2 です。

1 に設定されている場合、テスト手順は、**UserCount** パラメータで指定された値まで、順次ユーザ名に従って進みます。その後、繰り返しがすべて完了するか、時間が経過するまで、ユーザ名は「AAAAAA」に戻ります。

2 (デフォルト) に設定されている場合、ユーザはランダムに選択されます。

`.randomseed`

（スレッド制御ファイルで定義）ゼロ以外の整数値に設定されている場合、「AAAAAA」から「UserCount」パラメータの値によってインデックスを付けられたユーザ名までの範囲で、ユーザ名のセットがランダムに順序付けされます。これにより、各テストに対して、名前の疑似ランダム順を繰り返すことが（スレッド順序指定まで）可能になります。

`.randomseed` が指定されていない場合、現在の時刻が使用されます。

（オプション）安定したロードをシミュレートするためのポリシー サーバリクエスト間のスリープ時間の設定

リクエストが一定のレートでポリシー サーバに送信される安定したロードをシミュレートするには、スレッド制御ファイル内の以下のパラメータを設定します。

`.sleepbetweenrequests`

各スレッドのポリシー サーバへの各リクエスト間のスリープ時間（ミリ秒単位）を指定します。

たとえば、`.sleepbetweenrequests` パラメータを 5（ミリ秒）に設定した場合、各スレッドで、リクエストはポリシー サーバに 1 秒あたり約 200 回送信されます。

注: 実際のリクエスト レートは、リクエストの処理時間、各スレッドが取得する CPU 時間、および外部要因などの要因によって影響を受けます。

(オプション) CSV 形式でファイルに再生テストの結果を書き込むためのテストツールの設定

以下のシステム環境変数を設定することにより、カンマ区切り値 (CSV) 形式ファイルに再生テストの結果をすべて書き込むようにテスト ツールを設定できます。

SMTESTCSVFILE

テスト ツールが再生テストの結果を書き込む CSV ファイルのパス名を指定します。再生テストの結果は、先頭にヘッダ行付きで、指定されたファイルにカンマ区切り形式で書き込まれます。

ファイルがすでに存在する場合、そのファイルに追加されます。スレッド制御ファイルによって、複数の記録の再生が指定されている場合、結果が集約されます。

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

SiteMinder テスト ツールはユーザ認証および許可をシミュレートします。証明書ベースの認証計画では、追加の設定を必要とします。

別の証明書生成ツールでは、発行者 DN の形式および証明書の他の属性に影響する場合があります。

たとえば、IIS Web サーバ上の `certutil.exe` で生成された証明書用の発行者 DN は、状態を表すために `ST=` を使用する可能性があります。ただし、Oracle iPlanet Web サーバ上の `OpenSSL` ツールで生成された証明書用の発行者 DN は、状態を表すために `S=` を使用する可能性があります。

注: 特定の証明書生成ツールによって使用される実際の値の詳細については、証明書生成ツールのベンダーによって提供されているドキュメントを参照してください。

証明書ベースの認証方式をテストするには、別の証明書生成ツールで作成された証明書に対応するためにポリシー サーバで証明書マッピングを設定します。

カスタム マッピングを必要とする証明書属性

いくつかの共通の証明書属性は、証明書を生成するために使用されるサードパーティ ツール (`certutil.exe` または `OpenSSL` など) によって異なります。以下の属性間の違いにより、SiteMinder テスト ツールでエラーが発生する可能性があります。

電子メール アドレス

証明書生成ツールのベンダーによって、`E` または `Email` で表されます。

都道府県

証明書生成ツールのベンダーによって、`S` または `ST` で表されます。

ユーザ ID 番号

証明書生成ツールのベンダーによって、`UID` または `UserID` で表されます。

注: 特定の証明書生成ツールによって使用される実際の値の詳細については、証明書生成ツールのベンダーによって提供されているドキュメントを参照してください。

テスト用のカスタム属性マッピング

SiteMinder テスト ツールを証明書認証方式に対して使用すると、(ブラウザおよび Web サーバを介して) 正常に動作しても、失敗することがあります。認証ログには、証明書で使用される発行者 DN とは異なる形式の発行者 DN をテスト ツールが予期していることが示されます。

この状況が発生するのは、発行者 DN および他の属性が、使用される証明書生成ツールのタイプによって異なる場合です。たとえば、IIS Web サーバ上の `certutil.exe` プログラムは、発行者 DN の状態の名前を短縮するために `ST=` を使用する可能性があります。ただし、Oracle iPlanet Web サーバ上の OpenSSL ツールは、状態の名前を短縮するために `S=` を使用する可能性があります。

注: 特定の証明書生成ツールによって使用される実際の値の詳細については、証明書生成ツールのベンダーによって提供されているドキュメントを参照してください。

この状況は、「[カスタム マッピングが必要な証明書属性 \(P. 847\)](#)」にあるリストの他の属性に似ています。

この問題を解決するには、ポリシー サーバで管理者に各発行者 DN 形式のマッピングを作成してもらいます。これにより、ポリシー サーバは、別の証明書生成ツールによって作成された発行者 DN 形式を受け入れることができます。

発行者 DN マッピング

別の証明書生成ツール（certutil.exe や OpenSSL など）は少し異なる方法で発行者 DN を作成します。たとえば、1 つのツールは、以下のような発行者 DN を作成する可能性があります。

```
CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, S=Western Cape, C=ZA
```

別のツールは、以下のような発行者 DN を作成する可能性があります。

```
CN=Personal Freemail RSA 2000.8.30, OU=Certificate Services, O=Thawte, L=Cape Town, ST=Western Cape, C=ZA
```

複数の可能性に対応するには、ポリシー サーバで、環境内のすべての発行者 DN 形式のマッピングを管理者に作成してもらいます。

注: 特定の証明書生成ツールによって使用される実際の値の詳細については、証明書生成ツールのベンダーによって提供されているドキュメントを参照してください。

カスタム証明書マッピングの作成

SiteMinder 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が作成されていることを前提とします。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 \(P. 847\)](#)」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 で電子メール属性が異なって表示される、さまざまなタイプの証明書生成ツール (certutil.exe や OpenSSL など) および SiteMinder テスト ツールからのリクエストを処理できます。この処理は、「[カスタム マッピングが必要な証明書属性 \(P. 847\)](#)」に記載された、他のあらゆる属性に使用できます。

詳細情報:

[X.509 クライアント証明書認証方式の証明書マッピング \(P. 471\)](#)

第 25 章: エージェント設定のトラブルシューティング

英語以外の入力文字にジャンク文字が含まれる

UNIX/Linux 上で有効

症状:

英語以外の入力文字の一部がコンソール ウィンドウに正しく表示されません。 -

解決方法:

コンソール ウィンドウのターミナルの設定を確認します。コンソールが入力文字の高位 (8) ビットをクリアしないことを確認します。以下のコマンドを実行します。

```
stty -istrip
```

Advanced Authentication Server からのエラー メッセージ

症状:

私のユーザで以下のブラウザ エラーが発生しています。

内部エラーが発生しました。 ブラウザを閉じ、再度開いてください。 問題が解決されない場合は、テクニカル サポートにお問い合わせください。

解決方法:

以下の項目を確認します。

- ポリシー サーバで RiskMinder サービスが稼働中であることを確認します。
- 以下のログ ファイルをチェックして、RiskMinder が READY 状態であることを確認します。

`siteminder_installation_directory\..\%aas%\logs\cariskminderstartup.log`

- CA SiteMinder® for Secure Proxy Server サーバを再起動します。
- DeviceDNA™ が収集されていない場合は、以下の手順を実行します。
 - a. JSafeJCE セキュリティ プロバイダで JVM を設定します。
 - b. CA SiteMinder® for Secure Proxy Server サーバを再起動します。

詳細情報:

[DeviceDNA™ による拡張セッション保証に対する SiteMinder 環境の設定 \(P. 786\)](#)

"Token not Found" エラー

症状:

CA SiteMinder® for Secure Proxy Server からのフロー アプリケーション ログ ファイルに以下のエラーが表示されます。

Token is not found...

解決方法:

以下の項目を確認します。

- エージェントが有効であることを確認します。
- ポリシー サーバがセッション ストアに接続されていることを確認します。

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. 855)

[SSL 設定](#) (P. 856)

[SSL のトラブルシューティング](#) (P. 860)

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ことが必要です。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。これらのコンポーネントを個別に分析できるように、Web エージェントと対話せずに確認してください。

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。

以下の手順に従います。

1. SSL を介した認証方式を使用するレلمを保護している Web エージェントを無効にします。
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 サーバ)
 - `https://web_server_name:port/<SSL Virtual Directory>` (IIS Web サーバ)
 - `https://web_server_name:port` (Apache-ベースの Web サーバ)

SSL 接続が証明書を要求するよう設定されている場合は、証明書を選択することになります。

SSL 設定

SSL を設定し、本製品を使用する前に適切に機能することを確認します。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。この設定によって、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、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 接続を確立する手順を説明します。

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。

以下の手順に従います。

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)] オプション ボタンを選択して、[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。

SSL を使用する Netscape Web サーバの設定

証明書を要求することにより、SSL 用の Netscape Web サーバを設定します。

以下の手順に従います。

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、証明書を要求します。[Encryption Preferences] 設定を確認します。この設定では、[Require Certificates] がオンに設定されている必要があります。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 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。

以下の手順に従います。

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します。

SSL Web Server は、認証局ごとに証明書が必要です。主な認証局はすでにインストールされている可能性があります。証明書スナップインを使用して、**Windows** オペレーティング環境の証明書を設定できます。

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 [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、以下の手順に従って SSL Web サーバ上に認証局の証明書をインストールします。

以下の手順に従います。

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
2. 認証局の証明書を `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

証明書のプロンプトが表示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ます。エージェントがインストールされている場合は、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。

以下の手順に従います。

1. SSL を介した認証方式を使用するレلمを保護している Web エージェントを無効にします。
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、証明書を選択することになります。

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は、以下の手順に従います。

- すべての Firefox ブラウザが、毎回証明書を要求するよう設定されていることを確認する。
- すべての Web サーバが SSL を使用し、証明書を要求するよう設定されていることを確認する。
- 本製品によって使用される SiteMinder 仮想ディレクトリごとに、次の設定を確認する。
- Web サーバの証明書の有効期限を確認する。
- ブラウザ証明書が有効であることを確認する。

すべての Firefox 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。

Firefox Web 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。

以下の手順に従います。

1. Firefox ブラウザで、Firefox メニューから [オプション] を選択します。
2. [詳細] をクリックします。
3. [暗号化] タブをクリックします。
4. [証明書] セクションで、[毎回自分で選択する] オプション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

以下の手順に従います。

IIS Web サーバの場合

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 ベースの 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。

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. キーを選択します。プロパティを表示し、キーの有効期限が切れていないことを確認します。
5. 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

証明書がない、または証明書が無効な場合、ユーザが証明書プロンプトを受信できないことがあります。

Web ブラウザを開き、ブラウザ証明書の有効性を確認します。

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。証明書の一覧が表示されていないか、または有効期限が切れている場合は、新しい認証局の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、または有効期限が切れている場合は、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

ポリシー サーバと Web エージェントの設定を確認します。

以下の手順に従います。

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

この状況は、いくつかの設定エラーによって発生する場合があります。一般的なエラーの一部は、以下のとおりです。

- 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、本製品の認証処理を実行できません。Web エージェントのロギング オプションを有効して、この状況を確認します。適切なログには、ユーザが不明であることが示されます。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。ポリシーに関連付けられているユーザを確認します。
- Apache Web サーバの場合は、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

このエラーは、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。

詳細情報:

[認証方式](#) (P. 307)

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 \(P. 867\)](#)

[LanMan ディレクトリ接続の前提条件 \(P. 868\)](#)

[LanMan ディレクトリ接続の設定 \(P. 868\)](#)

[Windows ユーザ ディレクトリのフェイルオーバー \(P. 870\)](#)

[LanMan ユーザ ディレクトリの検索条件 \(P. 870\)](#)

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。このプロセスにより、**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、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** 接続では **PDC** がバイパスされます。

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します。

詳細情報:

[LanMan ディレクトリ接続の設定 \(P. 868\)](#)

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 \(P. 869\)](#)
2. LanMan ユーザ ディレクトリ接続の設定

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。

以下の手順に従います。

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。
2. **regedit** と入力し、[OK] をクリックします。
3. 次のレジストリキーを変更します。
 - HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥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¥Wow6432Node¥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¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥Ds¥Lanman_DC¥<NT_domain_name>
 例 :
 ¥HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥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。

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。

以下の手順に従います。

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] をクリックします。
3. [ユーザ ディレクトリの作成] をクリックします。
4. 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。
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。

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。唯一の違いは、実際のドメイン コントローラがリクエストを処理することです。この設定は、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。

詳細情報:

[ユーザ ディレクトリの検索](#) (P. 244)

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。

[CA SSO/WAC の統合の概要 \(P. 872\)](#)

[統合された SiteMinder および CA SSO のアーキテクチャ例 \(P. 873\)](#)

[SiteMinder と CA SSO の統合の前提条件 \(P. 880\)](#)

[SiteMinder から CA SSO へのシングルサインオンの設定 \(P. 881\)](#)

[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 \(P. 883\)](#)

[CA SSO から SiteMinder へのシングルサインオンの設定 \(P. 885\)](#)

[smetssocookie Web エージェント アクティブ レスポンス属性の設定 \(P. 887\)](#)

[smauthsso カスタム認証方式の設定 \(P. 888\)](#)

CA SSO/WAC の統合の概要

本製品には、CA SSO 環境へのシングルサインオン機能があります。ユーザは、いずれかの製品を使用する環境にログインします。認証されたら、両方の環境にアクセスできます。すべての認証されたユーザは、許可されている限り、認証情報を再入力せずに、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。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、両方の製品は、それら自身のセッションストアにそれぞれ、ユーザの認証情報を維持しています。さらに、それら独自のセッション認証情報も持っており、これは別々に維持されます。これらの認証情報は別のストア内に保管されているため、ポリシーサーバと CA SSO ポリシーサーバは、同じ Cookie ドメインの一部である必要があります。また、ユーザストアまたは認証ストアを共有している必要もあります。

このシングルサインオン設定では、両方のポリシーサーバを、同じマシンと別のマシンのいずれにも配置できます。本製品には、Web エージェント、CA SiteMinder® for Secure Proxy Server、または両方を含めることができます。使用している環境に基づいて、エージェントまたは CA SiteMinder® for Secure Proxy Server を使用してください。CA SSO 製品では、eTrust Web Access Control (WAC) エージェントが使用されます。本製品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。

詳細情報:

[統合された SiteMinder および CA SSO のアーキテクチャ例 \(P. 873\)](#)

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 サインオンの実例を示す例です。

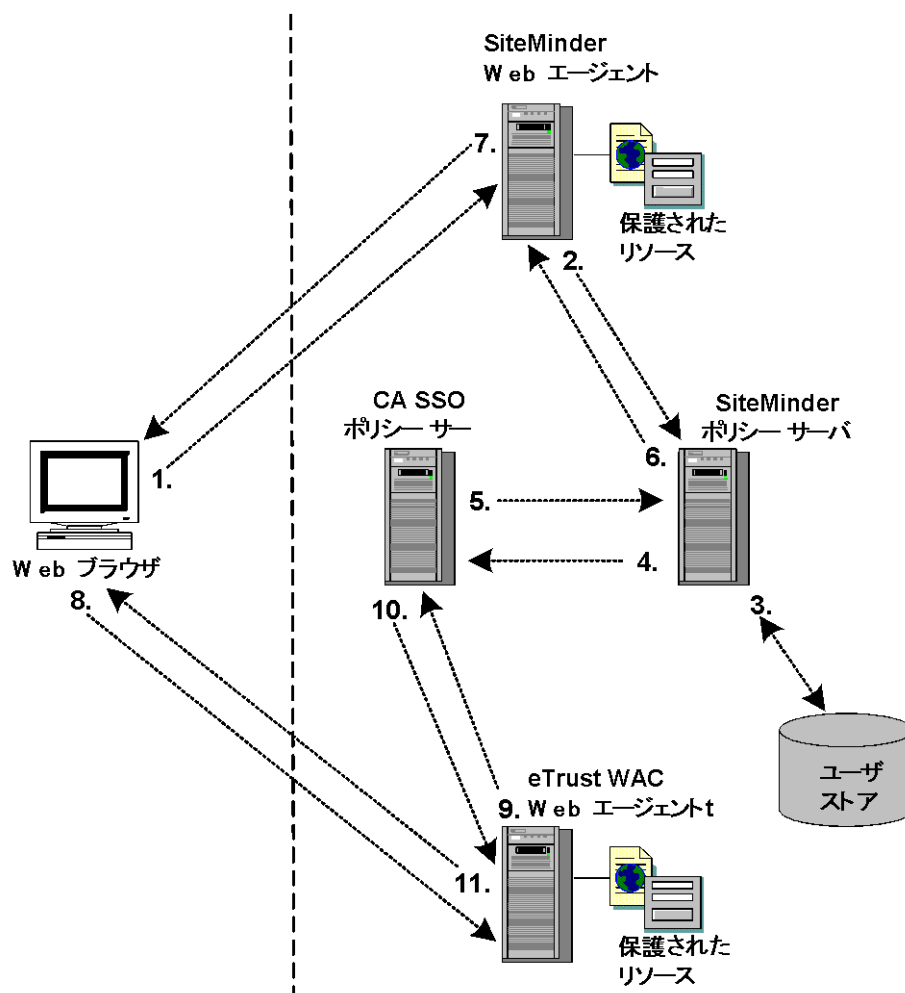
1. ユーザは、Web ブラウザを使用する SiteMinder で認証され、その後に、CA SSO で保護されたリソースにアクセスします（「[例 1： ユーザが、CA SSO の前に、SiteMinder で保護されたリソースにアクセス \(P. 874\)](#)」を参照）。
2. ユーザは、デスクトップ CA SSO クライアント経由で CA SSO で認証され、その後に、Web ブラウザを使用して、SiteMinder で保護されたリソースにアクセスします（「[例 2: 認証された CA SSO クライアント ユーザが SiteMinder リソースにアクセス \(P. 876\)](#)」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 \(P. 878\)](#)」を参照）。

詳細情報:

[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 \(P. 883\)](#)

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。

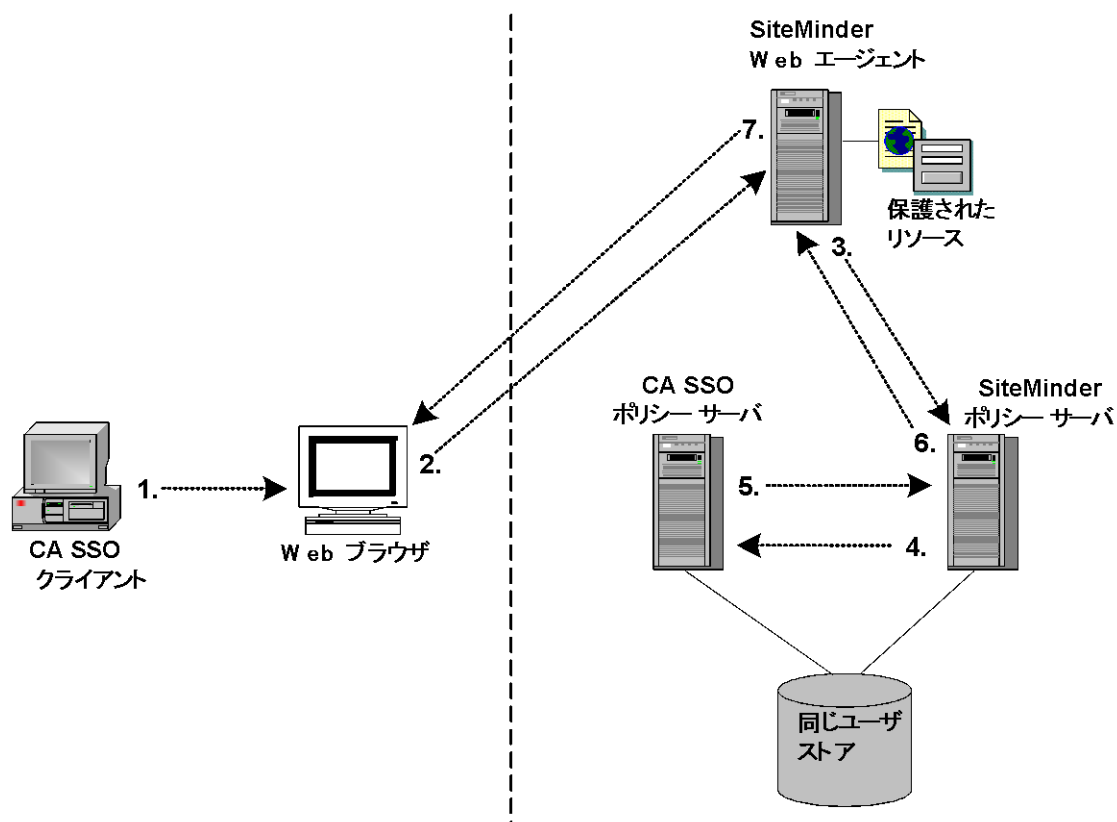


1. ユーザが SiteMinder で保護されたリソースにアクセスしようとする、SiteMinder Web エージェント/CA SiteMinder® for Secure Proxy Server はそのリクエストをインターセプトします。ユーザはエージェント/SPS に認証の認証情報を渡します。
2. Web エージェント/CA SiteMinder® for Secure Proxy Server は認証情報をポリシー サーバに転送し、検証します。
3. ポリシー サーバは、ユーザの認証情報が有効であることを確認します。
4. 認証に成功すると、ポリシー サーバは CA SSO ポリシー サーバに、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 認証情報をポリシー 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/CA SiteMinder® for Secure Proxy Server に CA SSO Web 認証情報を転送します。
7. SiteMinder Web エージェント/CA SiteMinder® for Secure Proxy Server は、ブラウザに CA SSO Web 認証と SiteMinder Cookie を設定します。 リソースがユーザに表示されます。
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 認証情報を 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認証情報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。



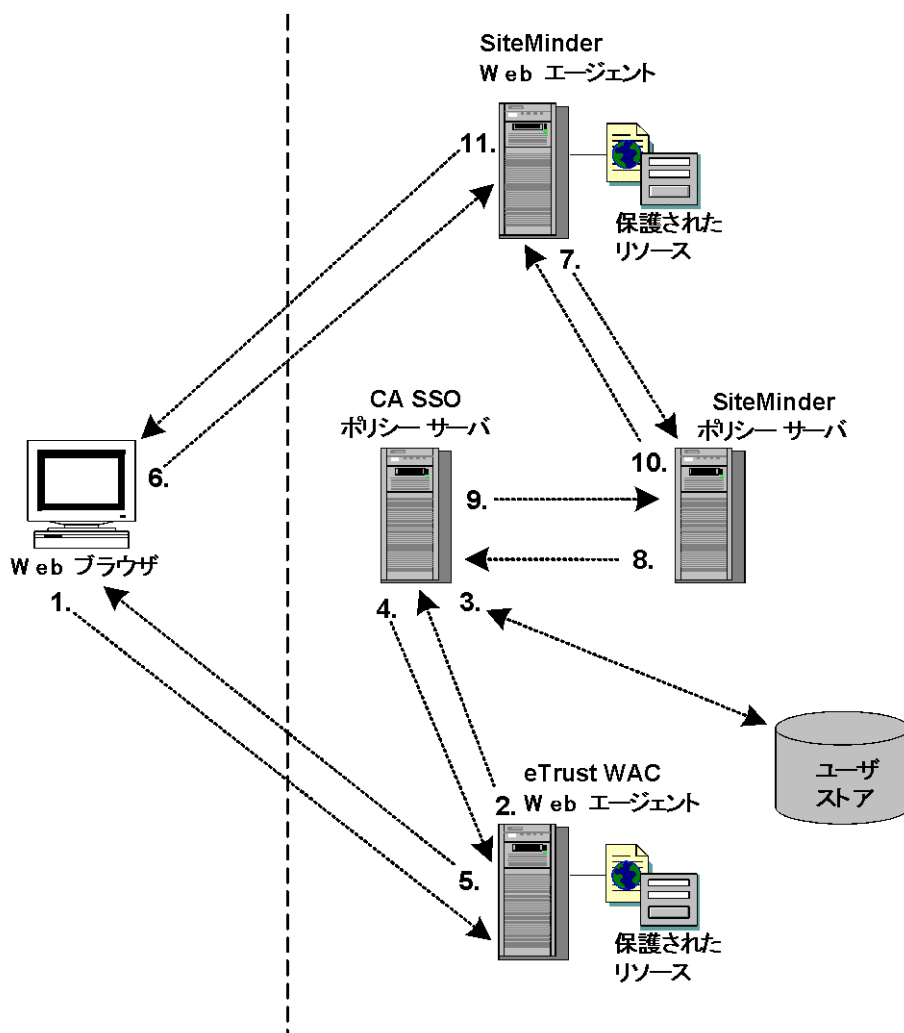
1. 認証済み CA SSO クライアント ユーザが Web ブラウザを起動します。この時、CA SSO クライアントはブラウザに CA SSO Web 認証 cookie を設定します。
2. ユーザが Web ブラウザを使用して、SiteMinder で保護されているリソースにアクセスしようとする、そのリクエストは SiteMinder Web エージェント/CA SiteMinder® for Secure Proxy Server によってインターセプトされます。
3. SiteMinder Web エージェント/CA SiteMinder® for Secure Proxy Server 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/CA SiteMinder® for Secure Proxy Server に返します。
7. SiteMinder Web エージェント/CA SiteMinder® for Secure Proxy Server 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 認証情報を持っています。

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、本製品の前に 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の認証情報を渡します。
2. エージェントは認証情報を **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、ユーザの認証情報が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** 認証情報を **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、**CA SSO Web** 認証 Cookie をユーザの Web ブラウザに設定します。
6. ユーザが保護されたリソースにアクセスしようとする、**Web** エージェント/**CA SiteMinder® for Secure Proxy Server** はそのリクエストをインターセプトします。
7. **Web** エージェント/**CA SiteMinder® for Secure Proxy Server** は、ユーザの **CA SSO Web** 認証情報をポリシー サーバに転送します。
8. ポリシー サーバは、ユーザの **CA SSO Web** 認証の認証情報を **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** 認証認証情報を検証します。次に、**CA SSO** ポリシー サーバは、ポリシー サーバにユーザ名を転送します。
10. ポリシー サーバは、返されたユーザ名をユーザストアで検証し、対応する Cookie を発行してエージェント/**CA SiteMinder® for Secure Proxy Server** に返します。
11. **Web** エージェント/**CA SiteMinder® for Secure Proxy Server** は、ユーザのブラウザに Cookie を設定します。このプロセスにより、ユーザはリクエストされたリソースにアクセスできます。

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 `smetssockie` 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、**12.52 SP1 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

本製品には、CA SSO 環境へのシングル サインオン機能があります。

SiteMinder Web エージェントまたは CA SiteMinder® for Secure Proxy Server を使用して SiteMinder から CA SSO へのシングル サインオンを有効にするには、以下の手順に従います。

エージェントまたは CA SiteMinder® for Secure Proxy Server と共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。

Apache 2.0 Web エージェントの場合

- WebAgent.conf ファイルの以下の行のいずれかからコメント（#）文字を削除します。
 - （Windows オペレーティング環境）
`#LoadPlugin=Path_to_eTSSOPlugin.dll_file`
 - （UNIX または Linux オペレーティング環境）
`#LoadPlugin=Path_to_libetssoplugin.so_file`

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 の CA SiteMinder® for Secure Proxy Server 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 CA SiteMinder® for Secure Proxy Server を再起動します。これで新しい設定が有効になります。

以下の手順に従います。

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 コンピュータ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「Default authentication method」を 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コンピュータをポイントします。
 - `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コンピュータの IP アドレスをポイントします。

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. 以下の設定を確認します。
 - 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。ポリシー サーバ内での `smetssockie` のアクティブ レスポンスには、このキーが必要です。

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。
2. 同じユーザ ストアまたは認証ストアの SiteMinder と CA SSO ポリシー サーバを設定します。

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. 管理 UI を使用して、ドメイン、レルム、およびルールを作成します。
SiteMinder Web エージェントを使用して、リソースを保護します。

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認証情報をユーザに設定します。
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認証情報を渡します。
4. 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。

認証情報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。

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 smauthetssso カスタム認証方式の設定
2. 管理 UI を使用して、ドメイン、レルム、およびルールを作成します。
SiteMinder Web エージェントを使用して、リソースを保護します。

3. リソースの `smauthetsso` カスタム認証方式を設定します。
4. CA SSO クライアントを保護しているブラウザアクセスできるユーザーに、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、以下の項目を設定します。

注: `SsoClnt.ini` ファイルは、CA SSO クライアント コンピュータ上の
`C:\Program Files\CA\CA SSO\Client` にインストールされます。

`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、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 から再認証を求められることなく、リソースが表示されます。

CA SSO から SiteMinder へのシングル サインオンの設定

SiteMinder には、CA SSO クライアントから本製品へのシングル サインオン機能があります。

以下の手順に従います。

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. ドメイン、レルム、およびルールを作成します。Web エージェントを使用して、リソースを保護します。
3. 保護されたリソースの `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: ドメインの値は、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 コンピュータの `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」パラメータと「The default authentication method」パラメータを SSO に設定します。
 - `WebServerName`、`PrimaryWebServerName`、`AgentName`、`NTLMPath` および `Secure` は、SSO Web Access Control がインストールされているコンピュータ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ように指定します。
 - 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。

SiteMinder Web エージェントまたは CA SiteMinder® for Secure Proxy Server の設定手順

1. エージェントまたは CA SiteMinder® for Secure Proxy Server と共にインストールされている SSO プラグインを有効にします。このプラグインにより SSO クライアント Cookie が認証されます。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 | path_to_libetssoplugin.so
```

注: WebAgent.conf ファイルは以下の場所にあります。

Apache 2.0 Web エージェント

6.0 CA SiteMinder® for Secure Proxy Server

```
SPS_install_dir¥proxy-engine¥conf¥defaultagent¥
```

SPS_install_dir

CA SiteMinder® for Secure Proxy Server 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、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ます。有効な認証情報を入力します。
3. 同じブラウザを使用して、Web エージェントによって同じブラウザで保護されているリソースにアクセスします。

認証情報の入力を求められることなく、リソースが表示されます。

smetssock Web エージェント アクティブ レスpons属性の設定

smetssock 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。

以下の手順に従います。

1. [ポリシー] - [ドメイン] をクリックします。
2. [レスpons] をクリックします。
3. [レスponsの作成] をクリックします。
4. リストからドメインを選択し、[次へ] をクリックします。
5. レスponsの名前と説明を定義します。
6. SiteMinder が選択されており、Web エージェントが [エージェント タイプ] 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。
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。
9. [属性] リストから [WebAgent-HTTP-Cookie-Variable] を選択します。
10. [属性の種類] 内の [アクティブ レスpons] を選択します。
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、**smetssock** と入力します。
13. [関数名] フィールドに、**GenEtssoCookie** 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、以下のトークンを定義します。

```
<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。

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。

16. [完了] をクリックします。

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、ポリシー サーバで CA SSO 認証情報を検証できるため、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、ログイン認証情報として受け取った 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 サーバにフェールオーバーするよう設定が可能です。フェールオーバーを設定するには、[認証方式] ページの [スキーム セットアップ] 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。

以下の手順に従います。

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] をクリックします。

3. [認証方式の作成] をクリックします。
[認証方式タイプの新しいオブジェクトの作成] が選択されていることを確認します。
[OK] ボタンをクリックします。
4. [認証方式のタイプ] リストから [カスタム テンプレート] を選択します。
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認証情報のタイプを指定します。選択肢は、Cookie、Cookieorbasic、Cookieorforms です。

Cookie

CA SSO Cookie のみが受理可能であることを指定します。

Cookieorbasic

CA SSO Cookie が提供されない場合にログイン名とパスワードを決定するために基本認証方式が使用されることを指定します。

Cookieorforms

CA SSO Cookie が提供されない場合にログイン名とパスワードを決定するためにフォーム認証方式が使用され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。

第 26 章: CA User Activity Reporting Module の統合

CA User Activity Reporting Module (CA UAR) には、SiteMinder と CA UAR の統合の設定方法について詳しく説明されている SiteMinder コネクタ ガイドがあります。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設定されているかによって決まります。

CA UAR コネクタ ガイドを検索するには、以下の手順に従います。

1. [CA User Activity Reporting Module Integration Matrix](#) に移動します。
2. [Product Integrations] の下にある [Authentication Service] をクリックします。

SiteMinder コネクタ ガイドは CA UAR で使用される LogSensor のタイプに基づいています。

3. 以下のいずれかの操作を行います。

- SiteMinder が監査情報をテキスト ファイルに格納する場合は、File LogSensor 用のコネクタ ガイドを使用します。
- SiteMinder が監査情報を ODBC データベースに格納する場合は、ODBC LogSensor 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 **CA UAR 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 D: 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 \(P. 893\)](#)

[RADIUS クライアント/サーバアーキテクチャ \(P. 894\)](#)

[ポリシー サーバを使用した RADIUS 認証の動作 \(P. 894\)](#)

[RADIUS 環境のポリシー \(P. 896\)](#)

[RADIUS ポリシー ドメインのレスポンス \(P. 901\)](#)

[RADIUS 環境への SiteMinder の展開 \(P. 910\)](#)

[RADIUS デバイスの保護に関するガイドライン \(P. 911\)](#)

[同機種 RADIUS 環境でユーザを認証する方法 \(P. 912\)](#)

[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 \(P. 914\)](#)

[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 \(P. 919\)](#)

[RADIUS エージェント グループの概要 \(P. 923\)](#)

[RADIUS エージェント グループのセットアップ \(P. 924\)](#)

[RADIUS レスポンスのグループ化 \(P. 925\)](#)

[RADIUS のトラブルシューティングとテスト \(P. 926\)](#)

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 監査を使用するには、別に RADIUS 監査サーバを設定する必要があります。ポリシー サーバは、監査サーバに 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、監査情報はファイルに記録できます。

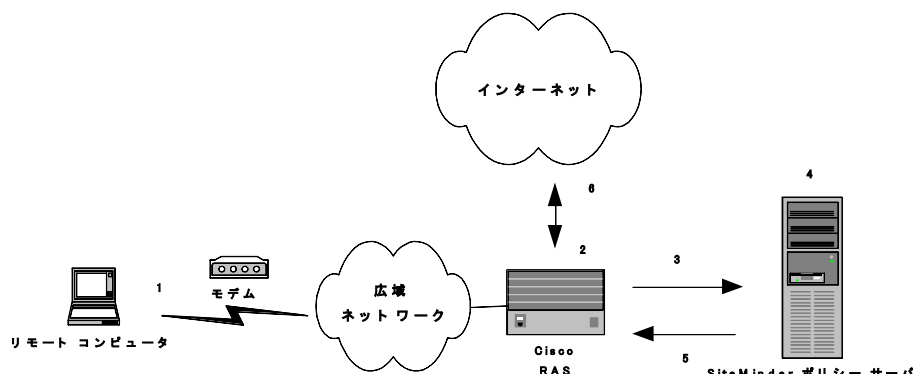
詳細情報:

[監査とデバッグのための RADIUS ログの生成 \(P. 927\)](#)

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。

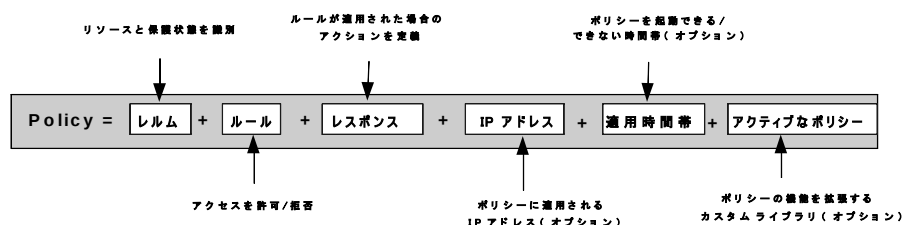
5. ポリシー サーバは、認証レスポンスを RAS に送信します。
 6. 次のいずれかが実行されます。
 - 認証が失敗すると、RAS は接続を拒否します。
 - 認証が成功すると、RAS は、ユーザの認証に対する SiteMinder のレスポンスから属性のリストを受け取ります。属性は、ユーザのネットワーク セッションを設定するユーザ プロファイルとして使用されます。
- RAS は、ポリシー サーバに、セッションが開始されたこと、およびいつセッションが終了するかを通知します。

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。

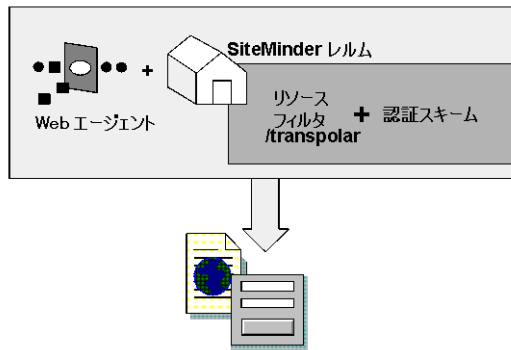


RADIUS ポリシーは、SiteMinder エージェントが使用するポリシーに含まれる要素と同じ要素で構成されていますが、RADIUS エージェントは、これらのコンポーネントを異なる方法で解釈します。ルール、レلم、およびレスポンスは、次の表に示すように、異なる機能を実行します。

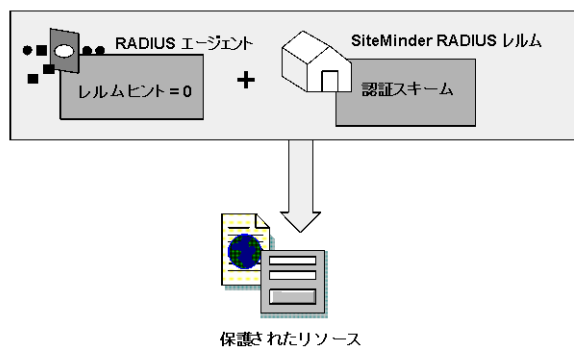
ポリシーのコンポーネント	RADIUS ポリシーでの機能	SiteMinder エージェント ポリシーでの機能
レلم	■ エージェントの識別 ■ 認証方式の識別 ■ セッションタイムアウトの定義	■ リソース フィルタ (SiteMinder エージェントが管理するドメイン内のディレクトリ) の定義 ■ エージェントの識別 ■ 認証方式の識別 ■ リソースのステータス (保護されているかいないか) の定義 ■ 処理するイベント (認証または許可) の識別 ■ セッションタイムアウトの定義
ルール	■ 認証のみ ■ アクセスの許可または拒否 ■ 時間制限またはアクティブルール制限の定義	■ リソース 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

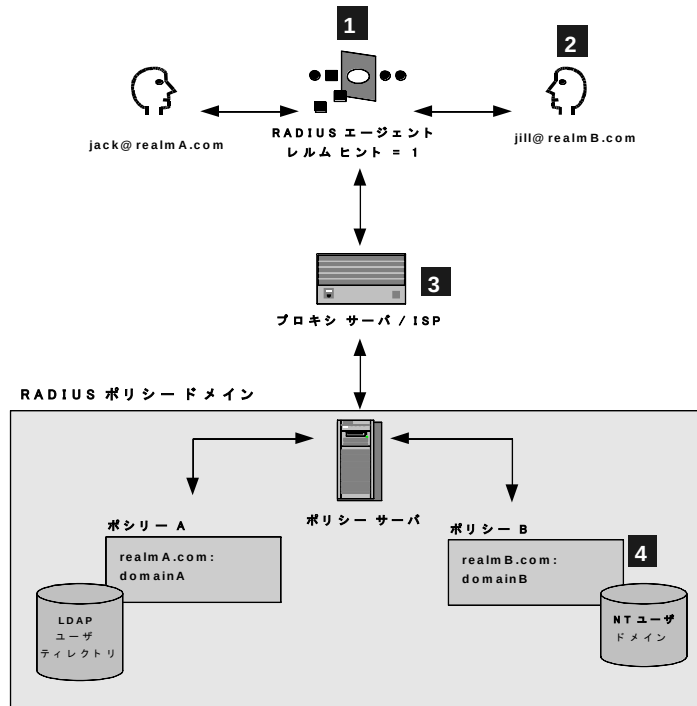
RADIUS エージェントは、異なるドメイン (domainA や domainB など) のユーザを認証する必要がある NAS デバイスをどのような方法で保護するのでしょうか。レルム ヒントは、ユーザを認証する適正なドメインを SiteMinder で判断できるようにする RADIUS 属性です。RADIUS エージェントに、次のいずれかのレルムヒント値を提供する必要があります。

- 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。

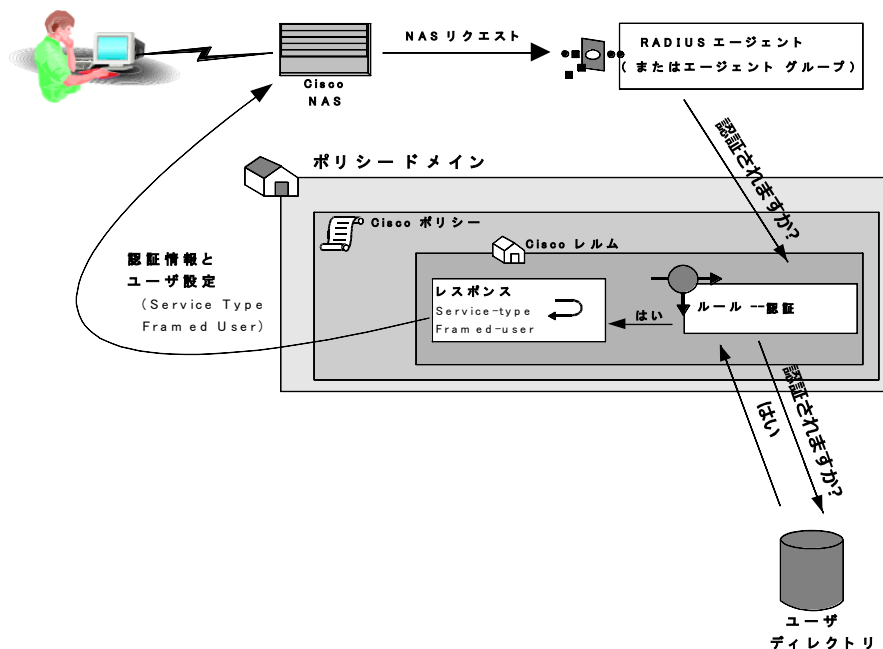
権限をユーザに割り当てるユーザ プロファイル情報を、レスポンスを使用して NAS デバイスに提供することができます。たとえば、同じリソースへのアクセスをあるユーザには制限なしで、また別のユーザには制限ありで許可することができます。本来 RADIUS は認証メカニズムにすぎませんが、レスポンスをこのように使用することで、ユーザに許可を与えることができるようになります。

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. 905)」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ァイル情報を返します。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 AV-pair、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。

詳細情報:

[エージェントタイプの属性の作成](#) (P. 905)

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¥Wow6432Node¥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] ページにあります。

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. [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. [エージェントタイプ] をクリックします。
[エージェントタイプ] ページが表示されます。
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の名前をクリックします。
[エージェントタイプの表示] ページが表示されます。
5. [変更] をクリックします。
設定とコントロールがアクティブになります。

6. [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 タイプ属性タイプの新規オブジェクトの作成] オプションが選択されていることを確認します。
7. [OK] をクリックします。
[エージェント タイプ属性の作成] ページが表示されます。
8. エージェント タイプの名前および説明を入力します。
9. [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。
10. [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。
11. [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 属性の識別子上書きすることはできませんが、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。

12. [RADIUS の動作] 内の各フィールドの RADIUS コードを選択します。
RADIUS コードは次のとおりです。

不可

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 に固有のフィールドは以下のとおりです。

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。

監査のリクエスト

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。

監査のレスポンス

監査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 の監査のレスポンスには、属性が含まれている必要はありません。

13. データ型が数値である場合は、[作成] をクリックします。

14. 対応するフィールドに属性の記号名および数値を入力し [OK] をクリックします。

[エージェント タイプ属性の変更] ページが再表示され、属性名/値ペアが追加されます。

注: 複数の属性名/値ペアを作成するには、手順 11 および 12 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。

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] ページが再表示され、エージェント タイプ属性が追加されます。

16. [サブミット] をクリックします。

属性が選択されたエージェント タイプに対して定義されます。

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. 914)」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](#) (P. 919)」で説明されています。

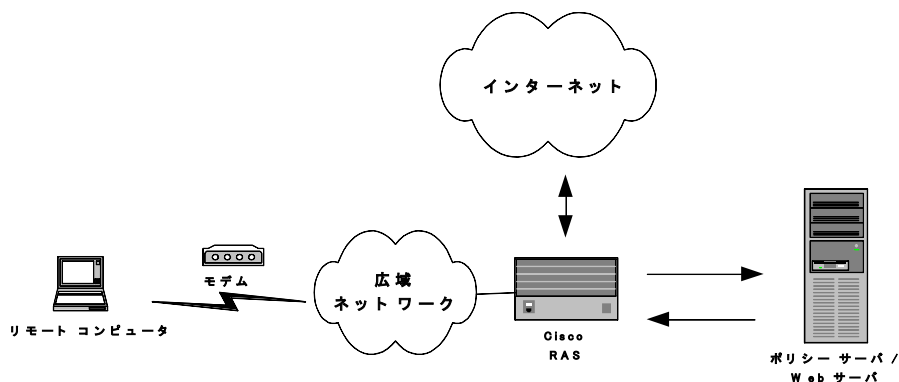
RADIUS デバイスの保護に関するガイドライン

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。



以下の手順に従います。

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」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」の説明に従って、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」の説明に従って、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」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. 「認証イベント アクション用のルールの設定」の説明に従って、RADIUS エージェントによって保護されているレルムに認証されたユーザがアクセスできるようにするルールを定義します。
 - d. 「レスポンスの設定」および「[RADIUS エージェント レスポンス属性 \(P. 636\)](#)」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」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。

詳細情報:

[ポリシー サーバを使用した RADIUS 認証の動作 \(P. 894\)](#)

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-enabled 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

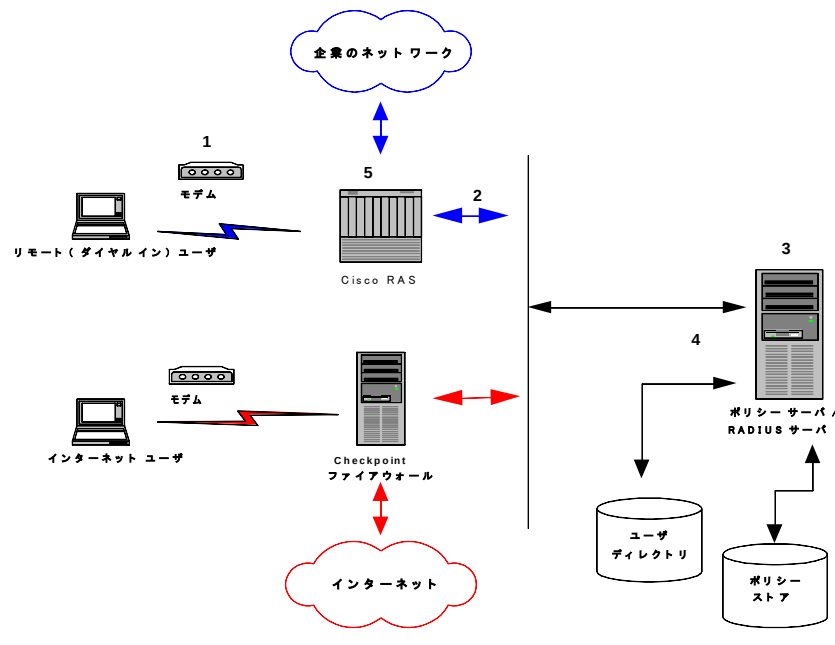
1つのユーザディレクトリを持つ異機種 RADIUS 環境でのユーザの認証

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 トポロジでは、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

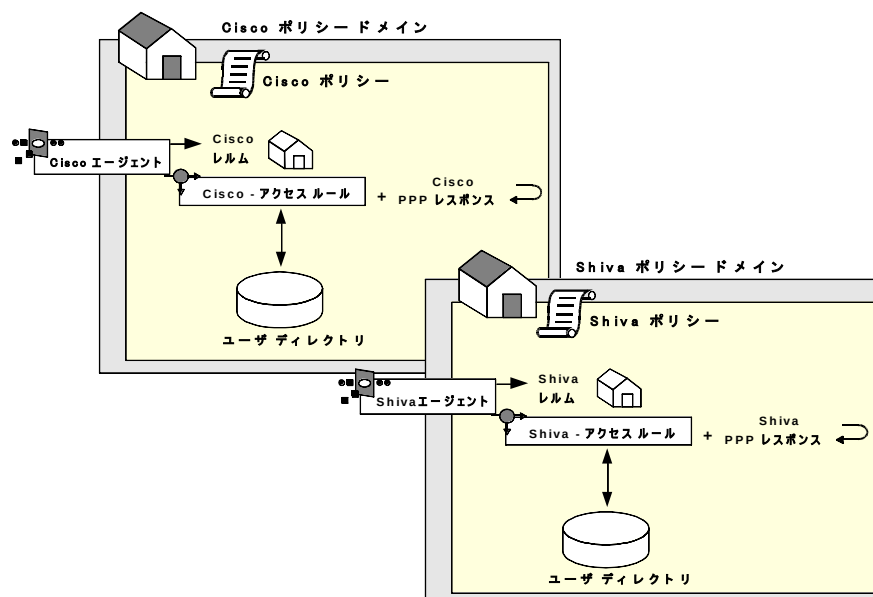
このシステム設定は、同機種環境と異なり、2つのエージェントを作成する必要があります。

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 \(P. 918\)](#)」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 \(P. 918\)](#)」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. ユーザプロファイルを 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つのみの場合、必要なポリシー ドメインは1つのみです。

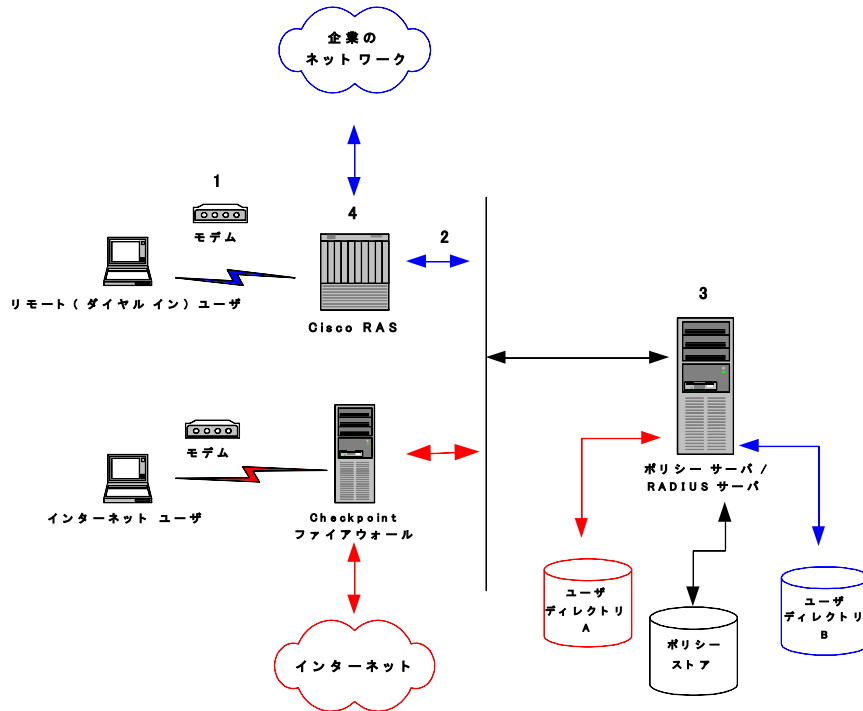
2つのユーザディレクトリを持つ異機種 RADIUS 環境でのユーザ認証方法

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、レルムヒントが必要なくなります。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。

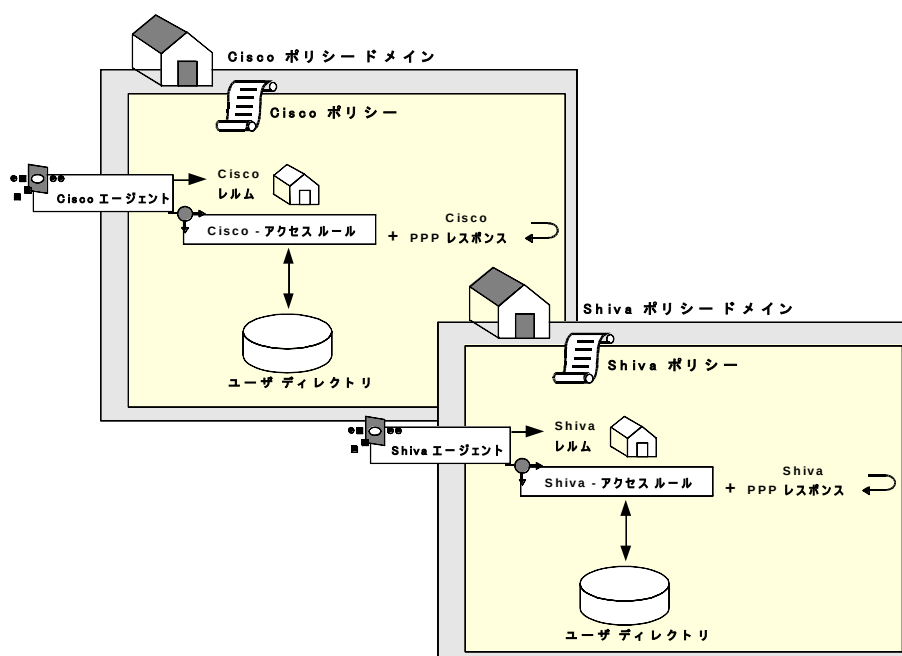
システムとポリシー ドメインを設定する方法

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 \(P. 923\)](#)」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 \(P. 923\)](#)」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 \(P. 923\)](#)」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. ユーザプロファイルを 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。

詳細情報:

[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 \(P. 918\)](#)

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ルムを作成して設定する必要がなくなります。1 つのレルムを使用する場合、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つのレルム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。

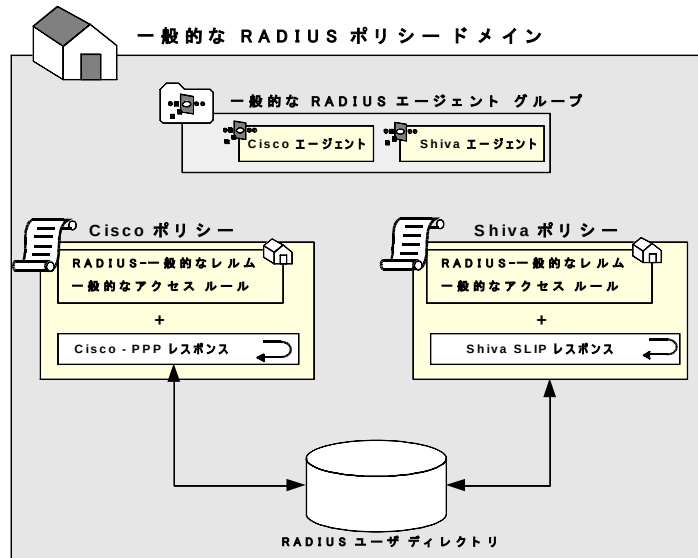
RADIUS エージェント グループのセットアップ

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あります。



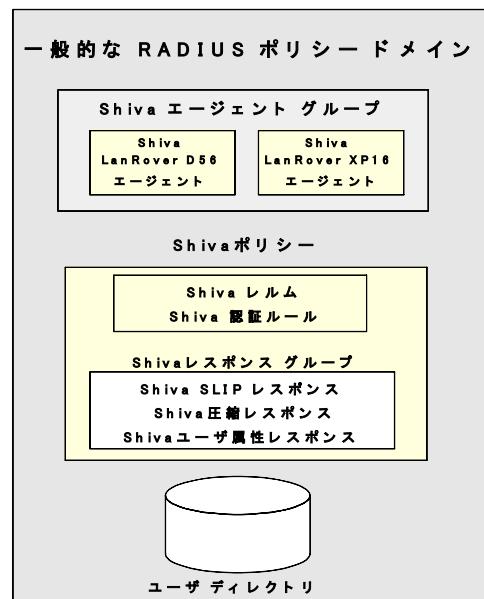
RADIUS エージェント グループをセットアップする手順は、「エージェント グループの設定」にあります。

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。

監査とデバッグのための 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監査情報を追跡します。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、このファイル名に日時が追加されます。

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にのみ更新されます。

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<監査ポート番号>

デフォルトポート (1646) 以外のポートを RADIUS 監査に使用する場合に、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 : <site minder_installation>\Bin

ここで、<site minder_installation> は、SiteMinder がインストールされている場所です。

- UNIX の場合 : <site minder_installation>/bin

ここで、<site minder_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  
-sssecret -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. 929)」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. 930)」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 インターフェイスにアクセスする場合は、管理者権限でショートカットを開きます。管理者としてシステムにログインしている場合でも、管理者権限を使用します。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する前に、管理者権限でコマンドライン ウィンドウを開きます。アカウントに管理者権限がある場合でも、このようにコマンドライン ウィンドウを開きます。
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 E: 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. 931)

[式の構文の概要](#) (P. 934)

[貼り付け](#) (P. 936)

[演算子](#) (P. 938)

[式の中に使用できる関数](#) (P. 956)

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もあります）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もあります）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 インターフェースで定義できます。

```
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、その後に 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。

演算子

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

1 = 2
結果 = FALSE

"sparrow" = "SPARROW"
結果 = FALSE

"sparrow" ~= "SPARROW"
結果 = TRUE

詳細情報:

[不等価演算子](#) (P. 940)

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
```

```
1 != 2  
結果 = TRUE
```

```
"sparrow" != "SPARROW"  
結果 = TRUE
```

```
"sparrow" ~!= "SPARROW"  
結果 = FALSE
```

詳細情報：

[等価演算子](#) (P. 939)

「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

詳細情報：

[「より大きい」演算子](#) (P. 941)

「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

詳細情報:

[「より小さい」演算子](#) (P. 940)

「以下」演算子

「以下」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 (`~<=`) 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**

詳細情報:

[「以上」演算子](#) (P. 943)

「以上」演算子

「以上」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、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 (`~>=`) 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
```

詳細情報：

[「以下」演算子](#) (P. 942)

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

詳細情報：

[末尾演算子](#) (P. 944)

[包含演算子](#) (P. 944)

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

詳細情報：

[先頭演算子](#) (P. 943)

[末尾演算子](#) (P. 944)

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。
¥	¥* 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][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

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 日から経過した秒の数として、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
```

DATEOSTRING 関数 -- 文字列への数値の変換

DATEOSTRING 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

DATEOSTRING 関数の構文は以下のとおりです。

```
DATEOSTRING(date_time[, format_string])
```

パラメータ

DATEOSTRING 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)
%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)

コード	返される日時の形式
%l	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、関数が想定する属性タイプに応じて、文字列または整数を返します。

例 1

```
Virtual User Attribute #GetCN set to RDN( STRING(%0),FALSE)

ENUMERATE(SM_USERGROUPS, #GetCN)
```

例 2

```
ENUMERATE(SM_USERGROUPS, STRING(RDN(%0, FALSE)))
```

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。

注釈

特権： はい

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(*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。

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"
```

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
```

詳細情報：

[KEY 関数 -- キーの検索](#) (P. 993)

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
<code>?</code>	任意の 1 文字と一致します。
<code>*</code>	任意の数（1 または 0 を含む）の文字と一致します。
<code>[set]</code>	指定したセット内の任意の 1 文字と一致します。
<code>[!set]</code> または <code>[^set]</code>	指定したセットにない任意の 1 文字と一致します。
<code>¥</code>	<code>¥*</code> 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。

詳細情報:

[HOUR24 関数 -- 時間への変換 \(P. 992\)](#)

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。

詳細情報:

[HOUR 関数 -- 時間への変換 \(P. 991\)](#)

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'
```

詳細情報:

[EXISTS 関数 -- ファイル名の検索 \(P. 985\)](#)

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(#virtual_user_attribute, start_value, end_value, [step,] )
```

パラメータ

LOOP 関数は、以下のパラメータを受け取ります。

#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。

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。

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(#Padset ,1, 5)
Return_set="001^002^003^004^005"
Return_set=LOOP(#Padset ,1, 5, 2)
Return_set="001^003^005"
Return_set=LOOP(#Padset ,5, 1, -1)
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 '
```

詳細情報:

[LTRIM 関数 -- 文字列の後続スペースの削除](#) (P. 1019)

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

詳細情報:

[MIN 関数 -- 2つのうち、より小さい値の特定](#) (P. 1004)

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す。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
```

詳細情報:

[MAX 関数 -- 2 つのうち、より大きい値の特定 \(P. 1000\)](#)

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
```

詳細情報:

[NOWGMT 関数 -- 秒単位での現在の時間の取得 \(P. 1008\)](#)

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。

詳細情報:

[NOW 関数 -- 秒単位での現在の時間の取得 \(P. 1007\)](#)

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)

LPAP 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
```

詳細情報:

[URL 関数 -- URL 文字列の要素の取得](#) (P. 1028)

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'
```

詳細情報:

[LTRIM 関数 -- 文字列の先頭にあるスペースの削除](#) (P. 1000)

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。

詳細情報:

[MINUTE 関数 -- 日付の「分」要素の取得 \(P. 1005\)](#)

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))`

詳細情報:

[GET 関数 -- ユーザ ディレクトリでの属性の検索 \(P. 990\)](#)

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。

注釈

特権： はい

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/xyzzy.jsp?id=12",  
"PROTOCOL")
```

```
Return_value="http:"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzzy.zip", "USER")
```

```
Return_value="joe"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip",  
"PASSWORD")  
Return_value="dog"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzyy.jsp?id=12",  
"SERVER")  
Return_value="www.myserver.com"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzyy.jsp?id=12",  
"PORT")  
Return_value="8080"
```

```
Return_value=URL("http://www.myserver.com:8080/app1/xyzyy.jsp?id=12",  
"DOMAIN")  
Return_value="myserver.com"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip", "URI")  
Return_value="/dir1/xyzyy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip", "PATH")  
Return_value="/dir1/xyzyy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip",  
"DIRECTORY")  
Return_value="/dir1"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip",  
"FILENAME")  
Return_value="xyzyy.zip"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.zip",  
"BASENAME")  
Return_value="xyzyy"
```

```
Return_value=URL("ftp://joe:dog@ftp.myserver.com/dir1/xyzyy.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"
```

詳細情報:

[URLENCODE 関数 -- 文字列のエンコード](#) (P. 1032)

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'
```

詳細情報:

[URLDECODE 関数 --URL 文字列のデコード](#) (P. 1031)

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
```

WARNING 関数 -- コンソール ログへの WARNING メッセージの書き込み

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。