SiteMinder フェデレーション

パートナーシップ フェデレーション ガイド

12.52 SP1

ca technologies

CA Technologies 製品リファレンス

このマニュアルが参照している CA Technologies の製品は以下のとおりです。

- SiteMinder

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。

マニュアルの変更点

SiteMinder の旧リリースで発見された問題の結果として、12.52 のドキュメントに更新が行われませんでした。

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。

- セッションを確立するための認証 URL の保護 (P. 27) -「IdP パートナーの設定」の下の新しいサブセクションです。この手順では、セッションを確立するための認証 URL の保護方法を示します。
- 認証 URL の保護によるセッションの要求 (P. 67) - セッションを確立するための認証 URL の保護の要件を詳細に示す新しいセクションです。この手順は、アサーティング パーティでの設定に必要です。

目次

第 1 章: パートナーシップ フェデレーションの概要

このセクションには、以下のトピックが含まれています。

製品および設定の概要

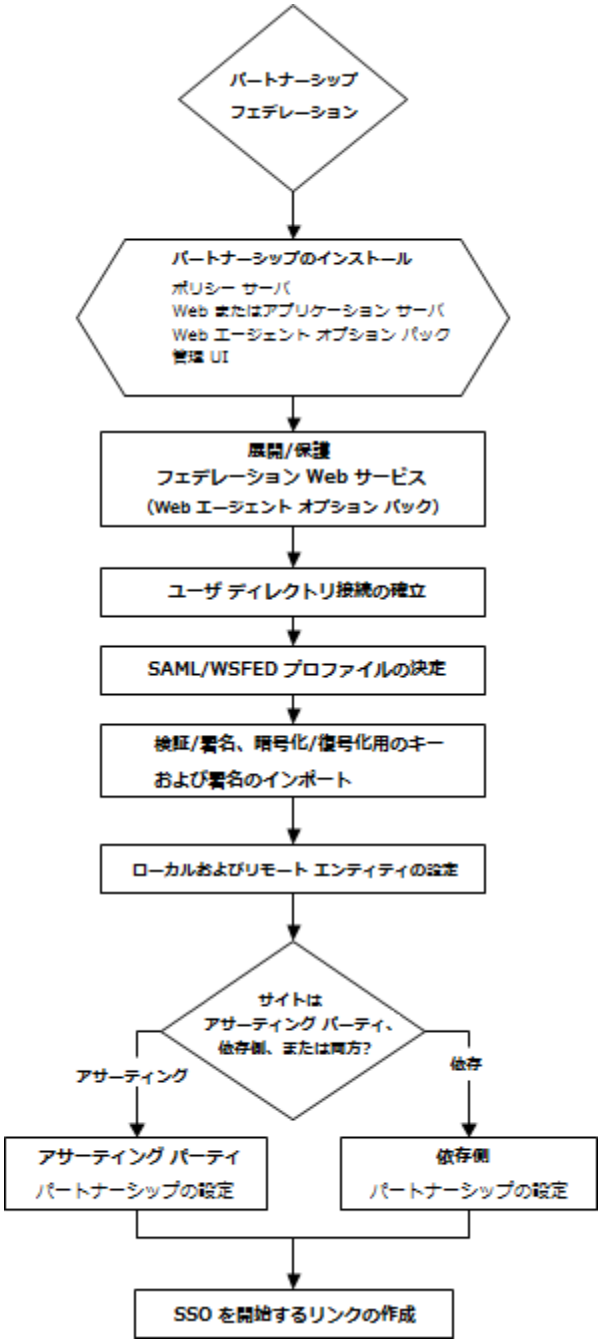
フェデレーション パートナーシップによって、識別情報を柔軟かつポータブルにすることができます。 パートナーシップ フェデレーションは、信頼できるビジネス パートナーのネットワーク全体で安全なシングル サインオンおよびシングル ログアウトを提供します。

SiteMinder パートナーシップ フェデレーション により、顧客は柔軟な方法でフェデレーション パートナーシップを Web アクセス管理システムと共に、または単独で確立できます。パートナーシップ フェデレーションは、標準ベースのフェデレーションの展開しやすいソリューションを提供します。 パートナーシップ フェデレーション を使用すると、組織はアサーティング パーティまたは依存パーティとして機能できます。 アサーティング パーティは、ユーザ認証および ID のアサーションを提供します。 依存パーティは、ユーザ ID を使用して Web リソースおよびサービスへのアクセスを許可します。

パートナーシップ フェデレーションでは以下のプロファイルをサポートします。

- SAML 1.1
- SAML 2.0
- WS-フェデレーション

以下のフローチャートは、パートナーシップ フェデレーション を設定する一般的なプロセスに焦点を当てています。



プログラマなしフェデレーション

プログラマなしフェデレーションは、安全な認証、ユーザの特定、検査および SAML アサーションの変更を可能にする HTTP ベースの方法です。 プログラマなしフェデレーションの利点は、アプリケーションがこれらのタスクを実行するために、言語固有の SDK または他のバインドを使用する必要がないということです。

プログラマなしフェデレーションは、HTTP/HTTPS のリクエストおよびレスポンスに依存します。これらのリクエストおよびレスポンスには、REST（Representational State Transfer）システム アーキテクチャ実装の Web サービスを使用して URL および HTML ベースのプロトコルでアクセスできます。

すべてのアプリケーションは HTTP リクエストの発行、HTTP レスポンスの読み取りが可能で、XML を解析してプログラマなし機能を利用できます。

プログラマなしフェデレーションで最も重要なのは、安全にデータを交換する機能です。 データを保護するために、SiteMinder はオープン形式の Cookie を使用します。 オープン形式の Cookie とは、強力な暗号化アルゴリズムをサポートする明確に定義された Cookie フォーマットのことです。暗号化された Cookie が、SiteMinder とローカル アプリケーションまたはリモート アプリケーションの間のレスポンスを保護します。この Cookie は、Perl または Ruby などのオープン形式の Cookie によってサポートされている同じ暗号化および復号のアルゴリズムをサポートするすべてのプログラミング言語で記述できます。

以下の パートナーシップ フェデレーション 機能はプログラマなしフェデレーションを実装しています。

分散代行認証

分散代行認証によって、SiteMinder はサードパーティ Web アクセス管理（WAM）システムを使用して、保護されているフェデレーション リソースを要求するすべてのユーザの認証を実行できます。 サードパーティ WAM は認証を実行して、SiteMinder にフェデレーション ユーザ ID を送信します。

HTTP/HTTPS のリクエストおよびレスポンスによって、プロビジョニング用の通信が容易になります。

依存パーティでのプロビジョニング

プロビジョニングとは、データおよびアプリケーションにアクセスするために必要なアカウント権限およびアクセス権限を持つクライアント アカウントを作成するプロセスのことです。 パートナーシップ フェデレーションのプロビジョニングによって、ユーザの新規アカウントを作成したり、SAML アサーションで送信される情報を既存のユーザ アカウントに登録したりできます。

リモート プロビジョニングは、SiteMinder のプロビジョニング方法の 1 つです。 リモート プロビジョニングでは、自律型プロビジョニング アプリケーションを使用してユーザ レコードを作成します。 アサーション データを渡すために、SiteMinder は、そのデータを含む暗号化された Cookie を作成します。 この Cookie は、ユーザ アカウントを作成するリモート プロビジョニング アプリケーションに送信されます。

HTTP/HTTPS のリクエストおよびレスポンスによって、プロビジョニング用の通信が容易になります。

対象読者

本書では、読者が以下の概念について理解していることを前提としています。

- 基本的な SAML および WS-フェデレーションの基礎
- フェデレーション バインディング。
- SSO （シングル サインオン）、SLO （シングル ログアウト）、およびシングル サインアウトなどのフェデレーション プロファイル
- 公開キー インフラストラクチャ（PKI）の基礎
- Secure Socket Layer 通信の基本

本書で使用される用語

連携 SAML および WS-フェデレーションのバインディングおよびプロファイルに関する標準的な用語に加えて、以下の用語がこのガイドでは使用されます。

パートナー エンティティの用語

このガイドでは、フェデレーション関係の両側を区別するために*アサーティング パーティ*と*依存するパーティ*という用語を使用しています。

アサーションを生成するパーティをアサーティング パーティと呼びます。アサーティング パーティとして機能できるパートナーは以下のとおりです。

- SAML 1.x プロデューサ
- SAML 2.0 ID プロバイダ（IdP）
- WS フェデレーション ID プロバイダ（IP）

認証の目的でアサーションを消費するパーティを依存するパーティと呼びます。 依存するパーティとして機能できるパートナーは以下のとおりです。

- SAML 1.x コンシューマ
- SAML 2.0 サービス プロバイダ（SP）
- WS フェデレーション リソース パートナー（RP）

1 つのサイトがアサーティング パーティ（プロデューサ/IdP/IP）および依存するパーティ（コンシューマ/SP/RP）として機能できます。

オープン形式 Cookie

ユーザ識別情報を含む Cookie。 FIPS 準拠または非 FIPS 準拠アルゴリズムを使用して、オープン形式の Cookie を生成方法に応じて暗号化できます。 CA SiteMinder® Federation SDK を使用してオープン形式の Cookie を作成できます。 または、UTF-8 エンコーディングをサポートしているプログラミング言語を使用して手動で作成できます。

FIPS で暗号化されたオープン形式の Cookie が必要な場合は、SDK を使用して Cookie の作成および読み取りを行います。 CA SiteMinder® Federation Java SDK では、FIPS 準拠（AES）アルゴリズムまたは非 FIPS 準拠（PBE）アルゴリズムを使用して Cookie を暗号化できます。 CA SiteMinder® Federation.NET SDK で Cookie を暗号化する場合は、FIPS 準拠のアルゴリズムのみを使用できます。

統一表現言語

統一表現言語（UEL）とは、主に Java Web アプリケーション用の特殊な Java の式構文のことです。Web ページに式を埋め込むために UEL を使用できます。パートナーシップ フェデレーション の場合、UEL は依存パーティでアサーション属性とアプリケーション属性間のマッピングを定義するために必要な言語です。

[パートナーシップ フェデレーション]ダイアログ ボックスのナビゲート

管理 UI は、パートナーシップ フェデレーション オブジェクトを作成および変更するための設定ウィザードを提供します。 設定ウィザードの手順に従って、オブジェクトの設定手順を進めます。

第 2 章: パートナーシップ フェデレーションの前提条件

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 アサーティング パートナーの前提条件

SiteMinder がアサーティング パートナーとして機能するには、以下の条件を確認する必要があります。

- ポリシー サーバがインストールされています。
- Web エージェントおよび Web エージェント オプション パックがインストールされています。 Web エージェントは、ユーザを認証し、SiteMinder セッションを確立します。 オプション パックはフェデレーション Web サービス アプリケーションを提供します。 ネットワーク内の適切なシステムに必ず FWS アプリケーションを展開してください。

  詳細については、「Web エージェント オプション パック ガイド」を参照してください。
- 秘密キーと証明書は、メッセージの署名および復号を必要とする機能に対してインポートされます。
- SQL クエリ方式および有効な SQL クエリを設定してから、パートナーシップのユーザ ディレクトリとして ODBC データベースを選択します。この前提条件が必要なのは、ODBC を使用する場合のみです。
- 依存パートナーは、フェデレーション ネットワーク内でセットアップされます。

SiteMinder 依存パートナーの前提条件

依存パートナーとして機能する SiteMinder については、以下の要件を満たすようにしてください。

- ポリシー サーバがインストールされています。
- Web エージェントおよび Web エージェント オプション パック。 Web エージェントは、ユーザを認証し、SiteMinder セッションを確立します。 オプション パックはフェデレーション Web サービス アプリケーションを提供します。ネットワーク内の適切なシステムに必ず FWS アプリケーションを展開してください。

  詳細については、「Web エージェント オプション パック ガイド」を参照してください。
- 検証およびメッセージの暗号化を必要とする機能のために秘密キーと証明書をインポートします。
- フェデレーション ネットワーク内でアサーティング パートナーをセットアップします。

第 3 章: 簡単なパートナーシップの概要

このセクションには、以下のトピックが含まれています。

Basic SAML 2.0 パートナーシップ

パートナーシップ フェデレーション を開始する 1 つの方法として、パートナーシップの設定があります。 この章では、基本的な SAML 2.0 フェデレーション パートナーシップをセットアップする方法について説明します（SAML 2.0 POST プロファイルによるシングル サインオン）。 基本的な設定から始めることによって、最少の手順で パートナーシップ フェデレーション がどのように機能するかを確認することができます。

注: このパートナーシップでは SAML 2.0 に焦点を当てていますが、全体的なプロセスは SAML 1.1 に共通しています。 パートナーシップの各手順での設定は、SAML プロトコルによって異なる場合があります。

本章では、実際の実稼働環境を反映するデジタル署名およびシングル ログアウトなどの追加機能の設定についても説明します。 Artifact バインディングを設定に追加することもできます。

この章で使用されるサンプル ネットワークは、SiteMinder がパートナーシップの両サイトでインストールされていることを前提としています。 ただし、一方のサイトで SiteMinder、およびもう一方のサイトで別の SAML 対応製品がインストールされている場合も、パートナーシップを始めることができます。

両サイトの SiteMinder で、パートナーシップの設定による見通しを理解しておく必要があります。 完全なパートナーシップを設定するには、指定されたサイトから通信する各方向のそれぞれのサイトで、*パートナーシップ定義*を定義することから始めます。 たとえば、ローカル サイトがアイデンティティ プロバイダ（IdP）である場合、ローカル IdP からリモート SP へのパートナーシップを設定します。 この設定はパートナーシップの一定義です。 パートナーシップ設定を完了するには、ローカル SP でローカル SP からリモート IdP への相互的なパートナーシップを設定します。

パートナーシップ定義では、必ずローカル エンティティとリモート エンティティを区別します。 ローカル エンティティとは、パートナーシップ フェデレーション の設定元サイトのエンティティです。 この環境は、必ずしも SiteMinder がインストールされている環境と同じではありませんが、ドメインは同じです。 リモート エンティティとは、パートナーシップ フェデレーション の設定元の別のドメインにあるパートナーのエンティティです。

以下のプロセスでは、SiteMinder が両方のサイトにある場合に基本的なパートナーシップを作成するための手順を示します。

1. ユーザ ディレクトリ接続を確立します。
2. セッションを確立するための認証 URL を保護します。
3. ローカル エンティティおよびリモート エンティティを作成します。
4. IdP でローカル IdP から SP へのパートナーシップ定義を設定します。
5. SP でローカル SP から IdP へのパートナーシップ定義を設定します。
6. パートナーシップをアクティブにします。
7. パートナーシップをテストします。

サンプル フェデレーション ネットワーク

作成する最初のパートナーシップは以下のサンプル ネットワークを表します。 手順およびサンプル ネットワークの URL は例であり、実際のサイトを指定しません。

ビジネス パートナー

- IdP1 というアイデンティティ プロバイダ
- SP1 というサービス プロバイダ

SAML プロファイルおよび機能

- POST プロファイルを含む SAML 2.0
- シングル サインオン
- 署名処理なし
- FIPS_COMPAT モード

IdP の SSO サービス URL

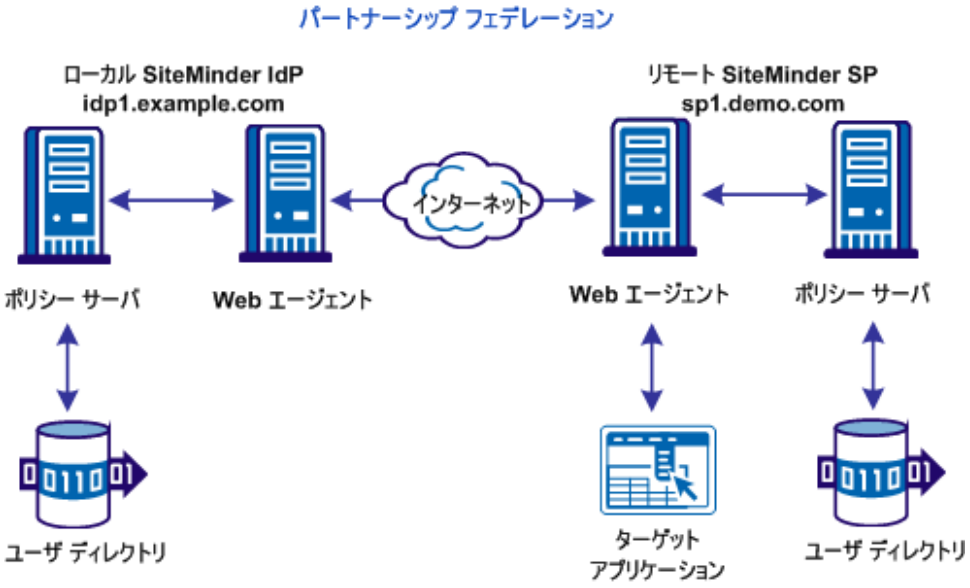
http://idp1.example.com:9090/affwebservices/public/saml2sso

SP のアサーション コンシューマ サービス URL

http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

注: このサンプル ネットワークを実装するためには、SiteMinder がインストールされた 2 つのシステムが必要です。

以下の図では、両方のパートナーに SiteMinder があるサンプル パートナーシップを示します。



必須コンポーネントのインストール確認

パートナーシップ フェデレーション を使用するには、以下のコンポーネントが必要です。

- ポリシー サーバ
- 管理 UI
- Web エージェント
- 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 パックには、フェデレーション Web サービス（FWS）アプリケーションが含まれています。 FWS は、フェデレーションの必須コンポーネントです。

  Web エージェント オプション パックのインストール、および FWS の展開方法については、「*Web エージェント オプション パック ガイド*」を参照してください。

この簡単なパートナーシップの展開例では、これらのコンポーネントがインストール済みで機能していることを前提としています。

IdP パートナーの設定

以下は、IdP1 の管理者から見た設定プロセスです。したがって IdP1 はローカル IdP です。

以下のプロセスによって IdP パートナーを確立します。

1. 管理 UI にログインします。
2. ユーザ ディレクトリ接続を確立します。
3. IdP エンティティおよび SP エンティティを識別します。
4. IdP から SP への SAML2 パートナーシップを作成します。
5. パートナーシップ ウィザードに従い、最低限必要な設定を行います。

IdP でのユーザ ディレクトリ接続の確立

ユーザ ディレクトリへの接続を定義した後でパートナーシップを確立できます。 IdP ユーザ ディレクトリは、アイデンティティ プロバイダがアサーションを生成する対象のユーザ レコードから構成されます。

以下の手順では、管理 UI でユーザ ディレクトリを設定する方法を説明します。IdP LDAP という名前のディレクトリにはユーザ 1 およびユーザ 2 が含まれます。

次の手順に従ってください:

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。

   ［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。

4. 以下のフィールドに値を入力します。

   名前

   IdP LDAP

   ネームスペース

   LDAP

   サーバ

   www.idp.demo:42088

5. ［LDAP 設定］セクションの以下のフィールドに入力します。

   ルート

   dc=idp,dc=demo

   その他の値はデフォルトのままにします。

   ［LDAP ユーザ DN の検索］の以下のフィールドに入力します。

   Start

   uid=

   End キー

   ,ou=People,dc=idp,dc=demo

6. ［内容の表示］をクリックして、ディレクトリの内容を表示できることを確認します。

7. ［サブミット］をクリックします。

セッションを確立するための認証 URL の保護

ポリシー サーバがアサーションを生成するには、ユーザは IdP ポリシー サーバでセッションを持つ必要があります。 セッションを確立するには、ユーザに認証チャレンジが提示されるように、ポリシーを使用して認証 URL を保護します。 これにより、ユーザがログインしてセッションが確立されます。

次の手順に従ってください:

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］-［エージェントの作成］を選択します。

   Agent1 という名前の Web エージェントを作成します。
3. ［ポリシー］-［ドメイン］-［ドメイン］-［ドメインの作成］を選択します。

   認証 URL 用のポリシー ドメインを作成します。チャレンジを要求されるユーザが含まれるユーザ ディレクトリを追加します。
4. ポリシー ドメインに含まれるリソースへのアクセス権が必要なユーザを選択します。
5. ［レルム］タブを選択し、次の値を使用してポリシー ドメインのレルムを定義します。

   エージェント

   Agent1

   リソース フィルタ

   /affwebservices/redirectjsp

   デフォルト リソース保護

   保護

   認証方式

   基本

   永続セッション

   HTTP-Artifact プロファイルのレルム ダイアログ ボックスの［セッション］セクションにある［永続］チェック ボックスをオンにして、セッション情報を格納します。セッション情報は、シングル ログアウトなどの機能、および属性機関に必要です。

6. レルム ダイアログ ボックスの［ルール］セクションで、［ルールの作成］をクリックします。 フィールドに以下の値を入力します。

   リソース

   /*

   アスタリスクは、ルールがレルム内のすべてのリソースに適用されることを意味します。

   許可/拒否と有効/無効

   アクセスを許可する

   ［有効］チェック ボックスはオン。

   アクション

   Web エージェント アクション

   Get、Post、Put

7. ［ポリシー］タブを選択し、次のコンポーネントが含まれるポリシーを作成します。
   - ユーザ ディレクトリで選択したユーザのセット。
   - redirectjsp アプリケーションおよび関連ルールが含まれるレルム。

これで、ポリシーにより認証 URL が保護されます。

パートナーシップ エンティティの設定

ユーザ ディレクトリ接続を確立した後で、パートナーシップの両側を識別します。 管理 UI では、パートナーはそれぞれエンティティと呼ばれます。

以下の手順では、ローカル エンティティおよびリモート エンティティに必要な値について説明します。実際のネットワーク設定では、両方でローカル エンティティを作成してメタデータ ファイルにエクスポートし、ファイルを交換することができます。 その後、両方でリモート エンティティを定義できます。

ローカル IdP を作成する方法

1. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
2. ［フェデレーション エンティティ リスト］で［エンティティの作成］をクリックします。
3. エンティティ ウィザードの最初の手順で、以下の選択を行ってから［次へ］をクリックします。

   エンティティ ロケーション

   ローカル

   新しいエンティティ タイプ（New Entity Type）

   SAML2 IDP

4. ウィザードの 2 番目の手順で、以下のフィールドに入力してから［次へ］をクリックします。

   エンティティ ID

   idp1

   この値によって、パートナーに対してエンティティが識別されます。

   エンティティ名

   idp1

   この値によって、エンティティ オブジェクトがデータベースで内部的に識別されます。 パートナーはこの値を認識しません。

   ベース URL

   http://idp1.example.com:9090

   他の設定はそのまま残します。

   注: ［エンティティ名］は［エンティティ ID］と同じ値にすることができます。 ただし、値をサイトの他のエンティティとは共有しないでください。

5. 最後の手順で設定を確認し、［完了］をクリックします。

［エンティティ］ウィンドウに戻ります。

SP エンティティを作成する方法

1. ［エンティティ］ウィンドウから始めます。
2. ［フェデレーション エンティティ リスト］で［エンティティの作成］をクリックします。

   ［エンティティの作成］ダイアログ ボックスが表示されます。

3. エンティティ ウィザードの最初の手順で、以下の選択を行ってから［次へ］をクリックします。

   エンティティ ロケーション

   リモート

   新しいエンティティ タイプ（New Entity Type）

   SAML2 SP

4. ウィザードの 2 番目の手順で、以下のようにフィールドに入力してから［次へ］をクリックします。

   エンティティ ID

   sp1

   この値によって、パートナーに対してエンティティが識別されます。

   エンティティ名

   sp1

   この値によって、エンティティ オブジェクトがデータベースで内部的に識別されます。 パートナーはこの値を認識しません。

   アサーション コンシューマ サービス URL

   インデックス

   0

   バインディング

   HTTP-Post

   URL

   http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

   デフォルト

   エントリのチェック ボックスをオンにします。

   他の設定はそのまま残します。

5. 最後の手順で設定を確認し、［完了］をクリックします。

リモート SP エンティティが設定されました。

ローカル エンティティおよびリモート エンティティを設定した後で、パートナーシップを作成します。

IdP から SP へのパートナーシップの作成

フェデレーション エンティティの作成後に、パートナーシップ ウィザードに従って IdP から SP へのパートナーシップを設定します。 ウィザードは基本的なパートナーシップ パラメータから始まります。

次の手順に従ってください:

1. ［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］を選択します。
2. ［パートナーシップの作成］をクリックします。
3. ［SAML2 IdP -> SP］を選択します。

   このオプションを選択することで、ユーザがローカル IdP であることを示します。

   パートナーシップ ウィザードの最初の手順を始めます。
4. フィールドに以下の値を入力します。

   パートナーシップ名

   TestPartnership

   ローカル IDP ID

   idp1

   （プルダウン リストから選択）

   リモート SP ID

   sp1

   （プルダウン リストから選択）

   ベース URL

   http://idp1.example.com:9090

   スキュー時間（秒）

   デフォルトを受け入れる
5. IDP LDAP ディレクトリを［使用可能なディレクトリ］リストから［選択されたディレクトリ］リストに移動します。
6. ［次へ］をクリックして［フェデレーション ユーザ］手順に進みます。

アサーション生成用のフェデレーション ユーザの指定

［フェデレーション ユーザ］ダイアログ ボックスで、IdP によってアサーションを生成されるユーザを選択します。

次の手順に従ってください:

1. デフォルトを受け入れます。
2. ［次へ］をクリックして続行します。

デフォルトを受け入れることによって、SiteMinder がユーザ ディレクトリのすべてのユーザのアサーションを生成できることを示します。

アサーションへの名前 ID の追加

［アサーションの設定］手順では、名前 ID のフォーマットと値、およびユーザを識別する属性を指定できます。 これらの属性はアサーションに含まれています。

注: 名前 ID は必ずアサーションに含まれています。

この設定では、［名前 ID］のみを指定します。 他の属性を追加しないでください。

次の手順に従ってください:

1. ［アサーションの設定］手順から、以下のフィールドの値を入力します。

   名前 ID 形式

   未指定

   名前 ID タイプ

   静的

   値

   GeorgeC

2. ［次へ］をクリックして続行し、シングル サインオン（SSO）をセットアップします。

IdP でのシングル サインオンのセットアップ

パートナー間のシングル サインオンを確立するには、SSO 設定を行います。

次の手順に従ってください:

1. パートナーシップ ウィザードの［SSO と SLO］手順から始めます。
2. ［認証］セクションで以下のエントリを指定します。

   認証モード

   ローカル

   認証 URL

   http://webserver1.example.com/affwebservices/redirectjsp/redirect.jsp

   この例では、webserver1 によって Web エージェント オプション パックを持つ Web サーバが識別されます。redirect.jsp ファイルは、アイデンティティ プロバイダ サイトでインストールされた Web エージェント オプション パックに含まれています。

   重要: アクセス制御ポリシーで認証 URL を保護します。

   AuthnContext の設定

   デフォルトを受け入れる

   認証クラス

   デフォルトを受け入れる

3. ［SSO］セクションで以下のエントリを指定します。

   SSO バインディング

   HTTP-POST

   アサーション コンシューマ URL

   http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

4. ［次へ］をクリックして［署名および暗号化］手順に移動します。

署名の処理を無効にする

この簡単なパートナーシップでは、署名処理を無効にします。 ただし、実稼働環境では、アイデンティティ プロバイダはアサーションに署名する必要があります。

次の手順に従ってください:

1. ［署名および暗号化］手順から、［署名の処理を無効にする］を選択します。
2. ［次へ］をクリックして次の手順に移動します。

IdP から SP へのパートナーシップ設定の確認

フェデレーション パートナーシップの一方に対するパートナーシップ定義が完了しました。 設定を確認します。

次の手順に従ってください:

1. ［確認］ダイアログ ボックスでパートナーシップの設定を確認します。
2. 設定を変更するには、いずれかのセクションで［変更］をクリックします。
3. 設定が終了したら、［完了］をクリックします。

パートナーシップの IdP 側が完了しました。 IdP システムとは異なるシステム上でパートナーシップの SP 側を定義します。

SP パートナーの設定

以下は、SP（この例では SP1）の管理者から見た設定プロセスです。 したがって SP1 はローカル SP です。

以下のプロセスによって SP パートナーを確立します。

1. 管理 UI にログインします。
2. ユーザ ディレクトリ接続を確立します。

3. IdP エンティティおよび SP エンティティを識別します。
4. SP から IdP への SAML2 パートナーシップを作成します。
5. パートナーシップ ウィザードに従い、最低限必要な設定を行います。

SP でのユーザ ディレクトリ接続の確立

SP ユーザ ディレクトリは、サービス プロバイダが認証に使用するユーザ レコードで構成されます。 以下の手順では、管理 UI でユーザ ディレクトリを設定する方法を説明します。 SP LDAP という名前のディレクトリにはユーザ 1 およびユーザ 2 が含まれます。

ユーザ ディレクトリを設定する方法

1. 管理 UI にログインします。
2. ［インフラストラクチャ］-［ディレクトリ］-［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。

   ［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。
4. 以下のフィールドに値を入力します。

   名前

   SP LDAP
5. ［ディレクトリのセットアップ］セクションで、以下のフィールドに入力します。

   ネームスペース

   LDAP

   サーバ

   www.sp.demo:32941
6. ［LDAP 検索］セクションで、以下のフィールドに入力します。

   ルート

   dc=sp,dc=demo

   その他の値はデフォルトのままにします。

7. ［LDAP ユーザ DN の検索］セクションで、以下のフィールドに入力します。

   Start

   uid=

   End キー

   ,ou=People,dc=sp,dc=demo

8. ［内容の表示］をクリックして、ディレクトリの内容を表示できることを確認します。
9. ［サブミット］をクリックします。

パートナーシップ エンティティの識別

ユーザ ディレクトリ接続を確立した後で、パートナーシップのローカル側およびリモート側を識別します。 管理 UI では、パートナーはそれぞれエンティティと呼ばれます。

以下の手順では、ローカル エンティティおよびリモート エンティティに必要な値について説明します。 通常、両方でローカル エンティティを作成してメタデータ ファイルにエクスポートし、ファイルを交換します。その後、両方でリモート エンティティを定義できます。

ローカル SP を作成する方法

1. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
2. ［エンティティの作成］をクリックします。
3. エンティティ ウィザードの最初の手順で、以下の選択を行ってから［次へ］をクリックします。

   エンティティ ロケーション

   ローカル

   新しいエンティティ タイプ（New Entity Type）

   SAML2 SP

4. 2 番目の手順で、以下のようにフィールドに入力してから［次へ］をクリックします。

   エンティティ ID

   sp1

   この値によって、パートナーに対してエンティティが識別されます。

   エンティティ名

   sp1

   この値によって、エンティティ オブジェクトがデータベースで内部的に識別されます。 パートナーはこの値を認識しません。

   ベース URL

   http://sp1.demo.com:9091

   注: エンティティ ID およびエンティティ名は、アイデンティティ プロバイダでリモート SP エンティティに対して指定したものと同じである必要があります。

5. 設定を確認して［完了］をクリックします。

［エンティティ］ウィンドウに戻ります。 リモート パートナーを設定します。

リモート IdP を作成する方法

1. ［エンティティ］ウィンドウから始めます。
2. ［エンティティの作成］をクリックします。
3. エンティティ ウィザードの最初の手順で、以下の選択を行ってから［次へ］をクリックします。

   エンティティ ロケーション

   リモート

   新しいエンティティ タイプ（New Entity Type）

   SAML2 IDP

4. ウィザードの 2 番目の手順で以下のようにフィールドに入力します。

   エンティティ ID

   idp1

   この値によって、パートナーに対してエンティティが識別されます。

   エンティティ名

   idp1

   この値によって、エンティティ オブジェクトがデータベースで内部的に識別されます。 パートナーはこの値を認識しません。

   注: エンティティ ID およびエンティティ名はアイデンティティ プロバイダ側と同じである必要があります。

   ［SSO サービス URL グループ］セクション

   バインディング

   HTTP リダイレクト

   URL

   http://idp1.example.com:9090/affwebservices/public/saml2sso

5. 設定を確認して［完了］をクリックします。

ローカル エンティティおよびリモート エンティティの設定後に、パートナーシップを作成できます。

SP から IdP へのパートナーシップの作成

パートナーシップ エンティティを作成したら、パートナーシップ ウィザードに従って SP から IdP へのパートナーシップを設定します。

次の手順に従ってください:

1. ［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］を選択します。
2. ［パートナーシップの作成］をクリックします。
3. ［SAML2 SP->IdP］を選択します。

   パートナーシップ ウィザードの最初の手順を始めます。

4. フィールドに以下の値を入力します。

   パートナーシップ名

   DemoPartnership

   ローカル SP ID

   sp1

   リモート IDP ID

   idp1

   ベース URL

   http://sp1.demo.com:9091

   スキュー時間（秒）

   デフォルトを受け入れる

5. SP LDAP ディレクトリを［使用可能なディレクトリ］から［選択されたディレクトリ］に移動します。
6. ［次へ］をクリックして［ユーザ識別］手順に進みます。

ユーザ識別属性の指定

ユーザを識別するアサーションの属性を指定します。 SiteMinder は ID 属性値を使用して、SP でユーザ ディレクトリ内のユーザ レコードを検索します。

ユーザ識別属性を指定する方法

1. ［ユーザ識別］手順に移動します。
2. ［アサーションからのアイデンティティ属性の選択］で、デフォルトの［名前 ID を使用］をそのまま使用します。

3. ［アイデンティティ属性のユーザ ディレクトリへのマップ］セクションで、以下のエントリを指定します。

   LDAP 検索仕様

   uid=%s

   このエントリによって、SiteMinder は変数（%s）をアサーションの［名前 ID］属性の値に置換します。 その後、SiteMinder はその値をサンプル ユーザ データベースの［名前］列と一致させます。 一致させると、ユーザは明確化され、ターゲット リソースへのアクセスを許可されます。

4. ［フェデレーション ユーザ］セクションではデフォルトを受け入れます。 ユーザ ディレクトリ内のすべてのユーザはフェデレーション ユーザであるとみなされます。

5. ［次へ］をクリックしてシングル サインオンを設定します。

SP でのシングル サインオンの設定

パートナー間のシングル サインオンを確立するには、SSO 設定を行います。

次の手順に従ってください:

1. ［SSO と SLO］手順から始めます。
2. ［SSO プロファイル］の［HTTP-POST］を選択します。
3. ［リモート SSO サービス URL］セクションで以下の値を指定します。

   バインディング

   HTTP リダイレクト

   URL

   http://idp1.example.com:9090/affwebservices/public/saml2sso

4. ［署名および暗号化］手順に到達するまで［次へ］をクリックします。

   ［AuthnContext の設定］手順をスキップします。

署名の処理を無効にする

この簡単なパートナーシップでは、署名処理を無効にします。 ただし、実稼働環境では、アイデンティティ プロバイダはアサーションに署名する必要があります。

次の手順に従ってください:

1. ［署名および暗号化］手順から、［署名の処理を無効にする］を選択します。
2. ［次へ］をクリックして次の手順に移動します。

SP のターゲットの指定

［アプリケーション統合］手順では、ターゲット リソース、および SiteMinder がユーザをターゲット リソースにリダイレクトする方法を指定します。

次の手順に従ってください:

1. ［リダイレクト モード］フィールドの［データなし］を選択します。
2. ［ターゲット］フィールドで SP のターゲット リソースを指定します。

   このサンプル パートナーシップでは、このターゲットは以下のとおりです。

   http://spapp.demo.com:80/spsample/welcome.html
3. ダイアログ ボックスの残りのセクションを無視します。
4. ［次へ］をクリックして［確認］手順に移動します。

SP パートナー設定の確認

フェデレーション パートナーシップのローカル SP 側のパートナーシップが完了しました。

次の手順に従ってください:

1. ［確認］ダイアログ ボックスで SP パートナーの設定を確認します。
2. 設定を変更するには、該当するセクションで［変更］をクリックします。
3. 設定が終了したら、［完了］をクリックします。

パートナーシップの SP 側が設定されました。

パートナーシップのアクティブ化

パートナーシップの両側が定義されたので、パートナーシップをアクティブ化できるようになりました。

SiteMinder がパートナーシップの両方のサイトでインストールされているので、IdP および SP でパートナーシップをアクティブ化する必要があります。

パートナーシップをアクティブ化する方法

1. ［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］を選択します。
2. ［フェデレーション パートナーシップ リスト］でアクティブ化するエントリを探します。［ステータス］列の値が［定義済み］であることを確認します。 ステータスが［未完了］の場合は、パートナーシップを編集します。すべての必要な設定が行われていることを確認します。
3. アクティブ化するパートナーシップ エントリの横の［アクション］-［アクティブ化］を選択します。

   ［アクティブ化の確認］ダイアログ ボックスが表示されます。
4. ［はい］をクリックします。

   パートナーシップがアクティブ化されて、［ステータス］列の値は［アクティブ］です。

パートナーシップのテスト（POST プロファイル）

パートナーシップの設定後に、2 つのパートナー間のシングル サインオンをテストします。

テストには以下が含まれます。

- シングル サインオンを開始する Web ページを作成する。
- 要求されたフェデレーション リソースとして機能するターゲット Web ページを作成する。
- シングル サインオンをテストする。

基本的なパートナーシップをテストした後で、サンプル設定に追加の変更を行うことができます。

シングル サインオンを開始する Web ページの作成

テストのために、シングル サインオンを開始するリンクを持つ独自の HTML ページを作成します。IdP または SP からシングル サインオンを開始できます。この例では SP によって開始されるシングル サインオンについて説明します。

次の手順に従ってください：

1. SP サイトでサンプル HTML ページを作成します。以下のように、SP の認証リクエスト サービスにハードコードされたリンクを含めます。

   ```
   <a href="http://sp1.demo.com:9091/affwebservices/public/
   saml2authnrequest?ProviderID=idp1.example.com>
   Link to Test POST Single Sign-on</a>
   ```

   このリンクによって、認証リクエスト サービスは、指定されたアイデンティティ プロバイダにユーザをリダイレクトして認証コンテキストを取得します。

2. Web ページを testsso.html という名前で保存します。
3. Web サーバ ドキュメント ルート ディレクトリの /spsample という名前のサブフォルダ以下に testsso.html をコピーします。

   このサンプル ネットワークでは、ターゲット Web サーバは http://spapp.demo:80 です。

ターゲット リソースの作成

シングル サインオンのテストに必要な最後の手順は、ターゲット リソースの作成です。

次の手順に従ってください：

1. SP サイトでサンプル HTML ページを作成して、以下のようなメッセージを含めます。

   SP1 へようこそ

    シングル サインオンに成功しました

2. Web ページを welcome.html という名前で保存します。
3. Web サーバ ドキュメント ルート ディレクトリのサブフォルダ /spsample 以下に welcome.html をコピーします。

   このサンプル ネットワークでは、ターゲット Web サーバは http://spapp.demo.com:80 です。

POST シングル サインオンのテスト

サンプル Web ページのセットアップ後、シングル サインオンをテストしてそのパートナーシップ設定が成功していることを確認します。

次の手順に従ってください:

1. パートナーシップの両側が 管理 UI でアクティブ化されていることを確認します。
2. ブラウザを開きます。

3. シングル サインオンをトリガするリンクを含む Web ページの URL を入力します。 この例では、以下の URL を入力します。

   http://spapp.demo.com:80/spsample/testsso.html

   URL を入力すると、POST シングル サインオンをテストするリンクを読み取るリンクと共にページが表示されます。

4. **POST シングル サインオンをテストするリンク**をクリックします。

   シングル サインオンが開始されます。 ユーザはサービス プロバイダからアイデンティティ プロバイダにリダイレクトされます。

アイデンティティ プロバイダはセッションを確立した後で、サービス プロバイダのターゲット リソース (welcome.html) にユーザを送り返します。 SP で作成したサンプル ウェルカム ページが表示されます。 表示されたページは、シングル サインオンが成功したことを示します。

署名処理の有効化

SAML 2.0 POST シングル サインオンではアサーションにデジタル署名を付ける必要があります。 署名および検証タスクについては、SiteMinder は秘密キー/証明書ペアを使用します。

トランザクションまたはランタイム アクションの前に、IdP1 の管理者は、証明書（公開キー）が含まれるファイルを SP1 に送信します。 このキーは秘密キーに関連付けられています。 IdP1 は公開キーを使用してアサーションに署名します。SP1 の管理者は、証明書を証明書データ ストアに追加します。

シングル サインオン トランザクションが発生した場合、IdP1 は秘密キーでアサーションに署名します。 SP1 はアサーションを受け取って、証明書データ ストアの証明書を使用してアサーション署名を確認します。

IdP での署名処理の設定

HTTP-POST シングル サインオンの場合、Idp1 はアサーションに署名する必要があります。 IdP は、証明書データ ストアに格納された秘密キーを使用してアサーションに署名する必要があります。

注: 例では、キー/証明書ペアをインポートできるファイルがあると仮定します。 または、秘密キー/証明書ペアがすでに証明書データ ストア内にあります。

署名を設定する方法

1. ［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］を選択します。
2. TestPartnership のエントリ（IdP から SP へのパートナーシップ）の横の［アクション］-［非アクティブ化］を選択します。

   編集する前に非アクティブ化する必要があります。
3. TestPartnership エントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードが開きます。
4. ［署名および暗号化］手順を選択します。
5. ［署名］セクションで、以下のタスクを完了します。
   a. ［署名の処理を無効にする］をクリアします。
   b. ［署名秘密キー エイリアス］フィールドの横の［インポート］をクリックします。

      ［証明書/秘密キーのインポート］ウィンドウが開きます。
6. 以下のようにインポート ウィザードを完了します。
   a. 秘密キー/証明書ペアのインポート元のファイルを選択します。
   b. pkcs#12 ファイルについては、ファイルを暗号化するパスワードを入力します。 このパスワードはすでにあります。
   c. インポートするファイルから証明書エントリを選択して、［エイリアス］に「cert1」などの値を入力します。
   d. 選択内容を確認して［完了］をクリックします。

   ［フェデレーション パートナーシップ］リストに戻ります。
7. パートナーシップ エントリの［アクション］-［変更］を選択します。

8. ［署名および暗号化］手順に進みます。ダイアログ ボックスで、インポートしたキー/証明書が［署名秘密キー エイリアス］ドロップダウン リストから選択できるようになったことに注目します。
9. 別名［cert1］を選択して［次へ］をクリックします。
10. ［確認］ダイアログ ボックスで設定を確認し、［完了］をクリックします。

    ［パートナーシップ］ウィンドウに戻ります。
11. TestPartnership エントリの横の［アクション］-［アクティブ化］を選択することによって、パートナーシップを再度アクティブ化します。

署名処理が IdP で設定されました。

SP での署名処理の設定

SP1 はアサーション署名を確認する必要があります。トランザクションの前に、SP1 は IdP1 から証明書（公開キー）を受信しています。この証明書は IdP1 がアサーションに署名するために使用した秘密キー用です。この証明書は SP1 証明書データ ストアにインポートされます。

署名検証を設定する方法

1. ［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］を選択します。

   ［パートナーシップ］ウィンドウが開きます。
2. DemoPartnership のエントリの横の［アクション］-［非アクティブ化］を選択します。

   編集する前に非アクティブ化する必要があります。
3. DemoPartnership エントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードが開きます。
4. ［署名および暗号化］手順を選択します。

5. ［署名］セクションで、以下のタスクを完了します。
   a. ［署名の処理を無効にする］をクリアします。
   b. ［検証証明書エイリアス］フィールドの横の［インポート］をクリックします。

      ［証明書/秘密キーのインポート］ウィンドウが開きます。
6. 以下のようにインポート ウィザードを完了します。
   a. 証明書のインポート元のファイルを選択します。
   b. インポートするファイルから証明書エントリを選択して、［エイリアス］に「cert1」などの値を入力します。
   c. 選択内容を確認して［完了］をクリックします。

   ［フェデレーション パートナーシップ リスト］に戻ります。
7. パートナーシップ エントリの［アクション］-［変更］を選択します。
8. ［署名および暗号化］手順に進みます。 ダイアログ ボックスで。 インポートしたキー/証明書が［署名秘密キー エイリアス］ドロップダウン リストから選択できるようになったことに注目します。
9. 証明書の別名［cert1］を選択して［次へ］をクリックします。
10. ［確認］ダイアログ ボックスで設定を確認し、［完了］をクリックします。

    ［パートナーシップ］ウィンドウに戻ります。
11. DemoPartnership エントリの横の［アクション］-［アクティブ化］を選択することによって、パートナーシップを再度アクティブ化します。

署名検証が SP で設定されました。

シングル ログアウトの追加

シングル ログアウト プロトコル（SLO）により、ログアウトを開始したブラウザのすべてのユーザ セッションが同時に終了します。 シングル ログアウトの設定によって、権限のないユーザがサービス プロバイダのリソースにアクセスできる開いたままのセッションを確実になくすことができます。

重要: SLO 設定を表示するには、ポリシー サーバ管理コンソールを使用してセッション ストアを有効にします。 管理コンソール使用の詳細については、「*ポリシー サーバ管理ガイド*」の手順を参照してください。

IdP でのシングル ログアウトの設定

Idp1 でシングル ログアウトを設定します。

次の手順に従ってください:

1. ［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］を選択します。

   ［パートナーシップ］ウィンドウが表示されます。

2. TestPartnership エントリの横の［アクション］-［非アクティブ化］を選択します。

   編集する前にパートナーシップを非アクティブ化します。

3. TestPartnership エントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードが開きます。

4. ［SSO と SLO］手順を選択します。

5. ［SLO］セクションで、以下のフィールドを設定します。

   SLO バインド

   HTTP リダイレクト

   SLO 確認 URL

   http://idp1.example.com:9090/idpsample/SLOConfirm.html

   このリンクは、シングル ログアウトを開始したサイト（この場合は IdP1）の確認ページです。 シングル ログアウトが正常に完了すると、ユーザはこのページにリダイレクトされます。

6. ［SLO サービス URL］テーブルの［行の追加］をクリックし、以下のフィールドに入力します。

   SLO ロケーション URL

   http://sp1.demo.com:9091/affwebservices/public/saml2slo

   このリンクは、シングル ログアウト リクエストがリモート SP に送信されることを示します。

7. ［選択］列で設定した行を選択します。
8. ウィザードの［確認］手順をクリックして、設定を確認します。
9. ［完了］をクリックします。

   ［パートナーシップ］ウィンドウに戻ります。

10. TestPartnership の横の［アクション］-［アクティブ化］を選択することによって、パートナーシップを再度アクティブ化します。

シングル ログアウトが IdP1 の設定に追加されました。

SP でのシングル ログアウトの設定

SP1 でシングル ログアウトを設定します。

SP でシングル ログアウトを設定する方法

1. ［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］を選択します。

   ［パートナーシップ］ウィンドウが表示されます。

2. Demo Partnership のエントリの横の［アクション］-［非アクティブ化］を選択します。

   編集する前にパートナーシップを非アクティブ化します。

3. DemoPartnership のエントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードの最初の手順のダイアログ ボックスが開きます。

4. ［SSO と SLO］手順をクリックします。
5. ［SLO］セクションで、以下のフィールドを設定します。

   SLO バインディング

   HTTP リダイレクト

   SLO 確認 URL

   http://sp1.demo.com:9091/spsample/SLOConfirm.html

   この URL は、ログアウトを開始したサイトのシングル ログアウト確認ページです。
6. ［SLO サービス URL］テーブルの［行の追加］をクリックし、以下のフィールドに入力します。

   SLO ロケーション URL

   http://idp1.example.com:9090/affwebservices/public/saml2slo

   この URL はシングル ログアウト リクエストが送信される場所です。
7. ［選択］列で設定した行を選択します。
8. ウィザードの［確認］手順をクリックして、設定を確認します。
9. ［完了］をクリックします。

   ［パートナーシップ］ウィンドウに戻ります。
10. ［フェデレーション パートナーシップ リスト］で、DemoPartnership エントリの横の［アクション］-［アクティブ化］を選択することによって、パートナーシップを再度アクティブ化します。

シングル ログアウトが SP で設定されました。

シングル ログアウトのテスト

シングル ログアウトの設定後に、それをテストします。 このテストの場合、シングル ログアウトは SP1 で開始されます。

SP からシングル ログアウトを開始するには、シングル ログアウトの開始および確認のために 2 つの Web ページが必要です。

- welcome.html を使用し、IdP1 のシングル ログアウト サービスにブラウザを送るリンクをこのページに追加します。 このリンクには以下の構文が含まれます。

  Log Me Out

- 以下のようなログアウト確認メッセージを含む SLOConfirm.html という名前の確認ページを作成します。

   正常にログアウトしました

Web サーバ ルート ディレクトリのサブフォルダ /spsample 以下に両方のページをコピーします。

注: SLO をテストできるように、SSO トランザクションを完了します。

次の手順に従ってください:

1. パートナーシップの両側が 管理 UI でアクティブ化されていることを確認します。
2. これまでに説明した手順に従ってシングル サインオンを設定およびテストします。

   シングル サインオンに成功すると、ウェルカム ページがブラウザに表示されます。
3. ブラウザを開いたままにして、ウェルカム ページで**ログアウトする**リンクをクリックします。

   成功すると、以下のメッセージを表示する確認ページにリダイレクトされます

   正常にログアウトしました。

SSO の Artifact プロファイルのセットアップ

基本的なパートナーシップはシングル サインオンの HTTP-POST バインディングから始まりました。 ただし、パートナーシップでは SAML 2.0 Artifact プロファイルを使用できます。

HTTP Artifact バインディングの設定は、POST バインディングの設定とウィザードの［SSO と SLO］手順までは同じです。

IdP での Artifact SSO の設定

この手順では、SSO の HTTP Artifact プロファイルを設定する方法について説明します。

次の手順に従ってください:

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］を選択します。

   ［パートナーシップ］ウィンドウが表示されます。

2. TestPartnership のエントリの横の［アクション］-［非アクティブ化］を選択します。

   編集する前に非アクティブ化する必要があります。

3. TestPartnership のエントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードが開きます。

4. ［SSO と SLO］手順をクリックします。
5. ［認証］セクションの既存の設定を残します。
6. ［SSO］セクションで以下のエントリを指定します。

   SSO バインド

   HTTP Artifact

   Artifact 保護タイプ

   パートナーシップ

   残りの設定はそのまま残します。

7. ［アサーション コンシューマ サービス URL］テーブルに行を追加し、以下の設定を使用します。

   バインディング

   HTTP Artifact

   URL

   http://sp1.demo.com:9091/affwebservices/public/saml2assertionconsumer

   この URL は POST プロファイルに使用されたものと同じです。

8. ［バック チャネル］セクションで、［受信設定］の以下の認証方法を選択します。

   認証方法

   認証なし

9. ダイアログ ボックスの他のセクションをスキップします。
10. ［確認］手順に進んで、設定を確認します。
11. ［完了］をクリックして、設定を終了します。

Artifact バインディングが Idp1 で設定されました。

SP での Artifact SSO の設定

この手順では、SSO の HTTP Artifact プロファイルを設定する方法について説明します。

次の手順に従ってください:

1. ［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］を選択します。

   ［パートナーシップ］ウィンドウが表示されます。

2. Demo Partnership のエントリの横の［アクション］-［非アクティブ化］を選択します。

   編集する前に非アクティブ化する必要があります。

3. DemoPartnership エントリの横の［アクション］-［変更］をクリックします。

   パートナーシップ ウィザードが開きます。

4. ［SSO と SLO］手順をクリックします。
5. ［SSO］セクションで以下のエントリを指定します。

   SSO プロファイル

   HTTP Artifact

   SSO サービス URL

   HTTP-POST シングル サインオン用に設定された URL と同様のままにします。
6. ［リモート SOAP Artifact 解決 URL］テーブルの［行の追加］をクリックします。 以下の設定を入力します。

   インデックス

   1

   URL

   http://idp1.example.com:9090/affwebservices/public/saml2ars
7. テーブルの［選択］列でこのエントリを選択します。
8. ［バック チャネル］セクションで、［送信設定］の以下の認証方法を選択します。

   認証方法

   認証なし
9. ［アプリケーション統合］手順に到達するまで［次へ］をクリックします。

SP のターゲットの指定

［アプリケーション統合］手順では、ターゲット リソース、および SiteMinder がユーザをターゲット リソースにリダイレクトする方法を指定します。

次の手順に従ってください:

1. ［リダイレクト モード］フィールドの［データなし］を選択します。
2. ［ターゲット］フィールドで SP のターゲット リソースを指定します。

   このサンプル パートナーシップでは、このターゲットは以下のとおりです。

   http://spapp.demo.com:80/spsample/welcome.html

3. ダイアログ ボックスの残りのセクションを無視します。
4. ［次へ］をクリックして［確認］手順に移動します。

パートナーシップのテスト（Artifact SSO）

パートナーシップの両側が動作している場合は、2 つのパートナー間のシングル サインオンをテストします。

IdP1 はリクエストを受信すると、アーティファクトを生成します。 その後、アーティファクトは SP1 に送信されます。

SP1 はアーティファクトを受信した後、IdP1 にリクエストをリダイレクトします。 IdP はアサーションを取得して SP1 にそれを返します。

シングル サインオン（Artifact）を開始する Web ページの作成

テストのために、シングル サインオンを開始するリンクを持つ独自の HTML ページを作成します。IdP または SP からシングル サインオンを開始できます。この例では SP によって開始されるシングル サインオンについて説明します。

次の手順に従ってください：

1. SP サイトでサンプル HTML ページを作成して、以下のように SP の認証リクエスト サービスにハードコードされたリンクを含めます。

   ```
   <a href="http://sp1.demo.com:9091/affwebservices/public/
   saml2authnrequest?ProviderID=idp1.example.com:9090&
   ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact>
   Link for ARTIFACT Single Sign-on</a>
   ```

   このリンクによって、認証リクエスト サービスは、指定されたアイデンティティ プロバイダにユーザをリダイレクトしてユーザ認証コンテキストを取得します。

2. Web ページを testartifact.html という名前で保存します。
3. Web サーバ ドキュメント ルート ディレクトリのサブフォルダ /spsample 以下に testartifact.html をコピーします。

   このサンプル ネットワークでは、ターゲット Web サーバは http://spapp.demo:80 です。

ターゲット リソースの作成

シングル サインオンのテストに必要な最後の手順は、ターゲット リソースの作成です。

次の手順に従ってください：

1. SP サイトでサンプル HTML ページを作成して、以下のようなメッセージを含めます。

   SP1 へようこそ

    シングル サインオンに成功しました

2. Web ページを welcome.html という名前で保存します。
3. Web サーバ ドキュメント ルート ディレクトリのサブフォルダ /spsample 以下に welcome.html をコピーします。

   このサンプル ネットワークでは、ターゲット Web サーバは http://spapp.demo.com:80 です。

Artifact シングル サインオンのテスト

サンプル Web ページのセットアップ後、シングル サインオンをテストしてパートナーシップ設定が成功していることを確認します。

次の手順に従ってください:

1. パートナーシップの両側がアクティブ化されていることを確認します。
2. ブラウザを開きます。

3. シングル サインオンをトリガする Web ページの URL を以下のように入力します。

   http://spapp.demo.com:80/spsample/testartifact.html

   注: ターゲット Web サーバは、SiteMinder があるサーバとは異なります。

   URL を入力すると、ARTIFACT シングル サインオンをテストするリンクを読み取るリンクと共にページが表示されます。

4. **ARTIFACT シングル サインオンをテストするリンク**をクリックすると、シングル サインオンが開始されます。

   ユーザは SP からアイデンティティ プロバイダにリダイレクトされます。

アイデンティティ プロバイダはセッションを確立した後で、サービス プロバイダのターゲット リソース (welcome.html) にユーザを送り返します。SP で作成したサンプル ウェルカム ページが表示されます。 表示されたページによって、シングル サインオンが成功したことを確認できます。

簡単なパートナーシップ以外の設定手順

簡単なパートナーシップでは、パートナーシップ フェデレーション を使用したフェデレーション パートナーシップ設定の概要について説明しています。

ガイドの残りの章では、実行できるすべてのタスクの詳細な手順について説明します。 詳細な設定手順については、管理 UI の［ヘルプ］と同様にこれらの手順も使用してください。

詳細情報:

フェデレーション エンティティ設定 (P. 71)
パートナーシップの作成およびアクティブ化 (P. 83)

第 4 章: セッション ストアを必要とするフェデレーション機能

セッション ストアには、以下のフェデレーション機能のデータが格納されます。

- HTTP Artifact シングル サインオン（SAML 1.x または 2.x）

  SAML アサーションおよび関連 Artifact は、アサーティング パーティで生成されます。Artifact によって、生成されたアサーションが識別されます。 アサーティング パーティは、依存パーティに Artifact を返します。 依存パーティは、Artifact を使用してアサーションを取得します。アサーションは、アサーティング パーティによってセッション ストアに保存されます。

  このプロセスが動作するには、永続セッションが必要です。

  注: SAML POST プロファイルでは、セッション ストアにアサーションを保存しません。

- HTTP-POST 使い捨てポリシー（SAML 2.0 および WS-フェデレーション）

  使い捨てポリシー機能は、依存パーティで別のセッションを確立するためにアサーションが再利用されるのを防止します。 依存パーティでは、アサーションに関する時間ベースのデータ（有効期限データと呼ばれます）がそのセッション ストアに保存されます。有効期限データにより、アサーションが 1 度だけしか使用されないようにできます。

  セッション ストアは依存パーティで必須ですが、永続セッションは必須ではありません。

- シングル ログアウト（SAML 2.0）

  シングル ログアウトが有効な場合、一方のパートナーがユーザ セッションに関する情報を保存できます。 セッション情報は、セッション ストアで保持されます。 シングル ログアウト リクエストが終了すると、そのユーザに関するセッション情報は削除され、セッションが無効化されます。

  アイデンティティ プロバイダおよびサービス プロバイダでは、永続セッションが必須です。

- サインアウト（WS-フェデレーション）

  サインアウトが有効な場合、ユーザ コンテキスト情報はセッション ストアに配置されます。ポリシー サーバでは、この情報を使用してサインアウト リクエストを生成します。 サインアウト リクエストが終了すると、そのユーザに関するセッション情報は削除され、ユーザ セッションが無効化されます。

  アイデンティティ プロバイダおよびリソース パートナーでは、永続セッションが必須です。

- 認証セッション変数永続性（すべてのプロファイル）

  依存パーティでフェデレーションを設定する際に、［永続認証セッション変数］オプションを選択できます。 このオプションは、認証コンテキスト データをセッション変数としてセッション ストアに保存するようにポリシー サーバに指示します。 ポリシー サーバは認証決定で使用されるこれらの変数にアクセスできます。

- アサーション属性の保持（すべてのプロファイル）

  依存パーティでのリダイレクト モードとして［属性の保持］を選択できます。リダイレクト モードにより、ユーザがターゲット アプリケーションにどのようにリダイレクトされるかが決定されます。 このモードは、HTTP ヘッダ変数として提供できるように、セッション ストアにアサーション属性を格納することをポリシー サーバに指示します。

- 認証リクエスト POST バインディング（SAML 2.0）

  IdP が HTTP-POST バインディングを使用して提供される認証リクエストを処理するには、IdP はセッション ストアにリクエストを格納する必要があります。

このタイプのユーザ セッション、アサーション、および有効期限データを保持するには、セッション ストアを有効にします。

セッション ストアの有効化

シングル サインオン、シングル ログアウト用に SAML Artifact を使用して、ポリシーの使い捨てを有効にするときに、データを保持するためにセッション ストアを有効にします。

セッション ストアの有効化は、ポリシー サーバ管理コンソールから行います。

次の手順に従ってください:

1. ポリシー サーバ管理コンソールにログインします。
2. ［データ］タブを選択します。
3. ［データベース］フィールドのドロップダウン リストから［セッション ストア］を選択します。
4. ［ストレージ］フィールドのドロップダウン リストから利用可能なストレージ タイプを選択します。

5. ［セッション ストアが有効です］チェック ボックスを選択します。

   1 つ以上のレルムで永続セッションを使用する予定がある場合は、［セッション サーバ］を有効にします。 セッション サーバを有効にすると、ポリシー サーバのパフォーマンスに影響します。

   注: ［ポリシー ストアを使用］データベース オプションは無効になります。パフォーマンス上の理由から、セッション サーバをポリシー ストアと同じデータベース上で動作させることはできません。

6. 選択したストレージ タイプに適した［データ ソース情報］を指定します。
7. ［OK］をクリックして設定を保存し、コンソールを終了します。
8. ポリシー サーバを停止してから再起動します。

共有セッション ストアを必要とする環境

以下の機能では、SAML アサーションおよびユーザ セッション情報を保存するために共有セッション ストアを必要とします。

クラスタ化されたポリシー サーバ環境にこれらの機能を実装するには、以下のように環境をセットアップします。

- HTTP-POST 使い捨てポリシー*以外*のすべての機能に関する永続セッションのログイン レルムを設定します。

  永続セッションは、レルム設定の一部です。

- HTTP Artifact シングル サインオンの場合、プロデューサ/アイデンティティ プロバイダ サイトのセッション ストアを、クラスタ内のすべてのポリシー サーバで共有します。

  セッション ストアを共有することにより、ポリシー サーバのそれぞれがアサーションに関するリクエストを受信するときに、すべてのポリシー サーバがアサーションにアクセス権があることを確認できます。

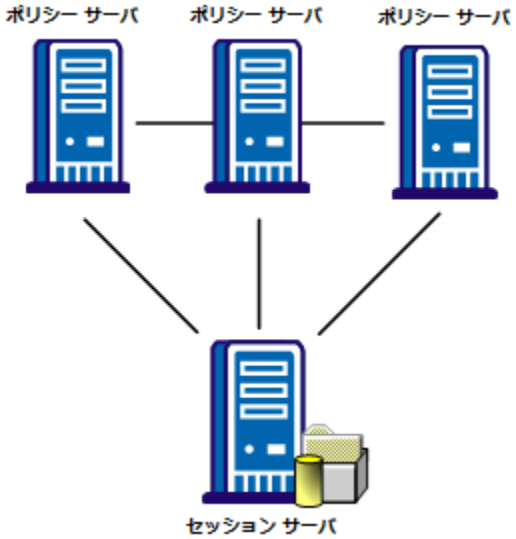
- SAML 2.0 シングル ログアウトおよび WS-フェデレーション サインアウトの場合、アサーティング パーティおよび依存パーティのセッション ストアを、クラスタ内のすべてのポリシー サーバで共有します。

  セッション ストアを共有することにより、ポリシー サーバのそれぞれがセッション ログアウトに関するリクエストを受信するときに、すべてのポリシー サーバがユーザ セッション データにアクセス権があることを確認できます。

- HTTP-POST および WS-フェデレーションの使い捨てポリシー機能の場合、依存パーティのセッション ストアを、クラスタ内のすべてのポリシー サーバで共有します。

アサーションを生成または消費するポリシー サーバや、永続的な SMSESSION Cookie を処理するポリシー サーバはすべて、共通のセッション ストアにアクセスできる必要があります。 たとえば、ユーザが example.com にログインし、そのドメインの永続セッション Cookie を取得するとします。 example.com に対するリクエストを処理しているすべてのポリシー サーバは、セッションが引き続き有効であることを確認できる必要があります。

次の図は、1 つのセッション ストアと通信するポリシー サーバ クラスタを示しています。



セッション ストアを共有するには、以下のいずれかの方法を使用します。

- すべてのポリシー サーバが 1 つのセッション ストアを参照するようにします。

  ポリシー サーバ管理コンソールで、指定のセッション ストアを使用するようにポリシー サーバを設定します。

- 複数のセッション ストアでセッション ストアを複製します。

  データベースの複製の手順については、使用しているデータベースのマニュアルを参照してください。

第 5 章: パートナーシップ フェデレーションのユーザ ディレクトリ接続

パートナーシップ フェデレーションは、ユーザ ディレクトリでエントリを検索して ID を確認し、指定されたプリンシパルのユーザ属性を取得します。 アサーティング パーティでは、フェデレーション パートナーが該当するユーザのアサーションを生成し、ユーザ ディレクトリに対して各ユーザを認証します。 依存するパーティでは、フェデレーション パートナーがアサーションから必要な情報を抽出し、ユーザ ディレクトリで該当するユーザ レコードを検索します。

管理 UI で［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］を選択することにより、既存のユーザ ディレクトリへの接続を設定します。 既存のユーザ ディレクトリへの接続のみを確立します。 新規ユーザ ディレクトリは設定しません。

注: フェデレーションの設定に ODBC データベースを使用する場合は、ユーザ ディレクトリとして ODBC データベースを選択する前に、SQL クエリ方式と有効な SQL クエリを設定してください。

必要に応じて、複数のディレクトリへの接続を設定します。 ディレクトリは、同じタイプである必要はありません。

ユーザ ディレクトリの詳細については、「ポリシー サーバ設定ガイド」を参照してください。

第 6 章: 認証 URL の保護による SiteMinder セッションの要求

ポリシー サーバがアサーションを生成するには、ユーザは IdP ポリシー サーバでセッションを持つ必要があります。 セッションを確立するために、IdP のシングル サインオン サービスが認証 URL 経由でユーザをアプリケーションにリダイレクトします。 ユーザに認証チャレンジが提示されるように、ポリシーを使用して認証 URL を保護します。 これにより、ユーザがログインしてセッションが確立されます。

認証 URL が redirect.jsp ファイルを指すことが必要です。 例：

http://webserver1.example.com/affwebservices/redirectjsp/redirect.jsp

この例では、webserver1 によって Web エージェント オプション パックを持つ Web サーバが識別されます。 redirect.jsp ファイルは、アイデンティティ プロバイダでインストールされた Web エージェント オプション パックに含まれています。

認証が成功した後、redirect.jsp アプリケーションは、アサーション生成のためにシングル サインオン サービスにユーザをリダイレクトして戻します。

セッションの作成を可能にするには、次の 2 つの手順が必要です。

1. redirect.jsp ファイル用のポリシーを作成します。 (P. 68)
2. パートナーシップで認証 URL を指定します (P. 70)。

Redirect.jsp 用のポリシーの作成

認証チャレンジをトリガするためには、ポリシーにより認証 URL が保護されている必要があります。

次の手順に従ってください:

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］-［エージェントの作成］を選択します。

   アサーティング パーティ Web サーバに対して定義されたレルムにバインドするには、Web エージェントを作成します。Web サーバに対して一意のエージェント名を割り当てます。
3. ［ポリシー］-［ドメイン］-［ドメイン］-［ドメインの作成］を選択します。

   認証 URL 用のポリシー ドメインを作成します。チャレンジを要求されるユーザが含まれるユーザ ディレクトリを追加します。
4. ポリシー ドメインに含まれるリソースへのアクセス権が必要なユーザを選択します。
5. ［レルム］タブを選択し、次の値を使用してポリシー ドメインのレルムを定義します。

   エージェント

   アサーティング パーティ Web サーバのエージェント。 このエージェントは、手順 2 で作成しています。

   リソース フィルタ

   /affwebservices/redirectjsp

   このリソース フィルタは Web エージェントおよび SPS フェデレーション ゲートウェイに適用されます。

デフォルト リソース保護

保護

認証方式

基本

永続セッション

HTTP-Artifact プロファイルのレルム ダイアログ ボックスの［セッション］セクションにある［永続］チェック ボックスをオンにして、セッション情報を格納します。セッション情報は、シングル ログアウトなどの機能、および属性機関に必要です。

6. レルム ダイアログ ボックスの［ルール］セクションで、［ルールの作成］をクリックします。 フィールドに以下の値を入力します。

リソース

/*

アスタリスクは、ルールがレルム内のすべてのリソースに適用されることを意味します。

許可/拒否と有効/無効

アクセスを許可する

［有効］チェック ボックスはオン。

アクション

Web エージェント アクション

Get、Post、Put

7. ［ポリシー］タブを選択し、次のコンポーネントが含まれるポリシーを作成します。
   - ユーザ ディレクトリで選択したユーザのセット。
   - redirectjsp アプリケーションおよび関連ルールが含まれるレルム。

これで、ポリシーにより認証 URL が保護されます。 ユーザがこの URL にリダイレクトされると、認証チャレンジがトリガされます。 最終的に、セッションが作成されます。

パートナーシップでの認証 URL の指定

認証 URL を保護するポリシーを設定した後、IdP から SP へのパートナーシップなどのアサーティング パーティから依存パーティへのパートナーシップでこの URL を指定します。

認証 URL はシングル サインオン設定の一部として設定されます。 ダイアログ ボックスの［認証］セクションで、［認証モード］フィールドに［**ローカル**］を選択し、完全な認証 URL を入力します。 例：

http://webserver1.example.com/affwebservices/redirectjsp/redirect.jsp

この例では、webserver1.example.com によって Web エージェント オプション パックを持つ Web サーバが識別されます。

第 7 章: フェデレーション エンティティ設定

このセクションには、以下のトピックが含まれています。

エンティティを作成する方法

フェデレーション パートナーシップの各パートナーは、*フェデレーション エンティティ*であるとみなされます。 パートナーシップを確立する前に、ローカル パートナーを表すローカル エンティティ、およびリモート パートナーを表すリモート エンティティを定義します。

フェデレーション エンティティを設定する 2 つの方法は以下のとおりです。

- メタデータを使用せずにエンティティを作成します (P. 71)。
- メタデータのインポートによりエンティティを作成します (P. 77)。

メタデータを使用しないエンティティの作成

以下のプロセスを使用して、メタデータなしでエンティティを作成します。

1. エンティティ タイプを示します。
2. そのエンティティ タイプに関する詳細を設定します。
3. エンティティ設定を確認します。

エンティティ タイプ選択

エンティティ設定の最初の手順は、エンティティ タイプを確立してエンティティ ロールを決定することです。

エンティティ タイプを確立する方法

1. 管理 UI にログインします。
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
3. ［エンティティの作成］をクリックします。

   ［エンティティの作成］ダイアログ ボックスが表示されます。

   注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。
4. 以下の*いずれか*のオプションを選択します。

   ローカル

   サイトに対してローカルなエンティティを作成することを示します。

   リモート

   リモート サイトのパートナーを表すエンティティを設定することを示します。
5. 残りのフィールドを設定します。

   新しいエンティティ タイプ（New Entity Type）

   アサーティング パーティまたは依存パーティを選択します。

   SAMLToken Type（WS-FED **のみ**）

   トークン タイプ（ユーザ認証情報が含まれる暗号化されたトークンの SAML 形式を定義する）を選択します。 WS-Federation 1.0 の SAML トークン タイプに準拠したトークンを望む場合にのみ、［レガシー］オプションを選択します。
6. エンティティに関する詳細を設定するために［次へ］をクリックします。

詳細なローカル エンティティ設定

エンティティ タイプを指定した後で、エンティティの詳細を設定します。ローカル エンティティについては、以下の情報を定義します。

- エンティティに関する識別情報
- 署名および暗号化オプション
- 名前 ID 形式と属性

次の手順に従ってください:

1. ［エンティティの設定］手順から始めます。
2. 設定するローカル エンティティ タイプの機能およびサービスに必要なフィールドに入力します。

   フィールドの説明については、［ヘルプ］をクリックしてください。
3. ［次へ］をクリックします。

   ［確認］ダイアログ ボックスが表示されます。

以下の機能に注意してください。

エンティティ ID およびエンティティ名の設定

エンティティ ID がリモート パートナーを表す場合、その値は一意である必要があります。 エンティティ ID がローカル パートナーを表す場合、同じシステム上で再利用できます。

エンティティ名は、ポリシー ストアのエンティティ オブジェクトを識別します。［エンティティ名］は一意の値にする必要があります。 この値は内部使用のみです。リモート パートナーはこの値を認識しません。

注: ［エンティティ名］は［エンティティ ID］と同じ値にすることができますが、その値を同じサイトの他のエンティティとは共有しないでください。

署名と暗号化機能

署名と暗号化機能については、証明書データ ストアに適切なキー/証明書エントリを持つ必要があります。 適切なキー/証明書エントリがない場合は、［インポート］をクリックしてローカル システム上のファイルから秘密鍵/証明書ペアをインポートします。 また、信頼された証明書もインポートできます。

注: SAML 2.0 POST プロファイルを使用している場合は、アサーションの署名が必要です。

WSFED 属性（WS フェデレーションのみ）

通信する WS フェデレーション エンティティのさまざまなサービス URL および ID を指定できます。

名前 ID 形式

フェデレーション エンティティがサポートする識別子タイプを指定できます。

アサーション属性設定（アサーティング パートナーのみ）

アサーティング パーティがアサーションを生成するときに特定のアサーション属性を含めるように、アサーティング パーティを設定できます。これらの属性はエンティティ レベルで定義することをお勧めします。エンティティはパートナーシップ用のテンプレートとして機能するため、そのエンティティについて定義するすべてのアサーション属性はパートナーシップに伝達されます。エンティティでアサーション属性を定義する利点は、複数のパートナーシップでエンティティを使用できることです。

パートナーシップのアサーション属性を追加または削除する場合は、エンティティ レベルではなく、パートナーシップ レベルでそのような変更を行います。

詳細なリモート エンティティ設定

エンティティ タイプを指定した後で、エンティティの詳細を設定します。リモート エンティティ タイプについては、以下の情報を定義します。

- エンティティに関する識別情報
- 署名および暗号化オプション
- 名前 ID および属性情報

次の手順に従ってください:

1. ［エンティティの設定］手順から始めます。
2. ［アサーション コンシューマ サービス URL］を指定します。 例：
   - SP が Google などのサイトである場合、URL は以下のようになります。

     https://www.google.com/a/example.com/acs
   - SP が Salesforce.com などのサイトである場合、URL は以下のようになります。

     https://login.salesforce.com/?saml=EK05LGnm40H7
   - SP が別のビジネス パートナーである場合、URL は以下のようになります。

     http://myserver.forwardinc.com:9080/samlsp/acs
3. リモート エンティティ タイプの機能およびサービスの他の必須フィールドに入力します。

   フィールドの説明については、［ヘルプ］をクリックしてください。
4. ［次へ］をクリックします。

   ［確認］ダイアログ ボックスが表示されます。

以下の機能に注意してください。

エンティティ ID およびエンティティ名の設定

エンティティ ID がリモート パートナーを表す場合、その値は一意である必要があります。 エンティティ ID がローカル パートナーを表す場合、同じシステム上で再利用できます。

エンティティ名は、ポリシー ストアのエンティティ オブジェクトを識別します。［エンティティ名］は一意の値にする必要があります。 この値は内部使用のみです。リモート パートナーはこの値を認識しません。

注: ［エンティティ名］は［エンティティ ID］と同じ値にすることができますが、その値を同じサイトの他のエンティティとは共有しないでください。

署名と暗号化機能

署名と暗号化機能については、証明書データ ストアに適切なキー/証明書エントリを持つ必要があります。 適切なキー/証明書エントリがない場合は、［インポート］をクリックしてローカル システム上のファイルから秘密鍵/証明書ペアをインポートします。 また、信頼された証明書もインポートできます。

注: SAML 2.0 POST プロファイルを使用している場合は、アサーションの署名が必要です。

WSFED 属性（WS フェデレーションのみ）

通信する WS フェデレーション エンティティのさまざまなサービス URL および ID を指定できます。

名前 ID 形式

フェデレーション エンティティがサポートする識別子タイプを指定できます。

アサーション属性設定（アサーティング パートナーのみ）

アサーティング パーティがアサーションを生成するときに特定のアサーション属性を含めるように、アサーティング パーティを設定できます。これらの属性はエンティティ レベルで定義することをお勧めします。エンティティはパートナーシップ用のテンプレートとして機能するため、そのエンティティについて定義するすべてのアサーション属性はパートナーシップに伝達されます。エンティティでアサーション属性を定義する利点は、複数のパートナーシップでエンティティを使用できることです。

パートナーシップのアサーション属性を追加または削除する場合は、エンティティ レベルではなく、パートナーシップ レベルでそのような変更を行います。

エンティティ設定の確認

エンティティ設定を保存する前に確認します。

次の手順に従ってください:

1. エンティティ ダイアログ ボックスで設定を確認します。
2. ［戻る］をクリックしてこのダイアログ ボックスから設定を変更します。
3. 設定が終了したら、［完了］をクリックします。

新しいエンティティが設定されました。

パートナーシップからのエンティティ設定変更

単一のパートナーシップ設定のコンテキスト内からリモート エンティティのエンティティ ID 値を変更できます。 ただし、パートナーシップ レベルでエンティティ ID を変更しても、パートナーシップは別のエンティティに関連付けられず、元のエンティティも更新されません。 エンティティへの変更はエンティティからパートナーシップへの一方向の伝達です。 パートナーシップ レベルでのエンティティ ID への変更は元のエンティティに伝達されません。

注: 指定するエンティティ ID は、リモート パートナーが使用しているものと一致する必要があります。

エンティティ設定はテンプレートと見なされます。 パートナーシップはエンティティ テンプレートに基づいて作成されるので、パートナーシップの変更によって元のエンティティ テンプレートが変更されることはありません。

パートナーシップ内のエンティティの詳細については、「パートナーシップからエンティティを編集する (P. 87)」を参照してください。

メタデータのインポートによるエンティティの作成

メタデータ ファイルからデータをインポートしてフェデレーション エンティティを作成できます。メタデータのインポートによって、パートナーシップを作成するための設定の量が減少します。

以下の方法でメタデータを使用できます。

- リモート パートナーからデータをインポートして新しいリモート エンティティを作成します。
- リモート パートナーからデータをインポートして既存のリモート エンティティを更新します。
- ローカル エンティティからデータをインポートして新しいローカル エンティティを作成します。

  このオプションは、別のフェデレーション製品からの移行を容易にするために役立ちます。

  注: フェデレーションは、既存のパートナーシップおよびローカル エンティティを更新またはリストアするメタデータ インポートをサポートしません。既存のローカル エンティティを更新するには、エンティティを編集して変更が必要な設定を変更します。*新しい*ローカル エンティティを作成するためにのみメタデータをインポートできます。

メタデータ ベースのエンティティを作成するプロセスは以下のとおりです。

1. 新しいエンティティを設定するためのメタデータ ファイルを選択します。
2. メタデータ ファイルからエンティティ エントリを選択します。 ファイルには複数のエンティティを含めることができますが、1 つのファイルに 1 つのエンティティを含めることをお勧めします。
3. （オプション）証明書データ ストアにインポートする証明書を選択します。 証明書はメタデータ ファイル内にある必要があります。

   これらの証明書は、認証リクエスト検証、シングル ログアウト レスポンス検証（SAML 2.0）、および暗号化（SAML 2.0）に使用できます。
4. エンティティ設定を確認します。

これらの手順についての詳細は、次のセクションで説明します。

メタデータ ファイル選択

メタデータからエンティティを作成する最初の手順は、メタデータ ファイルを選択することです。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
3. ［メタデータのインポート］をクリックします。

   ［メタデータのインポート］ダイアログ ボックスが開きます。

   フィールドの説明については、［ヘルプ］をクリックしてください。
4. エンティティの作成に使用するメタデータ ファイルを参照します。
5. 新しいローカルまたはリモート エンティティの作成、または既存のリモート エンティティの更新のいずれかを選択します。

   注: ポリシー サーバでは、既存のパートナーシップおよびローカル エンティティを更新するためのメタデータ インポートはサポートされていません。 新しいローカル エンティティのみを作成できます。 既存のローカル エンティティを更新するには、エンティティを編集して変更が必要な設定を変更します。既存のリモート エンティティを更新する、または新しいリモート エンティティを作成することができます。
6. ファイルからエンティティを選択するために［次へ］をクリックします。

期限切れエントリを含むメタデータ ファイルを選択すると、UI が表示する次のダイアログ ボックスには期限切れエントリが一覧表示されたセクションが含まれます。 参照用に表示されるこれらの期限切れエントリは選択できません。 メタデータ ファイル内のすべてのエンティティが期限切れである場合、エンティティは表示されません。 この場合、新しいドキュメントをアップロードします。

インポートするエンティティの選択

この手順では、エンティティを作成するためのメタデータ ファイルをすでに選択していることを前提としています。 ファイルからエンティティを選択します。

次の手順に従ってください:

1. ［ファイルに定義されるエンティティの選択］ダイアログ ボックスで新しいエンティティの名前を指定します。

   エンティティを作成するためにローカル インポートを実行する場合は、パートナーシップ名を定義します。

2. オプション ボタンをクリックしてエンティティを選択します。
3. ［次へ］をクリックします。

   リモート エンティティおよびドキュメント用のメタデータのインポートに証明書データが含まれる場合、［証明書のインポート］ダイアログ ボックスが表示されます。

   インポートしたメタデータ ファイルに証明書エントリが含まれる場合、これらのエントリをインポートできます。

証明書インポート

署名済みのアサーションを確認するために、メタデータに証明書が含まれる場合は、証明書をインポートします。 メタデータに証明書が含まれない場合、この手順をスキップして［確認］手順に進みます。

次の手順に従ってください:

1. ［証明書のインポート］手順で、インポートするメタデータ ファイルから証明書エントリ（複数可）を選択します。

   無効なエントリを含む証明書ファイルを選択すると、次のダイアログ ボックスには期限切れエントリが一覧表示されたセクションが含まれます。これらの期限切れエントリは選択できません。それらは参照用に表示されます。 ファイル内のすべてのエントリが無効である場合、インポート ウィザードは証明書選択の手順をスキップします。

   選択した各エントリに対して一意の別名を指定します。

2. ［次へ］をクリックします。

   ［確認］ダイアログ ボックスにはエントリのテーブルが表示されます。

   同じ証明書を含むメタデータ ファイルから 2 つのエントリを選択できます。 SAML 1.1 および WS-フェデレーションのメタデータについては、SAML 1.1 がデータを暗号化しないので、すべてのエントリは証明書の使用状況として［署名］を示します。

   SAML 2.0 については、各エントリが示す証明書の使用状況は異なる場合があります（たとえば、1 つが署名、1 つは暗号化）。 ［確認］手順に到達すると、ウィンドウには単一の証明書エントリを含むテーブルが表示されます。 証明書使用状況は［署名］および［暗号化］として一覧表示されます このエントリは、前に選択した 2 つのエントリの組み合わせです。 さらにこのエントリは、選択した証明書エントリに対して指定した最初の別名を使用します。

   同じ証明書が両方の用途でメタデータ ファイルに一覧表示された場合にのみ、この状況が発生します。 ファイルに 2 つの個別の証明書が含まれる場合、確認手順で、両方のエントリがテーブルに示されます。

   たとえば、メタデータ ファイルから 2 つのエントリを選択しても、それらが同じ証明書であることはわかりません。 最初の使用状況は［署名］です。それに別名 **cert1** を割り当てます。2 つ目の使用状況は［暗号化］です。それに別名 **cert2** を割り当てます。 インポートの確認時に、以下のようなエントリを含む［選択された証明書データ］というタイトルのテーブルが表示されます。

エイリアス	発行先	使用状況
cert1	Jane Doe	署名および暗号化

   使用状況がメタデータ ファイルで指定されていない場合、使用状況はデフォルトで［署名］および［暗号化］になります。

3. ［次へ］をクリックして設定を終了します。

エンティティ設定の確認

エンティティ設定を保存する前に確認します。

次の手順に従ってください:

1. エンティティ ダイアログ ボックスで設定を確認します。
2. ［戻る］をクリックしてこのダイアログ ボックスから設定を変更します。
3. 設定が終了したら、［完了］をクリックします。

新しいエンティティが設定されました。

第 8 章: パートナーシップの作成およびアクティブ化

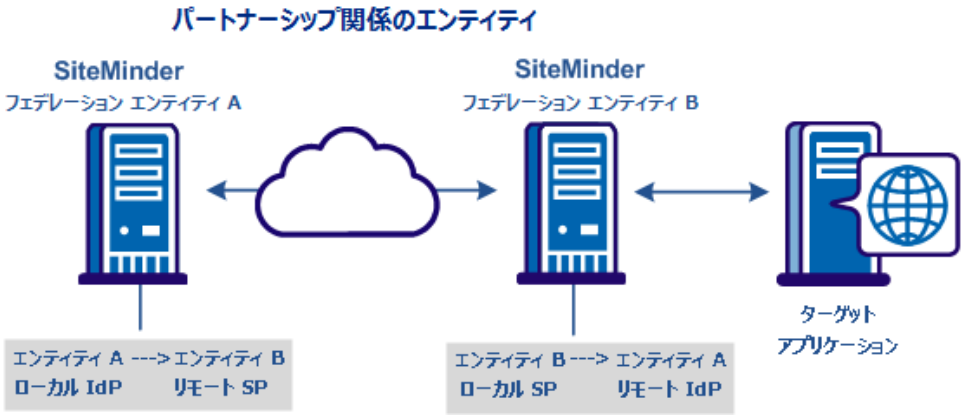
このセクションには、以下のトピックが含まれています。

パートナーシップ作成

パートナーシップ フェデレーション の主な目的は、2 つの組織の間でパートナーシップを確立することにより、ユーザ識別情報を共有してシングル サインオン（SSO）を容易にすることです。 パートナーシップは異なるサイトの 2 つのエンティティ（ローカルとリモート）で構成されます。 両方のエンティティが、アサーションを生成する側であるアサーティング パーティ、またはアサーションを消費する側である依存するパーティの役割を担うことができます。

SiteMinder が両方のサイトでインストールされている場合、各サイトでパートナーシップを定義する必要があります。1 つのサイトのローカル アサーティング パーティから依存するパーティへの各パートナーシップに対して、パートナー サイトにはローカル依存パーティからアサーティング パーティへの相互的なパートナーシップが必要です。 たとえば、エンティティ A でのパートナーシップ設定については、エンティティ A はローカル ID プロバイダ（IdP）で、エンティティ B はリモート サービス プロバイダ（SP）です。 エンティティ B でのパートナーシップ設定については、エンティティ B はローカル サービス プロバイダ（SP）で、エンティティ A はそのリモート ID プロバイダ（IdP）です。このパースペクティブは、ローカル エンティティに基づきます。

以下の図は、パートナーシップのエンティティ関係を示しています。



注: アサーティング パーティは複数の依存するパーティとのパートナーシップを持つことができ、依存するパーティは複数のアサーティング パーティとのパートナーシップを確立できます。

パートナーシップを作成するには、パートナーシップ ウィザードに従って必要な設定手順を実行します。

パートナーシップ定義

フェデレーション パートナーシップ定義で、ローカルのフェデレーション パートナーおよびリモートのフェデレーション パートナーを指定します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［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］を選択します。

   ［フェデレーション パートナーシップ］ダイアログ ボックスが表示されます。
3. ［フェデレーション パートナーシップ リスト］の［パートナーシップの作成］をクリックします。

4. 以下のいずれかのパートナーシップを選択します。
   - SAML2 IDP->SP（アイデンティティ プロバイダはローカルです）。
   - SAML2 SP->IDP（サービス プロバイダはローカルです）。
   - SAML1.1 プロデューサからコンシューマ（プロデューサはローカルです）。
   - SAML1.1 コンシューマからプロデューサ（コンシューマはローカルです）。
   - WSFED IP->RP（アイデンティティ プロバイダはローカルです）。
   - WSFED RP から IP （リソース パートナーはローカルです）

パートナーシップ ウィザードの最初の手順で［パートナーシップ］ダイアログ ボックスが開きます。

パートナーシップの識別および設定

ウィザードの［パートナーシップの設定］手順で、パートナーシップを命名し、ローカルまたはリモート エンティティを指定して、パートナーシップを設定します。

注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。

次の手順に従ってください:

1. パートナーシップの名前を入力します。 名前には英数字、アンダースコア、ハイフン、およびピリオドを使用できます。 スペースは使用できません。
2. （オプション）説明を入力します。
3. すでにエンティティを設定している場合、ローカル リストからローカル エンティティを選択します。設定していない場合は、［ローカル エンティティの作成］をクリックします。

4. すでにエンティティを設定している場合、リモート リストからリモート エンティティを選択します。設定していない場合は、［リモート エンティティの作成］をクリックします。

   注: 後でメタデータをインポートしてリモート エンティティを作成する予定がある場合は、この手順を先送りできます。

5. （オプション）ベース URL を指定します。

6. （オプション）スキュー時間を秒単位で入力します。

   スキュー時間とは、ローカル システム上のシステム時間とリモート システム上のシステム時間の差です。通常、システム クロックの誤差によってこの状態が生じます。 現在の時刻から秒数を引くことにより、スキュー時間数を決定します。

   システムは、スキュー時間および SSO 有効期間を使用して、アサーションが有効な期間を決定します。

7. ［使用可能なディレクトリ］リストから 1 つ以上のユーザ ディレクトリを選択して、［選択されたディレクトリ］リストに移動します。

   1 つのユーザ ディレクトリのみを設定する場合、そのディレクトリは自動的に［選択されたディレクトリ］リストに配置されます。

   重要: ODBC データベースをユーザ ディレクトリとして使用するには、SQL クエリ スキームおよび有効な SQL クエリを定義します。ユーザが ODBC データベースをユーザ ディレクトリとして選択できるようになるには、これらの手順が必要です。

8. パートナーシップ ウィザードを続行し完了するには、［次へ］をクリックします。 このウィザードの手順によって、パートナーシップのさまざまな機能を設定できます。一部の機能は必須で、一部の機能はオプションです。 これらの機能の設定の詳細は、このガイドの以降のセクションに記述されています。

注: パートナーシップを編集する場合、このフィールドの横の［更新の取得］をクリックしてエンティティ情報を更新できます。 エンティティ設定からの最新情報はパートナーシップに伝達されます。 ただし、パートナーシップから直接エンティティ情報を編集する場合は、変更は個々のエンティティ設定に伝達されません。

パートナーシップからエンティティを編集する

ローカル エンティティおよびリモート エンティティのフィールドの横の［更新の取得］をクリックして、エンティティに関する情報を更新できます。 ［更新の取得］の選択時に、エンティティから最新情報を取り込むように求められます。

確認後、編集中のパートナーシップは最新のエンティティ情報でリフレッシュされます。 パートナーシップ ウィザードを完了するときに、変更が保存されます。 更新を確認しない場合、パートナーシップ設定は変更されません。

［エンティティ名］によってポリシー ストアのエンティティ オブジェクトが識別されます。 ［エンティティ名］の値は、製品によってエンティティを区別するために内部的に使用されるので、一意の識別子である必要があります。 この値は外部的には使用されず、リモート パートナーはこの値を認識しません。

エンティティ ID がリモート パートナーを表す場合、その値は一意である必要があります。 エンティティ ID がローカル パートナーを表す場合、同じシステム上で再利用できます。

注: ［エンティティ名］は［エンティティ ID］と同じ値にすることができますが、その値を他のエンティティとは共有しないでください。

エンティティはフェデレーション パートナーシップの主要なコンポーネントです。 エンティティの変更によってパートナーシップには著しい変更が加えられてしまうので、管理 UI ではパートナーシップに取り込まれた後のエンティティを置換できません。 エンティティを置換するには、パートナーシップを作成します。

エンティティ ID ではエンティティが一意に識別されないので、パートナーシップ設定内の柔軟性のためにエンティティ ID を変更できます。パートナーシップ レベルでエンティティ ID を変更してもパートナーシップは別のエンティティに関連付けられません。 パートナーシップ内の元のエンティティは変更されません。 エンティティへの変更はエンティティからパートナーシップへの一方向の伝達です。 パートナーシップでのエンティティ ID への変更は元のエンティティに伝達されません。

エンティティ設定はテンプレートと見なされます。 パートナーシップはエンティティ テンプレートに基づいて作成されるので、パートナーシップの変更によって元のエンティティ テンプレートが変更されることはありません。

パートナーシップ確認

パートナーシップ設定を保存する前に確認します。

次の手順に従ってください:

1. パートナーシップ ウィザードの［確認］手順で設定を確認します。
2. 設定を変更するには、各グループ ボックスの［変更］をクリックします。
3. 設定が終了したら、［完了］をクリックします。

パートナーシップ設定が完了しました。

パートナーシップ アクティブ化

パートナーシップに対して必要なすべての設定を行ったら、それをアクティブにして使用します。 また、同じプロセスでパートナーシップを非アクティブ化できます。

次の手順に従ってください:

1. ［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］を選択します。

   ［パートナーシップ］ダイアログ ボックスが開きます。

2. ［アクション］メニューから、対象パートナーシップの横の［アクティブ化］または［非アクティブ化］を選択します。

   確認のダイアログ ボックスが表示されます。

   注:［アクティブ化］は［定義済み］または［非アクティブ］ステータスのパートナーシップにのみ使用できます。［非アクティブ化］は［アクティブ］ステータスのパートナーシップにのみ使用できます。

3. ［はい］をクリックして選択内容を確認します。

   パートナーシップのステータスが設定され、表示がリフレッシュされます。

重要: 変更する前にパートナーシップを非アクティブ化します。

パートナーシップのエクスポート

リモート エンティティの作成、およびパートナーシップの作成のベースとしてメタデータを使用できます。 エンティティの多くの特徴がすでにメタデータ ファイルで定義されているので、メタデータによってパートナーシップ設定の効率は向上します。 パートナーシップまたはリモート エンティティを作成するためにファイルをインポートできます。

エクスポートする前にパートナーシップを完了する必要はありません。パートナーシップの一部を設定した後でエクスポートできます。

管理 UI で、既存のパートナーシップ エントリからメタデータをエクスポートできます。

注: 管理 UI で、既存のローカル アサーティング エンティティまたは依存エンティティからメタデータをエクスポートできます。 SAML 1.1 データをエクスポートする場合、結果のメタデータ ファイルで使用される用語は SAML 2.0 の用語です。この規則は SAML 仕様の一部です。SAML 1.1 データをインポートする場合、用語は SAML 1.1 の用語を使用して正確にインポートされます。

パートナーシップからエクスポートするとき、選択されたパートナーシップはエクスポートのベースとして使用されます。 新しいパートナーシップ名を定義することは許可されていません。 SiteMinder は、選択されたパートナーシップの名前を使用します。

次の手順に従ってください:

1. ［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］を選択します。

   ［パートナーシップ］ダイアログ ボックスが表示されます。

2. リスト内で該当するエントリの横の［アクション］プルダウン メニューをクリックし、［メタデータのエクスポート］を選択します。

   ［メタデータのエクスポート］ダイアログ ボックスが開きます。

3. ダイアログ ボックスのフィールドに入力します。

   ［アクティブ］ステータスのパートナーシップをエクスポートしている場合、ほとんどのフィールドは読み取り専用です。 ［有効期間］フィールドおよびエイリアス ドロップダウン リストのみが変更可能です。

4. ［エクスポート］をクリックして終了します。

5. メタデータ ファイルを開く、または保存することを要求するダイアログ ボックスが表示されます。 メタデータ ファイルを開いて表示できます。

6. ローカル システム上の XML ファイルにデータを保存します。

メタデータは指定された XML ファイルにエクスポートされます。

第 9 章: パートナーシップのフェデレーション ユーザ の識別

このセクションには、以下のトピックが含まれています。

アサーティング パーティでのフェデレーション ユーザ設定

ローカル エンティティがアサーティング パーティである場合、［フェデレーション ユーザ］ダイアログ ボックスはパートナーシップ ウィザードの 2 番目の手順です。この手順では、リモート サイトでターゲット リソースへのアクセスを許可するユーザを指定できます。

次の手順に従ってください:

注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。

1. ［Federation ユーザ］グループ ボックスのテーブルの［ディレクトリ］列のリストから、ユーザ ディレクトリを選択します。

   プルダウン リストは、前のダイアログ ボックスで指定したディレクトリの数に応じて、1 つ以上のディレクトリ エントリで構成されています。

2. ［ユーザ クラス］列のユーザ クラスを選択します。このエントリは、認証できる個別ユーザまたはユーザ グループのカテゴリを指定します。 このフィールドのオプションは、ユーザ ディレクトリのタイプ（LDAP または ODBC）に依存します。各ユーザ クラスの説明および例については、「ユーザ クラス」の表を参照してください。

3. 名前を入力するか、または［ユーザ名/フィルタ条件］列でフィルタします。この列の値を使用して、フェデレーション ユーザを認証する元のユーザ グループまたはユーザを特定することができます。このエントリは、［ユーザ クラス］列で選択する値によって異なります。 名前およびフィルタの例については、この手順の最後の表を参照してください。

4. （オプション）エントリの［除外］を選択すると、このユーザ クラスを除外することを示すことができます。デフォルトはディレクトリ内のすべてのユーザを含めることです。

   注: 2 つの条件が競合した場合、除外条件は、包含条件より常に優先されます。

5. （オプション）同じディレクトリまたは別のユーザ ディレクトリに対して別のユーザ クラスを指定するには、［行の追加］をクリックします。

ユーザの選択が完了します。

ユーザ クラス エントリの例

LDAP の例

エントリを指定する場合は、LDAP フィルタ構文を使用します。

ユーザ クラス	有効なエントリ
ユーザ	ユーザの識別名。 例： uid=user1,ou=People,dc=example,dc=com
グループ	リストから選択されたグループ。 例： ou=Sales,dc=example,dc=com
組織単位	リストから選択された組織単位。 例： ou=People,dc=example,dc=com
ユーザ プロパティのフィルタ	LDAP フィルタ。 現在のユーザは検索の出発点です。 **例 1：** mail=user@example.com **例 2：** (|(mail=*@.example.com)(memberOf=cn=Employees,ou=Groups,dc=example,dc=com))

ユーザ クラス	有効なエントリ
グループ プロパティのフィルタ	LDAP フィルタ。 現在のユーザがフィルタに一致するグループの 1 つのメンバである場合、現在のユーザが許可されます。SiteMinder レジストリで設定されているグループ用のオブジェクト クラスは、フィルタと組み合わされます。 **例 1：** ビジネス カテゴリが「CA Support」であるグループのメンバであるユーザを許可するには、「businessCategory=CA Support」と入力します。 **例 2：** 説明に「Administrator」が含まれ、ビジネス カテゴリが「Administration」であるグループのメンバであるユーザを許可するには、「(|(description=*Administrator*)(businessCategory=Administration))」と入力します。 **注**: グループ作業の属性には、検索条件として機能しないものもあります。
OU プロパティのフィルタ	LDAP フィルタ。現在のユーザがフィルタに一致する組織単位に属する場合、現在のユーザが許可されます。 SiteMinder レジストリで設定されている組織単位用のオブジェクト クラスは、フィルタと組み合わされます。 **例 1：** 郵便番号が「12345」の組織単位内のユーザを許可するには、「postalCode=12345」と入力します。 **例 2：** 優先配布方法が「phone」で終わり、市区町村が「London」の組織単位内のユーザを許可するには、「(|(preferredDeliveryMethod=*phone)(l=London))」と入力します。
任意の項目のフィルタ	LDAP フィルタ。 現在のユーザがフィルタに一致する場合、現在のユーザが許可されます。 **例 1：** 部門が「CA Support」のユーザを許可するには、「department=CA Support」と入力します。 **例 2：** グループ「Administrators」のメンバで、部門番号が「123」または「789」のユーザを許可するには、「(&(memberof=cn=Administrators,ou=Groups,dc=example,dc=com)(|(departmentNumber=123)(departmentNumber=789)))」と入力します。

ODBC の例

クエリを指定する場合、SQL 構文を使用します。

ユーザ クラス	有効なエントリ
ユーザ	ユーザの［名前］列の値。 現在のユーザがエントリに一致する場合、現在のユーザが許可されます。 例： user1
グループ	ユーザ グループの［名前］列の値。 現在のユーザがクエリに一致するグループのメンバである場合、現在のユーザが許可されます。 例： 管理者
クエリ	SQL SELECT ステートメント。 現在のユーザがクエリに一致する場合、現在のユーザが許可されます。 **例 1：** ユーザ ID が「user1」 エントリ： SELECT * FROM SmUser 結果のクエリ： SELECT * FROM SmUser WHERE Name = 'user1' **例 2：** ユーザ ID が「user1」 エントリ： SELECT * FROM SmUser WHERE Status LIKE 'Active%' 結果のクエリ： SELECT * FROM SmUser WHERE Status LIKE 'Active%' AND Name = 'user1' **例 3:** ユーザ ID が「user1」 エントリ： FROM SmUser WHERE Location IN ('London', 'Paris') 結果のクエリ： SELECT * FROM SmUser WHERE Location IN ('London', 'Paris') AND Name = 'user1'

依存パーティでのユーザ識別

依存パーティでは、パートナーはローカル ユーザ ディレクトリでユーザを特定できる必要があります。 ユーザ ディレクトリでユーザを特定することは、特定のプロセスです。［ユーザ識別］ダイアログ ボックスでユーザ特定用の ID 属性を設定します。

ポリシー サーバは、以下のいずれかの特定プロセスのメソッドを使用できます。

- アサーションから［名前 ID］の値を抽出します。
- アサーションの特定の属性の値を使用します。
- Xpath クエリが取得する値を使用します。

  Xpath クエリによってアサーションから［名前 ID］以外の属性が特定および抽出されます。

アサーションから抽出される属性が決定したら、この属性を検索仕様に含めます。 特定プロセスが成功した後で、ポリシー サーバはユーザのセッションを生成します。

SAML 2.0 の場合は、アサーティング パーティによるユーザ識別子の作成を可能にする AllowCreate 機能 (P. 97)を設定することもできます。

依存パーティでのユーザ識別の設定

依存パーティがローカル ユーザ ディレクトリでユーザを特定できるように、ユーザ識別を設定します。

次の手順に従ってください:

1. 特定に使用する以下のいずれかの属性を選択します。
   - 名前 ID
   - 以前に生成したドロップダウン リストの属性

     リモート アサーティング エンティティが、属性を含むメタデータに基づいて作成された場合、リストが生成されています。
   - 入力する属性。

     このオプションは、メタデータを使用できず、かつリモート アサーティング エンティティに属性が含まれていない場合に、最も使用される可能性があります。
   - Xpath クエリ

   フィールドの説明については、［ヘルプ］をクリックしてください。

2. （オプション - SAML 2.0 のみ）［IDP に新規ユーザ識別子の作成を許可する］を選択します。

   この属性によりアサーティング パーティは名前 ID の新しい値を生成します（この機能がアサーティング パーティで有効になっている場合）。 アサーティング パーティの［名前 ID 形式］エントリは永続識別子である必要があります。

3. （オプション - SAML 2.0 のみ）識別子を上書きするクエリ パラメータを選択します。

   この設定によって、依存パーティは AllowCreate クエリ パラメータを送信して認証リクエストで設定された AllowCreate 属性の値を上書きします。 識別子の代わりにクエリ パラメータを使用することにより、パートナーシップ設定を変更せずに、AllowCreate 属性の値を変更できます。

   注: アイデンティティ プロバイダがこのクエリ パラメータ設定を適用できるように、［IDP に新規ユーザ識別子の作成を許可する］チェック ボックスを選択します。

4. リストされた各ディレクトリに対してディレクトリ検索仕様を指定します。 検索仕様の 2 つの例を以下に示します。

   LDAP の例

   uid=%s

   ODBC の例

   name=%s

5. ［次へ］をクリックして、パートナーシップ設定を続行します。

ユーザ識別用 AllowCreate の採用（SAML 2.0）

SAML 2.0 AllowCreate 機能は、SP の［ユーザ識別］設定のオプション設定です。 AllowCreate 属性を認証リクエストに含めることによって、アイデンティティ プロバイダは SP 用のユーザ識別子を作成できます。

SP は認証リクエストをアイデンティティ プロバイダに送信することで、シングル サインオンを開始できます。 リクエストの一部として、サービス プロバイダは、true に設定されている AllowCreate という名前の属性を含めることができます。 サービス プロバイダは、ユーザの ID を取得する必要があります。認証リクエストを受信するとすぐに、アイデンティティ プロバイダはアサーションを生成します。 アイデンティティ プロバイダは、［名前 ID］として使用されるアサーション属性に適したユーザ レコードを検索します。 アイデンティティ プロバイダが名前 ID 属性の値を見つけられない場合、名前 ID 用の一意の永続識別子を生成します。 識別子を生成させるために、アイデンティティ プロバイダで 許可/作成機能を有効にします。 アイデンティティ プロバイダは、アサーションを一意の識別子と共に SP に返します。

AllowCreate クエリ パラメータを有効にして AllowCreate 属性の値と取り換えられます。 クエリ パラメータの使用によって、パートナーシップの非アクティブ化、編集、および再アクティブ化を行わずに設定済みの AllowCreate 設定を上書きできます。 クエリ パラメータにより機能の実装がより柔軟になります。

第 10 章: アサーティング パーティでのアサーションの設定

このセクションには、以下のトピックが含まれています。

アサーション設定

パートナーシップ ウィザードの［アサーションの設定］手順では、以下の設定を定義します。

名前 ID

必須のアサーション属性である名前 ID 属性によって、一意の方法でユーザが識別されます。名前 ID 形式は、フェデレーション パートナーがサポートする識別子タイプを示します。名前 ID タイプは、名前 ID 形式に関連付けられているユーザ プロファイル属性を指定します。ユーザ プロファイル属性は、ユーザ ストアまたはセッション ストアにあります。

アサーション属性

サーブレット、Web アプリケーションまたは他のカスタム アプリケーションは、属性を使用して、カスタマイズされたコンテンツを表示する、または他のカスタム機能を有効にすることができます。 属性が Web アプリケーションで使用されると、依存パーティでのユーザのアクティビティが制限される場合があります。 たとえば、上限金額（Authorized Amount）という名前の属性変数は、ユーザが依存パーティで使用できる上限金額に設定されます。

属性は、<AttributeStatement> エレメントまたは <EncryptedAttribute> エレメントで指定されます。 属性の形式は、名前/値のペアになっています。 属性はまた、HTTP ヘッダまたは HTTP Cookie として利用できるようになります。

注: 属性ステートメントはアサーションに必要ありません。

属性ステートメントに対して別の種類の属性を設定できます。 属性の種類には以下のものが含まれます。

- ユーザ属性
- DN 属性
- 静的データ
- セッション属性

  セッション属性 (P. 103)は、それらがセッション ストアに保持されている場合のみアサーションで利用可能です。

また、式を設定してアサーション属性を変換することもできます。 この機能は、クレーム変換 (P. 111)と呼ばれます。

依存パーティはアサーションを受け取ると、その属性値をアプリケーションで使用できるようにします。

アサーション ジェネレータ プラグイン

通常、属性はユーザ ディレクトリ レコードに含まれますが、外部データベースまたはアプリケーション コンテンツなどの他のソースの属性がアサーションに含まれる場合があります。 さまざまなソースから属性を取り込むアサーション ジェネレータ プラグインを作成できます。 アサーション ジェネレータ プラグインは、アサーション ジェネレータ プラグインのインターフェースに従って作成するカスタム コードの一部です。

プラグイン作成の詳細については、「*Programming Guide for the Federation Java SDK*」を参照してください。

アサーション オプションの設定

アサーティング パーティでアサーション オプションを設定します。

次の手順に従ってください:

1. パートナーシップ ウィザードの［アサーションの設定］手順に移動します。
2. ［名前 ID］セクションの設定を行います。

   依存パーティは、これらの値を使用してアサーション内の名前 ID 値を解釈します。

   選択した［名前 ID タイプ］オプションに応じて、エントリに適切な値を入力します。

   スタティック属性

   ［値］フィールドに定数の文字列を入力します。

   ユーザ属性

   ［値］フィールドに、有効なユーザ ストア属性を入力します。 たとえば、「mail」を入力します。

   セッション属性

   ［値］フィールドに、有効なセッション ストア属性を入力します。

   DN 属性（LDAP のみ）

   ［値］フィールドに、有効な LDAP ユーザ ディレクトリ属性を入力します。 また、DN 指定フィールドに有効な DN を入力します。 たとえば、DN 属性は cn=JaneDoe で、指定は ou=Engineering,o=ca.com です。
3. （オプション - SAML 2.0 のみ）アサーティング パーティが名前 ID の値を作成できるように、［ユーザ識別子の作成を許可］を選択します。この機能を動作させるには、依存パーティからの認証リクエストに AllowCreate 属性が含まれている必要があります。

   注: このオプションを選択する場合、［名前 ID 形式］の値が［永続 ID］である必要があります。

4. （オプション）［アサーション属性］テーブルの［行の追加］をクリックして、アサーションの 1 つ以上の属性を指定します。オプションで、属性を暗号化できます。

   テーブルの入力のヘルプについては、いくつかのアサーション属性の例 (P. 102)を参照してください。属性テーブルの列に関する詳細については、［ヘルプ］をクリックします。

   注: LDAP ユーザ ストア属性については、アサーションに複数の値を持つユーザ属性を追加できます。［ヘルプ］では、複数値のユーザ属性を指定する方法を説明します。

5. （オプション） CA SiteMinder® Federation Java SDK を使用して、アサーション ジェネレータ プラグインを作成した場合は、［アサーション ジェネレータ プラグイン］セクションのフィールドに入力します。

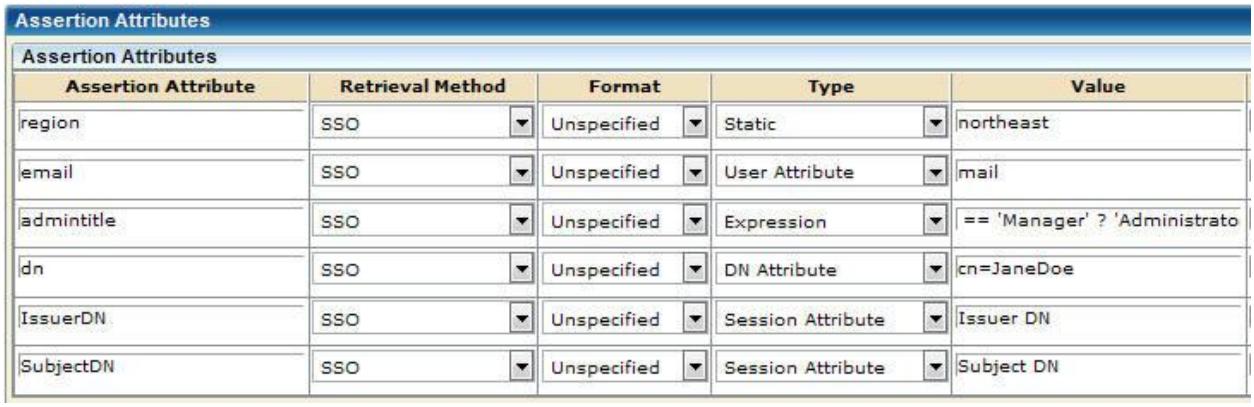
   プラグインの作成については、「*Programming Guide for the Federation Java SDK*」を参照してください。

6. ［次へ］をクリックして、パートナーシップ設定を続行します。

アサーション属性の設定の例

以下の画像は、アサーション属性エントリのいくつかの例を示します。この画面は、SAML 2.0 パートナーシップ用です。 SAML 1.1 の場合の画面もこれに似ていますが、［取得方法］列と［フォーマット］列がありません。代わりに、［ネームスペース］列が存在します。

注: DN 属性の例には［DN 指定］列が含まれており、エントリは ou=Engineering,o=ca.com です。 この列は、この画像では表示されていません。



セッション属性をアサーションに追加する方法

ポリシー サーバは、ユーザの認証後の動的なユーザ情報を保持するためにセッション ストアを使用します。 格納された情報には、認証コンテキスト情報、SAML 属性、ユーザを認証するサードパーティ IdP、および OAuth 認証からのクレームなどがあります。 ポリシー サーバは、ユーザ トークンの生成またはポリシーの決定にこの情報を使用できます。

フェデレーション シングル サインオンの場合、ポリシー サーバは、リクエストされたアプリケーションをカスタマイズするために属性をセッション ストアからアサーションに追加できます。

セッション属性は、以下の展開で格納されます。

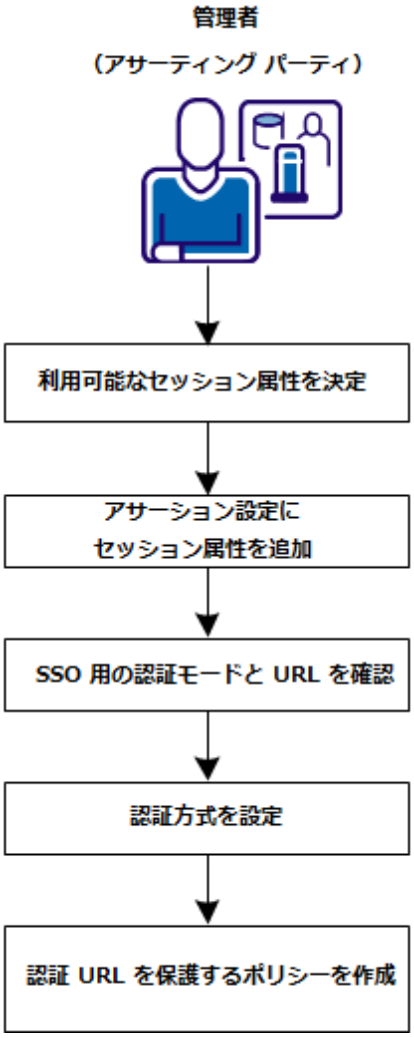
- 委任認証以外の展開。

  ローカル システムまたは外部のサードパーティはユーザを認証しますが、システムはそれをローカル認証と見なします。 ローカル認証の展開は、認証モードがシングル サインオン設定でローカルであることを必要とします。また、アクセス ポリシーによって認証 URL を保護する必要があります。 ポリシーの認証方式は、セッション属性を保持するように設定されます。

- 委任認証の展開

  外部のサードパーティがユーザを認証できます。 サードパーティのパートナーは、セッション ストアに格納されるユーザ情報を返します。

以下の図は、セッション属性を設定し、アサーションに追加するために必要な手順を示しています。



セッション属性のサポートに対して、以下の手順を実行します。

1. 利用可能にするセッション属性を決定します。（P. 105）
2. セッション属性をアサーション設定に追加します。（P. 105）
3. SSO の認証モードと URL を確認します（P. 106）。
4. セッション属性を保持するための認証方式を設定します。（P. 107）
5. 認証 URL を保護するポリシーを作成します。（P. 108）

利用可能なセッション属性の特定

フェデレーション管理者として、パートナーシップによって使用されるセッション属性を識別します。 データベースやユーザ ディレクトリなどの認証ソースを操作し、使用可能な属性をよく理解してください。

アサーション設定へのセッション属性の追加

セッション属性をアサーション設定に追加します。IdP から SP へのパートナーシップなどの設定は、アサーティング パーティにあります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. パートナーシップ ウィザードの［アサーションの設定］手順に移動します。
3. ［アサーション属性］セクションで、［行の追加］をクリックします。
4. セッション属性を設定するには、テーブル内の設定を完了します。例:

   アサーション属性

   IssuerID

   取得メソッド

   SSO

   形式

   未指定

   タイプ

   セッション属性

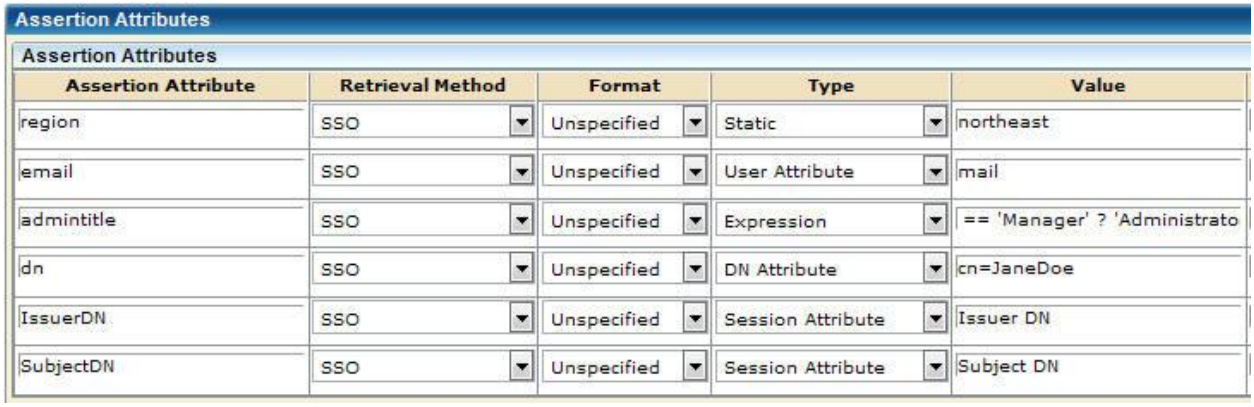
   値

   IssuerID

   属性テーブルの詳細については、［ヘルプ］をクリックします。
5. 必要数のエントリ用に行を追加します。
6. （オプション）。 ［暗号化］を選択して属性を暗号化します。
7. ［次へ］をクリックして、［SSO と SLO］手順に移動します。

管理 UI でのセッション属性の例

以下の図の最後の 2 つのエントリは、セッション属性エントリの例を表しています。 この画面は、SAML 2.0 パートナーシップ用です。 SAML 1.1 の場合の画面もこれに似ていますが、［取得方法］列と［フォーマット］列がありません。 代わりに、［ネームスペース］列が存在します。



SSO の認証モードと URL の確認

パートナーシップの認証モードおよび認証 URL が正しく設定されていることを確認します。

注: この手順では、その他の必要な SSO 設定が設定済みであると仮定しています。

次の手順に従ってください:

1. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
2. ［認証］セクションで、以下のフィールドの設定を確認します。

   認証モード

   ローカル

認証 URL

この URL は、たとえば以下のように redirect.jsp ファイルを指している必要があります。

http://*myserver*.idpA.com/siteminderagent/redirectjsp/redirect.jsp

myserver

Web エージェント オプション パックまたは SPS フェデレーション ゲートウェイで Web サーバを識別します。redirect.jsp ファイルは、アサーティング パーティでインストールされる Web エージェント オプション パックまたは SPS フェデレーション ゲートウェイに含まれています。

ポリシーでこのリソースを保護します。

3. ［確認］手順に移動し、［完了］をクリックします。

セッション属性を保持するための認証方式の設定

認証 URL を保護する認証方式の設定 セッション属性を保持するための方式を有効にします。 この手順は、システムでセッション属性を格納する場合に必要です。

次の手順に従ってください:

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. ［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］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. セッション属性を保持できる認証方式テンプレート（ユーザ名およびパスワードのみでなく、もっと詳しい情報が必要）を選択します。

   たとえば、X.509 証明書認証方式には、証明書の SubjectDN と IssuerID が必要です。OAuth 認証方式では、名および姓などの情報が必要です。この情報は、セッション ストアに保持し、アサーションに追加することができます。

使用できる認証方式テンプレートは次のとおりです。

- OpenID
- OAuth
- すべての X.509 認証テンプレート
- カスタム方式

5. 方式に固有のフィールドおよびコントロールに入力します。

   フィールドの説明については、［ヘルプ］をクリックしてください。

6. ダイアログ ボックスの［方式のセットアップ］セクションにある［認証セッション変数を保持する］を選択します。
7. ［サブミット］をクリックして方式を保存します。

認証 URL を保護するポリシーの作成

認証 URL を保護するポリシーで、セッション属性を保持する認証方式を使用します。 保護されているリソースをユーザがリクエストした場合、ポリシーによってユーザの認証に必要なアクションがトリガされます。 システムは、ユーザによってセッション変数として指定された認証情報を格納します。

まずアサーティング パーティのポリシー ドメインを作成し、ユーザを割り当てます。 また、既存のアサーティング パーティ ドメインを変更できます。

次の手順に従ってください:

1. ［ポリシー］-［ドメイン］-［ドメイン］をクリックします。

   ［ドメイン］ページが表示されます。

2. アサーティング パーティのドメインを選択し、それを変更します。

3. ユーザ ディレクトリがドメインの一部になっていることを確認します。 そうでない場合は、［追加/削除］をクリックしてユーザ ディレクトリを追加します。

   ［使用可能なメンバ］リストから 1 つ以上のユーザ ディレクトリを選択できます。 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、次に 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。

   注: ユーザ ディレクトリを作成し、それをドメインに追加するには、［作成］をクリックします。

4. ［サブミット］をクリックします。

ドメインが設定されます。

認証 URL ポリシーのレルムおよびルールの作成

フェデレーション ドメインについては、レルムを作成し、それを Web エージェントに関連付けます。

次の手順に従ってください:

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. ［レルムの作成］をクリックします。
3. 変更するドメインを選択して、［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。

   レルムが SSO 認証 URL 用であることを示す名前を指定します。

5. ［エージェント/エージェント グループの検索］をクリックして、エージェントを選択します。
6. 適切な Web エージェントを選択し、［OK］をクリックします。
7. redirect.jsp のリソース フィルタの指定 例：

   /siteminder/redirectjsp/redirect.jsp

8. 以下のフィールドに値を入力します。

   デフォルト リソース保護

   保護されている

   認証方式

   認証 URL を保護するために設定した認証方式を選択します。 この方式は、セッション属性を保持するために設定したものです。

9. ［ルール］セクションでルールを作成します。
   a. ルールの名前を指定します。
   b. その他の設定はデフォルトのままにします。
10. 他の設定オプションをスキップします。
11. ［完了］をクリックします。

レルムおよびルームの設定は完了です。

認証 URL ポリシーの作成

認証 URL を保護するためのポリシーを作成します。 ポリシー コンポーネントは一緒に動作し、リソースを保護します。

ポリシーを作成した後、ユーザおよびルールを追加します。

次の手順に従ってください:

1. ［ポリシー］-［ドメイン］-［ドメイン］をクリックします。
2. ドメインを検索します。

   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. アサーティング パーティのドメインを選択します。
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。

   ［ポリシー］ページが表示されます。
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. ［ユーザ］タブから、個々のユーザ、ユーザ グループ、または両方を追加します。ユーザはドメインに関連付けられたユーザ ディレクトリのメンバです。

   各ユーザ ディレクトリ グループ ボックスから、［メンバの追加］、［エントリの追加］、［すべて追加］を選択します。 使用するメソッドに応じて、ダイアログ ボックスが表示され、ユーザの追加が可能になります。

   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、自動的には表示されません。 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には、検索ユーティリティを使用します。

   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。

9. ［ルール］タブからルールを追加します。
10. 認証 URL 用に作成したルールを選択し、［OK］をクリックします。

    ルールのレスポンスを設定する要求はありません。
11. ［サブミット］をクリックして、設定を完了します。

ポリシー設定が完了しました。

アサーション属性、シングル サインオン、およびポリシー設定は連携して動作し、アサーションでセッション属性を使用できるようにします。

アサーティング パーティでクレーム変換を設定する方法

クレーム変換では、連携したシングル サインオン トランザクションの際にクレームを操作します。 クレームは属性とも呼ばれ、属性のカスタマイズおよびパートナーでのユーザ操作性の向上を支援します。

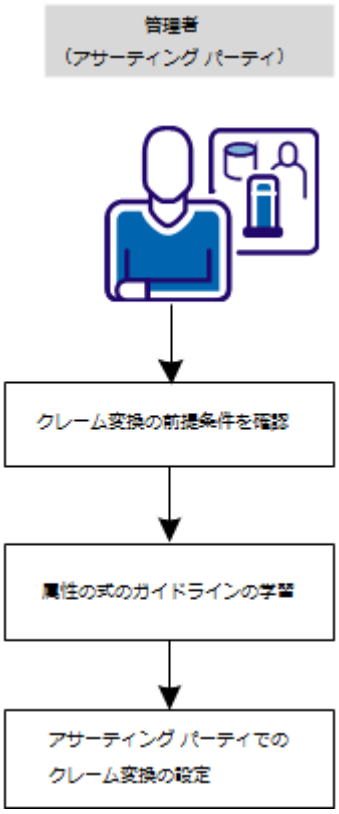
アサーション属性を変更すると、依存パーティがユーザ情報に対応し、ターゲット アプリケーションでユーザ情報を使用できるようになります。たとえば、クレーム変換によって別のドメインにある別のパートナーでロールを関連付けることができます。 あるドメインで、ユーザはエンジニアリング マネージャであり、EngineerAdmins という名前のグループに属しています。ただし、依存パーティは同じロールを DevelAdmins と識別します。 アサーティング パーティは、アサーションを発行する前にロール属性を変更します。この時点で、このユーザは依存パーティのアプリケーションで認識できる DevelAdmins ロールで識別されるようになります。

クレーム変換は、アサーションの作成時にローカルのアサーティング パーティで行われます。 この機能はパートナーシップ単位で設定します。ローカル パーティまたはリモート パーティのどちらでアサーションを生成するかを変更できます。 クレームは、パートナーシップに対して設定する式に基づいて変換されます。 この式は、ユーザ ストアおよび SiteMinder セッション ストアからのユーザ情報に依存します。

ソフトウェアでは、アサーション属性に対して以下の 3 つの変更を実行できます。

- **変換：** アサーション属性の値を別の値に変更します。
- **追加：** アサーション属性が存在しない場合に、アサーション属性を追加します。
- **削除：** 条件に基づいてアサーション属性を削除します。

以下の図は、設定手順を示しています。



クレーム変換を設定するには、以下の手順に従います。

1. クレーム変換の前提条件を確認します。（P. 113）
2. 属性式のガイドラインについて学習します。（P. 114）
3. アサーティング パーティでクレーム変換を設定します。（P. 115）

クレーム変換の前提条件

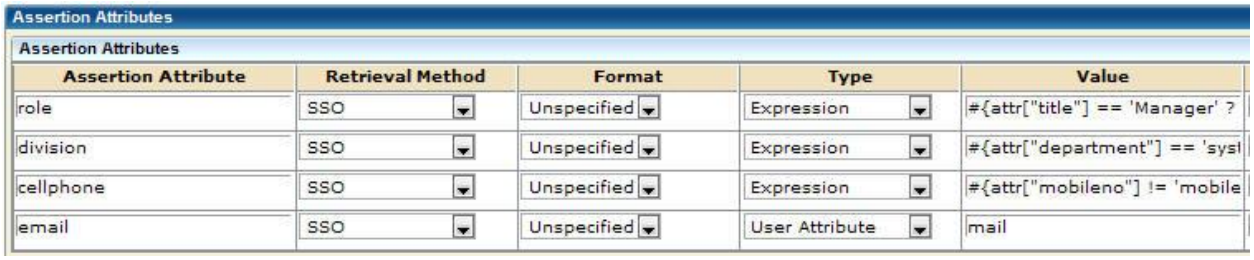
クレーム変換を設定する前に、以下の前提条件を確認してください。

- 使用可能なユーザ ストア属性およびセッション ストア属性に精通している必要があります。
- 依存パーティがアサーションで受信する属性を特定する。
- Unified Expression Language のオープン ソース バージョンである Java Unified Expression Language （JUEL）を理解している。

属性式のガイドラインについての説明

式はソフトウェアにアサーション属性を操作する方法を指示するためのルールです。 式は、アサーション属性の変更、追加、削除をソフトウェアに指示します。 式は Java Unified Expression Language（JUEL）を使用して作成します。 JUEL 式エバリュエータは設定された式を確認し、結果としてアサーション属性を生成します。

管理 UI の［アサーション属性］テーブルで式を定義します。 このテーブルを表示するには、パートナーシップ ウィザードの［アサーションの設定］手順に移動します。 このテーブルを以下の図に示します。



アサーション属性テーブルの［値］列に式を入力します。 式内の属性はすべて、ユーザ ストア属性またはセッション ストア属性です。

通常、式は、条件に基づいて作動します。 条件が満たされた場合は、指定されたクレームの変更が行われます。 たとえば、受信アサーションには「ロール」属性が含まれます。 「role」アサーション属性を変更する式は以下のとおりです。

#{attr["title"] == 'manager' ? 'administrator' : attr["title"]}

式 **#{attr["title"] == 'manager'** の最初の部分では、ログインされたユーザーの役職が「マネージャ」であるかどうかを特定するようにソフトウェアに指示します。ユーザ ディレクトリで検索が行われます。 この条件が満たされた場合は、式の次の部分である **? 'administrator' :** で role アサーション属性に値「administrator」を割り当てます。 この条件が満たされなかった場合は、式の最後の部分の **attr["title"]}** で、ユーザ属性「title」の値を「manager」のままにします。この値「manager」はアサーション属性「role」に割り当てられます。

注: attr["title"] という構文の代わりに静的な値を使用することもできます。前の例における 'administrator' が静的な値です。

この例では、「role」属性がすでにアサーションにあると仮定しています。そのため、この式は既存の属性の変換です。「role」がアサーションの一部でない場合、ソフトウェアは role 属性をアサーションに追加します。

式の構文

式は適切な構文を使用して作成します。

- ユーザ ストア属性は文字列 **attr**["*attribute_name*"] で表します。
- セッション ストア属性は文字列 **session_attr**["*attribute_name*"] で表します。
- クレームの削除には引数「DELETE」を使用します。

attr および **session_attr** プレフィックスには、小文字を使用します。 属性名では、大文字と小文字は区別されません。

また、以下の JUEL の条件付き演算子に注意してください。

オペレータ	意味
条件値 ? value1 : value2	条件値は value1 または value2 のいずれかに対応します。
!=	等しくない
==	等しい

重要: 式に含まれる属性は、ユーザ ディレクトリまたはセッション ストアで使用可能である必要があります。属性が正しくないと、システムによって対応する属性として空白が挿入されます。 アサーションの生成は失敗しません。

式の例については、「アサーティング パーティでのクレーム変換の設定 (P. 115)」を参照してください。

アサーティング パーティでのクレーム変換の設定

パートナーシップ レベルで式を定義します。 これらの式の結果により、アサーションの属性が変更、追加、削除されます。 ルールが定義された後、アサーションは変更され、依存パーティに送信されます。 クレーム変換を設定しない場合は、アサーション属性が依存パーティにそのまま渡されます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［Federation］-［パートナーシップ］を選択します。
3. 変更するパートナーシップを選択します。選択できるパートナーシップには以下のものがあります。
   - ローカル プロデューサからリモート コンシューマ
   - ローカル IdP からリモート SP
   - ローカル IP からリモート RP
4. パートナーシップ ウィザードの［アサーションの設定］手順に移動します。

   ［アサーション属性］セクションで、［行の追加］をクリックします。
5. 行の以下のフィールドに特に注意してください。各フィールドの詳細な説明については、［ヘルプ］をクリックしてください。

   アサーション属性

   アサーション属性を入力します。 この列の値はすべてアサーション属性です。 すでにアサーション内にある属性はそのままアサーションに残りますが、その属性は、設定された式に基づいて新しい値に設定されます。DELETE 式を設定した場合のみ、属性はアサーションから削除されます。

   取得メソッド

   デフォルトの SSO のままにします。

   形式

   アサーションに追加される属性用の形式を指定します。 フォーマット オプションはエンティティの SAML プロファイルによって異なります。

   タイプ

   式

   クレーム変換には常にこの値を使用します。

値

アサーション属性に対する変更を反映する式を入力します。

クレームの式の作成 (P. 114)に関するガイドラインと以下の例を確認してください。

- アサーションのクレーム変換。 (P. 117)
- アサーションへのクレームの追加 (P. 119)。
- アサーションからのクレームの削除 (P. 121)。

6. （SAML 2.0 およびトークン タイプが SAML 2.0 の WSFED のオプション）。アサーション属性を暗号化するには、［暗号化］を選択します。アサーティング パーティは、パートナーシップ設定で指定された証明書を使用してアサーションを暗号化します。

   依存パーティは、証明書と関連付けられている秘密キーを使用してアサーション属性を復号化します。

7. 設定するアサーション属性に対して行を必要なだけ追加します。

クレーム変換は、パートナーシップ内の設定されたエントリに基づいて実装されます。

アサーションのクレーム変換

クレーム変換によって、アサーション属性の値が別の値に変更されます。

注: 以下の例では、アサーション属性、タイプ、および値のエントリのみを示しています。

変換例 1

以下の例では、アサーションに「title」属性があると仮定しています。 この表はユーザ ストアのユーザ属性を示しています。

ユーザ ディレクトリ属性	属性値
role	admin
admintitle	SeniorAdmin
supertitle	SuperUser

以下の設定を使用して、既存の役職属性の値を変換します。

アサーション属性

title

タイプ

式

値

#{attr[“role”] == ‘admin’ ? attr[“admintitle”] : attr[“supertitle”]}

結果： この式は、「role」ユーザ属性が「admin」と設定されているという条件を表しています。この条件が満たされた場合、アサーション属性「title」には「admintitle」属性の SeniorAdmin という値が設定されます。ロールが「admin」以外である場合は、「title」属性は「supertitle」属性の値である SuperUser になります。

変換例 2

以下の例では、アサーションに ContactNo 属性があると仮定しています。

ユーザ ディレクトリ属性	属性値
homephone	555-3344
mobile	555-8888

以下の設定を使用して、既存の役職属性の値を変換します。

アサーション属性

ContactNo

タイプ

式

値

#{attr["homephone"] == '555-3344' ? attr["mobile"] : attr["homephone"]}

結果： この式は、ログイン ユーザの「homephone」ユーザ属性が 555-3344 に設定されているという条件を表しています。 この条件が満たされた場合、アサーション属性は「mobile」属性の値である 555-8888 に設定されます。 条件が満たされない場合、「homephone」の値は変更されません。

注: セッション属性を使用する式を設定するには、attr["*attribute_name*"] を session_attr["*attribute_name*"] に置き換えます。 以下に例を示します。

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

アサーションへのクレームの追加

アサーション属性がまだ存在しない場合に、アサーション属性を追加することができます。

追加例 1

以下の例は、アサーション属性「役職」がアサーションにないと仮定しています。

ユーザ ディレクトリ属性	属性値
role	admin
admintitle	director
supertitle	executive

以下の設定では title 属性をアサーションに追加します。

アサーション属性

title

タイプ

式

値

#{attr[“role”] == ‘admin’ ? attr[“admintitle”] : attr[“supertitle”]}

結果： この式は、ログイン ユーザの「role」属性が admin に設定されているという条件を表しています。 この条件が満たされた場合、アサーション属性「title」がアサーションに追加され、その値は「admintitle」属性の値である「director」に設定されます。role が「admin」以外である場合は、アサーション属性「title」が追加されますが、その値は「supertitle」属性の値である「executive」になります。

追加例 2

以下の例は、アサーション属性「smtitle」がアサーションにないと仮定しています。

ユーザ ディレクトリ属性	属性値
title	manager

アサーション属性

smtitle

タイプ

式

値

#{attr["title"] == 'manager' ? 'federation administrator' : attr["title"]}

結果： ログイン ユーザの title が「manager」である場合は、「smtitle」がアサーションに追加され、その値が「federation administrator」に設定されます。疑問符の後ろには、構文 attr["attribute_name"] を使用する代わりに静的な値を入力することもできます。 この例では、静的な値は federation administrator です。

注: セッション属性を使用する式を設定するには、attr["*attribute_name*"] を session_attr["*attribute_name*"] に置き換えます。 以下に例を示します。

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

アサーションからのクレームの削除

アサーション属性を削除できます。

削除例 1

2 つのエントリを設定して、admintitle および supertitle アサーション属性を削除します。

ユーザ ディレクトリ属性	属性値
role	admin または superuser

ユーザ ディレクトリ属性	属性値
title	administrator
su	superuser

アサーション属性

admintitle

タイプ

式

値

#{attr[“role”] == ‘superuser’ ? ‘DELETE’ : attr[“title”]}

結果： この式は、「role」ユーザ属性に基づく条件です。 ログイン ユーザのロールが superuser の場合は、アサーション属性「admintitle」を削除します。ロールが superuser でない場合は、タイトル アサーション属性を、タイトル ユーザ ディレクトリ属性の値である、値「administrator」に設定します。

アサーション属性

supertitle

タイプ

式

値

#{attr[“role”] == ‘admin’ ? ‘DELETE’ : attr[“su”]}

結果： この式は、「role」ユーザ属性に基づく条件です。 ログイン ユーザ ロールが "admin" である場合は、アサーション属性 "supertitle" を削除します。 ロールが "admin" でない場合は、supertitle アサーション属性を su ユーザ ディレクトリ属性の値である、値「superuser」に設定します。

削除例 2

以下の例では、1 つの式に追加と削除を組み合わせています。

ユーザ ディレクトリ属性	属性値
title	manager

アサーション属性

ManagerName

タイプ

式

値

#{attr["title"] != 'Manager' ? attr["manager"] : 'DELETE'}

結果： ログイン ユーザのユーザ属性 title が「manager」でない場合、ManagerName 属性をアサーションに追加します。 ただし、ログイン ユーザの title が manager である場合は、ManagerName がアサーションの一部であると想定して、ManagerName 属性を削除します。

注: セッション属性を使用する式を設定するには、attr["*attribute_name*"] を session_attr["*attribute_name*"] に置き換えます。 以下に例を示します。

#{session_attr[“att1”] == ‘admin’ ? session_attr[“attr2”] : attr[“attr3”]}

アサーション コンテンツのカスタマイズ化

AssertionGeneratorPlugin の実装

カスタム アサーション ジェネレータ プラグインの作成の最初の手順は、AssertionGeneratorPlugin インターフェースの実装です。以下の要件が実装クラスに適用されます。

- 実装では、パラメータが含まれないデフォルトのパブリック コンストラクタ メソッドを提供します。
- 実装はステートレスである必要があり、その結果、多くのスレッドで単一のプラグイン クラスが使用可能となります。
- 実装には、customizeAssertion メソッドへのコールが含まれる必要があります。 要件に示されているように、これらのメソッドの既存の実装は上書きできます。 サンプル プログラムを参照してください。
- 構文の要件および customizeAssertion メソッドに渡されるパラメータ文字列の使用は、カスタム オブジェクトで設定されます。

注: フォルダ *federation_sdk_home*¥¥sample¥com¥ca¥federation¥sdk¥plugin¥sample には 2 つのサンプル実装クラスが含まれています。

アサーション ジェネレータ プラグインの展開

AssertionGeneratoPlugin インターフェースの実装クラスをコード化した後、それをコンパイルし、CA SiteMinder® Federation が実行可能ファイルを検索できることを確認します。

アサーション ジェネレータ プラグインを展開する方法

1. 以下のいずれかの方法でアサーション プラグイン コードをコンパイルします。

   - サンプル プラグインを使用している場合は、プラットフォームのビルド スクリプトを使用してプラグインをコンパイルします。 ビルド スクリプトは、ディレクトリ *federation_sdk_home*¥sample にインストールされます。 ビルド スクリプトは次のとおりです。

     Windows: build_plugin.bat

     UNIX: build_plugin.sh

     コンパイルされたサンプル プラグイン、fedpluginsample.jar は、ディレクトリ *federation_sdk_home*¥jar にあります。

   - 独自のプラグインを書く場合は、プラグインをコンパイルするときに smapi.jar をインクルードします。

2. JVMOptions.txt ファイルで、プラグインのクラスパスをインクルードするように、-Djava.class.path 値を変更します。 ディレクトリ *federation_install_dir*¥siteminder¥config 内の JVMOptions.txt ファイルを見つけます。

   任意のディレクトリにプラグイン jar を配置し、JVMOptions.txt ファイルがそれを参照するよう設定できます。サンプル プラグインを使用するには、fedpluginsample.jar を参照するようクラスパスを変更しますが、smapi.jar 用のクラスパスは変更しないでください。

   注: プラグインで Apache Xerces または Xalan を使用するには、CA SiteMinder® Federation でインストールされた Xerces または Xalan のバイナリ ファイルを使用します。 バイナリは CA SiteMinder® Federation SDK でインストールされません。 互換性の理由でこれらのファイルを使用する必要があります。

3. CA SiteMinder® Federation サービスを再起動します。

   このサービスの再起動は、CA SiteMinder® Federation がアサーション ジェネレータ プラグインの最新バージョンを使用するのに役立ちます。

アサーション ジェネレータ プラグインの有効化

アサーション ジェネレータ プラグインを作成してコンパイルした後に、CA SiteMinder® Federation UI 内で設定することにより、このプラグインを有効にします。 UI パラメータにより、CA SiteMinder® Federation がプラグインの検索場所を認識できます。

プラグインを展開 (P. 124)するまで、プラグイン設定を実行しないでください。

アサーション ジェネレータ プラグインを有効にする方法

1. 管理 UI にログオンします。
2. 変更するパートナーシップのパートナーシップ ウィザードのアサーション設定手順に移動します。
3. 以下の後に［アサーション ジェネレータ プラグイン］設定の値を入力します。

プラグイン クラス

プラグインの Java クラス名を指定します。 名前を入力します。 このプラグインはランタイムで呼び出されます。

例： com.mycompany.assertiongenerator.AssertionSample

このプラグイン クラスはアサーションを解析および変更してから、最終処理のために CA SiteMinder® Federation に結果を返すことができます。各依存パーティのアサーション ジェネレータ プラグインを指定します。 コンパイルしたサンプル プラグインは SDK に含まれています。 コンパイルされたアサーション プラグインのサンプルは、ディレクトリ *federation_sdk_home*¥jar で参照できます。

注: ディレクトリ *federation_sdk_home*¥sample¥com¥ca¥federation¥sdk¥plugin¥sample で CA SiteMinder® Federation サンプル プラグインのソース コードを参照することもできます。

プラグイン パラメータ

（オプション）。 CA SiteMinder® Federation が実行時にパラメータとしてプラグインへ渡す文字列を指定します。 文字列にはあらゆる値を含めることができ、従う特定の構文はありません。

プラグインは、受信するパラメータを解釈します。 たとえば、パラメータは属性の名前などです。または、文字列には、何かを実行するようにプラグインに指示する整数を含めることができます。

参照情報（メソッドの署名、パラメータ、戻り値、データ型）、および UserContext クラスと APIContext クラスのコンストラクタが「*Javadoc Reference*」にあります。Javadoc の AssertionGeneratorPlugin インターフェースを参照してください。

第 11 章: シングル サインオンの設定

このセクションには、以下のトピックが含まれています。

シングル サインオン設定（アサーティング パーティ）

依存パーティにアサーションを配信する方法を指定するには、アサーティング パーティでシングル サインオンを設定します。

その後の手順では、シングル サインオンを有効にするための基本手順を提供します。 サインオン ダイアログ ボックスのすべての設定可能な機能に関する詳細については、後のトピックおよび 管理 UI ヘルプで説明します。

次の手順に従ってください:

1. パートナーシップ ウィザードの該当する手順から始めます。

   SAML 1.1

   シングル サインオン

   SAML 2.0

   SSO と SLO

   WSFED

   シングル サインオンおよびサインアウト

   リモート依存パーティの作成またはインポート中に定義されるすべての値が入力されます。

2. 以下の情報に注意して、［認証］セクションのフィールドに入力します。

   - ［認証モード］フィールドに対して［ローカル］を選択した場合は、redirect.jsp ファイルを指す認証 URL の URL を入力します。例：

     http://webserver1.example.com/affwebservices/redirectjsp/redirect.jsp

     この例では、webserver1 によって Web エージェント オプション パックを持つ Web サーバが識別されます。redirect.jsp ファイルは、アイデンティティ プロバイダ サイトでインストールされた Web エージェント オプション パックに含まれています。

     重要: アクセス制御ポリシーで認証 URL を保護 (P. 67)します。レルム、ルールおよびポリシーを設定します。セッション情報をアサーションに追加するには、［認証セッション変数を保持］チェックボックスをオンにします。

   - ［認証モード］に［委任済み］を選択する場合は、追加のフィールドを設定します。 委任認証 (P. 221)について、より詳しく学習します。

3. ［認証クラス］フィールド（SAML 1.1 および 2.0 のみ）に入力します。このフィールドに静的な URI を指定します。さらに SAML 2.0 に限っては、ソフトウェアは認証クラスを自動検出できます。 URI は、ユーザが認証される方法を示すアサーションの AuthnContextClassRef 要素に配置されています。

4. ［SSO］セクションのフィールドに入力します。 これらの設定によって以下の機能を管理できます。

   - シングル サインオン バインディング
   - アサーション有効期間

     ［SSO 有効期間］および［スキュー時間］によって、アサーションが有効なときが決定します。 これらの設定が連携する仕組みを理解するには、アサーション有効期間 (P. 138)に関する情報を参照してください。

   SAML 2.0 については、以下の機能を設定できます。

   - シングル サインオンを開始するパートナー
   - SP セッション有効期間
   - SP セッション持続期間
   - SP と識別情報を共有するユーザ許可

   フィールドの説明については、［ヘルプ］をクリックしてください。

5. アサーション コンシューマ サービスまたはセキュリティ トークン サービスの URL を指定します。このリモート依存パーティ サービスは、アサーションを消費して処理します。

   パートナーは、この URL をユーザに提供する必要があります。

6. ［HTTP-Artifact］を SAML バインディングとして選択した場合は、バック チャネル設定 (P. 142)を行います。

7. （オプション）。SAML 2.0 については、以下のタスクを実行できます。

   - IDP ディスカバリ プロファイル (P. 157)を有効にします。
   - 特定の HTTP エラー用のステータス リダイレクト URL (P. 137) を指定します。

詳細情報:

SAML 2.0 エンティティでのシングル サインオンの開始の許可 (P. 137)
HTTP エラー用ステータス リダイレクト（SAML 2.0 IdP） (P. 137)
HTTP Artifact バック チャネルのレガシー Artifact 保護タイプ (P. 133)

パートナーシップ フェデレーションの認証モード

パートナーシップ フェデレーションでは、フェデレーション シングル サインオンの認証モードを定義できます。

- **ローカル認証モード**

  ローカル認証は、主にローカル フェデレーション システムで発生します。ローカル認証では、認証方式として［ベーシック］または［フォーム］を選択できます。 ローカルで利用可能なメソッドは、この 2 つのオプションのみです。

  外部のサードパーティがユーザを認証する場合も、認証モードにローカルを選択できます。 サードパーティからユーザ情報が返されると、そのユーザ情報は、アサーションで後に使用するために、セッション ストアに格納されます。

- **委任認証モード**

  委任認証は、認証タスクをサードパーティ Web アクセス管理（WAM）システムに転送します。 サードパーティがユーザを認証する方法は、サードパーティがサポートする認証方式によって異なります。 サードパーティ WAM がユーザを認証した後、フェデレーション ユーザ ID が SiteMinder に送信されます。

HTTP Artifact バック チャネルのレガシー Artifact 保護タイプ

HTTP Artifact シングル サインオンの場合、［Artifact 保護タイプ］フィールドに対してレガシー オプションを選択できます。 レガシー オプションにより、アサーティング パーティの Artifact サービスへのバック チャネルを保護するレガシー方式を使用することを示します。

レガシー保護方式を実装するには、以下を実行します。

- FederationWebServicesAgentGroup エージェント グループに、FWS アプリケーションを保護する Web エージェントを追加します。
  - ServletExec の場合、このエージェントは、Web エージェント オプション パックがインストールされている Web サーバにあります。
  - WebLogic や JBOSS などのアプリケーション サーバの場合、この Web エージェントは、アプリケーション サーバ プロキシがインストールされている場所にインストールされています。 Web エージェント オプション パックは別のシステムに存在することもあります。
- Artifact サービスを保護するポリシーを適用します。ポリシーを適用するために、Artifact サービスへのアクセスが許可されているアサーティング パーティから依存パーティへのパートナーシップを示します。

Web エージェントをエージェント グループに追加するには、次の手順に従ってください:

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］-［エージェントの作成］を選択します。
3. 展開内の Web エージェントの名前を指定します。 ［サブミット］をクリックします。
4. ［インフラストラクチャ］-［エージェント グループ］を選択します。
5. ［FederationWebServicesAgentGroup］エントリを選択します。

   ［エージェント グループ］ダイアログ ボックスが表示されます。
6. ［追加/削除］をクリックします。［エージェント グループのメンバ］ダイアログ ボックスが表示されます。

7. Web エージェントを［使用可能なメンバ］リストから［選択されたメンバ］リストに移動します。
8. ［OK］ボタンをクリックして、［エージェント グループ］ダイアログに戻ります。
9. ［サブミット］クリックした後、［閉じる］クリックしてメイン ページに戻ります。

取得サービスを保護するポリシーを適用するには、次の手順に従ってください:

1. 管理 UI で、Artifact 保護タイプのレガシー方式を使用してパートナーシップを設定します。
2. このパートナーシップをアクティブ化します。
3. ［ポリシー］-［ドメイン］-［ドメイン ポリシー］を選択します。

   使用可能なドメイン ポリシーのリストが表示されます。
4. 鉛筆のアイコンを選択して該当する Artifact サービス ポリシーを編集します。

   SAML 1.1

   FederationWSAssertionRetrievalServicePolicy

   SAML 2.0

   SAML2FWSArtifactResolutionServicePolicy

   注: 提供されたポリシーはデフォルトのポリシーです。 Artifact サービスを保護するために作成したあらゆるポリシーを使用できます。
5. ［ユーザ］タブに進みます。

   フェデレーション カスタム ユーザ ストアが［ユーザ ディレクトリ］セクションに表示されます。
6. 変更するユーザ ストアの［メンバの追加］をクリックします。

   SAML 1.1

   FederationWSCustomUserStore

   SAML 2.0

   SAML2FederationCustomUserStore

7. レガシー Artifact 保護を設定したパートナーシップを選択します。

   例：

   - SAML 1.1 パートナーシップの名前が Acme である場合、affiliate:affiliate:Acme を選択する
   - SAML 2.0 パートナーシップの名前が Demo である場合、affiliate:samlsp:Demo を選択する

8. ［OK］をクリックします。

HTTP Artifact シングル サインオンのパートナーシップが、Artifact サービスへのアクセスを許可したので、依存パーティはアサーションを取得できます。

シングル サインオン設定（依存パーティ）

依存するパーティでシングル サインオンを設定するには、SAML バインディングおよび関連するその他の SSO 設定を指定します。

依存するパーティでは、システムはパートナーシップのスキュー時間を使用して、取得するアサーションが有効であるかどうかを特定します。 設定されたスキュー時間をシステムが使用する方法を理解するには、アサーション有効期間 (P. 138)についての詳細を参照してください。

その後の手順では、シングル サインオンを有効にするための基本手順を提供します。 サインオン ダイアログ ボックスのすべての設定可能な機能に関する詳細については、後のトピックおよび 管理 UI ヘルプで説明します。

次の手順に従ってください:

1. パートナーシップ ウィザードの該当する手順から始めます。

   SAML 1.1

   シングル サインオン

   SAML 2.0

   SSO と SLO

   WS-フェデレーション

   シングル サインオンおよびサインアウト

2. ダイアログ ボックスの［SSO］セクションの設定を行います。 これらの設定によって、シングル サインオン バインディングを制御できます。

   フィールドの説明については、［ヘルプ］をクリックしてください。

   SAML の場合、HTTP-Artifact または HTTP-POST プロファイルを設定します。依存するパーティがシングル サインオンを開始する場合、クエリ パラメータがリクエストに含まれます。このクエリ パラメータは、使用する SSO バインディングを示します。バインディングが指定されない場合、デフォルトは POST です。 アサーティング パーティがシングル サインオンを開始する場合、アサーティング パーティは特定のトランザクションに使用するバインディングを示します。

3. （オプション）。 SAML 2.0 については、次の設定を行えます。
   - リモート SSO サービス URLs
   - リモート SOAP Artifact URL
   - シングル サインオンを開始するパートナー

     サードパーティ IdP がホストでユーザ レコードを持たないコンシューマ ユーザを認証している場合、SSO は SP で開始されます。
   - ユーザ許可要件
4. HTTP-Artifact プロファイルを選択する場合は、ダイアログ ボックスの［バック チャネル］セクションでバック チャネルの認証方法を設定します。
5. 残りの設定については、デフォルトを使用します。

シングル サインオンの基本設定が完了しました。 その他の設定は SSO で利用可能です。 フィールドの説明については、［ヘルプ］をクリックしてください。

HTTP エラー用ステータス リダイレクト（SAML 2.0 IdP）

ID プロバイダについては、HTTP 500、400、または 405 エラーの発生時に SiteMinder がユーザをリダイレクトする方法を設定できます。 たとえば、リクエストの URL が間違ったターゲットを指すと、403 エラーが発生する場合があります。 このエラーが発生する場合、ユーザはさらなる処理を実行する特定の URL に送られます。

以下のようにリダイレクト オプションを選択します。

1. ［SSO と SLO］ダイアログ ボックスの［ステータス リダイレクト URL］セクションに移動します。
2. ［ステータス リダイレクト URL］セクションで、リダイレクトを求めるエラー状態のチェック ボックスをオンにします。
3. SiteMinder によるユーザのリダイレクト先の URL を入力します。
4. 各 URL については、リダイレクト方法の［302 データなし］または［HTTP Post］を選択します。

リダイレクト処理が設定されました。

SAML 2.0 エンティティでのシングル サインオンの開始の許可

SAML 2.0 パートナーシップの場合、IdP または SP 、または両方のいずれがシングル サインオンを開始できるかを決定できます。 パートナーシップの両側で許可されるトランザクションを設定できます。

トランザクションの開始を制限することによって、ユーザ認証コンテキスト情報の交換など、他のシングル サインオン機能に与える影響を考慮してください。

次の手順に従ってください：

1. 管理 UI にログインします。
2. 編集する SAML 2.0 パートナーシップを選択します。
3. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
4. ［許可されるトランザクション］フィールドで、プルダウン メニューからオプションを選択します。
5. ウィザードの［確認］手順にスキップして変更を保存します。

シングル サインオンのアサーション有効期間

シングル サインオンでは、［スキュー時間］および［SSO 有効期間］の値によって、アサーションの有効な期間が決まります。 ポリシー サーバはアサーションの生成および消費にスキュー時間を適用します。 アサーション ドキュメントで、NotBefore と NotOnOrAfter の値は、有効期間の開始と終了を表します。

アサーティング パーティで、ポリシー サーバはアサーションの有効性を設定します。 ポリシー サーバは、アサーションが生成されたシステム時間で有効期間の開始を決定します。 ソフトウェアはこの時間からアサーションの IssueInstant の値を設定します。 次に、ポリシー サーバは IssueInstant の値からスキュー時間を減算します。 その結果の時間が NotBefore 値になります。

NotBefore=IssueInstant - スキュー時間

有効期間の終了を決定するために、ポリシー サーバは、有効期間の値とスキュー時間を IssueInstant の値に加算します。 その結果の時間が NotOnOrAfter 値になります。

NotOnOrAfter=有効期間 + スキュー時間 + IssueInstant

時間は GMT が基準になります。

たとえば、アサーティング パーティでアサーションが 1:00 GMT に生成されたとします。 スキュー時間が 30 秒、有効期間が 60 秒とすると、アサーション有効期間は 12:59:30 GMT ～ 1:01:30 GMT となります。 この期間は、アサーションが生成される 30 秒前に開始され、その後 90 秒後に終了します。

依存側においても、ポリシー サーバは、アサーティング パーティで受信したアサーションが有効かどうかを判別するために実行する計算と同じ計算を実行します。

SiteMinder がパートナーシップの両側にある場合のアサーション妥当性期間の計算

アサーションが有効である総時間は、SSO 妥当性期間にスキュー時間の 2 倍を足した合計です。 計算式は次のとおりです。

アサーション妥当性期間 = 2 x スキュー時間（アサーティング パーティ）+ SSO 妥当性期間 + 2 x スキュー時間（依存側）

式の前半部分（2 x スキュー時間 + SSO 妥当性期間）は、アサーティング パーティでの妥当性期間を表します。 式の後半部分（2 x スキュー時間）は、依存パーティにおけるシステム クロックのスキュー時間を表します。 有効期間の NotBefore および NotOnOrAfter の両端を計算するために 2 を掛けます。

注: ポリシー サーバでは、［SSO 有効期間］はアサーティング パーティでのみ設定されます。

例

アサーティング パーティ

アサーティング パーティでの値は以下のとおりです。

IssueInstant=5:00PM

SSO 有効期間 =60 秒

スキュー時間 = 60 秒

NotBefore = 4:59PM

NotOnOrAfter=5:02PM

依存パーティ

依存側は NotBefore および NotOnOrAfter の値を取得し（つまり、アサーションで受け取る）、スキュー時間を適用して新しい値を算出します。

スキュー時間 = 180 秒（3 分）

NotBefore = 4:56PM

NotOnOrAfter=5:05PM

これらの値に基づいたアサーションの合計有効期間の計算は以下のとおりです。

120 秒（2x60） + 60 秒 + 360 秒（2x180） = 540 秒（9 分）

サービス プロバイダのセッション妥当性期間

サービス プロバイダの認証セッションの継続期間を管理できます。SessionNotOnOrAfter 属性は、IdP がアサーションの <AuthnStatement> に含めることができる任意属性です。セッション妥当性期間の設定は IdP で実行されます。

注: SessionNotOnOrAfter パラメータは NotOnOrAfter パラメータ（アサーションが有効な期間を決定する）とは異なります。

サードパーティ SP は SessionNotOnOrAfter の値を使用して、セッションが短すぎないことを確認できるように、自身のタイムアウト値を設定できます。ユーザ セッションが無効になった場合、ユーザはアイデンティティ プロバイダで再認証する必要があります。

重要: SiteMinder は、SP として機能している場合、SessionNotOnOrAfter 値を無視します。 代わりに、SiteMinder SP は、ターゲット リソースを保護する SAML 認証方式に対応するレルム タイムアウトからセッション タイムアウトを設定します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 変更する IdP から SP へのパートナーシップを選択します。
3. ［SSO と SLO］手順に移動します。
4. ［SSO］セクションで、［SP セッション有効期間］のオプションを選択します。カスタム オプションを選択する場合は、複数のオプションを選択できます。

   フィールドの説明については、［ヘルプ］をクリックしてください。
5. 変更が終了したら、［確認］手順を選択して［完了］をクリックします。

Artifact SSO のバック チャネル認証

Artifact シングル サインオンでは、依存側がアサーションを取得するアサーティング パーティに Artifact を送信する必要があります。アサーティング パーティは、Artifact を使用して正しいアサーションを取得し、バック チャネルで依存側にアサーションを返します。

バック チャネルへのアクセスを認証するためにエンティティを要求できます。必須ではありませんが、SSL を使用してバック チャネルを保護することができます。

SSL を使用してバック チャネルを保護するには、以下を実行する必要があります。

1. SSL を有効にする。

   SSL は基本認証には必要ありませんが、SSL を介して基本認証を使用できます。 SSL はクライアント証明書認証に必要です。

2. SAML 2.0 通信交換用に受信または送信バック チャネルを設定する。設定する方向は、ローカル エンティティのロールによって異なります。

   個別のチャネルの設定は SAML 2.0 に対してのみサポートされています。 SAML 1.1 Artifact シングル サインオン用のバック チャネル設定では、各パートナーシップの単一の設定を使用します。 SiteMinder は自動的に正しい方向を使用します（ローカル プロデューサに受信およびローカル コンシューマに送信）。

   設定しているエンティティに基づいて、SAML 2.0 シングル サインオンに対して設定する方向を選択します。

   - ローカル アサーティング パーティは受信チャネルを使用します。
   - ローカル依存側は送信チャネルを使用します。

   注: 1 つの受信および送信バック チャネルを設定できますが、チャネルに設定できるのは 1 つの設定のみです。 2 つのサービスが同じチャネルを使用する場合、これらの 2 つのサービスは同じバック チャネル設定を使用します。たとえば、ローカルのアサーティング パーティの受信チャネルが HTTP-Artifact SSO と SLO over SOAP をサポートする場合、これらの 2 つのサービスは同じバック チャネル設定を使用する必要があります。

3. 保護されているバック チャネルを介してアクセスできるように依存側用の認証タイプを選択する。 認証方法はチャネルごと（受信または送信）に適用されます。

   バック チャネル認証のオプションは以下のとおりです。

   - 基本
   - クライアント証明書
   - 認証なし

   管理 UI ヘルプではこれらのオプションを詳細に説明しています。

   重要: 受信バック チャネル用の認証方法は、パートナーシップの反対側の送信バック チャネル用の認証方法に一致する必要があります。認証方法の選択の一致は帯域外通信で処理されます。

SAML 2.0 属性クエリのサポート

SiteMinder IdP は SAML 2.0 アサーション クエリ/リクエスト プロファイルをサポートしており、属性クエリに応答できます。また、IdP はアサーション内またはメタデータ内にない属性のクエリを許可することにより、プロファイルの機能を拡張します。 IdP が属性クエリを受信すると、IdP は、属性を検索するためにまずそのユーザ ディレクトリを確認します。 属性が見つからない場合、ポリシー サーバがセッション ストアを確認します。セッション ストアは、外部の ID プロバイダからの属性、高度な認証方式から収集された属性、およびその他のソースを保持できます。

注: SiteMinder IdP のみがクエリ プロファイルをサポートします。 属性リクエスタとしての SiteMinder SP は、プロキシ化された属性クエリ機能 (P. 146)に対してのみサポートされます。

IdP には、SP がそのメタデータでリクエストできるユーザ属性がすべてあります。 SP は、2 つの方法でこれらの属性を取得できます。

- アサーションで送信される属性のセットを抽出します。

  ID プロバイダ アサーション設定により、含まれる属性のセットが決まります。 すべての属性のサブセットを定義すると、属性の数が最も不可欠なものに制限され、これにより処理のオーバーヘッドが軽減されます。

- IdP メタデータをインポートします。

メタデータ内の属性に加えて、SP は、アサーション、またはメタデータ内にない属性を必要とする場合があります。 その他の属性を取得するために、SP は IdP に属性クエリを送信します。

クエリ リクエスト プロファイルは、2 つのエンティティを使用します。

- SAML 属性機関
- SAML 属性リクエスタ

SiteMinder IdP は、属性機関としてのみ機能できます。 SiteMinder SP は属性リクエスタになりません。

以下の図は、属性機関の設定手順を示しています。

以下の手順を完了します。

- IdP から SP へのパートナーシップを設定するか変更します。 (P. 145)
- ID プロバイダで、SAML 2.0 属性機関を設定します (P. 145)。

SiteMinder がパートナーシップの両側にある場合、アサーション クエリ/レスポンス プロファイルを使用できません。

属性クエリ サポート用のパートナーシップの設定

IdP が属性クエリに応答するには、IdP から SP へのパートナーシップが存在する必要があります。 パートナーシップを作成するか、既存のパートナーシップを変更できます。

パートナーシップの作成手順は、以下のようなものです。

1. SAML 2.0 IdP および SP エンティティを作成します (P. 71)。
2. パートナーシップ用のユーザ ディレクトリへの接続を設定します (P. 65)。
3. SAML 2.0 の IdP から SP へのパートナーシップを作成します (P. 83)。
4. SAML 2.0 属性機関を設定します (P. 145)。

これらの手順についてこのガイドを通して説明します。

SAML 2.0 属性機関の設定

属性機関として機能するように IdP を設定できます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［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］を選択します。
3. 変更または新しく作成する IdP から SP へのパートナーシップを選択します。
4. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。

5. ダイアログ ボックスの［属性サービス］セクションの［有効］を選択します。
6. ［有効期間］に秒数を入力します。
7. （オプション）属性クエリの署名、および属性アサーションおよびレスポンスの署名を要求するかどうかを指定します。
8. ［ユーザの検索］セクションで、適切なユーザ ディレクトリ ネームスペースの検索指定を入力します。 属性機関は、この検索指定を使用してユーザを特定します。

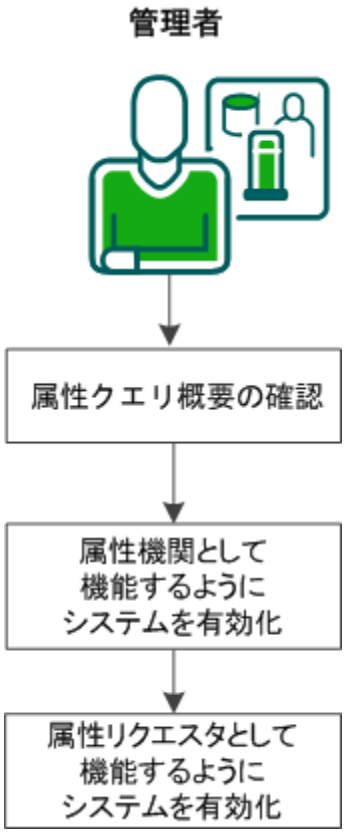
   LDAP ユーザ ディレクトリに対する例には、uid=%s などがあります。少なくとも 1 つの検索条件が必要です。
9. （オプション）［バック チャネル］セクションで［保護タイプ］に［パートナーシップ］を指定します。 認証方法を選択します。 バック チャネルの詳細については、［ヘルプ］をクリックしてください。
10. パートナーシップを保存してアクティブにします。

これでアイデンティティ プロバイダは属性機関として機能するように設定されました。 この機関は、サードパーティ SP からの属性クエリに応答できるようになりました。

サードパーティからのユーザ属性値の取得(SAML 2.0)

SAML 2.0 フェデレーション環境では、サービス プロバイダがアサーションで提供されていないユーザに関する情報を必要とする場合があります。サービス プロバイダは、事前定義されたユーザ属性の値をリクエストできます。 ID プロバイダにこれらの値がない場合は、サードパーティから値をリクエストできます。 SiteMinder 環境では、この機能はプロキシ化された属性クエリと呼ばれています。

以下の図は、プロキシ化された属性クエリを有効にするプロセスを示しています。



プロキシ化された属性クエリを有効にするには、以下のタスクを完了します。

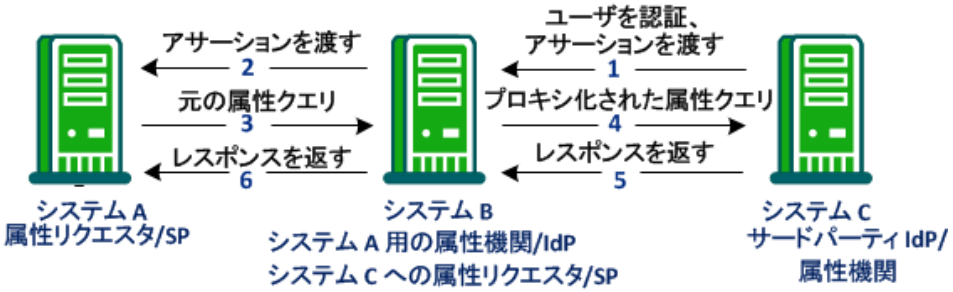
1. プロキシ化された属性クエリの概要を確認します。 (P. 147)
2. 属性機関として機能するシステムを有効にします (P. 149)。
3. 属性リクエスタとして機能するシステムを有効にします (P. 150)。

プロキシ化された属性クエリの概要

プロキシ化された属性クエリ機能は、SAML 2.0 アサーション クエリ/リクエスト プロファイルに基づいており、ユーザ属性の検索を拡張します。属性機関は、まずユーザ ディレクトリおよびセッション ストアで属性を検索します。 属性が見つからず、ユーザが最初サードパーティ IdP で認証されている場合、リクエストはサードパーティ IdP に転送できます。

プロキシ化された属性クエリを実装する場合、単一の SiteMinder システムは 2 つのリモート システム間の中継点として機能します。1 つのリモート システムから別のリモート システムにリクエストを中継する場合、単一のシステムが 2 つの役割を果たします。システムは、まず元の属性リクエスタの属性機関として機能します。システムは、サードパーティ IdP に対して属性リクエスタとしても機能します。 属性リクエスタとして、システムは元の IdP に対して属性クエリをプロキシ化します。

以下の図は、単一のシステムがプロキシ化されたクエリを処理する方法を示しています。



以下の手順では、プロキシ化された属性クエリのフローについて説明します。

1. ユーザは、始めにシステム C で、サードパーティ IdP を認証します。システム C はアサーションを生成し、それをシステム B へ渡します。
2. システム B はシステム A にアサーションを送信して、システム A、B、C 間の初期シングル サインオン トランザクションを完了します。このシングル サインオン トランザクションは、プロキシ化された属性クエリを処理するのに必要です。
3. システム A がアサーションを受信した後、システム A はそれがアサーション内にない他の属性を必要とするかどうかを決定します。 属性リクエスタとして、システム A は属性クエリをその属性機関/IdP、システム B に送信します。
4. システム B は、システム A がそのユーザ ディレクトリまたはセッション ストアにない属性を必要とするかどうかを決定します。属性を取得するために、システム B は新しいクエリ リクエストを生成します。システム B は、ユーザが最初に認証を行った、システム C（サードパーティ IdP）に新規クエリを送信します。 この新規クエリは、プロキシ化されたクエリです。
5. システム C は、システム B に属性を含むレスポンスを返します。 システム B は、そのセッション ストアに属性を保存します。

6. システム B は、属性機関として、システム A に属性を含む自身のレスポンスを返します。

重要: システム A の設定された属性名および名前形式（未指定、uri、または基本）は、システム C でのこれらの属性の名前と一致する必要があります。 この情報は、トランザクションが発生する前に通信されます。

属性機関として機能するシステムの有効化（IdP->SP）

プロキシ化されたクエリ トランザクションを実装するには、同じ SiteMinder システム上に 2 つのパートナーシップを設定します。

- IdP から SP へのパートナーシップ
- SP から IdP へのパートナーシップ

SiteMinder が属性機関として機能するには、既存の IdP から SP へのパートナーシップを変更するか、パートナーシップを作成します。 このパートナーシップで、SiteMinder はローカル IdP/属性機関であり、リモート パートナーは SP/属性リクエスタです。

注: このシステムは、SP から IdP へのパートナーシップで属性リクエスタとしても役立ちます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［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］を選択します。
3. 変更または新しく作成する IdP から SP へのパートナーシップを選択します。
4. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
5. ダイアログ ボックスの［属性サービス］セクションの［有効］を選択します。
6. ［有効期間］に秒数を入力します。
7. （オプション）属性クエリの署名、および属性アサーションおよびレスポンスの署名を要求するかどうかを指定します。
8. ［プロキシ化されたクエリの有効化］を選択します。

9. ［ユーザの検索］セクションで、適切なユーザ ディレクトリ ネームスペースの検索指定を入力します。 属性機関は、この検索指定を使用してユーザを特定します。

   LDAP ユーザ ディレクトリに対する例には、uid=%s などがあります。少なくとも 1 つの検索条件が必要です。

10. （オプション）［バック チャネル］セクションで［保護タイプ］に［パートナーシップ］を指定します。 認証方法を選択します。 バック チャネルの詳細については、［ヘルプ］をクリックしてください。

11. パートナーシップを保存してアクティブにします。

システムが、元の属性リクエスタに対する属性機関として機能できるようになります。

属性リクエスタとして機能するシステムの有効化(SP->IdP)

プロキシ化されたクエリ トランザクションを実装するには、同じ SiteMinder システム上に 2 つのパートナーシップを設定します。

- IdP から SP へのパートナーシップ
- SP から IdP へのパートナーシップ

注: パートナーシップ フェデレーションは、プロキシ化された属性クエリ機能に対する属性リクエスタとしてのみ、SP をサポートします。

SiteMinder が属性リクエスタとして機能するには、既存の SP から IdP へのパートナーシップを変更するか、新しいパートナーシップを作成します。このパートナーシップで、SiteMinder はローカル SP/属性リクエスタであり、リモート サードパーティはリモート IdP/属性機関です。

注: このシステムは、IdP から SP へのパートナーシップにおける属性機関としても機能します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［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］を選択します。
3. 変更する SP から IdP へのパートナーシップを選択するか、新しく作成します。

4. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
5. ［属性リクエスタ サービス］セクションで、［有効］および［プロキシ化されたクエリの有効化］を選択します。
6. ［属性サービス］セクションで、リモート IdP の URL を指定します。
7. 名前 ID の形式、タイプ、および値を指定します。
8. （オプション）バック チャネルの認証タイプを選択します。 バック チャネルについては、［ヘルプ］をクリックしてください。
9. パートナーシップを保存してアクティブにします。

これでサービス プロバイダは属性リクエスタとして機能できます。

SAML 2.0 IdP のユーザ許可

SiteMinder ID プロバイダは、SAML 2.0 のユーザ許可機能をサポートしています。 ユーザ許可では、ID プロバイダは、アサーションをパートナーに送信する前に、ユーザに許可を求める必要があります。 ID プロバイダでユーザ許可を有効にしていると、SiteMinder によってユーザは許可を求められます。 ID プロバイダは、アサーション内に許可値を渡します。

許可の妥当性期間は 5 分間です。 ID プロバイダがユーザを許可ページにリダイレクトすると、ユーザには、許可を与えて ID プロバイダに戻されるまでに 5 分間あります。 その後、ID プロバイダはアサーションを生成してサービス プロバイダに送信します。 これらのタスクを 5 分の間に完了する必要があります。 アサーションを生成する前に時間切れになると、ID プロバイダはユーザ ID を渡しません。

許可は単一のアサーションにのみ適用されます。ID プロバイダは、アサーションの生成後に、付与された許可のすべてのレコードを削除します。 5 分の妥当性期間が切れる前に、同じユーザは ID プロバイダに戻れますが、ID プロバイダはユーザにさらに許可を求めます。

注: 妥当性期間は設定できません。

例

ユーザ 1 は 2:00PM に MyWorkPlace.com にログインして認証します。 MyWorkPlace は ID プロバイダとして機能しています。 2:03PM に、ユーザは従業員の旅行サービスを実行するパートナー企業へのリンクを選択します。ユーザ 1 は、ExampleTravel.com に送られる前に許可を求めるフォームにリダイレクトされます。ユーザ 1 は許可フォームに入力する前に電話を受けます。現在は 2:10PM です。妥当性期間が切れたので、MyWorkPlace はアサーションを生成しません。

ユーザ 1 が速やかに許可を与えて、2:05PM までに ID プロバイダにリダイレクトされれば、ID プロバイダはアサーションを生成します。 許可およびアサーション生成の間で 2 分のみ経過しますので、妥当性期間はまだアクティブです。

ユーザ許可を設定するには以下を実行する必要があります。

- ユーザ許可を有効にします。
- ユーザ許可フォームの名前を入力します。

  ID プロバイダは、許可を得るためにカスタム フォームをユーザに送信します。

ID プロバイダがユーザ許可属性をアサーション レスポンスに含める場合、以下の URI のみが使用されます。

```
urn:oasis:names:tc:SAML:2.0:consent:obtained
```

ユーザ許可はサービス プロバイダでも設定できます。 サービス プロバイダは、ユーザ許可値をアサーション レスポンス内に渡すように ID プロバイダに要求できます。

ユーザ許可フォームのカスタマイズ

SiteMinder には、ca_defaultconsentform.html という名前のフォームを*連携する許可*が付属しています。 アイデンティティ プロバイダは、許可を得るためにカスタム フォームをユーザに送信します。 デフォルトの許可フォームは %NETE_WA_ROOT%¥customization ディレクトリ内にあります。 %NETE_WA_ROOT% は Web エージェント オプション パックの場所です。

デフォルトの許可フォームを使用して 管理 UI でフォームを指定する代わりにカスタム フォームを作成することができます。

次の手順に従ってください:

1. カスタム HTML フォームを作成します。 フォームを変更して以下の設定の値を置換します。

   $$userconsent_spid$$

   パートナーシップで設定された SP ID を表します

   $$userconsent_idpid$$

   パートナーシップで設定された IDP ID を表します。

2. フォームを %NETE_WA_ROOT%¥customization ディレクトリに配置します。

   NETE_WA_ROOT はシステム環境変数です。 %NETE_WA_ROOT% は Web エージェント オプション パックの場所です。 Web エージェントおよび Web エージェント オプション パックが同じシステム上にインストールされている場合、たとえば webagent¥customization など同じディレクトリにインストールされます。

3. 管理 UI にログインします。
4. ［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］に移動します。
5. 変更する IdP から SP へのパートナーシップを選択します。
6. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
7. ［SSO］セクションで以下の操作を行います。
   a. ［ユーザ許諾の有効化］チェック ボックスをオンにします。
   b. ［ユーザ許諾 Post フォーム］フィールドでカスタム フォームの名前を指定します。

   注:［ユーザ許諾サービス URL］はデフォルトで指定されています。この値は変更できません。
8. 設定が終了したら、［確認］手順に移動して［完了］をクリックします。

機能強化クライアントまたはプロキシ プロファイルの概要（SAML 2.0）

機能強化クライアントまたはプロキシ プロファイル（ECP）は、シングル サインオンのアプリケーションです。機能強化クライアントは、ECP 機能をサポートするブラウザやほかのいくつかのユーザ エージェントです。機能強化プロキシは、ワイヤレス デバイス用のワイヤレス アクセス プロトコル プロキシなどの HTTP プロキシです。

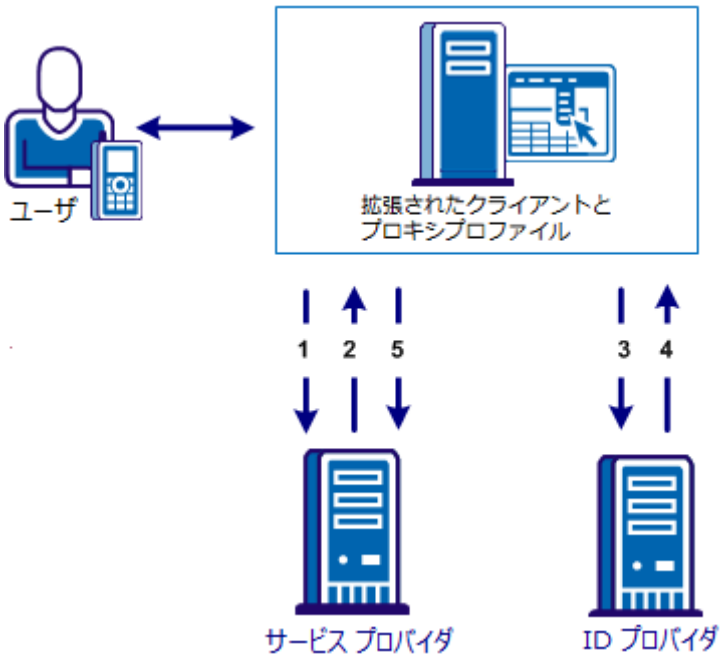
ECP プロファイルは、アイデンティティ プロバイダとサービス プロバイダが直接通信できない場合に、シングル サインオンを有効にします。ECP は、サービス プロバイダとアイデンティティ プロバイダの間で仲介する機能を果たします。

仲介として機能することに加えて、ECP プロファイルは以下の状況で役立ちます。

- このプロファイルを必要とする機能強化クライアントまたはプロキシをサービス プロバイダが提供する場合。
- 機能が制限されたモバイル デバイスの前のワイヤレス アクセス プロトコル（WAP）ゲートウェイなどのプロキシ サーバが使用中の場合。

ECP アプリケーションを入手または開発する必要があります。 SiteMinder は SAML 要件に準拠しており、ECP リクエストのみを処理し、ECP アプリケーションに対してのみ応答します。

ECP プロファイルのフローを、次の図に示します。



ECP 通信では、ユーザが携帯電話などからアプリケーションへのアクセスをリクエストします。アプリケーションはサービス プロバイダに存在し、ユーザの ID 情報はアイデンティティ プロバイダに存在します。 サービス プロバイダとアイデンティティ プロバイダは、直接通信を行いません。

呼び出しのフローを以下に示します。

1. ECP アプリケーションは、Reverse SOAP （PAOS）リクエストをサービス プロバイダに転送します。アイデンティティ プロバイダには、サービス プロバイダから直接アクセスできません。

   アイデンティティ プロバイダとは異なり、ECP エンティティは常に直接アクセスできます。

2. サービス プロバイダは、認証リクエストを ECP アプリケーションに送り返します。

3. ECP アプリケーションは、認証リクエストを処理および変更し、アイデンティティ プロバイダに送信します。

4. アイデンティティ プロバイダはリクエストを処理し、SOAP レスポンスを ECP アプリケーションに返します。 このレスポンスには、アサーションが含まれます。
5. ECP アプリケーションは、署名された PAOS レスポンスをサービス プロバイダに渡します。

シングル サインオンが続行され、アプリケーションへのアクセス権がユーザに付与されます。

アイデンティティ プロバイダでの ECP の設定

ECP を設定するには、アイデンティティ プロバイダおよびサービス プロバイダでこの機能を有効にします。SiteMinder アイデンティティ プロバイダのための手順を以下に示します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 変更するローカル アイデンティティ プロバイダ パートナーシップを選択します。
3. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
4. ［SSO］セクションで、［拡張されたクライアントまたはプロキシ プロファイルの有効化］チェック ボックスをオンにします。
5. ［確認］手順に移動して［完了］をクリックし、変更を保存します。

アイデンティティ プロバイダが、ECP 呼び出しを処理できるようになります。

注: 単一のサービス プロバイダ オブジェクトは、シングル サインオン リクエストの Artifact、POST、SOAP、および PAOS バインディングを処理できます。 SOAP と PAOS は、ECP プロファイルのバインディングです。 アイデンティティ プロバイダおよびサービス プロバイダは、リクエストのパラメータに基づいて使用するバインディングを決定します。

サービス プロバイダでの ECP の設定

ECP を設定するには、ID プロバイダおよびサービス プロバイダでこの機能を有効にする必要があります。 サービス プロバイダの手順を以下に示します。

次の手順に従ってください:

1. 保護されているリソースのリクエストをサービス プロバイダの 認証リクエスト サービスに送信します。 以下に URL の例を示します。

   https://*host*:*port*/affwebservices/public/saml2authnrequest

2. 管理 UI にログインします。
3. 関連するローカル サービス プロバイダ パートナーシップを変更します。
4. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。
5. ［SSO］セクションで、［拡張されたクライアントまたはプロキシ プロファイルの有効化］チェック ボックスをオンにします。
6. ［確認］手順に移動して［完了］をクリックし、変更を保存します。

サービス プロバイダが、ECP 呼び出しを処理できるようになります。

注: 単一のサービス プロバイダ オブジェクトは、シングル サインオン リクエストの Artifact、POST、SOAP、および PAOS バインディングを処理できます。 SOAP と PAOS は、ECP プロファイルのバインディングです。 アイデンティティ プロバイダおよびサービス プロバイダは、リクエストのパラメータに基づいて使用するバインディングを決定します。

IDP ディスカバリ プロファイル（SAML 2.0）

ID プロバイダ ディスカバリ （IPD）プロファイルは、共通の検出サービスを提供し、これを使用して、サービス プロバイダが認証用の固有の IdP を選択できます。パートナー間では前もって業務提携契約が確立され、ネットワーク内のすべてのサイトが ID プロバイダ ディスカバリ サービスとやり取りできるようになります。

このプロファイルは、複数のパートナーがアサーションを提供するフェデレーション ネットワークで役立ちます。 サービス プロバイダは、特定のユーザの認証リクエストを送信する ID プロバイダの決定ができます。

IdP ディスカバリ プロファイルは、2 つのフェデレーション パートナーに共通の Cookie ドメインを使用して実装されます。 合意されたドメインの Cookie には、そのユーザがアクセスしたことがある IdP のリストが含まれています。

アイデンティティ プロバイダでの IDP ディスカバリ設定

［SSO と SLO］ダイアログ ボックスの［IDP ディスカバリ］セクションで IDP ディスカバリ プロファイルを設定します。

注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。

次の手順に従ってください:

1. ［IDP ディスカバリの有効化］チェック ボックスをオンにします。
2. ［サービス URL］フィールドの値をアイデンティティ プロバイダ ディスカバリ プロファイル サーブレットに対して設定します。 SiteMinder の場合、この URL は以下のとおりです。

   http://*host:port*/affwebservices/public/saml2ipd

   ホスト

   ［共通ドメイン］フィールドで指定する共通のドメインを表します。

   ポート

   製品のインストール時に指定した Apache HTTP または HTTPS ポートを指定します。

   URL は、https で始まる場合もあります。
3. ［共通ドメイン］フィールドで Cookie ドメインを指定します。
4. （オプション）［永続的な Cookie の有効化］チェック ボックスをオンにしてブラウザ内の共通の Cookie を保存します。

IdP ディスカバリが IdP で有効になりました。

サービス プロバイダでの IDP ディスカバリ設定

IDP ディスカバリ プロファイルの場合、サービス プロバイダ（SP）は、認証リクエストの送信先のアイデンティティ プロバイダ（IdP）を特定する必要があります。 SP が認証するユーザは、以前にアイデンティティ プロバイダにアクセスし、認証している必要があります。

SP は、ユーザを自身の IdP ディスカバリ サービスにリダイレクトして、共通のドメイン Cookie を取得する必要があります。 Cookie には、ユーザがすでにアクセスしたアイデンティティ プロバイダのリストが含まれています。 このリストから、Cookie は正しい IdP を選択して、その IdP に認証リクエストを送信します。

IdP ディスカバリ プロセスは以下のとおりです。

1. ブラウザは SP のサイト選択ページを要求します。

   このサイト選択ページでは IDP ディスカバリ サービス URL が認識されます。

2. サイト選択ページはユーザを IDP ディスカバリ サービス URL にリダイレクトし、共通ドメイン Cookie を取得する必要があることを示します。
3. IDP ディスカバリ サービスは共通ドメイン Cookie を取得し、そのドメインで Cookie を読み取って、サイト選択ページにユーザをリダイレクトして返します。 ディスカバリ サービスはクエリ パラメータとして共通ドメイン Cookie を提供します。
4. SP は、サイト選択ページにユーザが以前に認証した IdP URL を読み込みます。
5. ユーザは IdP を選択してユーザ認証を実行します。

SP で IdP ディスカバリを設定する方法

1. SP の IdP ディスカバリ サービスに共通ドメイン Cookie を要求するサイト選択ページを作成します。

   SiteMinder には、IdPDiscovery.jsp という名前のサンプル サイト選択ページが付属しています。このページを使用して IdP ディスカバリを実装できます。 このページは、以下のディレクトリにあります。

   web_agent_home/affwebservices/public

   最初のリンクはブラウザを 1 つのドメインから共通ドメインの IdPDiscovery サービスにリダイレクトし、**_saml_idp** という名前の共通ドメイン Cookie を取得します。 SP の IdP ディスカバリ サービスがリクエストを受信すると、サービスは共通ドメイン Cookie を取得してクエリ パラメータとして追加します。 その後、IDP ディスカバリ サービスは、通常のドメインの IdPDiscovery.jsp サイト選択ページにユーザをリダイレクトして返します。 デフォルトでは、IdPDiscovery.jsp ページには、共通 Cookie から抽出した IdP の ID のリストのみが表示されます。このリストは静的です。関連 IdP との通信を開始する、リストに関連付けられた HTML リンクはありません。

2. SP サイトのサンプル ページで以下のリンクを編集します。 リンクの最初の部分は、saml2idp Cookie が配置された共通ドメインを指定します。 リンクの 2 番目の部分は、IdPDiscovery.jsp が配置された通常ドメインを指定します。

   例：

   ```
   <a href="http://myspsystem.commondomain.com/affwebservices/public
   /saml2ipd/?IPDTarget=/http://myspsystem.spdomain.com/affwebservices
   /public/IdpDiscovery.jsp&SAMLRequest=getIPDCookie">
   Retrieve idp discovery cookie from IPD Service</a>
   ```

   ターゲット サイト選択ページのある通常ドメインにユーザがリダイレクトされて戻ると、このページには共通 Cookie が取得されています。

3. （オプション）IdPDiscovery.jsp サイト選択ページに各 IdP の HTML リンクが表示されるように、このページを編集します。 それぞれのリンクが認証リクエストをトリガして、IdP がシングル サインオンを開始します。 デフォルトでは、IdPDiscovery.jsp ページには、共通 Cookie から抽出した IdP の ID のリストのみが表示されます。

4. 編集したサイト選択ページを使用して、IdP ディスカバリをテストします。

IdP ディスカバリを動作させたまま、選択する IdP のリストをサイト選択ページで参照できます。

Office 365 へのシングル サインオン

CA SiteMinder® Federation は、企業ユーザおよび Office 365 サービスの間のシングル サインオンを有効にします。 Office 365 にフェデレーションすることで、ホスト サービスの負荷がローカルで軽減されます。 たとえば、企業ユーザはデスクトップの電子メール クライアントにログインしますが、サービスがクラウドにあることに気づきません。 Office 365 のユーザ操作性は、オンプレミス アプリケーションに接続されている場合と同じです。

次のプロファイルが Office 365 へのシングル サインオンで利用できます。

WS-フェデレーション パッシブ リクエスタ プロファイル

WS-フェデレーション パッシブ リクエスタ プロファイルは、パッシブ リクエスタ（主に Web ブラウザまたは HTTP をサポートするブラウザ ベースのアプリケーション）と連動します。パッシブ プロファイルにより、これらのクライアントと Microsoft Office 365 の間のシングル サインオンが有効になります。

WS-フェデレーション アクティブ リクエスタ プロファイル

Security Token Service （STS）は WS-フェデレーション アクティブ リクエスタ プロファイルを実装します。このプロファイルにより、SOAP が有効なデスクトップ クライアントと以下の Office 365 サービスの間のシングル サインオンが有効になります。

- Exchange オンライン（Outlook）
- Lync オンライン
- Dynamics CRM オンライン

クライアントは HTTP-POST リクエストおよびレスポンスを使用して SOAP メッセージを送受信します。 ユーザは会社の認証情報を使用して社内ネットワークにサインインし、Outlook や Lync へアクセスできます。

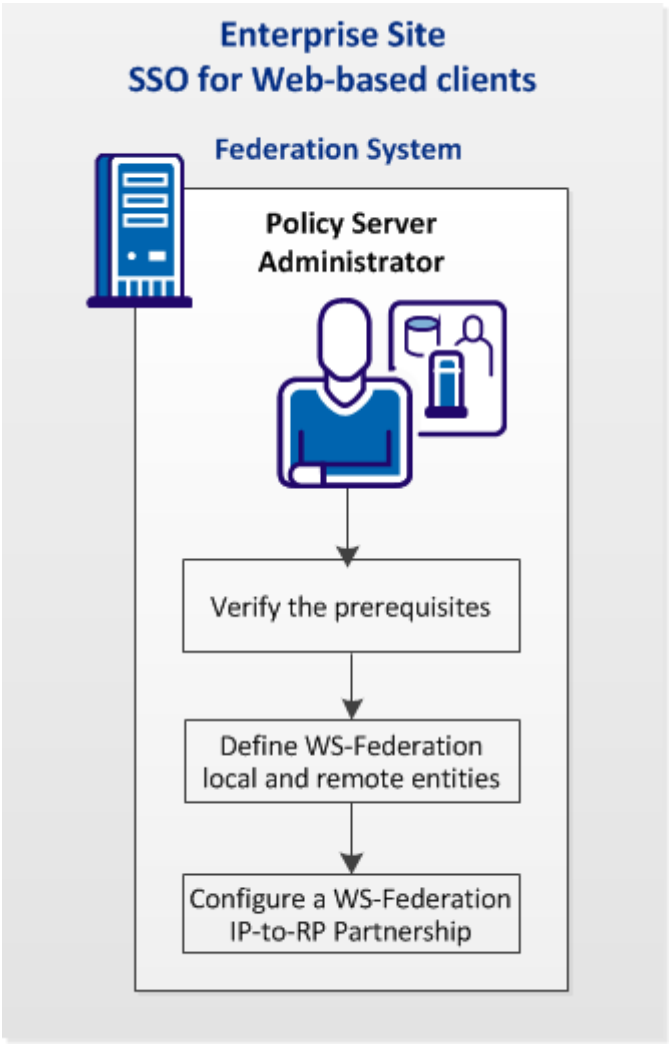
CA SiteMinder® Federation は STS サービスを提供します。このサービスは、Office 365 認証のアイデンティティ プロバイダとして機能します。STS サービスは、Office 365 サービスが消費できるセキュリティ トークンを発行します。

Office 365 へのシングル サインオンを実装するには、両方の WS-フェデレーション プロファイルで、フェデレーション システムで設定された WS-フェデレーション IP-to-RP パートナーシップが必要です。

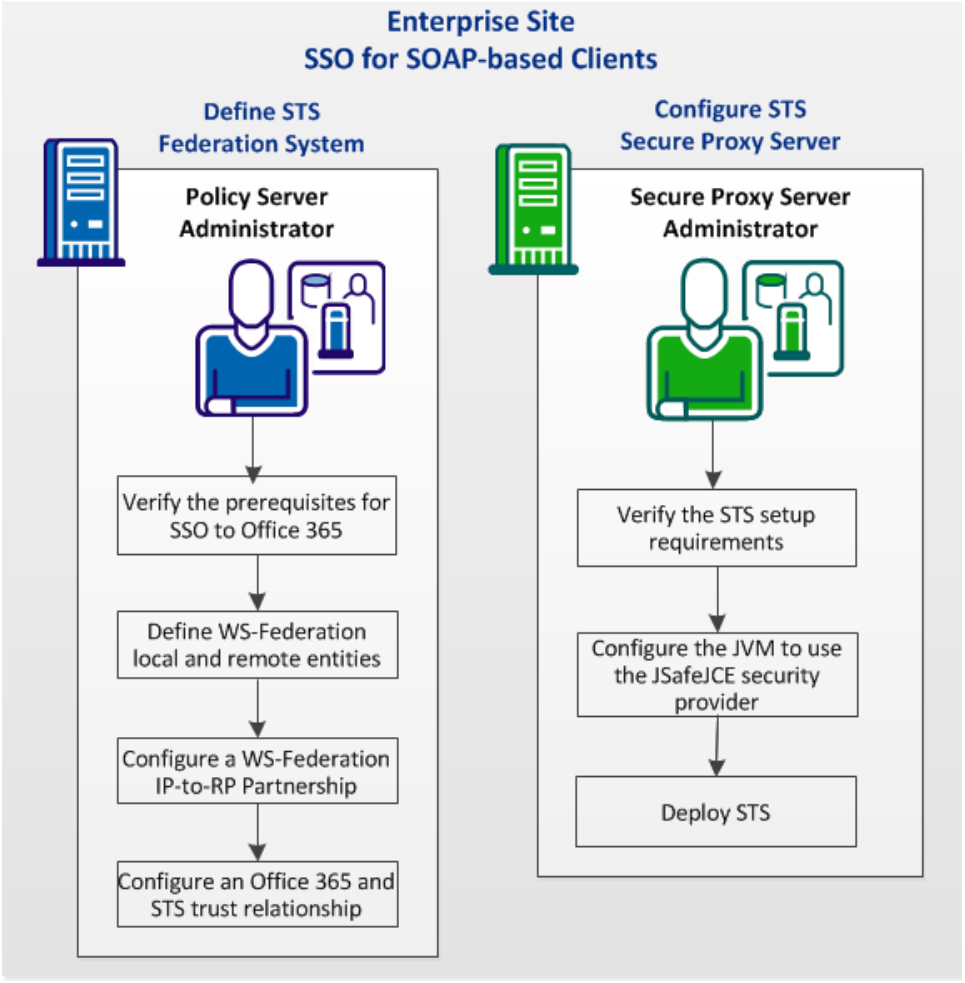
WS-フェデレーション アクティブ リクエスタ プロファイルについては、以下の追加のコンポーネントも必要です。

- パートナーシップで有効化された STS サービス
- SiteMinder Secure Proxy Server （SPS）で設定された STS。

次の図は、Web ベース クライアント SSO （パッシブ リクエスタ プロファイル）の必要な設定手順を示したものです。



次の画像は、SOAP ベースのクライアント SSO（アクティブ リクエスタ プロファイル）の必要な設定手順を示したものです。



フェデレーション システムで以下のタスクを完了します。

1. Office 365 への SSO の前提条件を確認します (P. 164)。
2. WS-フェデレーション ローカル IP (P. 167) とリモート RP (P. 170) エンティティを定義します。
3. WS-フェデレーション IP-to-RP パートナーシップを設定します (P. 172)。
4. Office 365 と STS の間の信頼関係を設定します。 (P. 174) (SOAP ベースの SSO のみ)

STS を展開するために、CA SiteMinder for Secure Proxy Server 上で以下のタスクを完了します（SOAP ベースの SSO のみ）。

1. STS セットアップの前提条件を確認します (P. 176)。
2. JSafeJCE セキュリティ プロバイダを使用するよう JVM を設定します (P. 177)。
3. CA SiteMinder for Secure Proxy Server で STS を展開します (P. 178)。

Office 365 へのシングル サインオンに関する設定のさらなる詳細については、SiteMinder Federation Cloud Runbook Library の適切なランブックを参照してください。 このコンテンツを表示するにはログインする必要があります。

Office 365 への SSO の前提条件を確認します

Office 365 へのシングル サインオンについては、次の要件に注意してください。

- Office 365 セットアップ
- SiteMinder ユーザ ディレクトリ

Office 365 セットアップ要件

- Office 365 ドメインを登録し、取得します。登録する計画はシングル サインオンをサポートする必要があります。
- 所有するドメインを登録します。
- Office 365 ドメインにドメインを追加します。
- 所有する Office 365 ドメインの DNS 記録を更新します。

Office 365 と連動するような展開の設定については、以下の方法に関する Microsoft のドキュメントを参照してください。

- Office 365 を登録し、購読します。
- オンプレミスと Office 365 ユーザ ディレクトリの間のディレクトリ同期を設定します。
- Office 365 と STS で設定されたオンプレミス Secure Proxy Server との信頼関係を確立します。

Office 365 へのシングル サインオンに関する設定のさらなる詳細については、SiteMinder Federation Cloud Runbook Library の適切なランブックを参照してください。 このコンテンツを表示するにはログインする必要があります。

SiteMinder ユーザ ディレクトリ要件

- 管理 UI でユーザ ディレクトリ接続を設定する際に、フェデレーション ユーザに対して ImmutableID および UPN 属性が存在することを確認します。 オンプレミス ユーザ ディレクトリのこれらの属性に対する値が、Office 365 ディレクトリにあるものに一致する必要があります。

  不変 ID と UPN が必要です。 WS-フェデレーション パートナーシップを設定するとき、これらの値を提供します。

詳細情報:

STS セットアップ要件の確認 (P. 176)

Office 365 との WS-フェデレーション パートナーシップの設定

Office 365 との WS-フェデレーション パートナーシップを設定します。WS-フェデレーション IP-to-RP のパートナーシップは Web ベースまたは SOAP ベースのクライアント SSO に必要です。

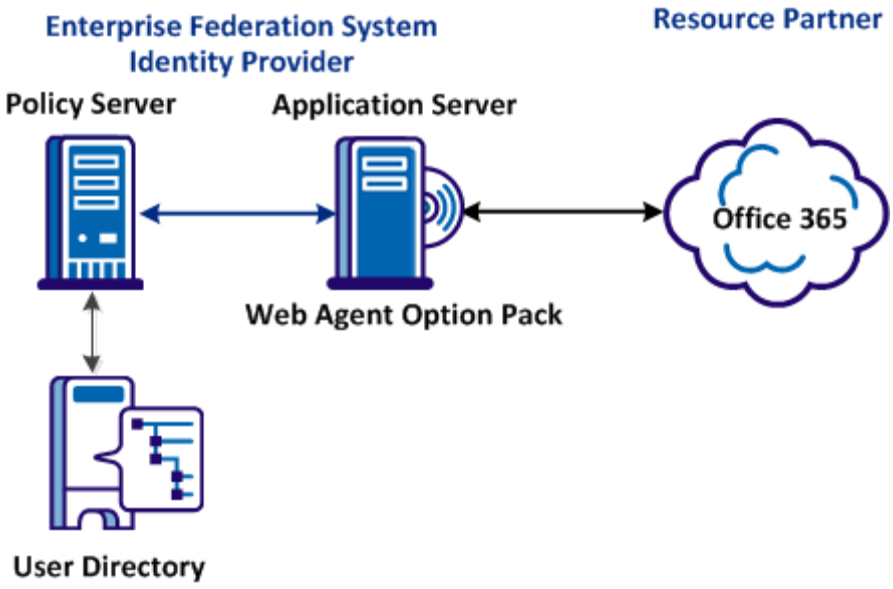
このパートナーシップでは：

- SiteMinder は ID プロバイダです （IP）
- Office 365 はリソース パートナーです （RP）

WS-フェデレーション パッシブ リクエスタ プロファイルおよびアクティブ リクエスタ プロファイルを設定するこのパートナーシップ間の違いは次のとおりです。

- アクティブ リクエスタ プロファイルの STS を有効にする
- サインアウトを設定します。これはオプションです。 サインアウトは WS-フェデレーション パッシブ リクエスタ プロファイルにのみ関連します。

次の画像は、このフェデレーション ソリューションに推奨される展開を示したものです。



Office 365 パートナーシップのローカル IP エンティティの定義

オンプレミス フェデレーション システムは Office 365 とのパートナーシップでのアイデンティティ プロバイダです。 アイデンティティ プロバイダとして、このシステムは SAML 1.1 アサーションを含むセキュリティ トークンを発行します。

SAML 1.1 トークンでローカル アイデンティティ プロバイダ エンティティを作成します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
3. ［エンティティの作成］をクリックします。

   ［エンティティの作成］ダイアログ ボックスが表示されます。
4. **ローカル**を選択し、サイトに対してローカルなエンティティを作成することを示します。
5. 残りのフィールドを設定します。

   新しいエンティティ タイプ

   WSFED アイデンティティ プロバイダを選択します。

   SAML トークン タイプ

   SAML 1.1

6. エンティティに関する詳細を設定するために［次へ］をクリックします。

   ［エンティティの設定］手順で、ダイアログ ボックスのすべての必須フィールドに入力します。 次のフィールドに特に注意します。

   エンティティ ID

   Office 365 ドメインで指定された IssuerURI を入力します。

   このローカル パートナーに対して、エンティティ ID は一意である必要がありません。

   エンティティ名

   このローカル IP を識別する名前を入力します。［エンティティ名］によって、ポリシー ストアのエンティティ オブジェクトが識別されます。この名前は一意の値である必要があります。 この値は内部使用のみです。リモート パートナーはこの値を認識しません。

   ベース URL

   システムの URL を指定します。 Office 365 との通信については、この URL が SSL 接続である必要があります。 たとえば https://fedserver.example.com になります。

   特定 ID（Office 365 に必要）

   同じ IP と RP の間に複数のパートナーシップがある場合に限り、この ID を設定します。そうすれば、会社には、Office 365 と独自の関係を持った個別の事業単位が与えられます。Office 365 は、RP としてそれ自体を識別するために単一の ID を使用します。 SiteMinder フェデレーションは、同じ IP または RP ID との複数のパートナーシップを許可しません。 特定 ID により、特定のパートナーに与えられたサービス URL の一意の論理パス サフィックスに基づきシステムはパートナーシップを区別できます。 フェデレーション サービスは 1 つですが、サフィックスを RP ID と組み合わせることで、一意のパートナーシップ ルックアップ キーが作成されます。

   例： microsoftonline

リクエストが正しいリモート パートナーに送信されるように、特定 ID はフェデレーション URL に追加されます。

例：

パッシブ リクエスタ サービス URL：
https://fedserver1.forwardinc.com/affwebservices/public/wsfeddispatcher/microsoftonline

「microsoftonline」は特定 ID です。

英数文字列は使用できますが特殊文字は使用できません。

サインアウト確認 URL

サインアウトを実行するアイデンティティ プロバイダの URL を指定します。

デフォルト：
http://*ip_server:port*/affwebservices/signoutconfirmurl.jsp

ip_server:port

アイデンティティ プロバイダ システムのサーバおよびポート番号を指定します。 システムは、フェデレーション ネットワークにどのコンポーネントがインストールされているかに応じて、Web エージェント オプション パックまたは SPS フェデレーション ゲートウェイをホストしています。

署名秘密キー エイリアス

署名と暗号化機能については、証明書データ ストアに適切なキーと証明書のペアをインポーとします。

重要: この秘密キーと関連付けられる公開証明書を Office 365 フェデレーション ドメインにインポートする必要があります。

サポートされる名前 ID 形式と属性

未指定

UPN 属性

UPN はユーザ プリンシパル名です。

アサーション属性

UPN

ネームスペース

http://schemas.xmlsoap.org/claims

注: Microsoft はこの値を指定します。 表示どおりにネームスペース値を入力します。 Office 365 はこの正確な値を必要とします。

不変 ID 属性

不変 ID は、オンプレミス Microsoft ディレクトリのユーザを区別する一意の属性です。

アサーション属性

不変 ID

ネームスペース

http://schemas.microsoft.com/LIVEID/Federation/2008/05

注: Microsoft はこの値を指定します。 表示どおりにネームスペース値を入力します。 Office 365 はこの正確な値を必要とします。

7. すべての必須フィールドに入力したら、［確認］をクリックします。
8. リモート エンティティを設定します。

Office 365 パートナーシップのリモート RP エンティティの定義

Office 365 を表すリモート リソース パートナーを作成します。 エンティティを定義するには、利用できる場合にメタデータをインポートするか、以下の手順を使用してエンティティを設定します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。

3. ［エンティティの作成］をクリックします。

   ［エンティティの作成］ダイアログ ボックスが表示されます。

   注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。

4. リモートを選択し、ロケーションに対してリモートなエンティティを作成していることを示します。

5. 残りのフィールドを設定します。

   新しいエンティティ タイプ

   WSFED リソース パートナーを選択します。

   SAML トークン タイプ

   SAML 1.1

6. エンティティに関する詳細を設定するために［次へ］をクリックします。

7. ［エンティティの設定］手順で、以下の必須フィールドに入力します。

   エンティティ ID

   urn.federation： MicrosoftOnline

   エンティティ名

   RP を識別する名前を入力します。

   リモート セキュリティ トークン サービス URL

   Office 365 セキュリティ トークン サービスの URL を指定します。 Microsoft からのこの URL を取得します。 この URL は SSL 接続である必要があります。 たとえば https://login.microsoftonline.com になります。

   リモート サインアウト URL（パッシブ リクエスタ プロファイルのみ）

   Office 365 サインアウト サービスの URL を指定します。 Microsoft からのこの URL を取得します。 この URL は SSL 接続である必要があります。 たとえば https://login.microsoftonline.com になります。

   注: Office 365 については、セキュリティ トークン消費者サービスの URL およびサインアウト URL が同じです。

   サポートされる名前 ID 形式

   未指定

8. 設定を見直したら［確認］をクリックします。

Office 365 との WS-フェデレーション パートナーシップの設定

ローカル IP とリモート RP エンティティを作成したら、WS-フェデレーション パートナーシップを設定します。 いずれかの WS-フェデレーション プロファイルに固有のステップに記録されます。

次の手順に従ってください:

1. 管理 UI にログオンします。
2. ［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］に移動します。
3. ［パートナーシップの作成］プルダウン メニューから、WSFED IP->RP を選択します。

   ダイアログ ボックスが開き、その上部にパートナーシップ ウィザードが表示されます。
4. パートナーシップ ウィザードの手順 1 で、標準のフェデレーション設定の必須フィールドに入力します。

   a. アクティブ リクエスタ プロファイルのみについては、［**WSFED アクティブ プロファイルの STS**］チェックボックスを選択します。
5. 必要に応じて、［メタデータ交換の有効化］を選択し、アクティブなプロファイル エンドポイントとデータを反映するフェデレーション メタデータ ドキュメントを作成します。このドキュメントの URL は次のとおりです。

   https://*sps_host*/affwebservices/public/FederationMetadata/*partnership_name*

   注: *sps_host* は STS が設定された Secure Proxy Server です。

   メタデータ ドキュメントは、WS-フェデレーション パッシブおよびアクティブ プロファイルのエンドポイントおよびデータなど、パートナーシップの詳細を提供します。
6. パートナーシップ ウィザードの手順 2 で、このパートナーシップにフェデレーション ユーザを選択します。

7. パートナーシップ ウィザードの手順 3 で、名前 ID に必要な設定を入力します。

   NameID 形式

   未指定

   名前 ID タイプ

   ユーザ属性

   名前 ID 値

   Office 365 によって割り当てられた 不変 ID

8. ウィザードの手順 3 にとどまり、アサーション属性設定に入力します。

   a. UPN および不変 ID アサーション属性がローカル IP エンティティ (P. 167)から継承されることを確認します。 エンティティ レベルでこれらの属性を追加しなかった場合は、ここでそれらを指定します。

   b. 両方の属性に対して［タイプ］フィールドを［ユーザ属性］に設定します。

   c. UPN および不変 ID 値をそれぞれ持つユーザ ディレクトリ属性に［値］フィールドを設定します。

9. パートナーシップ ウィザードの手順 4 で、［認証］セクションの次のフィールドに入力します。

   認証モード

   ローカル

   認証 URL

   WS-フェデレーション パッシブ リクエスタ プロファイルが使用中の場合に、このフィールドに入力します。 アクティブ リクエスタ プロファイルの場合、このフィールドは無視します。

   https://*web_agent_optionpack_system*/affwebservices/redirectjsp/redirect.jsp

10. ウィザードの手順 4 でとどまり、SSO セクションと SLO セクションの次のフィールドに入力します。

    オーディエンス

    urn:federation:MicrosoftOnline

    セキュリティトークン サービス URL

    https:// login.microsoftonline.com/

WS-フェデレーション パッシブ リクエスタ プロファイルの場合のみ、［サインアウト］フィールドに入力します。それらはアクティブ リクエスタ プロファイルには関連しません。

サインアウト確認 URL（オプション）

サインアウトが設定される場合、展開の URL を入力します。

サインアウト URL（オプション）

https:// login.microsoftonline.com/

11. 残りの設定にはデフォルトを受け入れて、次の手順に進みます。
12. パートナーシップ ウィザードの手順 5 で、署名処理を有効化し、適切な秘密キー/証明書ペアのエイリアスを選択します。
13. ［確認］手順に移動します。 設定を見直し、［完了］をクリックし、パートナーシップを保存します。

    ［パートナーシップ リスト］に戻ります。
14. ［アクション］、［アクティブ化］を選択し、パートナーシップをアクティブにします。

STS はフェデレーション パートナーシップに対して定義されます。 次に、CA SiteMinder for Secure Proxy Server で STS コンポーネントを設定します。

Office 365 と STS の間の信頼関係の設定（SOAP ベースの SSO）

SOAP ベースのクライアントと Office 365 の間のシングル サインオンを有効にするには、Office 365 Sign-In Service と STS のあるオンプレミス サーバとの間の信頼関係を設定します。Office 365 サブスクリプションを購入し、ディレクトリ同期を設定した後で、この関係をセットアップします。

Windows Powershell コマンドを使用して、信頼関係を設定します。 最初に信頼関係を設定するコマンドは Set-MsolDomainFederationSettings です。このコマンドを実行します。 正しい手順については、Windows Powershell に関する Microsoft のドキュメントを参照してください。

このコマンドは、次のようなコマンド引数をとることができます。

- ドメイン
- ActiveLogOnUri （オンプレミス STS の ws- ユーザ名エンドポイント）
- PassiveLogOnUri

- IssuerUri
- MetadataExchangeUri
- SigningCertficate

これらのコマンド引数だけで、Office 365 から STS のあるオンプレミス Secure Proxy Server システムへの信頼関係を確立できます。

コマンド引数の STS エンドポイントを決定するには、SiteMinder 管理 UI 内の既存の WS-フェデレーション パートナーシップを参照します。 これらのエンドポイントは、WS-フェデレーション IP-to-RP のパートナーシップの値に基づきます。 Office 365 を設定するときにこれらのエンドポイントを使用して、STS で Secure Proxy Server システムが信頼されるようにします。

注: 特定の WS-フェデレーション パートナーシップの設定を参照するには、そのパートナーシップの隣の［アクション］、［表示］を選択します。

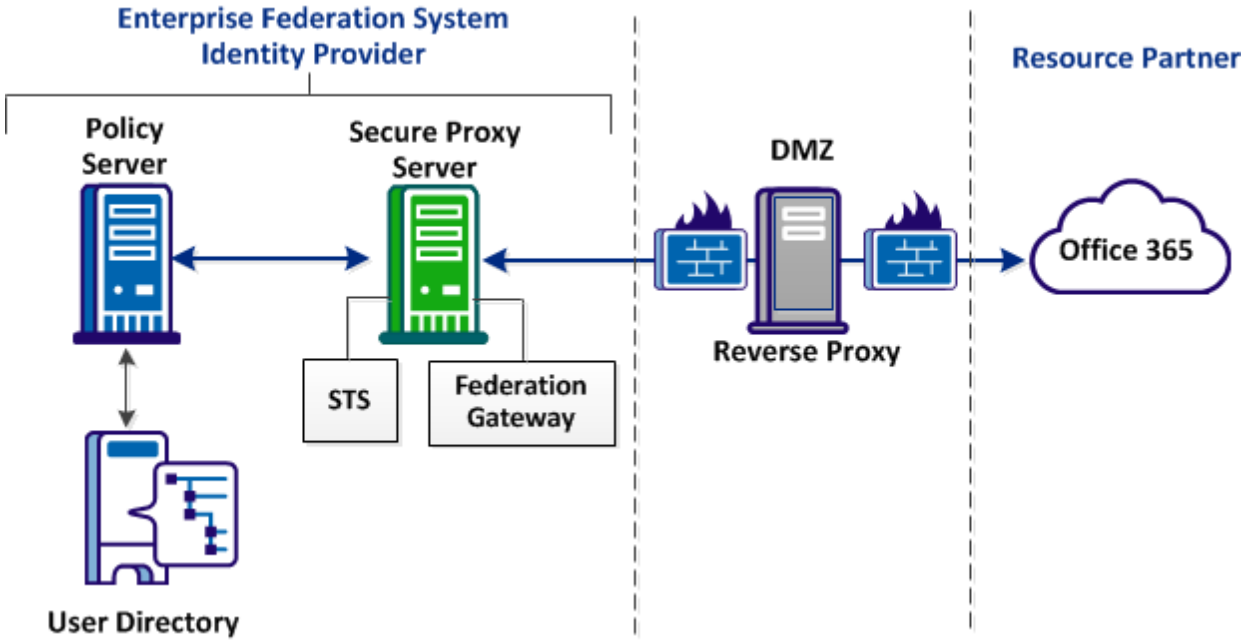
CA SiteMinder for Secure Proxy Server の設定

SiteMinder で WS-フェデレーション アクティブ プロファイルを使用してシングル サインオンを実装するには、STS Web サービスが必要です。Office 365 からのリクエストを管理するには、CA SiteMinder for Secure Proxy Server 上に STS を展開します。

注: STS Web サービスは、会社の CA SiteMinder for Secure Proxy Server 上でホストされている必要があります。 このサービスは、別の SiteMinder プラットフォームではホストできません。

クライアント アプリケーションが Office 365 に接続しようとすると、Office 365 は STS へのトークン リクエストを発行します。STS は、Office 365 が消費できる SAML 1.1 アサーションを持つセキュリティ トークンを発行します。クライアント アプリケーションは Office 365 にアクセスできます。

次の画像は、このフェデレーション ソリューションに推奨される展開を示したものです。さまざまな方法で STS にトラフィックを送信することができます。



次の手順に従ってください:

1. STS セットアップ要件を確認します。 (P. 176)
2. JSafeJCE セキュリティ プロバイダを使用するよう JVM を設定します (P. 177)。
3. STS を展開します (P. 178)。

STS セットアップ要件の確認

STS を CA SiteMinder for Secure Proxy Server 上に展開する前に、以下の要件を完了します。

- CA SiteMinder for Secure Proxy Server をインストールして設定します。

  注: ロード バランサの背後に 2 つ以上の Secure Proxy Server システムを配置することをお勧めしますが、これは必須ではありません。

- CA SiteMinder® Federation システムの管理者からのパートナーシップ名を取得します。 STS を展開する際に、［STS コンテキスト］設定にこの名前を指定します。
- CA SiteMinder for Secure Proxy Server で SSL を有効にします。
- Office 365 から始まるトラフィックが STS を持ったオンプレミス CA SiteMinder for Secure Proxy Server に到達できることを確認します。
- STS がある Secure Proxy Server システムは、内部トラフィック、および企業ファイアウォールの外からの外部トラフィックによってアクセス可能である必要があります。 外部トラフィックの転送には、プロキシの DMZ へのインストールが必要になる場合があります。 ファイアウォールの外から STS にトラフィックを転送できるようにプロキシを設定します。

JSafeJCE セキュリティ プロバイダを使用するための JVM の設定

暗号化を有効にするには、CA SiteMinder for Secure Proxy Server を実行している JVM を設定して、それが JSafeJCE セキュリティ プロバイダを使用するようにします。

次の手順に従ってください:

1. 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 の Web サイトからダウンロードします。
2. 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。

3. 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
4. java.security ファイルを開きます。
5. プロバイダ リスト セクションの以下の行に、2 番目のセキュリティ プロバイダとして JSafeJCE を追加します。

   ```
   security.provider.2=com.rsa.jsafe.provider.JsafeJCE
   ```
6. ほかのセキュリティ プロバイダの優先順序を 1 ずつ増やします。
7. 既存のセキュリティ プロバイダ リストの最後に以下の行を追加します。 この行により、JSafeJCE の初期 FIPS モードが設定されます。

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```
8. 変更を保存します。

以下の例は、JVM を設定した後の java.security ファイルの List of Providers セクションを示しています。

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=sun.security.rsa.SunRsaSign
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC
security.provider.10=sun.security.mscapi.SunMSCAPI
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

STS の展開

WS-フェデレーション アクティブ リクエスタ プロファイルをサポートするには、CA SiteMinder for Secure Proxy Server 上に STS を展開します。

次の手順に従ってください:

1. SPS 管理 UI を開きます。
2. ［Web サービス］-［セキュリティ トークン サービス］に移動します。
3. ［追加］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。

   STS 名

   STS Web サービスの名前を定義します。 管理 UI で定義されたパートナーシップ名を入力します。

   STS コンテキスト

   STS コンテキスト パスを定義します。 管理 UI で定義された WS-フェデレーション パートナーシップの名前を指定します。 */partnership_name* 構文を使用して、値を入力します。

   例： /Office365Cloud

5. ［OK］をクリックし、［保存］をクリックします。
6. システムを再起動します。
7. Office 365 へのシングル サインオンをテストします。

SSO から Office 365 へのテストおよびトラブルシューティングを行います（アクティブ リクエスタ プロファイル）

SSO のテスト

Lync または Outlook へサインインすることで、WS-フェデレーション設定および STS 展開を確認します。

次の手順に従ってください:

1. 企業のシステムで Lync または Outlook にログインします。
2. ログインできて、ローカルにインストールされているようにアプリケーションを使用できることを確認します。

SSO の問題のトラブルシューティング

WS-フェデレーション パートナーシップおよび接続の問題については、以下の調査方法を使用します。

- WS-フェデレーション パッシブ リクエスタのプロファイルを使用し、Microsoft オンラインとのシングル サインオンを確認します。

  Web ブラウザから、http://portal.microsoftonline.com または Microsoft Exchange オンラインに移動します。企業の認証情報でログインを試行します。 企業のクライアントからではなく、ブラウザから正常にログインできた場合は、オンプレミス STS のセットアップを確認します。

- Office 365 が SiteMinder についてフェデレーション パートナーとして認識していること、およびフェデレーション ユーザについて認識していることを確認します。パートナーシップおよびユーザの状態を確認するには、以下の Microsoft Powershell コマンドを実行します。

  Get-MsolDomainFederationSettings

  Microsoft がユーザのドメイン (つまりユーザの企業) について持っている情報が表示されます。 設定を参照し、正確かどうかを確認します。 情報が正しくない場合、フェデレーション通信の問題の原因である可能性があります。

  Get-MsolUser

  Microsoft が特定のユーザについて持っている情報が表示されます。ユーザ設定を参照し、正確かどうかを確認します。 情報が正しくない場合、フェデレーション通信の問題の原因である可能性があります。

- Microsoft リモート接続アナライザを使用して、企業と Microsoft の間の接続を検証します。 このツールを使用して、Outlook、Lync、Office 365 に関する接続の問題を特定できます。 このツールは https://www.testexchangeconnectivity.com/ にあります。

STS コンポーネントに関する問題については、以下のログおよびファイルを使用します。

- STS ログを参照し、STS が動作していること、および認証が失敗しているかどうかを確認します。 確認するログの場所は、*secure-proxy_install_dir*/proxy-engine/logs/*partnership_name*.log です。

  STS の初期化が完了したという内容のメッセージを探します。 このメッセージは、STS が実行されていることを示します。

- ログ設定は agent-log4j.xml 設定ファイル内で設定します。 システムが *partnership_name*.log に最も詳細な情報を記録するように、すべてのカテゴリのログ レベルを DEBUG に設定します。agent-log4j.xml ファイルは以下のディレクトリにあります。

  secure-proxy_install_dir/proxy-engine/conf/sts-config/*partnership_name*/config/

  また、チェックポイント ログ設定で、<category name="com.ca.CheckPointLogger," の優先度値を "INFO" に設定します。この設定は、認証アクティビティおよびアサーション生成に対してチェックポイント ログ メッセージを書き込みます。 チェックポイント ログ メッセージは、STS コンポーネントの操作を反映するコードを伴う説明メッセージです。

  セクション「フェデレーション トレース ログ (P. 317)」では、チェックポイント メッセージについて説明します。

- WSDL ファイルを参照し、オンプレミス STS が応答していることを確認します。 ブラウザを開き、http://*sts.company.com*/*partnership_name*?wsdl に移動します。 文字列 *sts.company.com* は、STS URL 用のプレースホルダです。 管理 UI で設定された WS-フェデレーション パートナーシップにおける STS URL を特定できます。

SAML 2.0 HTTP-POST バインディング設定

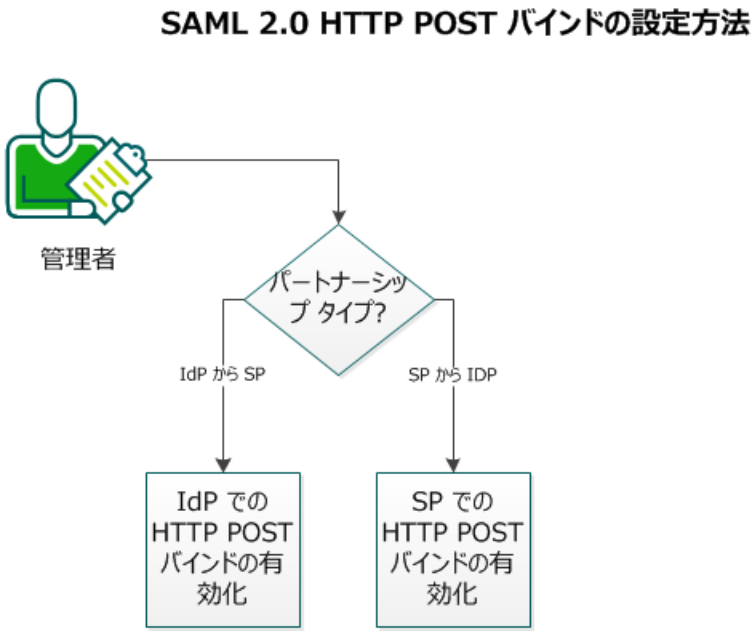
シングル サインオンおよびシングル ログアウトのリクエストの場合、リクエストとレスポンスを交換する方法として SAML 2.0 HTTP-POST バインディングを有効にできます。このバインディングは SAML プロトコルを標準メッセージ形式および通信プロトコルにマップします。

注: 認証リクエスト バインディングは SSO バインディングとは異なります。 SSO バインディングは、特定のユース ケースを処理する際のアサーション、プロトコルおよびバインディングの連携方法を指定するプロファイルを決定します。

この手順では、ユーザがフェデレーション環境に精通しており、以下のパートナーシップの 1 つ以上が作成およびアクティブ化されていることが前提です。

- IdP から SP
- SP から IdP

以下の図は、SAML 2.0 HTTP POST バインディングを有効にする方法を示しています。



次の手順に従ってください:

1. 該当するパートナーシップの種類に応じて適切なタスクを実行します。
   - IdP において HTTP POST バインディングを有効にします (P. 183)。
   - SP において HTTP POST バインディングを有効にします (P. 184)。

IdP での HTTP POST バインディングの有効化

IdP で HTTP POST バインディングを有効にできます。

重要: 認証リクエスト バインディングを設定する前に、セッション ストアを有効にします。 IdP が HTTP-POST バインディングを使用して提供される認証リクエストを処理するには、IdP はセッション ストアにリクエストを格納する必要があります。

セッション ストアの有効化

次の手順に従ってください:

1. ポリシー サーバ管理コンソールを開き、［データ］タブを選択します。
2. 以下のフィールドを設定します。

   データベース

   セッション ストア

   ストレージ

   ストレージ リポジトリを選択します。

   セッション ストア有効

   このボックスをオンにします。
3. データ ソース情報を完了します。
4. ［OK］をクリックして変更内容を保存します。

管理 UI でのバインディングの設定

次の手順に従ってください:

1. 管理 UI を開きます。
2. 変更するパートナーシップがアクティブな場合は、非アクティブにします。
3. ［変更］をクリックして、パートナーシップ ウィザードを開きます。
4. ［SSO と SLO］手順に移動します。

5. SSO セクションで、認証リクエスト バインディングに HTTP-POST を選択します。

   注: 認証リクエストに HTTP リダイレクトおよび HTTP-POST バインディングを共に選択できます。

6. （オプション）［SLO］セクションで［HTTP POST］チェックボックスをオンにします。

   注: 複数の SLO バインディングを選択できます。

7. SLO バインディングに一致するバインディングで SLO サービス URL を指定します。HTTP リダイレクトおよび HTTP-POST バインディングを選択した場合、各 SLO バインディングに 1 つずつ、2 つの SLO サービス URL を作成します。
8. 必要に応じて、その他のパートナーシップ情報を入力します。
9. 確認手順で［完了］をクリックします。

HTTP-POST バインディングが有効になりました。

SP での HTTP POST バインディングの有効化

SP で認証および SLO リクエストの HTTP-POST バインディングを有効にできます。

次の手順に従ってください:

1. 管理 UI を開きます。
2. 変更するパートナーシップがアクティブな場合は、非アクティブにします。
3. ［変更］をクリックして、パートナーシップ ウィザードを開きます。
4. パートナーシップ ウィザードで［SSO と SLO］タブに移動します。
5. SSO セクションで、認証リクエスト バインディングに HTTP-POST を選択します。

   注: 認証リクエストに HTTP リダイレクトおよび HTTP-POST バインディングを共に選択できます。

6. 認証リクエスト バインディングに一致するバインディングでリモート SSO サービス URL を指定します。 たとえば、HTTP リダイレクトおよび HTTP-POST バインディングを選択した場合は、各バインディングに 1 つずつ、2 つの SSO サービス URL を作成します。
7. （オプション）［SLO］セクションで［HTTP POST］チェックボックスをオンにします。

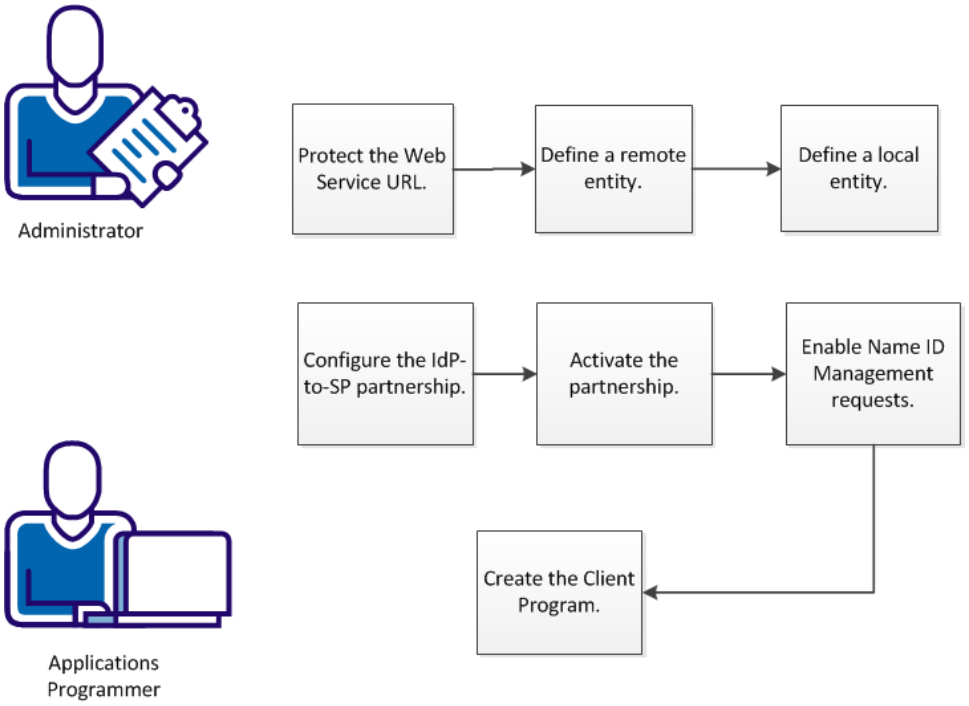
   注: 複数の SLO バインディングを選択できます。
8. SLO バインディングに一致するバインディングで SLO サービス URL を作成します。 たとえば、HTTP リダイレクトおよび HTTP-POST SLO バインディングを選択した場合は、各バインディングに 1 つずつ、2 つの SLO サービス URL を作成します。
9. 必要に応じて、その他のパートナーシップ情報を入力します。
10. 確認手順で［完了］をクリックします。

SSO HTTP-POST バインディングが有効になりました。

SAML 2.0 名前 ID 管理プロファイルの設定

SAML 2.0 名前 ID プロファイルを使用すると、フェデレーション パートナーシップから個別のユーザのプロビジョニングを解除できます。 さまざまな理由によりパートナーシップからユーザを削除できます。 たとえば、従業員が会社を辞めた場合や、サービス プロバイダで SSO 機能が必要なくなった場合などです。 プロビジョニング解除リクエストは、クライアント アプリケーション プログラムを使用して行います。

以下の図では、SAML 2.0 名前識別子プロファイルの実装プロセスを示します。



名前 ID 管理プロファイルを使用してユーザのプロビジョニングを解除するには、以下の手順が必要です。

1. Name Identifier Management Administration Web サービス URL を保護します。 (P. 187)
2. 名前 ID 管理に対してリモート エンティティを設定します。 (P. 187)
3. ローカル エンティティを作成します。 (P. 188)
4. 名前 ID 管理に対してパートナーシップを設定します。 (P. 188)
5. パートナーシップをアクティブにします。 (P. 190)
6. 名前 ID 管理リクエストを有効にします。 (P. 190)
7. Name Identifier Web サービスと対話するクライアント アプリケーションを作成します。 (P. 191)

Name Identifier Management Administration Web サービス URL を保護する

カスタマ アプリケーションは、Name Identifier Management Administration Web サービスを使用して、ユーザのプロビジョン解除をパートナーシップにリクエストします。この Web サービスは REST インターフェースを実装します。

このサービスの URL は /affwebservices/saml2nidws です。SiteMinder 基本認証情報を使用して、この URL を保護します。 ドメインに関連付けられたユーザ ディレクトリに、このサービスの任意のユーザを含めます。SiteMinder ポリシー管理者はデフォルトでは含まれていません。それらは、関連するディレクトリに手動で追加できます。

名前 ID 管理に対するリモート エンティティの設定

名前 ID 管理をサポートするパートナーシップを作成する最初の手順は、リモートおよびローカルのパートナーまたはエンティティを定義することです。 エンティティを手動で設定することも、XML メタデータをインポートすることできます。 以下の手順はマニュアル設定向けです。

次の手順に従ってください:

1. 管理 UI で、［フェデレーション］、［パートナーシップ フェデレーション］、［エンティティ］の順に移動します。
2. ［エンティティの作成］をクリックします。
3. ［リモート］（実装に応じて、IdP または SP のいずれか）を選択します。
4. エンティティに関する詳細を設定するために［次へ］をクリックします。
5. ［エンティティ ID］および［エンティティ名］（必須）の値を入力します。
6. ［名前 ID サービス URL の管理］の行にある［行の追加］をクリックします。
7. SOAP バインディングを選択します。 リモート エンティティは別のバインディングを指定できます。これはインポートされますが、未使用です。

8. ロケーション URL を入力します。これによって、名前 ID 管理サービスの URL が指定されます。 この値を以下に示します。

   http://*sp_server:port*/affwebservices/public/saml2nidsoap

9. ［レスポンス ロケーション URL］フィールドは空白のままにします。SOAP バインディングのレスポンス ロケーション URL はロケーション URL と同じです。
10. サポートされている名前 ID 形式をこのリストから選択します。
11. 実装に必要なフィールドがほかにあれば、入力します。
12. ［次へ］をクリックして、エンティティ設定を確認します。
13. ［完了］をクリックします。

ローカル エンティティの作成

名前 ID 管理をサポートするパートナーシップを作成する最初の手順は、リモートおよびローカルのパートナーまたはエンティティを定義することです。 エンティティを手動で設定することも、XML メタデータをインポートすることできます。 パートナーシップでのエンティティの作成に慣れていない場合は、「フェデレーション エンティティ設定 (P. 71)」を参照してください。

重要: ユーザ プロビジョニング解除またはリンク解除に対して名前 ID 形式を選択できます。 動的アカウント リンクは永続的な ID 形式のみをサポートします。 アカウント リンクとリンク解除を実装している場合は、永続的識別子名前 ID 形式を選択します。

名前 ID 管理に対するパートナーシップの設定

名前 ID 管理機能を有効にするには、新しいパートナーシップまたは既存のパートナーシップについて何らかの設定が必要です。 ローカル エンティティまたはリモート エンティティのいずれでも、ユーザのプロビジョニングを解除するリクエストをパートナーシップに出すことができます。

次の手順に従ってください:

1. ［SSO と SLO］ダイアログ ボックスに移動します。
2. ［認証］および［SSO］セクションがまだ設定されていない場合は、その設定を行います。
3. ［マネージャ名 ID］セクションに移動します。
4. ［MNI］フィールドで［SOAP］を選択します。

   この選択により、パートナーシップにおける名前 ID 管理が有効になります。これらのオプションの説明については、オンライン ヘルプを参照してください。
5. （必須）SOAP タイムアウト値を指定します。 この値は、リモート プロバイダへのリクエストがタイムアウトするまで、ランタイムが待機する秒数です。

   デフォルト： 60 秒
6. （必須）再試行回数を指定します。これは、失敗を宣言する前に、バックグラウンド リクエストが試行される回数です。デフォルト値は 3 です。
7. （必須）再試行境界を指定します。これは、再試行間隔の時間（分）です。 デフォルトは 15 分です。
8. ［通知を許可］オプションを選択している場合は、［通知 URL］を指定します。 この URL は、カスタマ アプリケーションに HTTP 通知を送信する場所です。 通知には、プロビジョニング解除リクエストの完了後のプロビジョニング解除リクエストのステータスが含まれます。
   - ステータス 1 - プロビジョニング解除成功
   - ステータス 0 - プロビジョニング解除失敗
9. ［通知タイムアウト］を指定します。これは、リクエストがタイムアウトと見なされるまで待機する秒数です。

   デフォルト： 60 秒
10. 通知認証タイプ（［認証なし］または［ベーシック］）を指定します。［ベーシック］を選択した場合は、ユーザ名とパスワードを指定します。

注: この機能が正しく機能するには、［MNI］セクションで［名前 ID の削除］または［通知の有効化］オプションまたはその両方を選択します。

これらの手順は、マネージャ名 ID 設定を完了します。

パートナーシップのアクティブ化

詳細については、「パートナーシップ アクティブ化 (P. 89)」を参照してください。

名前 ID 管理リクエストの有効化

非同期リクエスト プロセッサという名前の Web エージェント オプション パック内部コンポーネントは、名前 ID 管理サービスへのリクエストをすべて処理します。一度に 1 つの Web エージェント オプション パックだけがでこのサービスを実行できます。 管理 UI の設定に加えて、以下の場所で AffWebServices.properties ファイルで設定を指定することにより、名前 ID 管理の処理を有効にします。

- SPS: <SECURE_PROXY_HOME>/Tomcat/webapps/affwebservices/WEB-INF/classes
- WA+WAOP: <WEB_AGENT_HOME>/affwebservices/WEB-INF/classes

AffWebServices.properties ファイルには、名前 ID 管理に関連する以下の設定が含まれます。

ProcessBackgroundNameIDOperations

このシステムで名前 ID 操作を処理するかどうかを指定します。

デフォルト： False

重要: オプション パックまたは SPS に対して名前 ID 管理を有効にするには、この値を True に設定する必要があります。

BackgroundProcessingInterval

非同期プロセッサが名前 ID リクエストを確認する時間間隔を秒数で指定します。 この値は変更できます。

デフォルト： 60 秒

オプション パックまたは SPS をアップグレードする場合、インストーラはこれらの設定とそのデフォルト値を新しいプロパティ ファイルに追加します。

Name Identifier Web サービスと対話するクライアント アプリケーションの作成

クライアント アプリケーションのコンテンツは実装によって異なります。ユーザの削除をリクエストするには、Name Identifier Management Administration Web サービスを使用します。 Web サービスは以下の 2 つの HTTP メソッドを実現します。

- POST -- プロビジョニング解除リクエストを開始する。
- GET -- リクエストのステータスをポーリングする。

これらのメソッドは OData プロトコルに準拠しています。 これらのメソッドに関する詳細については後のセクションで説明されています。

フェデレーション メンバシップの終了

管理者は、以下の URL を使用することにより、ユーザのフェデレーション メンバシップを終了できます。

```
POST http://<server+port>/affwebservices/saml2nidws/terminate
```

この非同期リクエストによって、XPS に ManageNameID イベントが作成されます。

POST 本体には、以下の値が含まれています。

UserDN

SMSession がないので、ユーザを明確にします。 LDAP の DN はたとえば、uid=user0001,ou=Engineering,o=security.com となります。

OperationType

特定のユース ケースを示します。 有効値は以下のとおりです。

- sp - 特定のサービス プロバイダとのフェデレーションを終了する idp を示します。
- idp - 特定のアイデンティティ プロバイダとのフェデレーションを終了するサービス プロバイダを示します。

ProviderID

オペレーションの一部であるプロバイダを特定します。 sp 値および idp 値の場合、ProviderID はリモート プロバイダを識別します。 OperationType が 'sp' である場合、ProviderID はリモート サービス プロバイダ オブジェクトを表します。 OperationType が 'idp' である場合、ProviderID はリモート アイデンティティ プロバイダ オブジェクトを表します。

リクエストの POST 本文内の情報は JSON または AtomPub 形式です。 以下の例は JSON 形式です。

```
{
"UserDN":"uid=user0001,ou=Engineering,o=security.com",
"OperationType":"sp",
"ProviderID":"http://company.example.com/SPID"
}
```

このリクエストは、この永続化オブジェクトを表すリソースを返します。 たとえば、次のとおりです。

```
http://<server+port>/affwebservices/saml2nidws/terminate(<XID>)
```

<XID> は作成されたオブジェクトの XPS XID です。 クライアントは、このオブジェクトの変更をポーリングするためにこの URL を使用できます。

リクエストは、以下のような完全な AtomPub 形式にすることもできます。

```
<?xml version="1.0" encoding="utf-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
>
<title type="text"></title><author><name></name></author>
<category term="NameidProducer.terminate"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"><
/category>
<content type="application/xml">
<m:properties>
<d:UserDN>uid=user0001,ou=Engineering,o=security.com</d:UserDN><d:Provi
derID>http://company.example.com/SPID</d:ProviderID>
<d:OperationType>SP</d:OperationType>
</m:properties>
</content>
</entry>
```

POST サービスは以下の HTTP リターン コードを設定します。

HTTP ステータス	説明
201	リソースが作成されました
400	リクエストが正しくありません
415	メディア タイプがサポートされていません
500	内部サーバ エラー

ステータスのポーリング

管理者は、このサービスを使用することで、次の URL を使用して非同期リクエストのステータスをリクエストできます。

```
GET http://<server+port>/affwebservices/saml2nidws/terminate(<XID>)
```

リソース ステータスのポーリングに使用される URL。

レスポンスはリクエストのステータスを返します（PENDING、COMPLETED または FAILED のいずれか）。

重要: このリクエストを実行する前に、エージェント設定オブジェクトの CssChecking パラメータが NO に設定されていることを確認してください。この設定により、OData とクロス サイト スクリプティング攻撃の間の構文の潜在的な競合が回避されます。

GET サービスは以下の HTTP リターン コードを設定します。

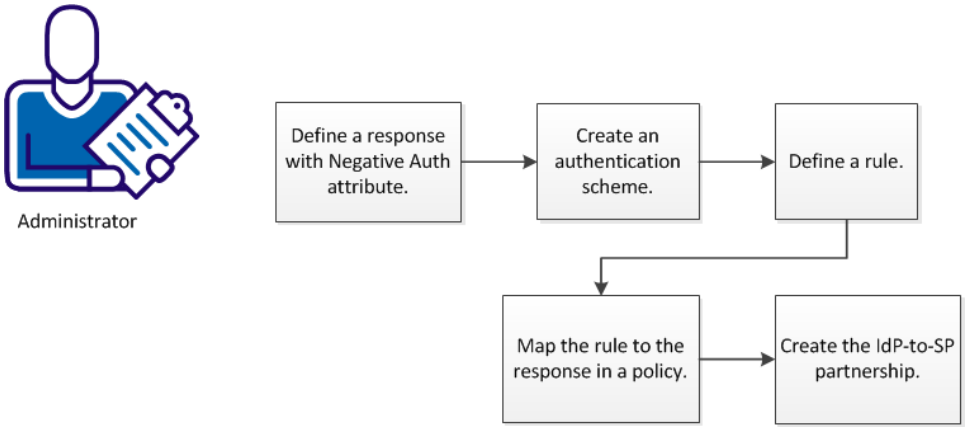
HTTP ステータス	説明
200	OK
400	リクエストが正しくありません
403	禁止されています（Web エージェントに対して CSS チェックがオンの場合）
415	メディア タイプがサポートされていません
500	内部サーバ エラー

認証失敗に対する SAML2.0 レスポンスの設定

このプロセスを使用して、認証失敗時のサービス プロバイダに対する非アサーション レスポンスを設定できます。 SAML 2.0 認証リクエストが成功した場合、サービス プロバイダへのレスポンスは認証アサーションを伴います。 認証リクエストが拒否された場合、以前はエンド ユーザにエラー メッセージが表示されるのみでした。 サービス プロバイダは、失敗ステータスの通知を取得しませんでした。 コントロールはサービス プロバイダに戻されるため、サービス プロバイダが、ユーザをリダイレクトすべきか、他の適切なアクションを実行すべきかを判断できます。

重要: この機能が動作するには、ポリシー サーバ、Web エージェント、および Web エージェント オプション パックがすべて SiteMinder r12.52 以降である必要があります。

以下の図は、この機能を設定するのに必要な手順を示します。



認証失敗時のサービス プロバイダに対するレスポンスを設定するプロセスには、以下の手順が含まれます。

1. 否定認証レスポンス属性を指定するレスポンスを定義します。 (P. 196)
2. 基本認証方式またはフォーム認証方式を作成します (P. 197)。
3. OnAuthReject アクションを指定するルールを定義します (P. 198)。
4. このルールをポリシー内で以前に定義されたレスポンスにマップします (P. 199)。
5. IdP から SP へのパートナーシップを設定して、否定認証レスポンスを有効にします (P. 199)。

否定認証レスポンス属性を指定するレスポンスの定義

WebAgent-OnReject-eGovNegResponse 属性タイプを使用してレスポンスの定義を開始します。 レスポンスを定義する場合は、ドメインが定義済であることを前提とします。

次の手順に従ってください：

1. ［ポリシー］-［ドメイン］-［レスポンス］の順に移動します。
2. ［レスポンスを作成］をクリックします。
3. 適切なドメインを選択するか、または新しいドメインを作成します。
4. ［次へ］をクリックします。
5. ［一般］セクションに、このレスポンスの名前および説明（オプション）を入力します。
6. 適切なエージェント タイプ（通常は、SiteMinder Web エージェント）を選択します。
7. ［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。
8. ［属性タイプ］セクションのドロップダウン リストから WebAgent-OnReject-eGovNegResponse を選択します。
9. ［相対ターゲットを使用］を選択するか、［属性フィールド］セクションに Web サーバ名を入力します。
10. （オプション）［SSL 接続を使用］を選択します。

    注: このセクションで選択した内容に基づいて、［詳細］セクションのペインにスクリプトが表示されます。 詳細については、オンラインヘルプを参照してください。
11. ［属性キャッシング］セクションで［キャッシュ値 値の再計算］を選択します。

12. ［OK］をクリックして、［レスポンスの作成： レスポンスの定義］ダイアログボックスに戻ります。
13. ［完了］をクリックします。

認証が失敗した場合に SP へのレスポンスを生成する適切な属性でレスポンスを定義しました。

基本認証方式またはフォーム認証方式を設定する

SP への認証失敗でレスポンスを生成するために、基本またはフォーム方式を設定できます。

次の手順に従ってください:

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。
4. ［OK］をクリックします。
5. 名前と保護のレベルを入力します。
6. ［認証方式のタイプ］リストから［基本またはフォーム テンプレート］を選択します。
7. ［サブミット］をクリックします。

   認証方式が保存され、これでレルムに割り当て可能になります。

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試みたときに発生するアクションを制御するルールを設定できます。 認証の失敗時の完全な SAML 2.0 応答については、OnAuthReject アクションを選択します。

レルムは認証イベントを処理できる必要があります。 ［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。 レルムの作成方法の詳細については、次のトピックを参照してください。

次の手順に従ってください:

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. 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: 認証イベントはレルム全体に適用されるため、［リソース］フィールドは無効になります。 ［アクセス許可］オプションおよび［アクセス拒否］オプションは認証イベントには適用されないため、これらのオプションも無効になります。
7. OnAuthReject アクションを選択します。
8. （オプション）［詳細］セクションで、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、指定されたレルムおよびリソースに適用されます。

OnAuthReject アクションを使用して適切なレスポンスにルールをマップする

OnAuthReject アクションを使用して作成したルールを、ポリシー内の eGovNegResponse 属性に関連付けます。

次の手順に従ってください：

1. ［ポリシー］-［ドメイン］-［ポリシー］の順に移動します。
2. ポリシーを選択します。
3. ルールに移動する
4. OnAuthReject アクションで作成したルールがルール リストに載っていることを確認します。
5. 該当するルールの横の［レスポンスの追加］をクリックします。
6. eGovNegResponse 属性タイプで指定したレスポンスを選択します。
7. 保存して終了します。

ルールが適切なレスポンスに関連付けられました。

IdP から SP へのパートナーシップを設定して否定認証レスポンスをサポートする

IdP から SP へのパートナーシップ設定手順で、否定認証レスポンスを有効にします。［否定認証レスポンスの有効化］チェック ボックスを選択します。

詳細については、「シングル サインオンの設定 (P. 129)」を参照してください。

第 12 章: ソーシャル サインオンの設定

CA SiteMinder® Federation（フェデレーション システム）は、ユーザがフェデレーション システム認証情報の代わりにソーシャル ネットワーキング認証情報を使用してフェデレーション リソースにサインオンできるように設定することができます。

ソーシャル サインオン機能は以下の機能から構成されます。

- Facebook などの OAuth 許可サーバを使用したユーザの認証。これにより、ユーザは OAuth 許可サーバ認証情報を使用してフェデレーション リソースにサインオンできます。
- 認証情報セレクタ ページの設定。このページでは、認証の選択肢として SAML 2.0 や Facebook などのさまざまなアイデンティティ プロバイダがユーザに提供されます。ユーザは、フェデレーション リソースにサインオンするための認証に対してアイデンティティ プロバイダを選択できます。

これらの機能は互いに依存するものではなく、いずれかの機能または両方の機能を実装するフェデレーション システムを設定できます。

OAuth 許可サーバを使用したユーザの認証

OAuth 許可サーバを使用してユーザを認証するには、フェデレーション システムと OAuth 許可サーバの間のシングル サインオンを設定します。

フェデレーション システムは、以下の OAuth 許可サーバのデフォルト サポートを提供します。

OAuth 1.0a

- Twitter

OAuth 2.0

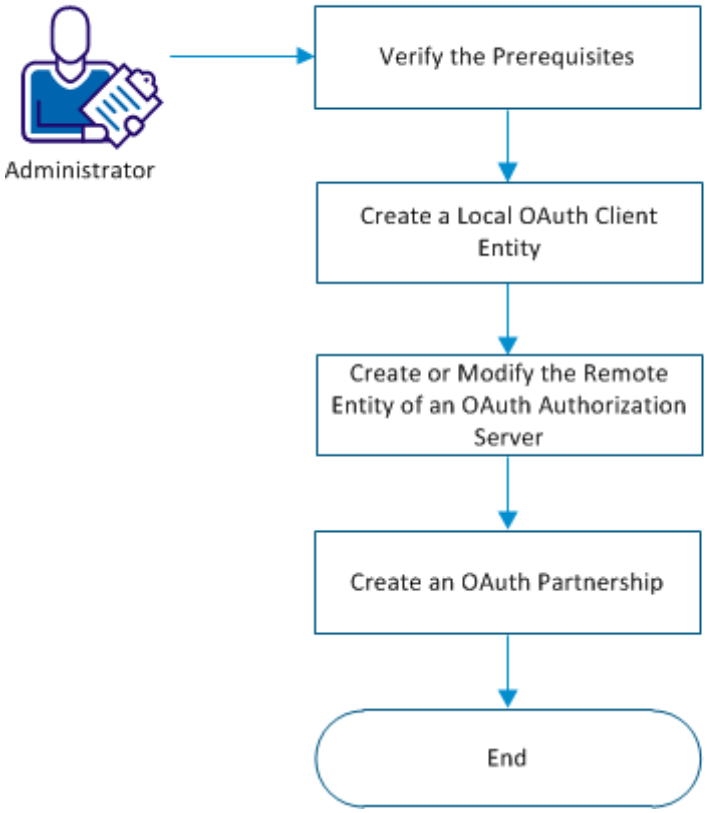
- Facebook
- Google
- LinkedIn
- Windows Live

以下のプロセスは、フェデレーション リソースにアクセスするためにフェデレーション システムがどのようにユーザ リクエストを処理するかを示しています。

1. フェデレーション システムは、ユーザ リクエストで指定された OAuth 許可サーバにユーザ リクエストをリダイレクトします。
2. OAuth 許可サーバは、ユーザを認証し、ユーザに関するクレームを持つ認証レスポンスをフェデレーション システムに送信します。
3. フェデレーション システムは認証レスポンスを確認し、認証プロセスを完了して、ユーザがフェデレーション リソースにアクセスすることを許可します。

以下のフローチャートは、OAuth 許可サーバを使用して、どのようにユーザを認証できるかを示しています。

Authenticate Users Using an OAuth Authorization Server



次の手順に従ってください:

1. 前提条件を確認します (P. 203)。
2. ローカルの OAuth クライアント エンティティを作成します (P. 204)。
3. (オプション) OAuth 許可サーバのリモート エンティティを作成または変更します (P. 204)。
4. シングル サインオン用の OAuth パートナーシップを作成します (P. 206)。

前提条件の確認

フェデレーション システムと OAuth 許可サーバの間のシングル サインオンを設定するには、パートナーシップを設定する前に以下の手順に従います。

- フェデレーション システムで SSL を有効にします。
- フェデレーション システムがデフォルトでサポートする OAuth 許可サーバを使用するには、パートナーシップを呼び出す前に以下の手順に従います。
  - スタンドアロン展開では、OAuth 許可サーバのデフォルトの CA 証明書がインポートされていることを確認します。
  - 統合展開では、smkeytool を使用して OAuth 許可サーバのデフォルトの CA 証明書をインポートします。
- フェデレーション システムがデフォルトではサポートしない OAuth 許可サーバを使用するには、パートナーシップを呼び出す前に、OAuth 許可サーバの SSL CA 証明書を取得およびインポートします。

ローカルの OAuth クライアント エンティティの作成

フェデレーション システムと OAuth 許可サーバの間のパートナーシップについてローカル OAuth クライアント エンティティを作成します。

次の手順に従ってください:

1. ［フェデレーション］-［エンティティ］に移動し、［エンティティの作成］をクリックします。
2. ［エンティティ ロケーション］で［ローカル］を選択します。
3. ［新規エンティティ タイプ］から［OAuth クライアント］を選択します。
4. OAuth バージョンを選択して［次へ］をクリックします。
5. 必要な値を入力して［次へ］をクリックします。
6. 入力した値を確認し、［完了］をクリックします。

   リダイレクト URL が生成されます。 OAuth トランザクションを開始するためにこの URL を使用します。

許可サーバのリモート エンティティの作成または変更

システムは、デフォルトでサポートされている以下の OAuth 許可サーバそれぞれに対して、リモート エンティティを提供します。

OAuth 1.0a

- Twitter

OAuth 2.0

- Facebook
- Google
- LinkedIn
- Windows Live

各リモート エンティティの値は、エンティティの既知の値であらかじめ設定されています。 実際のフェデレーション環境にあわせて値を変更するか、または OAuth 許可サーバ用のリモート エンティティを作成できます。

次の手順に従ってください:

1. 以下の*いずれか*のタスクを実行します。

   新しいリモート エンティティを作成します。

   a. ［フェデレーション］-［エンティティ］-［エンティティの作成］に移動します。

   b. ［エンティティ ロケーション］で［リモート］を選択し、［新規エンティティ タイプ］として［OAuth Authz サーバ］を選択します。

   c. ［次へ］をクリックします。

   d. 値を入力して［次へ］をクリックします。

   リモート エンティティにあらかじめ入力されている値を変更します。

   a. ［フェデレーション］-［エンティティ］に移動し、変更するエンティティを検索します。

   b. エンティティの［アクション］オプションをクリックし、［変更］をクリックします。

   c. ［次へ］をクリックして［エンティティの設定］タブに移動します。

   d. 値を変更して［次へ］をクリックします。

2. 変更を確認し、［完了］をクリックします。

シングル サインオン用の OAuth パートナーシップの作成

フェデレーション システムがユーザ情報を許可サーバから取得できるようにするには、OAuth パートナーシップを、OAuth 許可サーバをアサーティング パーティ、フェデレーション システムを依存パーティとして作成します。

次の手順に従ってください:

1. ［フェデレーション］-［パートナーシップ］に移動し、［パートナーシップの作成］をクリックします。
2. ［OAuth クライアント -> Authz サーバ］のパートナーシップ タイプを選択します。
3. パートナーシップ情報を設定します。
4. 値を確認して［完了］をクリックします。

OAuth パートナーシップが設定され、ユーザは OAuth 許可サーバの認証情報を使用してフェデレーション リソースにサインオンできるようになります。

フェデレーション システムがユーザ リクエストを以下の形式で受信した場合、リクエストはパートナーシップ設定に従って処理されます。

https://*baseURL_of_the_partnership*/affwebservices/public/oauthtokenconsumer?AuthzServerID=*authorization_server_id*

または

https://*baseURL_of_the_partnership*/affwebservices/public/oauthtokenconsumer/*disambiguation_id*?AuthzServerID=<*authorization_server_id*>

フェデレーション システムは、ソーシャル サインオン機能を実装するように設定されています。

OAuth パートナーシップへの OAuth 認証方式セット アップの移行

OAuth プロバイダを使用してユーザを認証するように OAuth 認証方式を設定した場合、使用する認証方式セット アップをフェデレーション パートナーシップに移行できます。

次の手順に従ってください:

1. 以下の手順のいずれかを実行します。
   - OAuth 認証方式および OAuth パートナーシップの両方を同時に使用する場合は、OAuth 許可サーバにアプリケーションを登録し、以下の形式の新しいリダイレクト URL を既存の OAuth 認証方式リダイレクト URL に追加します。

     `https://server:port/affwebservices/public/oauthtokenconsumer`
   - OAuth 認証方式の代わりに OAuth パートナーシップを使用する場合は、OAuth 許可サーバで既存のリダイレクト URL を、以下の形式の適切なパートナーシップ リダイレクト URL に更新します。

     `https://server:port/affwebservices/public/oauthtokenconsumer`

     注: パートナーシップ リダイレクト URL で認証方式リダイレクト URL を更新した後は、OAuth 認証方式は機能しなくなります。
2. OAuth クライアントおよび OAuth 許可サーバ間のパートナーシップを作成します。
3. OAuth パートナーシップを開始するには以下の URL を使用する必要があることをアプリケーション ユーザに伝えます。

   `https://server:port/affwebservices/public/oauthtokenconsumer?AuthzServerID=AuthorizationServerID`

[認証情報セレクタ]ページの設定

ユーザが認証に対して Facebook や Twitter などのアイデンティティ プロバイダを選択できるようにパートナーシップを設定することができます。CA SiteMinder for Secure Proxy Server にインストールされた認証情報処理サービスでは、パートナーシップを設定することにより、ユーザ認証の選択肢として複数のアイデンティティ プロバイダを含む認証情報セレクタ ページを表示できます。

認証情報セレクタ ページを設定するには、以下のパートナーシップを作成します。

1. フェデレーション システムおよびアイデンティティ プロバイダ間のシングル サインオンを設定するためのパートナーシップ。 アイデンティティ プロバイダがアサーティング パーティとして、フェデレーション システムが依存パーティとして機能します。
2. フェデレーション リソースが存在する企業とフェデレーション システムの間のパートナーシップ。 フェデレーション システムがアサーティング パーティとして、企業が依存パーティとして機能します。

以下のプロセスでは、フェデレーション システムがユーザ リクエストをどのように処理するかを表しています。

1. 企業（依存パーティ）は、ユーザ リクエストをフェデレーション システム（アサーティング パーティ）にリダイレクトします。
2. フェデレーション システム（アサーティング パーティ）は、認証情報セレクタ ページを表示するようにパートナーシップが設定されているかどうかを確認します。 設定されている場合、ユーザ認証の選択肢として複数のアイデンティティ プロバイダを含む認証情報セレクタ ページが表示されます。
3. ユーザがフェデレーション システムで登録されている場合、以下の手順が実行されます。 ユーザが登録されていない場合は、次の手順に進みます。
    a. ユーザはアイデンティティ プロバイダを選択し、アイデンティティ プロバイダにサインオンします。
    b. アイデンティティ プロバイダは、アクセス トークンを生成し、ユーザをフェデレーション システム（依存パーティ）にリダイレクトします。
    c. フェデレーション システム（依存パーティ）は、アクセス トークンを確認し、ユーザ ストア内のユーザを特定しようとします。
    d. フェデレーション システム（依存パーティ）は、セッションを生成し、ユーザをフェデレーション システム（アサーティング パーティ）にリダイレクトします。
    e. フェデレーション システム（アサーティング パーティ）は、アサーションを生成し、ユーザを企業（依存パーティ）にリダイレクトします。
    f. 企業（依存パーティ）は、アサーションを確認し、フェデレーション リソースに対するユーザ アクセス権を付与します。

4. ユーザがフェデレーション システムで登録されていない場合、以下の手順が実行されます。

   a. ユーザは登録リンクをクリックします。

   b. フェデレーション システムは、パートナーシップがプロビジョニング サーバで設定されているアイデンティティ プロバイダのリストを表示します。

   c. ユーザはアイデンティティ プロバイダを選択し、アイデンティティ プロバイダにサインオンします。

   d. アイデンティティ プロバイダは、アクセス トークンを生成し、ユーザをフェデレーション システム（依存パーティ）にリダイレクトします。

   e. フェデレーション システム（依存パーティ）は、アクセス トークンを確認し、ユーザ ストア内のユーザを特定しようとします。

   f. フェデレーション システム（依存パーティ）は、パートナーシップで設定されたプロビジョニング サーバにユーザをリダイレクトします。

   g. プロビジョニング サーバは、ユーザを作成し、ユーザをフェデレーション システム（依存パーティ）にリダイレクトします。

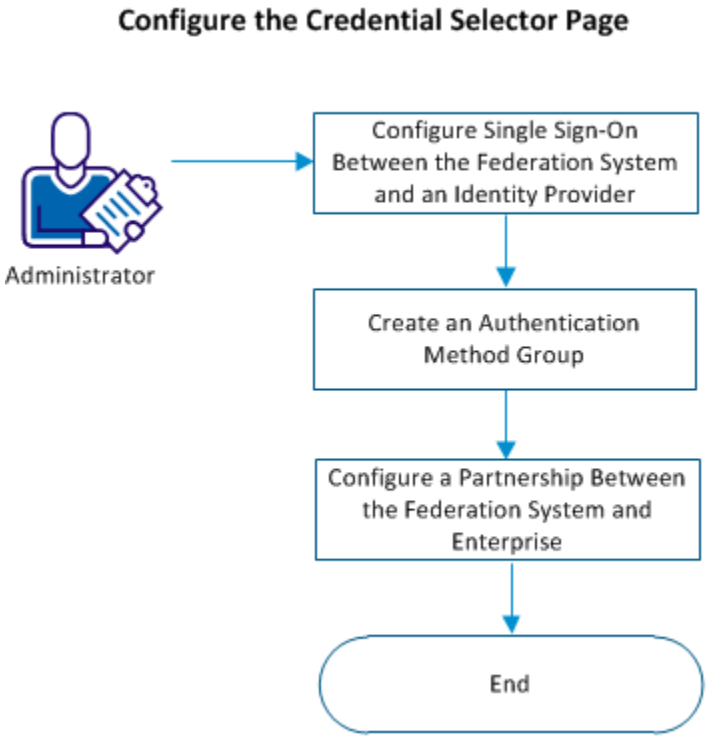
   h. フェデレーション システム（依存パーティ）は、セッションを生成し、ユーザをフェデレーション システム（アサーティング パーティ）にリダイレクトします。

   i. フェデレーション システム（アサーティング パーティ）は、アサーションを生成し、ユーザを企業（依存パーティ）にリダイレクトします。

   j. 企業（依存パーティ）は、アサーションを確認し、フェデレーション リソースに対するユーザ アクセス権を付与します。

ユーザ リクエストが処理されます。

以下のフローチャートは、認証情報セレクタ ページを設定する方法を示しています。



次の手順に従ってください:

1. フェデレーション システムとアイデンティティ プロバイダの間のシングル サインオンを設定します。 (P. 211)
2. 認証方式グループを作成します (P. 212)。
3. フェデレーション システムと企業の間のパートナーシップを設定します (P. 213)。

フェデレーション システムとアイデンティティ プロバイダ間のシングル サインオンの設定

認証情報セレクタ ページに表示する各アイデンティティ プロバイダに対して、アイデンティティ プロバイダとフェデレーション システム間のシングル サインオンを設定するためにパートナーシップを形成します。 アイデンティティ プロバイダがアサーティング パーティとして、フェデレーション システムが依存パーティとして機能します。

認証の選択肢として使用できるアイデンティティ プロバイダは、以下の認証プロトコルに基づいている必要があります。

- SAML 1.1
- SAML 2.0
- WS-フェデレーション
- OAuth

フェデレーション システムがアイデンティティ プロバイダとして機能するには、アサーティング パーティおよび依存パーティの両方として機能するシステムとのパートナーシップを作成します。

次の手順に従ってください:

1. ［フェデレーション］-［パートナーシップ］に移動します。
2. 認証情報セレクタ ページに表示する各アイデンティティ プロバイダに対してパートナーシップを作成します。

認証方式グループの作成

認証方式グループは、認証情報セレクタ ページに表示する必要があるアイデンティティ プロバイダのリストを定義します。SAML または Facebook など、認証情報セレクタ ページに表示する各アイデンティティ プロバイダは、認証方式グループの一部である必要があります。 認証方式グループを作成する場合、アサーティング パーティとして機能するアイデンティティ プロバイダとのすべてのパートナーシップのリストからアイデンティティ プロバイダを選択できます。

次の手順に従ってください:

1. ［インフラストラクチャ］-［認証方式グループ］に移動します。
2. ［認証方式グループの作成］をクリックします。
3. 認証の選択肢として表示するアイデンティティ プロバイダのパートナーシップを追加し、必要な値を入力します。
4. 変更を保存します。

フェデレーション システムと企業の間のパートナーシップの設定

ユーザがフェデレーション リソースにアクセスしようとする場合に、認証情報セレクタ ページを表示するフェデレーション システムとユーザの企業の間のパートナーシップを設定します。 フェデレーション システムがアサーティング パーティとして、企業が依存パーティとして機能します。 パートナーシップを作成するか、既存のパートナーシップを変更できます。

パートナーシップは以下のいずれかの認証プロトコルに基づいてる必要があります。

- SAML 1.1
- SAML 2.0
- WS-フェデレーション

次の手順に従ってください:

1. ［フェデレーション］-［パートナーシップ］に移動します。
2. 各手順に値を入力します。
3. ［シングル サインオン］、［SSO と SLO］、または［シングル サインオンおよびサインアウト］で以下の手順に従います。
   a. 認証情報セレクタとして［認証モード］を選択します。
   b. 認証ベース URL を定義します。
   c. 認証方式グループを選択します。
4. ［ターゲット アプリケーション］手順で以下のフィールドを選択します。
   - SAML 1.1： ターゲット
   - SAML 2.0 および WS フェデレーション： リレー状態を使用してターゲットをオーバーライドする
5. 変更を保存します。

ユーザがフェデレーション リソースにアクセスしようとした場合に認証情報セレクタ ページを表示するようにパートナーシップが設定されます。

フェデレーション システムは、ソーシャル サインオン機能を実装するように設定されています。

[認証情報セレクタ]ページでのヘッダおよびフッタのカスタマイズ

［認証情報セレクタ］ページに表示されるヘッダおよびフッタは、ユーザの要件に合わせてカスタマイズできます。

次の手順に従ってください:

1. フェデレーション システムで以下の場所に移動します。

   <*install_path*>¥CA¥secure-proxy¥Tomcat¥webapps¥chs¥jsps

2. header.jsp ファイルのコピーを作成し、新しいファイルの名前を header-custom.jsp にします。
3. footer.jsp ファイルのコピーを作成し、新しいファイルの名前を footer-custom.jsp にします。

   注: header-custom.jsp および footer-custom.jsp ファイルが存在する場合、フェデレーション システムはヘッダおよびフッタの表示にこのファイルを使用するように設定されます。
4. 認証情報セレクタ ページに表示される必要があるヘッダおよびフッタをカスタマイズするためにファイルを変更します。
5. 変更を保存します。
6. CA SiteMinder for Secure Proxy Server を再起動します。

   パートナーシップがアクティブな場合、カスタマイズされたヘッダおよびフッタは認証情報セレクタ ページに表示されます。

第 13 章: アサーション処理のカスタマイズ化（依存パーティ）

メッセージ コンシューマ プラグインは、Message Consumer Extension API を実装する Java プログラムです。 プラグインを使用することにより、アサーションを拒否したり、ステータス コードを返したりなど、アサーションを処理するための独自のビジネス ロジックを実装できます。 この追加の処理は、アサーションの標準的な処理と連携して動作します。

認証時、システムは、まず、ユーザをそのローカル ユーザ ストアにマップすることによりアサーションを処理しようと試みます。 そのユーザを検索できない場合、CA SiteMinder® Federation はメッセージ コンシューマ プラグインの postDisambiguateUser メソッドをコールします。

プラグインで正常にユーザが検索された場合、プロセスは認証の第 2 段階に進みます。 プラグインでユーザをローカル ユーザ ストアにマップできない場合、プラグインから UserNotFound エラーが返されます。 プラグインでは、オプションでリダイレクト URL 機能を使用できます。コンシューマ プラグインを使用しない場合、リダイレクト URL は、SAML 認証方式によって生成されるエラーに基づきます。

認証の第 2 段階では、システムはメッセージ コンシューマ プラグインの postAuthenticateUser メソッドをコールします（プラグインが設定されている場合）。 メソッドが成功した場合、CA SiteMinder® Federation はユーザをリクエストされたリソースにリダイレクトします。 メソッドが失敗する場合、ユーザを失敗ページに移動するようにプラグインを設定できます。失敗ページとして、認証方式設定で指定可能なリダイレクト URL の 1 つを使用できます。

参照情報（メソッドの署名、パラメータ、戻り値、データ型）、および UserContext クラスのコンストラクタが「*ava SDK Programming Reference*」にあります。 MessageConsumerPlugin インターフェースを参照してください。

プラグインを設定する方法：

1. CA SiteMinder® Federation SDK をインストールするには、以下の手順に従います。
2. MessageconsumerPlugin.java インターフェース（SDK に含まれています）を実装します。
3. メッセージ コンシューマ プラグイン実装クラスを展開します。
4. 管理 UI でメッセージ コンシューマ プラグインを有効にします。

MessageConsumerPlugin の実装

MessageConsumerPlugin.java インターフェースを実装するにより、カスタム メッセージ コンシューマ プラグインを作成します。実装クラスの最小要件は、以下の手順に示されています。

次の手順に従ってください:

1. パラメータが含まれない公のデフォルト コンストラクタ メソッドを提供します。
2. 実装がステートレスになるように、コードを提供します。 多数のスレッドが 1 つのプラグイン クラスを使用できる必要があります。
3. 現実の要件に応じて、インターフェース内のメソッドを実装します。

   MessageConsumerPlugin には、以下の 4 つのメソッドが含まれています。

   init()

   プラグインが必要とする初期化手順を実行します。 プラグインがロードされると、SiteMinder はプラグイン インスタンスごとに、このメソッドを 1 回コールします。

   release()

   プラグインが必要とするあらゆる要約手順を実行します。 SiteMinder のシャットダウン中、SiteMinder はプラグイン インスタンスごとに、このメソッドを 1 回コールします。

postDisambiguateUser()

認証方式がユーザの不明瞭解消処理を実行できない場合に、この処理を提供します。また、このメソッドは、新しいフェデレーション ユーザに関するデータをユーザ ストアに追加できます。このメソッドは、復号されたアサーションを受信します。 復号されたアサーションは、キー「_DecryptedAssertion」の下のプラグインに渡されるプロパティに追加されます。

postAuthenticateUser()

ポリシー サーバ処理が成功か失敗かにかかわらず、アサーション処理の結果を決定する追加のコードを提供します。

製品では、Message Consumer プラグイン クラスの以下のサンプルが提供されます。

- MessageConsumerPluginSample.java
- MessageConsumerSAML20.java

サンプルのデフォルトの場所は以下のとおりです。

Windows

C:¥Program Files¥FederationManager¥sdk¥java¥sample

パッケージ名は com¥ca¥federation¥sdk¥plugin¥sample です。

UNIX

/FederationManager/sdk/java/sample

パッケージ名は com/ca/federation/sdk/plugin/sample です。

メッセージ コンシューマ プラグインの展開

MessageConsumerPlugin インターフェースの実装クラスをコード化した後、それをコンパイルし、CA SiteMinder® Federation が実行可能ファイルを検索できることを確認します。

次の手順に従ってください:

1. MessageConsumerPlugin Java ファイルをコンパイルします。 このファイルには、以下の依存ライブラリが必要になります。それらのライブラリは、製品と共にインストールされています。

   federation_install_dir¥siteminder¥bin¥jars¥SmJavaApi.jar

   federation_install_dir は、CA SiteMinder® Federation をインストールしたディレクトリです。

2. フォルダまたは jar ファイルで、プラグイン クラスが利用可能な場合には、JVMOptions.txt ファイル内の -Djava.class.path 値を変更します。この手順により、変更したクラスパスを使用してプラグイン クラスがロードできるようになります。

   ディレクトリ *federation_mgr_installation_home*¥siteminder¥config に JVMOptions.txt ファイルを置きます。

   注: 既存の xerces.jar、xalan.jar、SmJavaApi.jar のクラスパスを変更しないでください。

3. MessageConsumerPlugin の最新のバージョンを取得するためのシステムの再起動 この手順は、プラグイン Java ファイルが再コンパイルされるごとに必要です。
4. プラグインを有効化します。

UI でのメッセージ コンシューマ プラグインの有効化

メッセージ コンシューマ プラグインを作成してコンパイルした後に、管理 UI 内で設定することにより、このプラグインを有効にします。 UI 設定により、CA SiteMinder® Federation にプラグインの検索場所が指定されます。

プラグインを展開 (P. 218)するまで、プラグイン設定を実行しないでください。

メッセージ コンシューマ プラグインを有効にする方法

1. 管理 UI にログオンします。

   変更するコンシューマ - プロデューサまたは SP - IdP パートナーシップを選択します。

2. パートナーシップ ウィザードの［ユーザ識別］手順に移動します。

3. ［メッセージ コンシューマ プラグイン］セクションで、以下のフィールドに入力します。

   プラグイン クラス

   プラグインの Java クラス名を指定します。たとえば、SDK に含まれるサンプル クラスは次のとおりです。

   com.ca.messageconsumerplugin.MessageConsumerPluginSample

   プラグイン パラメータ

   ［完全 Java クラス名］フィールドで指定されたプラグインに渡されるパラメータ文字列を指定します。

4. オペレーティング環境に応じて、フェデレーション サービスを再起動します。

   - **Windows**

     以下のように、停止および開始ショートカットを使用します。ローカル管理者ではなく、ネットワーク ユーザとしてログインした場合は、ショートカットを右クリックし、［管理者として実行］を選択します。

     a. ［スタート］、［すべてのプログラム］、［CA］、［Federation Standalone］、［サービスの停止］

     b. ［スタート］、［すべてのプログラム］、［CA］、［Federation Standalone］、［サービスの開始］

   - **UNIX**

     a. コマンド ウィンドウを開きます。

     b. 以下のスクリプトを実行します。

     federation_install_dir/fedmanager.sh stop

     federation_install_dir/fedmanager.sh start

     注: root ユーザとしてサービスを停止したり開始したりしないでください。

第 14 章: 分散代行認証

分散代行認証の概要

シングル サインオン設定に関する決定の 1 つに、ユーザを認証する方法の決定があります。

SiteMinder では認証の選択肢を 2 つ提供しています。

- ローカル認証

  SiteMinder はローカル サイトでユーザを認証します。 ユーザを認証にリダイレクトする認証 URL を 管理 UI で設定し、セッションを確立します。

- 分散代行認証

  SiteMinder は、SiteMinder が保護しないサードパーティ Web アクセス管理（WAM）アプリケーションを使用します。 サードパーティのアプリケーションは、保護されているフェデレーション リソースを要求するすべてのユーザを認証して、フェデレーション ユーザ ID を SiteMinder に転送します。 SiteMinder はユーザ識別情報を受信した後、ユーザ ディレクトリでユーザを特定して、依存パーティでフェデレーション プロセスを開始します。

分散代行認証リクエストはアサーティング パーティで行われ、サードパーティ WAM システム、または SiteMinder で開始できます。 認証リクエストを依存パーティで開始できますが、この状況は分散代行認証とみなされません。

以下のように認証を開始できます。

アサーティング パーティで SiteMinder によって開始される認証

SiteMinder はアサーティング パーティで認証リクエストを開始できます。 リクエストが SiteMinder に対して行われると、分散代行認証リクエストとして認識されます。その後、SiteMinder はユーザをサードパーティ WAM システムにリダイレクトします。

アサーティング パーティで WAM システムに直接ログインすることによって開始される認証

ユーザがアサーティング パーティで WAM システムにログインすると、認証リクエストが開始されます。 WAM システムによるユーザ認証が成功すると、識別情報が SiteMinder に転送されます。

依存パーティで開始される認証

依存パーティは認証リクエストを開始できますが、この状況は分散代行認証とみなされません。分散代行認証はアサーティング パーティでのみ実行されます。

フェデレーション リソースのリクエストは依存パーティに直接行われて、依存パーティは認証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。 SiteMinder はそれを分散代行認証リクエストとして認識し、ユーザをアサーティング パーティのサードパーティ WAM システムにリダイレクトします。 ユーザは認証リクエストを開始する WAM システムにログインします。WAM システムによるユーザ認証が成功すると、識別情報が SiteMinder に転送されます。

サードパーティ WAM システムは認証リクエストを受信すると、ユーザ ID を SiteMinder に渡します。 ユーザ ID を渡すために WAM システムが使用する方法は、分散代行認証方法が Cookie ベースまたはクエリ文字列ベースのどちらかによって異なります。

サードパーティ WAM がユーザ ID を渡す方法

サードパーティ WAM システムは、以下の 2 つの方法のいずれかを使用してフェデレーション ユーザ ID を SiteMinder に渡します。

- オープン形式の Cookie を使用する。

  オープン形式の Cookie を暗号化してデータのセキュリティを確保できます。

- ブラウザを SiteMinder に送信するリダイレクト URL に追加されるクエリ文字列を使用する。

  クエリ文字列はクリア テキストで送信されます。

  重要: 実稼働環境でクエリ文字列方式を使用しないでください。 クエリ文字列リダイレクト方式は、概念実証としてテスト環境でのみ使用します。

サードパーティ WAM システムが選択する方法は、ユーザ ID を SiteMinder に渡すために確立する設定によって異なります。

ユーザ ID を渡す方法は、以下のセクションで詳しく説明されています。

ユーザ ID を渡すための Cookie 方式

SiteMinder はオープン形式の Cookie を使用してユーザ ID を渡すことができます。 Cookie には値の 1 つとしてユーザ ログイン ID が含まれます。

WAM システムまたは SiteMinder で認証を開始できます。 SiteMinder で認証を開始する場合、ユーザは WAM システムにリダイレクトされます。 認証プロセスは、WAM システムで開始された場合と同じです。

分散代行認証プロセスは以下のとおりです。

1. 認証リクエストはサードパーティ WAM システムに送られます。
2. ユーザが認証されます。

3. サードパーティ WAM システムは、以下の 2 つの方法のいずれかで Cookie を取得します。

   - WAM システムは CA SiteMinder® Federation SDK を使用して、オープン形式の Cookie を作成します。 SDK は Cookie を作成して、リクエストで WAM システムに送り返します。

     注: FIPS 暗号化されたオープン形式の Cookie を作成するには、CA SiteMinder® Federation SDK を使用します。

     サードパーティ WAM アプリケーションは、Cookie の作成に使用している SDK と同じ言語を使用します。 CA SiteMinder® Federation Java SDK を使用している場合、サードパーティ WAM アプリケーションは Java 内にある必要があります。 .NET SDK を使用している場合、サードパーティ WAM アプリケーションは .NET をサポートしている必要があります。

   - WAM システムは手動で作成されたオープン形式の Cookie を使用します。

     CA SiteMinder® Federation SDK を使用せずに、オープン形式の Cookie を作成できます。 Cookie を手動で作成するには、UTF-8 エンコーディングをサポートしているプログラミング言語を使用します。 SiteMinder によってパスワード ベースの暗号化がサポートされている以下の PBE 暗号化アルゴリズムのいずれかを使用できます。

     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-128
     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-192
     - PBE/SHA1/AES/CBC/PKCS12PBE-1000-256
     - PBE/SHA256/AES/CBC/PKCS12PBE-1000-128
     - PBE/SHA256/AES/CBC/PKCS12PBE-1000-192
     - PBE/SHA256/AES/CBC/PKCS12PBE-1000-256
     - PBE/SHA1/3DES_EDE/CBC/PKCS12PBE-1000-3
     - PBE/SHA256/3DES_EDE/CBC/PKCS12PBE-1000-3

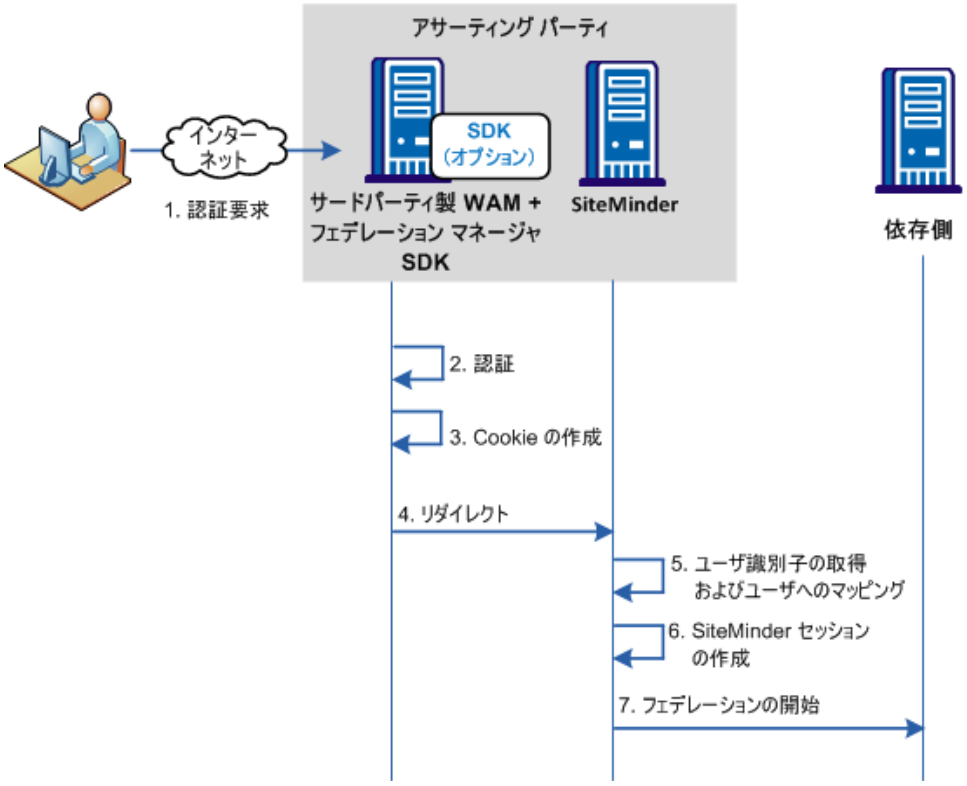
     オープン形式の Cookie がブラウザで設定されていることを確認します。

     完全な Cookie を書き込むには、オープン形式の Cookie のコンテンツについての詳細を確認してください。

   注: WAM システムおよび SiteMinder は同じ Cookie ドメイン内にある必要があります。

4. WAM システムはブラウザを SiteMinder にリダイレクトします。
5. SiteMinder は Cookie からログイン ID を抽出して、ユーザ ディレクトリでユーザを特定します。
6. SiteMinder は SiteMinder セッションを作成します。
7. セッションの作成後、依存パーティとのフェデレーション通信が行われます。

以下の図では、認証がサードパーティ WAM で開始された場合の Cookie 方式を示します。 SiteMinder は WAM アプリケーションを保護していません。



重要: SDK によって作成されたオープン形式の Cookie を使用するには、サードパーティは CA SiteMinder® Federation SDK をインストールする必要があります。SDK は SiteMinder から別々にインストールされたコンポーネントです。 インストール キットには、分散代行認証用に SDK を使用する方法を説明したドキュメントが含まれます。

ユーザ ID を渡すためのクエリ文字列方式

サードパーティ WAM システムはリダイレクト URL 上にクエリ文字列を追加することによって、ユーザ ID を SiteMinder に渡すことができます。この方式を使用するには、サードパーティ WAM システムは、認証後にフェデレーション ユーザを SiteMinder にリダイレクトする URL を設定する必要があります。

重要: 実稼働環境でクエリ文字列方式を使用しないでください。クエリ文字列リダイレクト方式は、概念実証としてテスト環境でのみ使用します。

WAM システムで認証を開始する場合、クエリ文字列を使用する分散代行認証プロセスは以下のとおりです。

注: SiteMinder または依存パーティで認証を開始することもできます。

1. サードパーティ WAM システムが認証リクエストを受信します。
2. ユーザが認証されます。
3. サードパーティ WAM システムはリダイレクト URL を構成して、ログイン ID およびハッシュされたログイン ID の値を LoginID=*LoginID*&LoginIDHash=*hashed_LoginID* 形式でクエリ文字列に追加します。

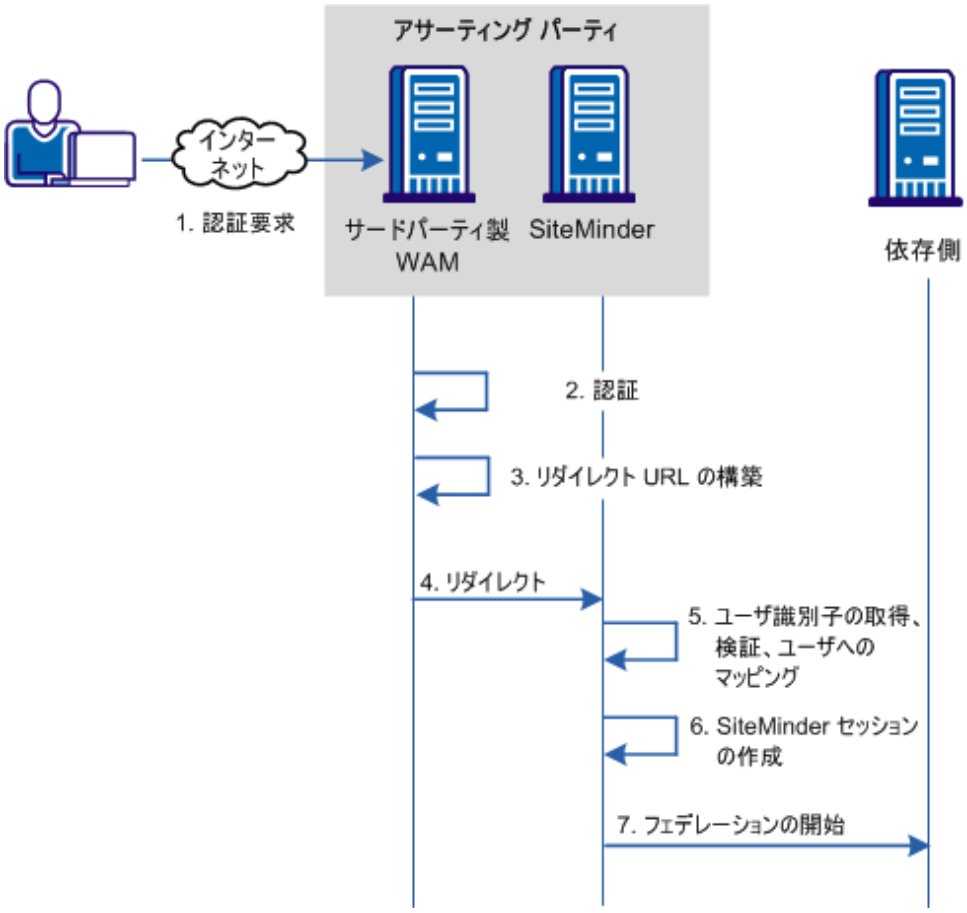
   重要: LoginID および LoginIDHash のパラメータでは大文字と小文字が区別されます。 必ずそれらを例で示されたとおりにリダイレクト URL に含めてください。

   ハッシュ メカニズムによって、SiteMinder はユーザ ID が変更されずに受信されたことを確認できます。

   リダイレクト URL の例

   http://idp1.example.com:9090/affwebservices/public/saml2sso?SPID=FmSP&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST&LoginID=jdoe&LoginIDHash=454d3bd5cb839168eeffcf060ae0b9c28ed6eec0
4. WAM システムはブラウザを SiteMinder にリダイレクトします。
5. SiteMinder は URL からログイン ID およびハッシュされたログイン ID を抽出して、ハッシュされた値の使用によって識別子を検証し、ユーザ ディレクトリでユーザを特定します。
6. SiteMinder はユーザ セッションを作成します。
7. セッションの作成後、依存パーティとのフェデレーション通信が行われます。

以下の図では、認証がアサーティング パーティで開始された場合のクエリ文字列方式を示します。



分散代行認証設定

分散代行認証は、認証されたユーザ ID に基づいてアサーションが生成されるアサーティング パーティで設定されます。

分散代行認証を設定する方法

1. サードパーティ WAM がユーザ ID を渡すために使用する方式（Cookie またはクエリ文字列）を決定します。

   注: このクエリ文字列では FIPS 準拠のパートナーシップは作成されません。

2. パートナーシップ ウィザードの該当する手順に進み、分散代行認証をセットアップします。

重要: SDK によって作成されたオープン形式の Cookie を使用するには、サードパーティは CA SiteMinder® Federation SDK をインストールする必要があります。 SDK は別々にインストールされたコンポーネントです。 インストール キットには、委任認証用に SDK を使用する方法を説明したドキュメントが含まれます。

Cookie 委任認証のサンプル セットアップ

以下は SAML 2.0 IdP から SP へのパートナーシップの観点から見たサンプル設定です。委任認証の設定は、パートナーシップ ウィザードの［SSO と SLO］の手順で行います。

このサンプル設定は SAML 2.0 設定を反映します。 アイデンティティ プロバイダは http://idp1.xyz.com、およびサードパーティ WAM システムは http://wamservice.xyz.com です。

Cookie 委任認証を設定する方法

1. パートナーシップを作成するか、または既存のパートナーシップを編集します。

   注: パートナーシップを非アクティブ化してから編集します。

2. パートナーシップ ウィザードの［SSO と SLO］手順に移動します。

3. ［認証］セクションで、以下のようにフィールドを設定します。

   認証モード

   委任

   委任認証タイプ

   オープン形式の Cookie

   Web アクセス管理アプリケーションと併用する場合。 CA SiteMinder® Federation SDK を使用して Java または .NET のアプリケーションを作成できます。または、手動でオープン形式の Cookie を作成すれば、別の言語で書き込まれたアプリケーションを使用できます。

   FIPS 140-2 暗号化が必要な場合は、CA SiteMinder® Federation Java または .NET SDK を使用してオープン形式の Cookie を作成します。

   委任認証 URL

   http://wamservice.xyz.com

   ユーザを認証し、CA SiteMinder® Federation SDK を使用して Cookie を作成するサードパーティ WAM システムの URL。

   認証クラス

   サードパーティで使用される認証方法を入力します。 例：

   urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos

4. すべてのオープン形式の Cookie 設定をサードパーティ WAM システムに伝達します。

   SiteMinder は Cookie の作成時にこれらの値を使用します。

5. パートナーシップ設定を続行します。

クエリ文字列の委任認証のセットアップ例

以下は SAML 2.0 IdP から SP へのパートナーシップの観点から見たサンプル設定です。委任認証の設定は、パートナーシップ ウィザードの［SSO と SLO］の手順で行います。

注: クエリ文字列方式は FIPS 準拠のパートナーシップを作成しません。

このサンプル設定は SAML 2.0 設定を反映します。 アイデンティティ プロバイダは http://idp1.xyz.com、およびサードパーティ WAM システムは http://wamservice.xyz.com です。

重要: 実稼働環境でクエリ文字列方式を使用しないでください。クエリ文字列リダイレクト方式は、概念実証としてテスト環境でのみ使用します。

クエリ文字列委任認証を設定する方法

1. パートナーシップを作成するか、または既存のパートナーシップを編集します。

   注: パートナーシップを非アクティブ化してから編集します。

2. パートナーシップ ウィザードの該当する手順に移動します。
3. ［認証］セクションで、以下のようにフィールドを設定します。

   認証モード

   委任

   委任認証タイプ

   クエリ文字列

   委任認証 URL

   http://wamservice.xyz.com

   ユーザを認証し、クエリ パラメータで SiteMinder に戻すリダイレクト URL を構成するサードパーティ WAM システムの URL。

   ハッシュ秘密キー

   FederatedAuth1

   サードパーティ WAM システムはこの秘密キーを使用して、ログイン ID をハッシュします。

   ハッシュ秘密キーの確認

   FederatedAuth1

   認証クラス

   サードパーティで使用される認証方法を入力します。 例：

   urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos

4. パートナーシップ設定を続行します。

Cookie 分散代行認証用のサードパーティ WAM 設定

分散代行認証を成功させるには、サードパーティ WAM ではそのフェデレーション アプリケーションを以下のように調整する必要があります。

- 認証されたユーザ ログイン ID を Cookie によって伝達するために、サードパーティ WAM システムは Cookie を生成する必要があります。
  - Java アプリケーションの場合、WAM では CA SiteMinder® Federation Java SDK を使用して、レガシー Cookie またはオープン形式の Cookie を作成できます。
  - .NET アプリケーションの場合、WAM では CA SiteMinder® Federation .NET SDK を使用して、オープン形式の Cookie を作成できます。
  - Java および .NET 以外の言語の場合、WAM ではオープン形式の Cookie を手動で作成できます。

  必要なクラスおよび方法の実装についての詳細は、「CA SiteMinder® Federation Java SDK ガイド」または「CA SiteMinder® Federation .NET SDK ガイド」を参照してください。各ガイドは SDK と共にインストールされます。オープン形式の Cookie を手動で作成する場合は、Cookie の必要なコンテンツに関する詳細を確認してください。
- サードパーティは、SiteMinder アサーティング パーティで設定されている以下の 管理 UI 設定の値を認識している必要があります。
  - 暗号化パスワード
  - オープン形式の Cookie 名
  - オープン形式の Cookie 暗号化変換

  Cookie の作成時に、SiteMinder はこれらの値を使用します。 これらの設定は、パートナーシップ ウィザードの［シングル サインオン］（SAML 1.x）および［SSO と SLO］（SAML 2.0）の手順にあります。

- サードパーティ WAM システムは、ユーザを SiteMinder に送り返すリダイレクト URL を作成する必要があります。 この URL はユーザを SiteMinder シングル サインオン サービスに送り返す必要があります。SiteMinder 管理者は帯域外通信でこの URL についてサードパーティに伝える必要があります。

  重要: サードパーティ WAM システムは SiteMinder から認証リクエストを受信した後、既存のクエリ文字列をキャプチャして再送信する必要があります。 受信リクエストには、クエリ文字列内に SiteMinder リクエスト情報が含まれる場合があり、リクエストを変更せずに渡す必要があります。

  注: Cookie を渡すためには、サードパーティ WAM システムがアサーティング パーティの SiteMinder と同じ Cookie ドメイン内にある必要があります。

クエリ文字列分散代行認証用のサードパーティ WAM 設定

アサーティング パーティのサードパーティ WAM システムおよび SiteMinder は、クエリ文字列内のログイン ID を伝達します。 WAM システムは、以下の 2 つの属性をリダイレクト URL 内のクエリ文字列に追加する必要があります。

LoginID

サードパーティ WAM システムに対してユーザを識別する値を指定します。

重要: LoginID パラメータは大文字と小文字を区別します。

LoginIDHash

LoginID のハッシュ。

LoginIDHash 値を生成するために、LoginID はハッシュ秘密キーの先頭に付けられて、値全体が SHA-1 ハッシュ アルゴリズムを使用して実行されます。ハッシュ秘密キーはアサーティング パーティの SiteMinder 設定で指定されます。

SiteMinder はクエリ文字列から認証情報を取得すると、これらの値を組み合わせてハッシュします。 ハッシュが等しい場合、SiteMinder はログイン ID が有効であるとみなし、フェデレーション リクエストを続行します。

重要: LoginIDHash パラメータは大文字と小文字を区別します。

サードパーティ WAM システムは、ユーザを SiteMinder シングル サインオン サービスに送り返すリダイレクト URL を構成するためにフェデレーション アプリケーションを設定する必要があります。 そのため、SiteMinder 管理者は帯域外通信でシングル サインオン サービスをサードパーティに伝達する必要があります。

重要: サードパーティ WAM システムは SiteMinder から認証リクエストを受信した後、既存のクエリ文字列をキャプチャして再送信します。 受信リクエストがクエリ文字列内に SiteMinder リクエスト情報を含んでいる場合、WAM システムはそれを変更せずに渡す必要があります。

クエリ文字列の構文は以下のとおりです。

?*existing_query_string*&LoginID=*LoginID*&LoginIDHash=*hashed_LoginID*

例

```
https://johndoe3227.b.com/affwebservices/public/saml2sso?SPID=sp1&
LoginID=user1&LoginIDHash=de164152ed6e8e9a7f760e47d135ecf0c98a
3e4e&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

第 15 章: シングル サインオンを開始する URL

シングル サインオンを開始するサーブレットへのリンク

フェデレーション コンテンツ用サイトを設計する場合、そのサイトにはシングル サインオンをトリガする特定のリンクを持つページが含まれます。 これらのリンクは、シングル サインオン サービスまたは認証リクエスト サービス用のサーブレットへの URL です。

シングル サインオンを開始するために、ユーザはアサーティング パーティまたは依存パーティから始めることができます。 各サイトでシングル サインオン操作を開始するための適切なリンクを設定します。

プロデューサによって開始される SSO（SAML 1.1）

プロデューサで、ユーザをコンシューマ サイトに導くリンクが含まれるページを作成します。 それぞれのリンクは、サイト間転送 URL を表します。 ユーザはサイト間の転送 URL にアクセスする必要があります。 ユーザがコンシューマ サイトにリダイレクトされる前に、URL によってプロデューサ側 Web エージェントへの要求が行われます。

SAML Artifact および POST プロファイルの場合、サイト間の転送 URL の構文は以下のとおりです。

```
http://producer_host:port/affwebservices/public/intersitetransfer?
CONSUMERID=consumer_entity_ID&TARGET=http://consumer_site/target_url
```

このサイト間の転送 URL の変数とクエリ パラメータは以下のとおりです。

producer_host:port

ユーザが認証されるサーバおよびポート番号を指定します。

CONSUMERID

（必須）コンシューマを識別します。 プロデューサ側で、プロデューサからコンシューマへのパートナーシップには名前があり、リモート コンシューマ エンティティには ID があります。 CONSUMERID はリモート コンシューマのエンティティ ID です。 エンティティ ID は大文字と小文字を区別します。管理 UI に表示される通りに正確に入力してください。

CONSUMERID の代わりにパラメータ NAME を使用できますが、両方を使用することはできません。

NAME を使用する場合は、プロデューサで定義されているプロデューサからコンシューマへのパートナーシップの名前を指定します。

consumer_entity_ID

ユーザがプロデューサ サイトからアクセスする必要があるコンシューマ サイトを識別します。 エンティティ ID は大文字と小文字を区別します。 管理 UI に表示される通りに正確に入力してください。

TARGET

（オプション）コンシューマで要求されたターゲット リソースを識別します。

TARGET パラメータはオプションです。 ターゲットを定義する必要がありますが、コンシューマ側のパートナーシップでサイト間の転送 URL の代わりに定義することができます。 ターゲットはパートナーシップ ウィザードの［アプリケーション統合］手順で定義します。 必ず URL またはパートナーシップでターゲットを定義してください。

consumer_site

コンシューマ サイトのサーバを指定します。

target_url

コンシューマ サイトのターゲット アプリケーションを示します。

注: SAML Artifact バインディング用のクエリ パラメータは HTTP エンコーディングを使用する必要があります。

Artifact および POST プロファイル用のサイト間の転送 URL の例は以下のとおりです。

```
http://www.smartway.com/affwebservices/public/intersitetransfer?
CONSUMERID=ahealthco&TARGET=http://www.ahealthco.com:85/
smartway/index.jsp
```

IdP によって開始される SSO（SAML 2.0 Artifact または POST）

ユーザがサービス プロバイダの前に SiteMinder アイデンティティ プロバイダにアクセスする場合、アイデンティティ プロバイダで未承認応答が開始される必要があります。 未承認応答を開始するには、SiteMinder が受理する HTTP Get リクエストを生成するハードコードされたリンクを作成します。 この HTTP Get リクエストには、サービス プロバイダ ID を提供するクエリ パラメータが含まれる必要があります。 アイデンティティ プロバイダは SAML アサーション レスポンスを生成する必要があります。ユーザはこのリンクをクリックして、未承認応答を開始します。

注: この情報は Artifact バインディングまたは POST バインディングに適用されます。

未承認応答で Artifact または POST プロファイルを使用するように指定するには、未承認応答リンクに以下の構文を使用します。

http://*idp_server:port*/affwebservices/public/saml2sso?SPID=*SP_ID*&ProtocolBinding=*URI_for_binding*&RelayState=*target_URL*

idp_server:port

SiteMinder をホストしている Web サーバおよびポートを識別します。

SP_ID

パートナーシップで定義されたサービス プロバイダのエンティティ ID を指定します。エンティティ ID は大文字と小文字を区別します。管理 UI に表示される通りに正確に入力してください。

URI_for_binding

ProtocolBinding 要素用の POST バインディングまたは Artifact バインディングの URI を識別します。 SAML 2.0 仕様によりこの URI が定義されます。

- SAML 2.0 仕様によって指定されている Artifact バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- SAML 2.0 仕様によって指定されている POST バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  このパラメータを HTTP-POST シングル サインオンに対して設定する必要はありません。

注: また、バインディングは、リクエストの実行のためにパートナーシップに対して有効である必要があります。

target_URL

サービス プロバイダのフェデレーション リソース ターゲットの URL を指定します。

以下の点に注意してください。

- リンクに ProtocolBinding クエリを含めない場合は、サービス プロバイダ プロパティで設定された 1 つのバインディングを使用します。
- Artifact および POST がサービス プロバイダ プロパティで有効な場合、POST がデフォルトです。 したがって、Artifact バインディングのみを使用する場合は、リンクに ProtocolBinding クエリ パラメータを含めます。

重要: アサーション コンシューマ サービスにインデックス付きエンドポイント サポートを設定する場合、ProtocolBinding クエリ パラメータの値はアサーション コンシューマ サービスのバインディングを上書きします。

IdP によって使用される未承認応答のクエリ パラメータ

IdP からシングル サインオンを開始する未承認応答には、以下のクエリ パラメータが含まれることがあります。

SPID

（必須）アイデンティティ プロバイダが未承認応答を送信するサービス プロバイダの ID を指定します。 エンティティ ID は大文字と小文字を区別します。 管理 UI に表示される通りに正確に入力してください。

ProtocolBinding

未承認応答内の ProtocolBinding 要素を指定します。 この要素は、アサーション レスポンスをサービス プロバイダに送信するためのプロトコルを指定します。 指定されたプロトコル バインドをサービス プロバイダがサポートするように設定されていない場合、リクエストは失敗します。

RelayState

サービス プロバイダのターゲット リソースの URL を示します。 このクエリ パラメータを含めることによって、IdP はサービス プロバイダの適切なリソースにユーザをリダイレクトします。 このクエリ パラメータは、シングル サインオンの設定時にターゲット URL を指定する代わりに使用できます。

ProtocolBinding クエリ パラメータの必須使用

Artifact バインディングおよび POST バインディングがサービス プロバイダ プロパティに対して有効な場合にのみ、ProtocolBinding クエリ パラメータは必要です。 さらに、ユーザは Artifact バインディングのみを使用する必要があります。

- Artifact バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- POST バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  このパラメータを HTTP-POST シングル サインオンに対して設定する必要はありません。

注: クエリ パラメータをコード化する HTTP は必要ありません。

ProtocolBinding クエリ パラメータの任意使用

ProtocolBinding クエリ パラメータを使用*しない* 場合、以下の情報が当てはまります。

- サービス プロバイダに対して有効なバインディングが 1 つのみで、ProtocolBinding が未承認応答で指定されていない場合、有効なバインディングが使用されます。
- 両方のバインディングがサービス プロバイダに対して有効で、ProtocolBinding が未承認応答で指定されていない場合、POST バインディングがデフォルトです。

例: ProtocolBinding のない未承認応答

リンクはユーザをシングル サインオン サービスにリダイレクトします。SPID クエリ パラメータによって指定されるサービス プロバイダ ID がこのリンクに含まれています。ProtocolBinding クエリ パラメータは存在しません。 ユーザはこのハードコードされたリンクをクリックした後、シングル サインオン サービスにリダイレクトされます。

http://fedsrv.fedsite.com:82/affwebservices/public/saml2sso?
SPID=http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90

例: ProtocolBinding が含まれる未承認応答

リンクはユーザをシングル サインオン サービスにリダイレクトします。SPID クエリ パラメータによって指定され、Artifact バインディングを使用しているサービス プロバイダ ID がこのリンクに含まれています。 ユーザはこのハードコードされたリンクをクリックした後、ローカル シングル サインオン サービスにリダイレクトされます。

http://idp-ca:82/affwebservices/public/saml2sso?SPID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&
ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

IdP での ForceAuthn および IsPassive 処理

サービス プロバイダがシングル サインオンを開始する場合、サービス プロバイダは ForceAuthn または IsPassive のクエリ パラメータを認証リクエスト メッセージに含めることができます。

サービス プロバイダが ForceAuthn または IsPassive を認証リクエストに含める場合、SiteMinder アイデンティティ プロバイダは以下のようにこれらのクエリ パラメータを処理します。

ForceAuthn の処理

サービス プロバイダが ForceAuthn=True を認証リクエスト メッセージに含める場合、SiteMinder アイデンティティ プロバイダはユーザに認証情報を要求します。 SiteMinder セッションが存在する場合にも要求します。

IsPassive の処理

SiteMinder IdP はパッシブ認証をサポートしません。 サービス プロバイダが IsPassive を認証リクエストに含み、アイデンティティ プロバイダがそれを受け取れない場合、IdP はこれらの SAML レスポンスのいずれかを送り返します。

- 認証リクエスト メッセージに IsPassive=True が含まれ、セッションがない場合、アイデンティティ プロバイダはエラー メッセージを返します。 SiteMinder にはセッションが必要です。
- 認証リクエスト メッセージに IsPassive=True が含まれ、セッションがある場合、アイデンティティ プロバイダはアサーションを返します。
- 認証リクエスト メッセージに IsPassive および ForceAuthn が含まれ、両方が true に設定されている場合、SiteMinder アイデンティティ プロバイダはエラーを返します。 IsPassive と ForceAuthn は相互に排他的です。

SP によって開始される SSO（SAML 2.0）

SP によって開始される SSO では、サービス プロバイダの HTML ページに、サービス プロバイダの認証リクエスト サービスへハードコードされたリンクが含まれている必要があります。 リンクはユーザをアイデンティティ プロバイダにリダイレクトし、アイデンティティ プロバイダは認証されて認証リクエスト自体に含まれているものを特定します。

この情報は Artifact バインディングまたは POST バインディングに適用されます。

ユーザが選択するハードコードされたリンクには、認証リクエスト サービスへの HTTP GET リクエストで使用される特定のクエリ パラメータが含まれている必要があります。

注: これらのハードコードされたリンクを持つページは、保護されていないレルムに存在する必要があります。

Artifact バインドまたはプロファイル バインドをトランザクションに使用するように指定するには、次のリンクの構文を使用します。

http://*sp_server:port*/affwebservices/public/saml2authnrequest?ProviderID=*IdP_ID*&ProtocolBinding=*URI_of_binding*&RelayState=*target_URL*

sp_server:port

CA SiteMinder® Federation をホストしているサービス プロバイダのサーバおよびポート番号を指定します。

IdP_ID

アイデンティティ プロバイダに割り当てられている ID を指定します。エンティティ ID は大文字と小文字を区別します。 管理 UI に表示される通りに正確に入力してください。

URI_of_binding

ProtocolBinding 要素用の POST バインディングまたは Artifact バインディングの URI を識別します。 SAML 2.0 仕様によりこの URI が定義されます。

- Artifact バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- POST バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  このパラメータを HTTP-POST シングル サインオンに対して設定する必要はありません。

また、リクエストの実行のためにパートナーシップに対してバインディングを有効にします。

target_URL

サービス プロバイダのフェデレーション ターゲットの URL を指定します。

以下の情報に注意してください。

- ProtocolBinding クエリ パラメータを認証リクエスト リンクに含めない場合、デフォルトのバインディングはパートナーシップに対して定義されたバインディングです。 両方のバインディングをパートナーシップで定義している場合、バインディングは認証リクエスト内に渡されません。その結果、アイデンティティ プロバイダのデフォルトのバインディングが使用されます。
- Artifact バインディングおよび POST バインディングがパートナーシップに対して有効だが、Artifact バインディングのみを使用する場合は、ProtocolBinding クエリ パラメータをリンクに含めます。

SP によって使用される認証リクエスト クエリ パラメータ

SiteMinder SP が認証リクエスト サービスへのリンクで使用できるクエリ パラメータは以下のとおりです。

ProviderID（**必須**）

認証リクエスト サービスが認証リクエスト メッセージを送信するアイデンティティ プロバイダのエンティティ ID。 エンティティ ID は大文字と小文字を区別します。管理 UI に表示される通りに正確に入力してください。

ProtocolBinding

認証リクエスト メッセージの ProtocolBinding 要素を指定します。この要素は、アイデンティティ プロバイダからの SAML レスポンスを返すためのプロトコルを指定します。指定したアイデンティティ プロバイダが指定したプロトコル バインディングをサポートするように設定されていない場合、リクエストは失敗します。

このパラメータを認証リクエストで使用する場合、AssertionConsumerServiceIndex パラメータを同時に含めることはできません。 これらは、相互に排他的です。

ForceAuthn

既存のセキュリティ コンテキストに依存せずにユーザを直接認証する必要があることをアイデンティティ プロバイダに示します。アイデンティティ プロバイダがサードパーティのフェデレーション ソフトウェアを使用せずに CA SiteMinder® Federation を使用している場合、このクエリ パラメータを使用します。

- SP が認証リクエスト メッセージで ForceAuthn=True を設定していて、セッションが特定のユーザに対して存在する場合、アイデンティティ プロバイダはユーザに認証を要求します。 ユーザが正常に認証された場合、IdP では、その識別情報を既存のセッションからアサーション内に送信します。 アイデンティティ プロバイダは再認証用に生成するセッションを破棄します。
- SP が認証リクエスト メッセージで ForceAuthn=True を設定していて、セッションがない場合、IdP はユーザに認証を要求します。ユーザが正常に認証されると、セッションが確立されます。

例

http://sp1.demo.com:81/affwebservices/public/saml2authnrequest?ProviderID=idp1.example.com&ForceAuthn=yes

IsPassive

ユーザのログイン時に認証情報をユーザに要求しないか、任意の方法でユーザと対話するようアイデンティティ プロバイダに指示します。ユーザにセッションがない限り、SiteMinder アイデンティティ プロバイダはこのクエリ パラメータを考慮しません。ユーザにセッションがない場合、アイデンティティ プロバイダからエラーが返されます。

AssertionConsumerServiceIndex

アサーション コンシューマ サービスとして機能するエンドポイントのインデックスを指定します。 インデックスによって、アイデンティティ プロバイダにアサーション レスポンスの送信先が指定されます。

認証リクエストでこのパラメータを使用する場合、ProtocolBinding パラメータは含めないでください。 このパラメータと ProtocolBinding パラメータは相互に排他的です。 アサーション コンシューマ サービスにはそれ自身のプロトコル バインディングがあり、ProtocolBinding パラメータと競合する可能性があります。

RelayState

サービス プロバイダのターゲット リソースの URL を示します。 このクエリ パラメータを含めることによって、サービス プロバイダにユーザの送信先を示します。 含めなければ、パートナーシップのデフォルトのターゲットが使用されます。

ProtocolBinding クエリ パラメータの必須使用

Artifact バインディングおよび POST バインディングがパートナーシップに対して有効で、ユーザが Artifact バインディングのみを使用する場合、ProtocolBinding パラメータが必要です。

- Artifact バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- POST バインディング用の URI は以下のとおりです。

  urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

  このパラメータを HTTP-POST シングル サインオンに対して設定する必要はありません。

ProtocolBinding の任意使用

ProtocolBinding クエリ パラメータを使用 *しない* 場合、以下の条件が当てはまります。

- パートナーシップに対して有効なバインディングが 1 つのみで、ProtocolBinding クエリ パラメータが指定されていない場合、パートナーシップに対して有効なバインディングが使用されます。
- 両方のバインディングが有効で、ProtocolBinding クエリ パラメータが指定されていない場合、デフォルトとして POST バインディングが使用されます。

注: クエリ パラメータを HTTP エンコードする必要はありません。

例：ProtocolBinding クエリ パラメータのない認証リクエスト リンク

このサンプル リンクは、認証リクエスト サービスに移動します。 このリンクは、ProviderID クエリ パラメータのアイデンティティ プロバイダを指定します。

http://ca.sp.com:90/affwebservices/public/saml2authnrequest?
ProviderID=http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90

ユーザがサービス プロバイダのリンクをクリックすると、SiteMinder は認証リクエスト メッセージのリクエストを渡します。

例：ProtocolBinding クエリ パラメータを含む認証リクエスト リンク

http://ca.sp.com:90/affwebservices/public/saml2authnrequest?
ProviderID=http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&
ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

ユーザがサービス プロバイダのリンクをクリックすると、SiteMinder は認証リクエスト メッセージのリクエストを渡します。

IP で開始するシングル サインオン（WSFED）

ユーザは、リソース パートナー（RP）に移動する前にアイデンティティ プロバイダ（IP）にアクセスできます。 ユーザが先にアイデンティティ プロバイダにアクセスする場合は、リンクで HTTP Get リクエストが生成される必要があります。 ハードコードされたリンクは、IP のパッシブ リクエスタ サービスを指しています。 リクエストには、RP Provider ID および必要に応じて他のパラメータが含まれます。

このリンクの構文は、次のとおりです。

https://*ip_server:port*/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=*rp_id*

ip_server:port

ID パートナーでシステムのサーバおよびポート番号を指定します。システムは、フェデレーション ネットワークにどのコンポーネントがインストールされているかに応じて、Web エージェント オプション パックまたは SPS フェデレーション ゲートウェイをホストしています。

rp_id

RP の ID。エンティティ ID は大文字と小文字を区別します。管理 UI に表示される通りに正確に入力してください。

RP で開始するシングル サインオン（WSFED）

ユーザが RP でシングル サインオンを開始する場合、通常は IP のリストから選択します。 サイト選択ページは、保護されていないレルムにあります。

サイト選択ページのリンクは、IP のパッシブ リクエスタ サービスを指しています。リンクを選択した後、RP はアサーションを取得するためにユーザを IP にリダイレクトします。

第 16 章: ユーザ セッションのログアウト

シングル ログアウトの概要（SAML 2.0）

シングル ログアウト（SLO）により、ログアウトを開始したブラウザのすべてのユーザ セッションが同時に終了します。 すべてのユーザ セッションを閉じることにより、権限のないユーザが SP のリソースにアクセスできないようにします。

シングル ログアウトにより、特定ユーザに関するすべてのセッションが必ずしも終了するとは限りません。たとえば、ブラウザを 2 つ開いているユーザは、独立した 2 つのセッションを持つことができますが、ログアウトを開始したブラウザのセッションだけが、そのセッションのすべてのフェデレーション サイトで終了します。 もう一方のブラウザのセッションは、引き続きアクティブです。

シングル ログアウト バインディングによって、シングル ログアウト メッセージと共に送信されるもの、および受信した各メッセージを処理する方法が決まります。

重要: シングル ログアウトを設定するには、ポリシー サーバ管理コンソールを使用してセッション ストアを有効にします。 管理コンソール使用の詳細については、「*ポリシー サーバ管理ガイド*」の手順を参照してください。

以下の 2 つのバインディングをシングル ログアウト操作で使用できます。

HTTP リダイレクト

HTTP リダイレクト バインディングでは、ブラウザを使用して各ログアウト トランザクションを実行します。 シングル ログアウト メッセージは常に GET リクエストです。ブラウザはすべてのリクエストおよびレスポンスに関与します。 ブラウザの関与は、HTTP リダイレクト バインディングによってブラウザ セッション データが提供されることを意味します（SOAP バインディングでは提供されません）。

HTTP リダイレクト バインディングのデメリットは、メッセージ内のデータがクエリ文字列で送信できるものに制限されることです。 また、HTTP リダイレクト バインディングは非同期処理なので、タイムアウトはほとんど発生しません。 ただし、リダイレクトが失敗すると、全シングル ログアウトが停止します。

SOAP

SOAP バインディングでは、POST リクエストを使用してシングル ログアウト トランザクションを実行します。 POST リクエストによって HTTP リダイレクト バインディングより多くのデータを送信できます。SOAP によって、暗号化の方法および他の機能でより多くのことを実行することもできます。

SOAP は同期処理です。 IdP はより制御性があり、1 つの SP での問題がプロセス全体に干渉することを防ぐことができます。 SOAP 通信はバック チャネルで行われます。1 つのログアウト失敗によって、IdP が残りの SP からログアウトすることを中断されることはありません。

SOAP はバック チャネル接続を使用しますので、最初のシングル ログアウト コールおよびレスポンスの後、ブラウザは関与しません。SOAP バインディングでは、ログアウト プロセスの一部としてリモート エンティティの Cookie をクリーンアップしません。 Cookie はローカル エンティティでのみクリーンアップされます。 Cookie の削除が必要な場合は、HTTP リダイレクト バインディングを使用します。

HTTP リダイレクトおよび SOAP を使用してネットワーク全体のシングル ログアウトを管理する

ネットワークには、HTTP リダイレクト バインディングをサポートするサイトおよび SOAP バインディングをサポートするサイトがある場合があります。 IdP は複数のバインディングを管理する必要がありますが、SP は 1 つのログアウト リクエストのみを送信または受信します。

以下のセクションでは、バインディングが混在する環境に対応するための設定ガイドラインについて説明します。

SiteMinder が IdP にある場合の SLO 設定

SiteMinder が IdP にある場合、HTTP リダイレクト ベースの SLO サービス URL および SOAP ベースの SLO サービス URL を含めるようにパートナーシップを設定します。

IdP の SiteMinder は、セッションの各 SP の設定を確認して、SOAP が有効なすべてのログアウトを最初に処理します。 その後 SOAP をサポートしない SP の HTTP リダイレクト ログアウトが続きます。

SiteMinder が SP にある場合の SLO 設定

SiteMinder が SP にあり、SP がシングル ログアウトを開始する場合は、HTTP リダイレクト バインディングでログアウトを開始することをお勧めします。ユーザ セッションの他の SP は SOAP をサポートしない可能性があります。

HTTP リダイレクトは、ブラウザ セッションを使用してすべてのリダイレクトを処理します。 このため、HTTP リダイレクトは、IdP が HTTP リダイレクトのみをサポートする SP をログアウトするために必要なデータを送信します。 SP が HTTP リダイレクトでプロセスを開始する場合、IdP はそれをサポートするすべての SP と共に SOAP を使用できます。残りの SP については HTTP リダイレクト バインディングに切り替えます。

SOAP バインディングでシングル ログアウトを開始する場合、ブラウザ セッション データは存在しません。

SP によって開始されるログアウトで確実に HTTP リダイレクトが使用されるように、SP のローカル サーブレットを指す HTTP リダイレクト リンクをページまたはアプリケーションに埋め込みます。 SiteMinder 用のリンクは以下のとおりです。

http://*sp_host:port*/affwebservices/public/saml2slo

この埋め込みリンクによって、SiteMinder は IdP の SLO サービスに送る SAML <LogoutRequest> メッセージを生成します。 ユーザがログアウトするときは、まず SP のログアウトが実行され、次にログアウト リクエストが IdP に送信されます。 その後、IdP は、ユーザ セッションに関連する他のすべての SP と共にログアウト プロセスを完了します。

SLO リクエスト有効期間に関するスキュー時間の概要

ログアウト リクエストの有効な期間の計算には、2 つの値が関連します。これらの値は IssueInstant 値および NotOnOrAfter 値です。 SLO レスポンスでは、NotOnOrAfter 値になるまでシングル ログアウト リクエストが有効です。 シングル ログアウト リクエストの生成時に、SiteMinder はシステム時間を取得します。 この時間がリクエスト メッセージの IssueInstant 設定になります。 ログアウト リクエストが期限切れになるときを特定するために、SiteMinder は現在のシステム時間を取得し、［スキュー時間］および［SLO 有効期間］を加えます。その結果の時間が NotOnOrAfter 値になります。

注: 時刻は GMT を基準にしています。

たとえば、ログアウト リクエストが 1:00 （GMT）にアサーティング パーティで生成されます。 スキュー時間は 30 秒、SLO 有効期間は 60 秒です。したがって、そのリクエストは 1:00 GMT から 1:01:30 GMT までの間が有効です。 IssueInstant 値は 1:00 GMT です。シングル ログアウト リクエスト メッセージはその後 90 秒で無効になります。

シングル ログアウトの設定

シングル ログアウトを設定するには、ポリシー サーバ管理コンソールを使用して、セッション ストアを有効にする必要があります。管理コンソール使用の詳細については、「*ポリシー サーバ管理ガイド*」の手順を参照してください。 セッション ストアが有効にされていない場合、管理 UI でシングル ログアウト設定を確認できません。

シングル ログアウトの設定時には、以下の情報に注意してください。

- パートナーが HTTP リダイレクトを使用して SAML <LogoutRequest> メッセージを受信する場合、送信元に返すレスポンスでは HTTP リダイレクト バインディングを使用する必要があります。
- パートナーが SOAP を使用して SAML <LogoutRequest> メッセージを受信する場合、送信元に返すレスポンスは SOAP を経由する必要があります。
- パートナーがサポートしていないバインディングによって SLO リクエストを受信すると、シングル ログアウトは失敗します。
- シングル ログアウト ユーザ セッションに HTTP リダイレクト バインディングおよび SOAP バインディングを使用するパートナーが含まれる場合は、両方のバインディングをサポートするように SiteMinder を設定します。 IdP がログアウトを続行すると、SOAP を使用するすべての SP からログアウトしてから、HTTP リダイレクト バインディングを使用するすべての SP からログアウトします。
- SiteMinder SP がシングル ログアウトを開始する場合は、SP が SOAP をサポートしていても、HTTP リダイレクト バインディングを使用して開始します。

  SOAP および HTTP リダイレクトがサポートされている混在環境でのシングル ログアウトの管理 (P. 251)に関するガイドラインを確認してください。

次の手順に従ってください:

注: SLO 環境設定は IdP と SP で同じです。

1. パートナーシップ ウィザードの［SSO と SLO］手順から始めます。
2. ［SLO］セクションで、1 つまたは両方の SLO バインディングを選択します。

   SLO バインディングによってシングル ログアウトが有効になり、ローカル エンティティで使用しているバインディングが示されます。さらに SLO バインディングによって、ローカル エンティティがシングル ログアウト リクエストを受信するときに使用するバインディングが示されます。

   SOAP を選択する場合、SOAP メッセージ内の名前 ID を暗号化できます。暗号化オプションはパートナーシップ ウィザードの［署名および暗号化］手順で設定されます。

   バインディングとして［SOAP］を選択する場合、［バック チャネル］の［受信および送信設定］はアクティブになります。 SLO リクエストおよびレスポンスはバック チャネルを介して送信されます。各ローカル パートナーは、リモート パートナーによる認証を要求することによってバック チャネルを保護できます。

   SLO のバック チャネル設定 (P. 255)に関する詳細を確認できます。
3. 他の SLO 設定のいずれかを設定します。
   - SLO 確認 URL
   - SLO 有効期間（秒）
   - リレー状態を使用して SLO 確認 URL をオーバーライドする

   フィールドの説明については、［ヘルプ］をクリックしてください。

4. ［SLO サービス URL］のテーブルに入力します。 1 つ以上のエンティティを入力する必要があります。選択されたリモート エンティティに定義された値は、すでにテーブルに入力されています。

   SLO サービス URL はシングル ログアウトを開始し、その後、ポリシー サーバをトリガして SAML <LogoutRequest> メッセージを生成します。さらに、SLO サービス URL は、ログアウト リクエスト メッセージの送信先をポリシー サーバに指示します。

   以下のように、サポートされている各 SLO バインディングに SLO サービス URL を指定します。

   - HTTP-Redirect 対応 - HTTP-Redirect をバインディングとして 1 つの URL を選択します。
   - SOAP 対応 - SOAP をバインディングとして 1 つの URL を選択します。
   - リダイレクトおよび SOAP 対応 - 1 つは HTTP リダイレクト、もう 1 つは SOAP に設定して 2 つの URL を選択します。

   注: ［レスポンス ロケーション URL］フィールドはオプションです。

シングル ログアウト設定が完了しました。

シングル ログアウト用バック チャネル設定

SOAP バインディングを使用するシングル ログアウトでは、ログアウト リクエストおよびレスポンスがバック チャネルを介して送信されます。バック チャネルへのアクセスを認証するためにエンティティを要求できます。 必須ではありませんが、SSL を使用してバック チャネルを保護することができます。

SSL を使用してバック チャネルを保護するには、以下を実行する必要があります。

- SSL を有効にする。

  SSL は基本認証には必要ありませんが、SSL を介して基本認証を使用できます。 SSL はクライアント証明書認証に必要です。

- シングル ログアウト通信交換用に受信および送信バック チャネルを設定します。ローカル エンティティは、送信チャネルでメッセージを送信し、受信チャネルでメッセージを受信できる必要があります。

  注: 1 つの受信および送信バック チャネルを設定できますが、チャネルに設定できるのは 1 つの設定のみです。 2 つのサービスが同じチャネルを使用する場合、これらの 2 つのサービスは同じバック チャネル設定を使用します。たとえば、ローカルのアサーティング パーティの受信チャネルが HTTP-Artifact SSO と SLO over SOAP をサポートする場合、これらの 2 つのサービスは同じバック チャネル設定を使用する必要があります。

- 保護されているバック チャネルを介してアクセスできるようにリモート エンティティ用の認証タイプを選択する。認証方法はチャネルごと（受信または送信）に適用されます。

  バック チャネル認証のオプションは以下のとおりです。

  基本

  基本認証方式がバック チャネルを保護していることを示します。

  注: SSL がバック チャネル接続に対して有効な場合も、基本認証を選択できます。

  クライアント証明書

  X.509 クライアント証明書を含む SSL がアサーティング パーティ バック チャネルを保護することを示します。

  認証方法として［クライアント証明書］を選択する場合、すべてのエンドポイント URL が SSL 通信を使用する必要があります。 これは、URL が **https://** で始まる必要があることを意味します。エンドポイント URL によって、サーバ上でシングル サインオン、シングル ログアウトおよびアサーション コンシューマ サービスなどさまざまな SAML サービスが特定されます。

認証なし

依存パーティが認証情報を提供する必要がないことを示します。バック チャネルは保護されません。 このオプションでも SSL を有効にできます。バック チャネル トラフィックは暗号化されますが、認証情報は保証機関と依存パーティの間で交換されません。

認証なし（NoAuth）オプションは、実稼働用ではなくテスト用にのみ使用します。SiteMinder が SSL 対応のフェールオーバを実装するプロキシ サーバの背後にある場合は例外です。 クライアント証明書認証がバック チャネルの保護に使用される場合、プロキシ サーバは認証を処理します。 IdP から SP へのすべてのパートナーシップは認証タイプとして［NoAuth］を使用できます。

重要: 受信バック チャネル用の認証方法は、パートナーシップの反対側の送信バック チャネルに一致する必要があります。認証方法の選択の一致は帯域外通信で処理されます。

シングル ログアウト用バック チャネルを保護する方法

1. パートナーシップ ウィザードの［SSO と SLO］手順の［バック チャネル］セクションから始めます。
2. ［SLO］セクションで［SOAP］を選択します。 ［認証方法］フィールドはアクティブになります。
3. 受信および送信バック チャネルに対して認証方法のタイプを選択します。 その他の設定するフィールドが、基本およびクライアント証明書方式用に表示されます。

   認証方法として［認証なし］を選択する場合、これ以上の手順は必要ありません。
4. 選択する認証方法に応じて、設定するフィールドがさらに表示されます。

すべての必要なフィールドに値を入力したら、バック チャネル設定は完了です。

サインアウトの概要（WS-フェデレーション）

サインアウトでは、サインアウトを開始したブラウザのすべてのユーザ セッションが同時に終了します。 すべてのユーザ セッションを閉じることにより、権限のないユーザがリソース パートナーのリソースにアクセスできないようにします。

サインアウトにより、特定ユーザに関するすべてのセッションが必ずしも終了するとは限りません。たとえば、ブラウザを 2 つ開いているユーザは、独立した 2 つのセッションを持つことができますが、サインアウトを開始したブラウザのセッションのみが、そのセッションに関するすべての連携したサイトで終了します。 もう一方のブラウザのセッションは、引き続きアクティブです。

ポリシー サーバは、signoutconfirmurl.jsp を使用してサインアウトを実行します。このページは、アイデンティティ プロバイダ システムにあります。アイデンティティ プロバイダ パートナーは、ユーザに代わってサインアウト リクエストを開始します。JSP は、指定されたブラウザ セッション中にユーザがサインオンした各サイトに、サインアウト リクエストを送信します。 その後、ユーザはサインアウトされます。

ユーザは、アイデンティティ プロバイダでのみサインアウト リクエストを開始できます。 リクエストは、該当するサーブレットを指すリンクをクリックすることによってトリガされます。 サインアウトの確認ページは、アイデンティティ プロバイダ サイト上の、保護されていないリソースである必要があります。

注: ポリシー サーバは、サインアウトに関して WS-フェデレーション パッシブ リクエスタ プロファイルのみをサポートします。

WSFED サインアウトの有効化

サインアウトを設定するための要件

- アイデンティティ プロバイダでサインアウトを有効にするには、ポリシー サーバ管理コンソールを使用してセッション ストアを有効にします。

  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。

- サインアウトには、有効な SiteMinder 永続セッションが必要です。 これは、シングル サインオン中に確立されます。 リソース パートナーで、認証 URL などの保護されたリソースを持ったレルムで永続セッションを設定します。

  レルムについては、「ポリシー サーバ設定ガイド」を参照してください。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 変更する WS-Federation パートナーシップを選択します。
3. パートナーシップ ウィザードの［シングル サインオンおよびサインアウト］手順に移動します。
4. ［サインアウト］セクションで、以下のフィールドを設定します。
   - サインアウトの有効化
   - サインアウト確認 URL （IP のみ）
   - サインアウト URL

   各 URL に、https:// または http:// で始まるエントリが入力されていることが必要です。
5. ［確認］手順に移動して［完了］をクリックし、変更を保存します。

サインアウトが設定されます。

SP でのローカル ログアウト（SAML 2.0）

SP としての SiteMinder は、スタンドアロン アプリケーションのローカル ログアウトをサポートします。 ローカル ログアウトによって、ユーザをローカル SP 側のアプリケーションでログアウトできるようになります。 SP のセッションは削除されますが、IdP または他の SP との通信には影響しません。 IdP および他の SP のセッションはアクティブなままです。

SP のアプリケーションにログアウト リンクを含める場合、SP はログアウト リクエストをローカルのシングル ログアウト サービスに送信します。 SP は、リクエストを受信するとユーザをログアウトします。 SP のアプリケーションはログアウト成功の確認メッセージを送信します。

SiteMinder では、**localLogout** という名前のクエリ パラメータを使用してローカル ログアウトを実行できます。このパラメータを使用するために、アプリケーションには、以下の例のようなページがある可能性があります。

```
demoapp への登録を完了しました。
セッションを安全に終了するには、［LOGOUT］を選択します。
```

以下のサンプル文字列は、［LOGOUT］ボタンのリンクを表します。

```
<http://sp1server.demo.com:8080/affwebservices/public/saml2slo?Loc
alLogout=true
```

第 17 章: 認証コンテキスト処理(SAML 2.0)

*認証*コンテキストは、アイデンティティ プロバイダでユーザが認証した方法を示します。 アイデンティティ プロバイダは、サービス プロバイダのリクエストで、またはアイデンティティ プロバイダの設定に基づいて、認証コンテキストをシングル サインオン アサーションに含めます。 サービス プロバイダは、リソースへのアクセス権を付与する前にアサーションの信頼性を確立するために認証プロセスに関する情報を必要とする場合があります。

認証コンテキストの要求

認証コンテキストを要求するには、SiteMinder サービス プロバイダが、アイデンティティ プロバイダへの認証リクエストに <RequestedAuthnContext> 要素を含める必要があります。 サービス プロバイダは、SP から IdP へのパートナーシップの設定に基づいて、この要素をリクエストに追加します。

認証コンテキストの取得

SiteMinder アイデンティティ プロバイダは、以下の 2 つの方法の*いずれか*で認証コンテキストを取得します。

- IdP から SP へのパートナーシップ設定で静的 AuthnContext URI を指定します。

  フェデレーション パートナーが AuthnContext リクエストをサポートしない SiteMinder サービス プロバイダである場合は、管理 UI に手動で URI を入力します。

- AuthnContext URI は設定された認証コンテキスト テンプレートを使用して動的に決定します。

  ポリシー サーバは、ポリシー サーバで定義された認証レベルに認証コンテキスト URI をマッピングします。 認証レベルは、確立されたユーザ セッションの認証コンテキストの強度を示します。 レベルにより、認証コンテキストをアイデンティティ プロバイダのユーザ セッションから導出できるようになります。

アイデンティティ プロバイダはリクエストを受信すると、<RequestedAuthnContext> 要素の値を認証コンテキストと比較します。この比較は、サービス プロバイダからのリクエストの比較値に基づいています。比較が成功した場合、アイデンティティ プロバイダはサービス プロバイダに返すアサーションに認証コンテキストを含めます。サービス プロバイダで検証が設定されている場合、サービス プロバイダはリクエストした値を持つ受信認証コンテキストを検証します。

IdP によって開始される SSO の認証コンテキスト処理

シングル サインオンが IdP で開始される場合、認証コンテキスト処理では以下の手順に従います。

1. ユーザ リクエストは IdP でシングル サインオンをトリガします。
2. ユーザは認証されて、ユーザ セッションが生成されます。認証方式で設定された保護レベルがセッションと関連付けられます。
3. IdP の認証コンテキスト設定に応じて、以下の*いずれか*の状態が発生します。
   - 自動検出が発生する

     設定された認証コンテキスト テンプレートに基づいて、AuthnContext クラスはセッションの保護レベルにマッピングされます。
   - 事前定義済み認証クラスが使用されます。

     指定するハードコードされた URI がアサーションに追加されます。
4. IdP はアサーションを生成して認証コンテキストを追加します。 その後、アサーションは SP に送信されます。
5. SP では、そのアサーションの認証コンテキスト クラスと SP で設定された認証コンテキスト クラスの間で別の比較が行われます。この比較が成功すると、認証トランザクションは完了です。

SP によって開始される SSO の認証コンテキスト処理

シングル サインオンが SP で開始される場合、認証コンテキスト処理では以下の手順に従います。

1. SP は、<RequestedAuthnContext> 要素および比較演算子を含む認証リクエストを送信します。要素は SP から IdP へのパートナーシップの設定に基づいて含まれています。
2. IdP がリクエストを受信すると、IdP はユーザを認証し、ユーザ セッションが生成されます。 認証方式用の保護レベルがセッションと関連付けられます。
3. IdP の認証コンテキスト設定に応じて、以下の*いずれか*の状態が発生します。
   - 自動検出が発生する

     設定された認証コンテキスト テンプレートに基づいて、AuthnContext クラスはセッションの保護レベルにマッピングされます。
   - 事前定義済み認証クラスが使用される

     指定するハードコードされた URI がアサーションに追加されます。
4. IdP は、AuthnContext をユーザ セッションの認証クラスと比較します。この比較は、リクエストで送信される比較演算子に基づいています。各比較演算子が処理に及ぼす影響の例については、この手順に続く表を参照してください。

   SP が複数の認証コンテキスト URI をリクエストに含める場合、クラスは一つずつ順番にセッションのコンテキストと比較されます。 最初に比較が成功した時点で、IdP はセッション認証コンテキストをアサーションに追加します。
5. 比較が成功すると、認証コンテキストが SP に送信されるアサーションに追加されます。

   比較が成功しない場合、トランザクションは「noauthncontext」ステータス レスポンスで終了します。

6. SP では、アサーションの認証コンテキストと SP で設定された認証コンテキストの間で次の比較が行われます。 この比較が成功すると、認証トランザクションは完了です。

以下の表では、認証コンテキスト リクエストで送信される比較属性に応じて、認証コンテキストが処理される例を示します。

SP によって要求される認証コンテキスト	比較属性値	IdP によって設定される認証コンテキスト	Status Response
パスワード	exact	InternetProtocol	NoAuthnContext
パスワード	minimum	InternetProtocol	NoAuthnContext
パスワード	better	InternetProtocol	NoAuthnContext
InternetProtocol	exact	InternetProtocol	Success
InternetProtocol	minimum	InternetProtocol	Success
InternetProtocol	maximum	InternetProtocol	Success
InternetProtocol	maximum	パスワード	NoAuthnContext
InternetProtocol	better	パスワード	Success

認証コンテキスト テンプレートの概要

認証コンテキスト テンプレートによって、パートナーがサポートする特定の SAML 2.0 AuthnContext URI が定義されます。 各 URI により、特定のコンテキスト クラスに保護レベルが割り当てられることが識別され、保護レベルは強度レベルにマッピングされます。

パートナーシップごとにテンプレートを選択することができ、複数のパートナーシップで 1 つのテンプレートを使用できます。

テンプレートには、各パートナーでの以下の個別の機能があります。

IdP

IdP が SP リクエストの認証コンテキストを自動的に検出するように設定されている場合は、IdP で認証コンテキスト テンプレートが必要です。

テンプレートは URI をユーザ セッションに関連付けられた保護レベルにマッピングします。 保護レベルは、ポリシー サーバでの認証方式の強度（1 から最も強い 1000 まで）を示します。 管理者は、ユーザを認証してユーザ セッションを確立する認証方式を設定する際に、保護レベルを割り当てます。

IdP は最初にテンプレートを使用して、ユーザ セッションの強度を判定します。 次に、テンプレートを使用して、SP 認証リクエストの URI の強度を判定します。 これで、これらの強度レベルが比較されます。

SP

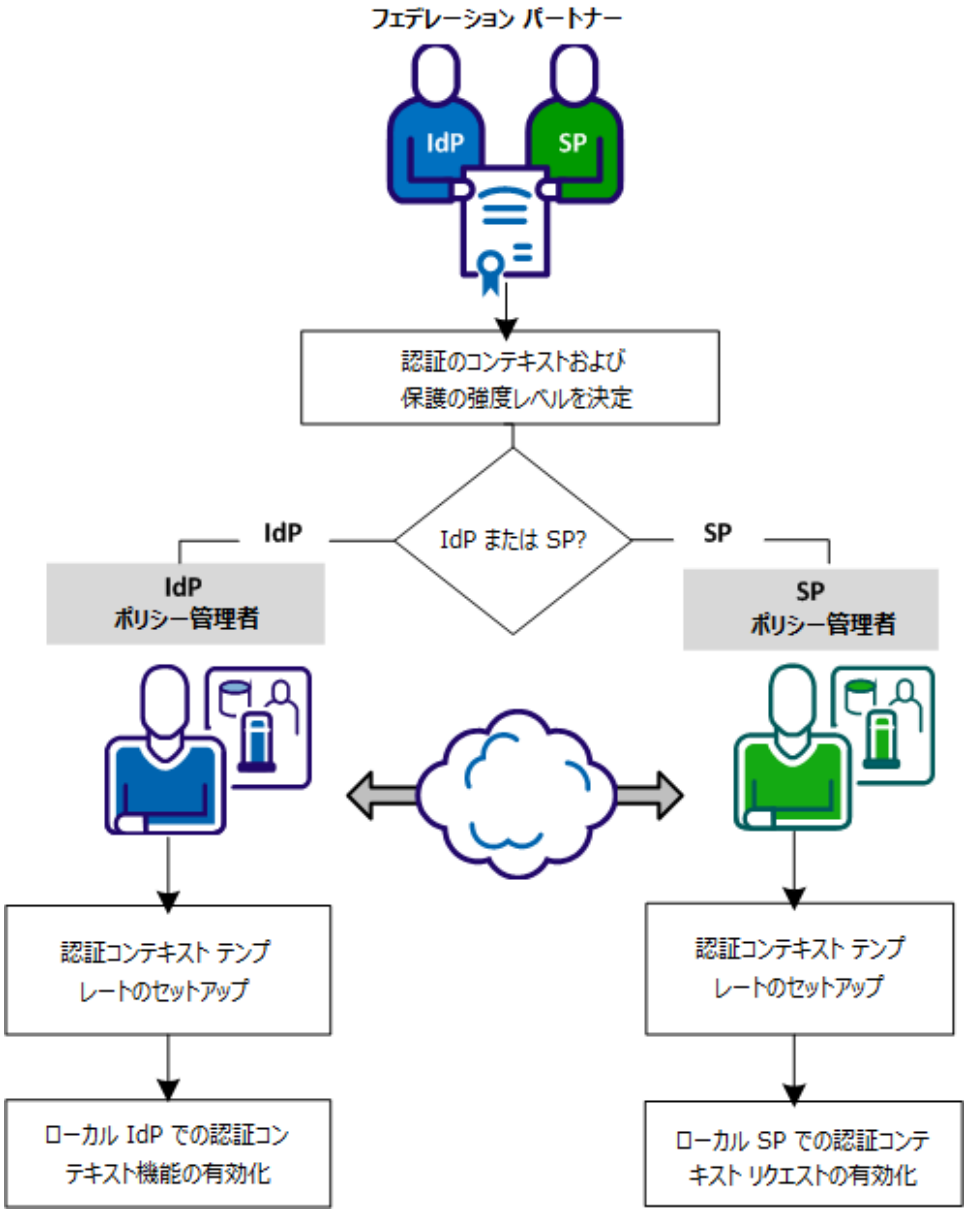
SP での認証コンテキスト テンプレートは、認証リクエストで送信される認証コンテキストを生成するために必要です。 SP はリクエストの生成後、IdP にそれを送信します。 テンプレートは、受信したアサーションが認証コンテキスト リクエストを満たしていることを SP が検証するためにも必要です。

設定を進める前に、以下の最小限の知識要件を満たしていることを確認します。

- 認証コンテキスト処理に関連する SAML 2.0 標準に精通している。
- フェデレーション設定オブジェクトについての理解。
- 管理 UI のアクセス方法および使用方法についての知識。

認証コンテキスト テンプレートの設定

-以下の図は、各パートナーの設定プロセスを示しています。 各サイトで SiteMinder Federation をインストールする必要はありません。



認証コンテキスト処理を設定するには、以下の手順に従います。

1. パートナーとの認証コンテキストおよび強度レベルを決定します。
2. 認証コンテキスト テンプレートをセットアップします。
3. サイトのタスクを実行します。
   - ローカル IdP パートナーシップで認証コンテキスト処理を有効にします。
   - ローカル SP パートナーシップで認証コンテキスト リクエストを有効にします。

パートナーとの認証コンテキストと強度レベルの決定

SP は、リクエストされたリソースへのアクセスを許可する前に、特定の認証コンテキストと強度レベルを必要とすることがあります。 SP でのリソースの感度に基づいて、SP は IdP から受け取るアサーションに確信を持つ必要があります。

IdP および SP の管理者は、サポートされる認証コンテキストおよび各認証コンテキスト URI の相対的強度のガイドラインを確立する必要があります。 IdP での URI の順序は、関連付けられた強度レベルと共に、IdP が SP にどのように応答するかに影響します。

たとえば、SP が、X.509 証明書および完全一致の比較値の認証コンテキストをリクエストするとします。 IdP はリクエストしたユーザを適切な強度レベルで認証し、認証コンテキストの評価中に比較値を満たす必要があります。

認証コンテキスト テンプレートのセットアップ

認証コンテキスト処理を実装するために認証コンテキスト テンプレートをセットアップします。 この手順はアイデンティティ プロバイダまたはサービス プロバイダで同じです。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［認証コンテキスト テンプレート］を選択します。

   ［認証コンテキスト テンプレートの表示］ウィンドウが開きます。

3. ［テンプレートの作成］を選択します。

   テンプレート ウィザードで最初の手順が開きます。

4. テンプレートの名前を入力します。

5. 以下の*いずれか*のアクションを実行します。

   - 手動で URI を入力し、［URI の追加］をクリックします。
   - ［デフォルト URI のロード］をクリックして事前定義済みリストから URI を選択します。 ［使用可能な URI］から［選択された URI］リストに URI を移動します。

6. 強度レベルで、選択された URI を並べ替えます。 強度レベルは、最強の URI が一番上で、最弱の URI が一番下の降順になります。

7. ［次へ］をクリックします。

8. （オプション）同じ強度レベルを必要とする URI を、前の URI の下に URI をインデントすることによってグループ化します。 ［グループ化の変更］矢印を使用して URI をグループへ、またはグループから移動させます。

9. ［保護レベルの有効化］をクリックします。

   保護レベルを認証方式から URI にマッピングします。 保護レベルは、1 から最も強い 1000 までの範囲で認証方式の強度を示します。個々の URI が一意の保護レベルを持つことができますが、URI のグループ化とは、それらが同じ強さのレベルを持つことを意味します。

   保護レベルを割り当てる場合は、以下の情報を考慮してください。

   - 保護レベルを降順に割り当てます。最強のコンテキストを上部に、および最弱のコンテキストを下部にして一覧表示します。
   - 最大の保護レベルを変更することができ、管理 UI によって最小が計算されます。 管理 UI は、各保護レベルに URI が関連付けられるように、レベルの範囲にギャップがないことを確認します。

   保護レベル割り当てについての詳細を参照してください。

10. ［完了］をクリックして設定を確認します。

テンプレートが完成しました。

コンテキスト テンプレート用の保護レベル割り当て

保護レベルは、認証の強度を示します。 選択した各認証コンテキスト URI に保護レベルを割り当てます。リストの各 URI の最大のレベルを指定します。 最小の保護レベルは、リスト内の後続の URI の最大レベルに基づいて自動的に決定されます。 この範囲は保護レベルを反映します。

保護レベルの割り当ては、設定されたポリシー サーバ認証方式の保護レベルを反映している必要があります。 たとえば、ポリシー サーバは 20 の保護レベルで X.509 認証方式を使用できます。テンプレートが指定する保護レベル範囲には 20 を含む必要があります。最後に、ポリシー サーバは、保護レベルに基づいて URI の強度レベルを生成します。

例

ポリシー サーバで設定される認証方式	保護レベル
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	20
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract	15
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	10
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	5

各 URI について、ポリシー サーバは保護レベルを URI の強度レベルに自動的にマッピングします。範囲は、認証方式の保護レベルを対象にします。例：

- X509 方式では、保護レベル 16 ～ 1000 を対象にします
- MobileTwoFactorContract では 11 ～ 15 の保護レベルを対象にします。
- インターネットプロトコルでは 6 ～ 10 を対象にします
- パスワードでは 1 ～ 5 を対象にします

URI	保護レベル最大	URI 強度
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	4
urn:oasis:names:tc:SAML:2.0:ac:classes: MobileTwoFactorContract	15	3
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	10	2

URI	保護レベル最大	URI 強度
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	4
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	5	1

複数の URI をグループ化すると、グループ化によって、異なる保護レベルを持つ URI が同じ URI 強度を持つことができます。 この強度は、URI が同等であると見なされることを意味します。

以下の変更されたテーブルは、X.509 URI および MobileTwoFactorContract URI のグループ化を示します。

URI	保護レベル最大	URI 強度
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	3
urn:oasis:names:tc:SAML:2.0:ac:classes:MobileTwoFactorContract	800	3
urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol	700	2
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	200	1

強度レベルの範囲は、リスト内の総グループ数を反映します。 たとえば 3 つのグループがある場合、強度レベルの範囲は 1 から総グループ数の 3 です。

ローカル IdP パートナーシップでの認証コンテキスト処理の有効化

IdP として機能しているポリシー サーバは、以下の 2 つの方法でアサーションの認証コンテキストを取得できます。

- 事前定義済み認証クラスを使用します

  認証クラスに対して URI を指定し、SP のコンテキスト リクエストを無視します。 ハードコードされたエントリは、IdP によって開始されたシングル サインオンのデフォルト認証コンテキストとして機能できます。

- 認証クラスを自動的に検出します。

  ポリシー サーバは認証コンテキスト テンプレートを使用して、ユーザ セッション認証コンテキストを自動的に検出します。

  SP の認証リクエストに <RequestedAuthnContext> 要素が含まれていなくても、IdP はテンプレートを使用します。 要素が存在すると、IdP による追加の評価がトリガされて、アサーションに追加できる選択肢が制限されます。

  認証コンテキスト処理 (P. 262)のフローに関する詳細を参照できます。

次の手順に従ってください:

1. IdP から SP へのパートナーシップ ウィザードの［SSO と SLO］手順に移動します。
2. ［認証］セクションで、認証コンテキストの取得方法を指定します。事前定義済み認証クラスまたは認証コンテキスト テンプレートで自動検出されたクラスを使用します。
3. 先の手順で選択した方法の手順に従います。
   - 事前定義済みクラスをアサーションに含めるには、［認証クラス］プルダウン メニューから URI を選択します。
   - セッション コンテキストおよびテンプレートからのクラスを含めるには、［認証コンテキスト テンプレート］フィールドからテンプレートを選択するか、［テンプレートの作成］をクリックします。
4. （オプション）。 認証コンテキストの取得方法によっては、［RequestedAuthnContext を無視］チェック ボックスをオンにすることもできます。

以下の表では、［AuthnContext の設定］および［RequestedAuthnContext を無視］設定がどのように連携するかを示します。

AuthnContext の設定	RequestedAuthnContext を無視	SP が AuthnContext を要求する	結果
事前定義済みクラス	選択	はい	IdP は <RequestedAuthnContext> を無視してアサーション内の定義された値を使用します。
事前定義済みクラス	選択	いいえ	デフォルトによって、IdP は定義された値をアサーション内に返します。

AuthnContext の設定	RequestedAuthnContext を無視	SP が AuthnContext を要求する	結果
事前定義済みクラス	選択なし	はい	IdP が認証コンテキスト リクエストを処理するように設定されていないので、トランザクションは失敗します。IdP はエラー メッセージを SP に返します。
事前定義済みクラス	選択なし	いいえ	デフォルトによって、IdP は定義されたクラス値をアサーション内に返します。
自動検出クラス	選択	はい	IdP は認証方式の保護レベルを認証コンテキスト テンプレートと比較し、一致する認証 URI をアサーション内に返します。IdP は SP リクエストの値を無視します。
自動検出クラス	選択	いいえ	IdP は認証方式の保護レベルを認証コンテキスト テンプレートと比較し、一致する認証 URI をアサーション内に返します。IdP は SP リクエストの値を無視します。
自動検出クラス	選択なし	はい	IdP は保護レベルを SP が送信する認証コンテキスト クラスと比較します。IdP は認証コンテキスト テンプレートを使用して、アサーションに配置する認証 URI を決定します。
自動検出クラス	選択なし	いいえ	IdP は認証方式の保護レベルを認証コンテキスト テンプレートと比較し、一致する認証 URI をアサーション内に返します。

ローカル SP パートナーシップでの認証コンテキスト リクエストの有効化

認証コンテキストはアサーション認証ステートメントの一部であり、ユーザが IdP で認証した方法を示します。SP は、リソースへのアクセス権を付与する前にアサーションの信頼性を確立するために認証プロセスに関する情報を必要とする場合があります。

認証コンテキスト URI は、<AuthnContext> 要素内の <AuthnContextClassRef> 要素の値です。各 URI によって、SP が IdP にアサーション内に返させるコンテキスト クラスが識別されます。

SP の認証コンテキスト テンプレートによって以下の情報が定義されます。

- SP が IdP から受信する必要がある URI。 送信リクエストの場合、テンプレート内の URI は、要求されたリソースへのアクセスを許可する前に、SP が受理できる認証コンテキストを示します。
- リクエスト内の URI を IdP で定義された URI と比較する方法。
- SP が URI を使用する方法。SP は URI を送信認証リクエストに含めることができます。SP は受信アサーション レスポンス内の URI を検証することもできます。 両方の機能に対して URI 使用状況を設定できます。

パートナーシップごとにテンプレートを選択することができ、かつ複数のパートナーシップで 1 つのテンプレートを使用できます。

認証コンテキスト リクエストを有効にする前に、または SP パートナーシップの設定中に、認証コンテキスト テンプレートを設定します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 編集する SP から IdP へのパートナーシップを選択します。
3. パートナーシップ ウィザードの［AuthnContext の設定］手順に移動します。

   ［設定］ダイアログ ボックスが開きます。
4. ［認証コンテキスト処理の有効化］チェック ボックスをオンにします。

5. ダイアログ ボックスの以下のフィールドに入力します。 フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。

   以下の情報に注意してください。

   - 認証コンテキスト テンプレートが存在しない場合は、［テンプレートの作成］を選択します。
   - ［比較］フィールドでは、SP 認証リクエスト内の URI をアイデンティティ プロバイダで設定された URI と比較する方法を表します。

     ［ヘルプ］には、各比較演算子の詳細が記載されています。
   - ［使用可能な URI］リストから URI を選択している場合、使用可能な URI は選択されたテンプレートに対して設定された URI を反映します。 事前定義済みテンプレートがない場合は、［テンプレートの作成］をクリックして設定します。

認証コンテキスト リクエストはアイデンティティ プロバイダに送信された認証リクエストに含まれています。

第 18 章: フェデレーション メッセージの署名および暗号化

このセクションには、以下のトピックが含まれています。

フェデレーションのためのキーと証明書の管理

アサーションの保護およびアサーション内のデータの暗号化は、パートナーシップ設定の重要な部分です。 フェデレーション環境で、キー/証明書ペアおよびスタンドアロン証明書は多くの機能に役立ちます。

- アサーションの署名/検証（3 つのすべてのプロファイル）
- 認証リクエストの署名/検証（SAML 2.0 のみ）
- シングル ログアウト リクエストおよびレスポンスの署名/検証（SAML 2.0）
- HTTP-Artifact SSO のバックチャネル リクエストおよびレスポンスの署名 (SAML 1.1 および 2.0)
- アサーション全体またはアサーションの一部の暗号化/復号化（SAML 2.0）
- Artifact シングル サインオン用のバックチャネル全体のクライアント認証情報 (SAML 1.1 および 2.0)

「*ポリシー サーバ設定ガイド*」には、キーおよび証明書の管理に関する情報と手順が記載されています。

SSL サーバ証明書を使用して、以下のタスクを実行できます。

- SSL 接続でのフェデレーション トラフィックを管理する。
- Artifact シングル サインオンでのバックチャネルの通信のセキュリティを保護する。

SiteMinder Web エージェントがインストールされている Web サーバに対して SSL を有効にする手順を参照してください。

注: SSL を有効にすると、Base URL パラメータも含めて、すべてのサービスの URL に影響があります。具体的には、すべてのサービス URL が https:// で始まる必要があります。

SAML 2.0 署名アルゴリズム

SAML 2.0 の場合、タスクに署名するための署名アルゴリズムを選択するオプションがあります。 アルゴリズムを選択する機能は以下のユース ケースをサポートします。

- IdP が RSAwithSHA1 または RSAwithSHA256 アルゴリズムで、アサーション、レスポンスおよび SLO-SOAP メッセージに署名する IdP から SP へのパートナーシップ。
- SP が RSAwithSHA1 または RSAwithSHA256 アルゴリズムで、認証リクエストおよび SLO-SOAP メッセージに署名する SP から IdP へのパートナーシップ。

署名検証によって、署名済みドキュメントで使用中のアルゴリズムを自動検出して、それを確認します。 署名検証の設定は必要ありません。

SAML 1.1 プロデューサおよび WSFED IP での署名設定

［署名］手順では、ポリシー サーバが SAML アサーションまたは WS-フェデレーション トークン レスポンスを署名するために秘密キーおよび証明書を使用する方法を定義できます。 SAML 1.1 の場合は、アサーション レスポンスの代わりにアサーションのみ署名することを選択できます。

SAML 1.1 および WS-フェデレーションは、暗号化をサポートしていません。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

注: システムが FIPS_COMPAT または FIPS_MIGRATE モードで動作している場合、すべての証明書およびキー エントリはプルダウン リストから利用可能です。 システムが FIPS 専用モードで動作している場合は、FIPS が承認した証明書およびキー エントリのみが選択可能です。

次の手順に従ってください:

1. 管理 UI へのログイン
2. 変更するアサーティング パーティから依存パーティへのパートナーシップを選択します。
3. パートナーシップ ウィザードの［署名］手順に移動します。
4. ［署名］セクションで、［署名秘密キー エイリアス］フィールドのプルダウン リストから別名を選択します。

   証明書データ ストアに秘密キーがない場合は、［インポート］をクリックしてキーをインポートします。 または、［生成］をクリックして証明書リクエストを作成します。

   このフィールドの入力によって、アサーティング パーティがアサーションおよびレスポンスに署名するために使用する秘密キーを示します。
5. （SAML 1.1 のみ）［Artifact］および［Post］署名オプションでは、署名が必要な特定のコンポーネント（アサーション、レスポンス）を選択します。

テスト環境で SiteMinder を使用している場合、署名処理を無効にしてテストを簡略化できます。 ［署名の処理を無効にする］チェック ボックスをクリックします。

署名設定が完了しました。

SAML 1.1 コンシューマおよび WSFED RP での署名検証

［署名］手順では、ポリシー サーバが SAML アサーションまたは WS-フェデレーション トークン レスポンスを検証するために秘密キーおよび証明書を使用する方法を定義できます。 SAML 1.1 の場合は、アサーションのみ検証することを選択できます。

SAML 1.1 および WS-フェデレーションは、暗号化をサポートしていません。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

注: システムが FIPS_COMPAT または FIPS_MIGRATE モードで動作している場合、すべての証明書およびキー エントリはプルダウン リストから利用可能です。 システムが FIPS 専用モードで動作している場合は、FIPS が承認した証明書およびキー エントリのみが選択可能です。

次の手順に従ってください:

1. 管理 UI へのログイン
2. 変更する依存パーティからアサーティング パーティへのパートナーシップを選択します。
3. パートナーシップ ウィザードの［署名］手順に移動します。
4. ［検証証明書エイリアス］フィールド用に証明書データ ストアから別名を選択します。

   このフィールドの入力によって、署名済みアサーションまたはレスポンス、または両方を確認する証明書を示します。証明書データ ストアに証明書がない場合は、［インポート］をクリックして証明書をインポートします。 または、［生成］をクリックして証明書リクエストを作成します。

注: テスト環境で製品を使用している場合、署名処理を無効にしてテストを簡略化できます。 ［署名の処理を無効にする］チェック ボックスをクリックします。

署名設定が完了しました。

SAML 2.0 IdP での署名の設定

パートナーシップ ウィザードの［署名および暗号化］手順では、以下の署名機能に対して製品が秘密キーおよび証明書を使用する方法を定義します。

- SAML アサーション、アサーション レスポンスおよび認証リクエストに署名して確認します。

  SAML 2.0 POST バインディングの場合は、アサーションに署名する必要があります。

- シングル ログアウトのレスポンスおよびリクエストに署名します（HTTP リダイレクト バインディングおよび SOAP バインディング）。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

注: システムが FIPS_COMPAT または FIPS_MIGRATE モードで動作している場合、すべての証明書およびキー エントリはプルダウン リストから利用可能です。 システムが FIPS 専用モードで動作している場合は、FIPS が承認した証明書およびキー エントリのみが選択可能です。

署名オプションを設定する方法

1. パートナーシップ ウィザードの［署名および暗号化］手順を選択します。
2. ［署名］セクションで、［署名秘密キー エイリアス］フィールド用にエイリアスを選択します。 使用できる秘密キーがない場合は、［インポート］をクリックして秘密キーをインポートします。 または、［生成］をクリックして証明書リクエストを作成します。

   このフィールドの入力によって、アサーティング パーティがアサーションおよびシングル ログアウトのリクエストおよびレスポンスに署名するために使用する秘密キーを示します。

   注: フィールドの説明については、［ヘルプ］をクリックしてください。

3. ［署名アルゴリズム］フィールドでデジタル署名用のハッシュ アルゴリズムを選択します。 IdP は、指定されたアルゴリズムを使用してアサーション、レスポンスおよび SLO-SOAP メッセージに署名します。

   最も用途に適したアルゴリズムを選択してください。

   RSAwithSHA256 の方が、結果として生成される暗号化ハッシュ値に使用されるビット数が多いため、RSAwithSHA1 より安全です。

   選択したアルゴリズムがすべての署名機能に使用されます。

4. 証明書データ ストアまたは［検証証明書エイリアス］フィールドからエイリアスを選択します。

   このフィールドの入力によって、署名済み認証リクエスト、またはシングル ログアウトのリクエストまたはレスポンスを確認する証明書を示します。 データベースに証明書がない場合は、［インポート］をクリックして証明書をインポートします。

5. （オプション）アサーションまたはレスポンス、または両方に対して［Artifact］および［POST］署名オプションを指定します。

6. （オプション）シングル ログアウトを使用している場合、ログアウト リクエスト、ログアウト レスポンスまたは両方に対して［SLO SOAP］署名オプションを指定します。

7. （オプション）［署名された認証リクエストが必要］チェック ボックスをオンにします。 このチェック ボックスの選択によって、アサーティング パーティが依存パーティから署名済みリクエストのみを受理することが確認されます。

   すべての設定変更を有効にしてパートナーシップが使用できるようにするために、パートナーシップをアクティブ化します。 サービスの再起動のみでは不十分です。

製品をテスト環境で使用している場合は、署名の処理を無効にしてテストを簡略化できます。 ［署名の処理を無効にする］チェック ボックスをクリックします。

重要: SAML 2.0 実稼働環境で署名処理を有効にします。

SAML 2.0 IdP での暗号化の設定

パートナーシップ ウィザードの［署名および暗号化］手順では、ポリシー サーバが以下のタスクを実行するために秘密キーおよび証明書を使用する方法を定義できます。

- SAML アサーション、アサーション レスポンスおよび認証リクエストに署名して確認します。

  SAML 2.0 POST バインディングの場合は、アサーションに署名する必要があります。

- シングル ログアウトのレスポンスおよびリクエストに署名します（HTTP リダイレクト バインディングおよび SOAP バインディング）。
- すべてのアサーション、名前 ID および属性を暗号化および復号します。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

暗号化オプションを設定する方法

1. ［暗号化］セクションで、以下のチェック ボックスのいずれか、または両方を選択して暗号化するアサーション データを指定します。
   - 名前 ID の暗号化
   - アサーションの暗号化
2. ［暗号化証明書エイリアス］用に証明書データ ストアから証明書の別名を選択します。

   この証明書はアサーション データを暗号化します。使用できる証明書がない場合は、［インポート］をクリックして証明書をインポートします。
3. ［暗号化ブロック アルゴリズム］および［暗号化キー アルゴリズム］フィールドの値を選択します。

   以下のブロック/キー アルゴリズムの組み合わせの場合、証明書に必要な最小キー サイズは 1024 ビットです。

   - 暗号化ブロック アルゴリズム： 3DES

     暗号化キー アルゴリズム： RSA-OEAP

- 暗号化ブロック アルゴリズム： AES-256

  暗号化キー アルゴリズム： RSA-OEAP

  注: AES-256 ビット暗号化ブロック アルゴリズムを使用するには、Java Cryptography Extension （JCE） Unlimited Strength Jurisdiction ポリシー 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。

暗号化の設定が終了しました。

SAML 2.0 SP での署名の設定

パートナーシップ ウィザードの［署名および暗号化］手順では、ポリシー サーバが以下のタスクを実行するために秘密キーおよび証明書を使用する方法を定義できます。

- SAML アサーション署名およびアサーション レスポンスを確認して認証リクエストに署名します。

  注: SAML 2.0 POST バインディングの場合は、IdP はアサーションに署名する必要があります。

- シングル ログアウトのレスポンスおよびリクエストに署名します（HTTP リダイレクト バインディングおよび SOAP バインディング）。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

注: システムが FIPS_COMPAT または FIPS_MIGRATE モードで動作している場合、すべての証明書およびキー エントリはプルダウン リストから利用可能です。 システムが FIPS 専用モードで動作している場合は、FIPS が承認した証明書およびキー エントリのみが選択可能です。

署名オプションを設定する方法

1. まず、パートナーシップ ウィザードの［署名および暗号化］手順を選択します。
2. ［署名］セクションで、［署名秘密キー エイリアス］フィールド用に証明書データ ストアから別名を選択します。データベースに秘密キーがない場合は、［インポート］をクリックして秘密キーをインポートします。または、［生成］をクリックしてキー ペアを作成および証明書リクエストを生成します。

   このフィールドの入力によって、依存パーティが認証リクエスト、シングル ログアウトのリクエストおよびレスポンスに署名するために使用する秘密キーを示します。

   注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。
3. ［署名アルゴリズム］フィールドでデジタル署名用のハッシュ アルゴリズムを選択します。SP は、指定されたアルゴリズムを使用して認証リクエストおよび SLO-SOAP メッセージに署名します。

   最も用途に適したアルゴリズムを選択してください。

   RSAwithSHA256 の方が、結果として生成される暗号化ハッシュ値に使用されるビット数が多いため、RSAwithSHA1 より安全です。

   SiteMinder は、すべての署名機能に対して、選択されたアルゴリズムを使用します。
4. ［検証証明書エイリアス］フィールド用に証明書データ ストアから別名を選択します。

   このフィールドの入力によって、依存パーティが署名済みアサーションまたはシングル ログアウトのリクエストおよびレスポンスを確認するために使用する証明書を示します。データベースに証明書がない場合は、［インポート］をクリックして証明書をインポートします。

5. （オプション）SP がすべての認証リクエストに署名するように、［認証リクエストに署名］を選択します。 リモート アサーティング パーティが認証リクエストへの署名を必要とする場合は、このオプションをオンにします。

   すべての設定変更を有効にしてパートナーシップが使用できるようにするために、パートナーシップをアクティブ化します。 サービスの再起動のみでは不十分です。

テスト環境で SiteMinder を使用している場合、署名処理を無効にしてテストを簡略化できます。 ［署名の処理を無効にする］チェック ボックスをオンにして機能を無効にします。

重要: SAML 2.0 実稼働環境で署名処理を有効にします。

SAML 2.0 SP での暗号化の設定

［署名および暗号化］手順では、アサーション、名前 ID および属性の暗号化および復号など、SP が秘密キーおよび証明書を使用する方法を設定できます。

証明書データ ストアには複数の秘密キーおよび証明書がある場合があります。 複数のフェデレーション パートナーが存在する場合、それぞれのパートナーに異なるキー ペアを使用できます。

注: システムが FIPS_COMPAT または FIPS_MIGRATE モードで動作している場合、すべての証明書およびキー エントリはプルダウン リストから利用可能です。 システムが FIPS 専用モードで動作している場合は、FIPS が承認した証明書およびキー エントリのみが選択可能です。

暗号化オプションを設定する方法

1. ［暗号化］セクションで、アサーションで正しいデータが暗号化されるように、以下のチェック ボックスのいずれか、または両方を選択します。

   - 暗号化された名前 ID を必要とする
   - 暗号化されたアサーションを必要とする

   注: AES-256 ビット暗号化ブロック アルゴリズムを使用するには、Sun Java Cryptography Extension （JCE） Unlimited Strength Jurisdiction ポリシー 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。

2. ［復号化秘密キー エイリアス］用に証明書データ ストアからエイリアスを選択します。

   この秘密キーは暗号化されたアサーション データを復号します。使用できる証明書がない場合は、［インポート］をクリックして証明書をインポートするか、［生成］をクリックしてキー ペアを作成および証明書リクエストを生成します。

暗号化の設定が終了しました。

第 19 章: フェデレーション環境の保護

このセクションには、以下のトピックが含まれています。

連携したトランザクションを保護する方法

いくつかのメカニズムは、アサーションの暗号化、およびパートナー サイト間の SSL 接続を使用するなど、連係したパートナー間のトランザクションの保護を支援します。

パートナーシップ フェデレーション を含むフェデレーション環境をセットアップする場合、環境の保護に関する以下の推奨事項があります。

- 使い捨てのアサーションを生成する。
- クロス サイト スクリプティングに対して保護します。

これらのトピックについては、以降の各セクションで説明します。

アサーションの使い捨ての適用

有効期間を過ぎたアサーションの再利用は、期限切れ認識情報による認証決定をもたらします。 再利用を防ぐために、SiteMinder は、SAML 1.x および 2.0 仕様に従って使い捨て用のアサーションを生成できます。 アサーションには、以降のトランザクション用にアサーションを保持しないように依存パーティに指示する要素が含まれており、アサーションの再利用による問題を防止します。

SiteMinder がアサーティング パーティ（プロデューサ/IdP）として機能している場合、アサーションの使い捨てを設定できます。 SAML 1.x プロデューサの場合、［**DoNotCache 条件の設定**］設定を選択できます。 SAML 2.0 IdP の場合、［**OneTimeUse 条件の設定**］設定を選択できます。 これらの環境設定によって、SiteMinder は使い捨て条件を示すアサーションに適切な要素を挿入できます。

注: アサーションの使い捨てと SAML 1.x および 2.0 の HTTP-POST シングルサインオン用使い捨てポリシーを混同しないように注意してください。 SiteMinder は、依存パーティとして機能するときに POST トランザクション専用に使い捨てポリシーを使用します。使い捨て機能は HTTP-Artifact および HTTP-POST 用です。

フェデレーション環境間の接続のセキュリティ保護

セキュリティ保護された接続で通信する場合、フェデレーション パートナー間またはパートナーとアプリケーション間で送信される ID 情報が最も厳重に保護されます。

依存パーティとターゲット アプリケーション間の接続のセキュリティ保護

依存パーティからクライアント サイト ターゲット アプリケーションへのデータ伝送のセキュリティを保護します。 セキュリティ保護された接続を通信チャネルとして使用することで、セキュリティ攻撃に対する環境の脆弱性が改善されます。

たとえば、アサーションには、依存パーティが抽出してクライアント アプリケーションに送信する属性が含まれることがあります。 依存パーティでは、HTTP ヘッダ変数や Cookie を使用して、これらの属性をアプリケーションに渡すことができます。 ヘッダや Cookie に保存された属性はクライアント側で上書きできるため、悪意のあるユーザが他のユーザになりすますことが可能になります。 SSL 接続を使用することで、この種のセキュリティ侵害から環境が保護されます。

ベスト プラクティスとして、該当するエージェント設定オブジェクト（ACO）に UseSecureCookies パラメータを設定することにより、この脆弱性から保護します。UseSecureCookies パラメータを設定すると、「secure」フラグが付けられた Cookie を生成するようにフェデレーション Web サービスに指定されます。このフラグは、Cookie が SSL 通信チャネルのみに送信されることを示します。

注: 変更対象となる ACO は、フェデレーション環境のセットアップに応じて異なります。 Web エージェントがインストールされているシステムと同じシステムにフェデレーション Web サービスを展開する場合は、Web エージェントに対応した ACO を編集してください。 Web エージェントとは異なるシステムにフェデレーション Web サービスを展開する場合は、フェデレーション Web サービスに対して作成した固有の ACO を編集してください。

SiteMinder アサーティング パーティでの初期認証のセキュリティ保護

SiteMinder アサーティング パーティでのユーザの初期認証は、潜在的な脆弱性を与えます。 ユーザがアサーティング パーティでユーザ セッションを確立するために最初に認証する際に、セッション ID Cookie がブラウザに書き込まれます。 cookie が非 SSL 接続の上で送信される場合、攻撃者は Cookie を取得してユーザの機密情報を盗むことができます。 そして、攻撃者は、情報を使用してインパーソネーションや個人情報の盗難を実行できます。

ベスト プラクティスとして、Web エージェント パラメータ UseSecureCookies を設定することにより、この脆弱性に対して保護します（このパラメータはエージェント設定オブジェクトで変更できます）。UseSecureCookies パラメータを設定すると、「secure」フラグが付けられた Cookie を生成するように Web エージェントに指定されます。 このフラグは、ブラウザが SSL 接続上のみで Cookie を渡すことを示し、その結果、セキュリティが向上します。概して、すべての URL に対して SSL 接続を確立することが推奨されます。

クロスサイト スクリプティングからフェデレーション ネットワークを保護する

クロスサイト スクリプティング（XSS）攻撃は、アプリケーションがブラウザからの入力テキストを表示するときに発生する場合があります。 アプリケーションは、実行可能なスクリプトを形成できる文字のテストに失敗した可能性があります。 これらの文字が表示されると、不要なスクリプトがブラウザ上で実行される結果をもたらす場合があります。

SiteMinder は、フェデレーション機能と併用できる複数の JSP を提供しています。 これらの JSP は、出力ストリーム内の安全でない情報がブラウザに表示されないように、リクエスト内のキャラクタをチェックします。

SiteMinder がリクエストを受信すると、以下の JSP はデコードされた値のクロスサイト スクリプティング文字をスキャンします。

- idpdiscovery.jsp

  アイデンティティ プロバイダ ディスカバリ用に依存パーティで使用されます。

- linkaccount.jsp

  動的なアカウント リンク用に依存パーティで使用されます。

- sample_application.jsp

  シングル サインオンを開始する IDP で使用されます。このサンプル アプリケーションを使用して、最初に SSO サービス、次にカスタム Web アプリケーションにユーザを送ることができます。 通常は独自のアプリケーションを使用します。

- signoutconfirmurl.jsp

  WS-フェデレーション サインアウト用にアカウント パートナーで使用されます。

- unsolicited_application.jsp

  ユーザが最初に SSO サービスではなく Web アプリケーションに直接送られる場合、IdP が開始するシングル サインオン用に使用されます。

ページはリクエスト内の以下の文字をスキャンします。

文字	説明
<	左山形かっこ
>	右山形かっこ
‘	一重引用符
“	二重引用符
%	パーセント記号
;	セミコロン
(	開き（左）かっこ
)	閉じ（右）かっこ
&	アンパサンド
+	プラス記号

各 JSP には、スキャンする文字を定義する変数が含まれます。これらの JSP を変更して文字セットの範囲を拡大します。

第 20 章: 依存パーティでのアプリケーション統合

このセクションには、以下のトピックが含まれています。

依存パーティとアプリケーションの相互作用

パートナーシップ ウィザードの［アプリケーション統合］手順は依存パーティにのみ適用できます。 この手順では、ユーザ ID を解決してユーザをターゲット アプリケーションに送るためのフェデレーション操作のさまざまな特徴を定義できます。

［アプリケーション統合］手順で設定できる機能は以下のとおりです。

- ターゲット アプリケーションへのユーザ リダイレクト
- アサーション属性のアプリケーション属性へのマッピング（SAML のみ）
- ユーザ ID のプロビジョニング
- 認証失敗時のユーザ リダイレクト

ユーザをターゲット アプリケーションにリダイレクトする

［アプリケーション統合］手順の［ターゲット アプリケーション］セクションでは、ユーザをターゲット アプリケーションにリダイレクトする方法を定義できます。 選択するリダイレクト方法は、ユーザと共にターゲット アプリケーションに渡すデータのタイプによって異なります。

次の手順に従ってください:

1. パートナーシップ ウィザードの［アプリケーション統合］手順に移動します。
2. ［リダイレクト モード］フィールド用にリダイレクト方法を選択します。 以下の情報に注意してください。
   - ［Cookie データ］を選択する場合、［URL エンコード属性 Cookie データ］チェック ボックスをオンにすることにより、Cookie 内の属性データを URL エンコードできます。 このオプションは SAML 1.1 および 2.0 でのみ使用できます。
   - オープン形式 Cookie またはオープン形式 Cookie ポスト オプションを選択する場合は、追加の必要な設定およびオプションの設定を行います。オープン形式 Cookie とは異なり、オープン形式 Cookie ポストは HTTP-POST リクエストの形でデータを送信します。

     依存パーティが複数の属性値を持つアサーションを受信する場合、ポリシー サーバはすべての値を Cookie でターゲット アプリケーションへ渡します。
   - FIPS 準拠のアルゴリズム（AES アルゴリズム）のいずれかを選択する場合は、CA SiteMinder® Federation SDK を使用してオープン形式の Cookie を生成します。 .NET SDK を使用する場合は、AES128/CBC/PKCS5Padding 暗号化アルゴリズムのみを使用します。

     ターゲット アプリケーションは Cookie を作成する SDK と同じ言語を使用する必要があります。 CA SiteMinder® Federation Java SDK を使用している場合、アプリケーションは Java 内にある必要があります。 .NET SDK を使用している場合、アプリケーションは .NET をサポートしている必要があります。
   - リダイレクト モードとして［HTTP ヘッダ］を選択する場合、SiteMinder は単一のヘッダ内に複数の属性値を提供できます。 カンマで各属性値を区切ります。 このオプションは SAML 1.1 および 2.0 でのみ使用できます。

     リダイレクト モードとしての［HTTP ヘッダ］ (P. 295)の使用、およびヘッダを保護する方法について把握してください。

   フィールドの説明については、［ヘルプ］をクリックしてください。

3. ［ターゲット］フィールドにターゲット アプリケーションの URL を入力します。

   ターゲット リソースを含むサーバの前にプロキシが配置されている場合は、プロキシ ホストの URL を入力します。 すべてのフェデレーション要求をプロキシがローカルに処理します。ターゲット サーバの前に配置された任意のシステムがプロキシ ホストとして機能できます。 また、SiteMinder がインターネットから 直接アクセスされる場合は、SiteMinder 自体がプロキシ ホストとして機能できます。 最終的には、プロキシと連係して動作する場合、ターゲットとして指定する URL は SiteMinder を経由する必要があります。 たとえば、ベース URL が fed.demo.com で、バックエンドのサーバ リソースが mytarget/target.jsp である場合、このフィールドの値は http://fed.demo.com:5555/mytarget/target.jsp となります。

   SAML 2.0 において、RelayState クエリ パラメータでこの値をオーバーライドする場合、このフィールドは空白のままでもかまいません。 RelayState クエリ パラメータは、シングル サインオンをトリガする URL の一部として含めることができます。 このオーバーライドを有効にするには、［リレー状態を使用してターゲットをオーバーライドする］チェック ボックスをオンにします。

ターゲットへのリダイレクトのセットアップが完了しました。

HTTP ヘッダを使用したアサーション データの受け渡し(SAML のみ)

SAML エンティティでは、ポリシー サーバは HTTP ヘッダを使用して、ID 属性をアサーションからバックエンド アプリケーションへ渡すことができます。 バックエンド アプリケーションは、シングル サインオン用のターゲット アプリケーションまたはユーザ プロビジョニング アプリケーションです。システムは、これらのヘッダを暗号化された Cookie で渡します。

ヘッダにはアサーション属性と同じ名前があります。 たとえば、アサーション属性が「address」である場合、アプリケーションは HTTP ヘッダ「ADDRESS」を検索します。

アサーション属性は大文字と小文字を区別しますが、HTTP ヘッダは区別しません。 ポリシー サーバは、大文字と小文字のみが異なる同じ属性を HTTP ヘッダに渡したり、マッピングしたりすることはできません。 たとえば、システムは、ヘッダとして「address」と「Address」を同時に渡すことができません。 通常、大文字と小文字の区別または形式のみが異なる同じ名前を持った属性を使用しないでください。

他に以下の値がヘッダとして渡されます。

- NAMEID
- FORMAT
- AUTHNCONTEXT

HTTP ヘッダを保護する

権限のないユーザがアサーション属性の名前を知った場合、そのユーザはブラウザでヘッダとしてこの名前を設定できます。 ヘッダ セットを使用すれば、悪意のあるユーザがターゲット アプリケーションにアクセスできます。ターゲット アプリケーションは、SiteMinder がアサーションを消費しなくても、予期されたヘッダ値を確認してリソースへのアクセス権を付与します。

FedHeaderPrefix の値を設定することによって、以下の事態を防ぎます。

1. 権限のないユーザが HTTP ヘッダの名前を把握します。 これらのヘッダ名にはプレフィックスが含まれます。
2. 悪意のあるユーザは、ポリシー サーバにヘッダを含めた受信リクエストを送信します。
3. ポリシー サーバは、プレフィックスを含むそのヘッダが受信リクエストのものであり、内部で生成されていないことを認識して、これらのヘッダを削除します。
4. システムは、自身の正式なヘッダをバックエンド アプリケーションに渡す前に、指定されたプレフィックスを各ヘッダに追加します。 その後、ヘッダはアプリケーションに渡されます。

アサーション データを渡す HTTP ヘッダの設定(SAML のみ)

SiteMinder は HTTP ヘッダを使用して、アサーション データを渡すことができます。

次の手順に従ってください:

1. フェデレーション トラフィックを処理している依存パーティ システムに SiteMinder Web エージェントがインストールされていることを確認します。
2. *web_agent_home*/conf に移動して WebAgent.conf ファイルを変更します。以下のエントリのコメントを解除すると次のように表示されます。

   Windows

   LoadPlugin="*path*¥SAMLDataPlugin.dll"

   UNIX

   LoadPlugin="*path*/SAMLDataPlugin.so"
3. (オプションだが推奨) **fedheaderprefix** 設定を Web エージェントの適切な[エージェント設定オブジェクト]に追加します。プレフィックスとして文字列を入力します。

   fedheaderprefix 設定により、SiteMinder が HTTP ヘッダに追加するグローバル プレフィックスが指定されます。プレフィックスの設定によって、SiteMinder がアサーションを消費する前に、権限のないユーザによって HTTP ヘッダが操作されることを防ぎます。その結果、正規ヘッダのみがターゲット アプリケーションに渡されます。HTTP ヘッダの保護 (P. 297)についての詳細を参照してください。
4. パートナーシップ ウィザードの[アプリケーション統合]手順で以下の*いずれか*のタスクを実行します。
   - ターゲット アプリケーションの[リダイレクト モード]として[HTTP ヘッダ]を選択します。
   - ユーザ プロビジョニングの[配信オプション]として[HTTP ヘッダ]を選択します。

HTTP ヘッダは属性データを渡すように設定されました。

アサーション属性のアプリケーション属性へのマッピング（SAML のみ）

SAML 1.1 コンシューマまたは SAML 2.0 SP では、アサーション属性のセットを送信アプリケーション属性のセットにマップできます。 その後、アプリケーション属性がターゲット アプリケーションに渡されます。 属性マッピングでは、ターゲット アプリケーションを変更せずに、カスタマイズされたユーザ操作を提供できます。 属性はパートナーシップ単位でマッピングされるので、依存するパーティのアプリケーションを複数のアサーティング パーティに対して使用できます。

使用可能なマッピングのタイプは、以下のとおりです。

- アサーション属性名をアプリケーション属性名に変換します。

  例

  受信アサーション属性は Region=US です。 この属性を送信アプリケーション属性 ServiceLocation=US に変換できます。

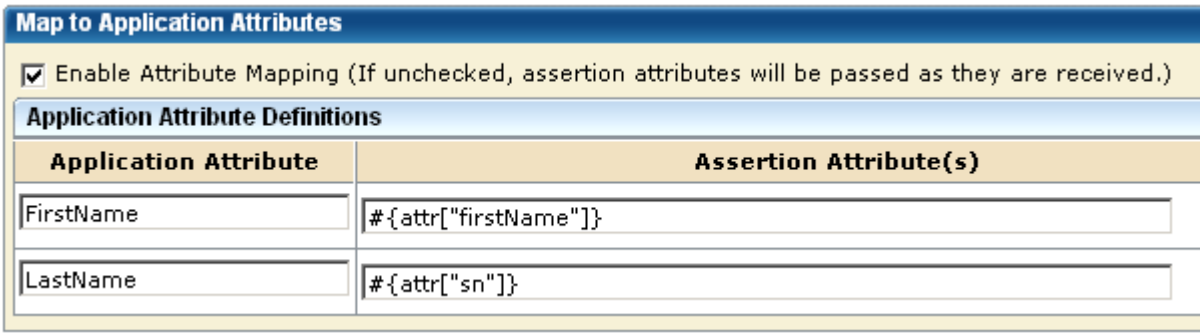
- 個々の属性およびそれらの値を単一の属性に変換します。

  例

  Name=Bob および LastName=Smith の 2 つの属性がアサーションに含まれています。これらの 2 つの属性を FullName =Bob Smith に変換できます。

アプリケーション属性定義テーブルを使用する

［アプリケーション統合］ダイアログ ボックスのアプリケーション属性定義テーブルで属性マッピング ルールを定義します。 このテーブルを以下の図に示します。



［アプリケーション属性］列および［アサーション属性］列はリモートのプロデューサまたは IdP エンティティのアサーション属性を使用して入力します。 このローカル依存パーティでこれらの属性を設定します。 アサーション属性名は［アプリケーション属性］列に入力します。 相当する統一表現言語（UEL）文字列は［アサーション属性］列に入力します。

依存パーティの管理者またはアプリケーション インテグレータには、属性マッピングを設定するために以下の情報が必要です。

- ターゲット アプリケーション属性の名前。
- アサーション内の属性の名前。
- アサーション属性とターゲット アプリケーション属性のマッピング関係。 マッピング関係を理解するということは、使用できるアサーション属性を必要なアプリケーション属性に変換する方法を知っているということです。

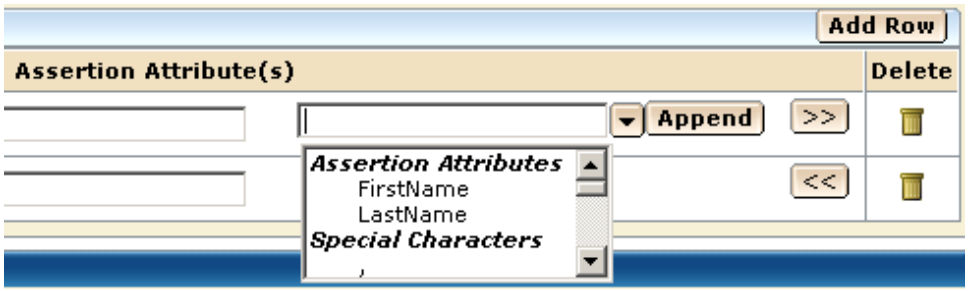
属性マッピングをセットアップする前に、必要なパーティからアプリケーション属性およびアサーション属性の名前を収集します。

アプリケーション属性はターゲット アプリケーションが使用する属性を反映する必要があるので、アプリケーションに適するようにデフォルト値を変更することが必要です。 アプリケーション管理者との帯域外通信によってアプリケーション属性を取得します。

式ビルダ使用によるマッピング ルールの作成

UI は、マッピング ルールの作成に使用できる式ビルダを提供します。［アサーション属性］フィールドの右側のスライダ ボタン（<<）を選択して式ビルダにアクセスします。 スライダ ボタンにより空白のフィールドおよびプルダウン矢印が表示されます。 矢印を選択してマッピングの構成に使用できるアサーション属性および特殊文字のリストを表示します。 スライダ ボタン（>>）をクリックして式ビルダを非表示にします。

以下の図に［式ビルダ］メニューを示します。



式ビルダの［アサーション属性］リストは、リモートのプロデューサまたは IdP エンティティのアサーション属性から作成されます。このローカル依存パーティでこれらの属性を設定します。 属性がアサーション内にあることを知っていれば、エントリを手動で指定できます。 式ビルダ メニューのオプションのみを使用する必要はありません。

［特殊文字］リストには、マッピング ルールの構築に使用できるカンマおよびパーセント記号などの文字が含まれます。 リストから文字を選択する、または文字を手動で入力することができます。

重要: このテーブルにアサーション属性を入力すると、リモート アサーティング パーティで指定されたアサーション属性に関連して大文字と小文字が区別されます。 大文字と小文字の区別が一致する必要があります。SiteMinder がパートナーシップの両側にある場合、リモート IdP のパートナーシップ ウィザードの［名前 ID と属性］手順で属性が指定されます。パートナーとの帯域外通信、またはメタデータのインポートによってアサーション属性を取得します。

マッピング ルールの定義後に、SiteMinder は、レガシー Cookie、オープン形式の Cookie または HTTP ヘッダにデータを配置します。 その後、SiteMinder はアプリケーションにデータを送信します。［アプリケーション統合］ダイアログ ボックスの［ターゲット アプリケーション］セクションで配信方法を指定します。

マッピングの変更および削除

［アプリケーション属性定義］テーブルでいつでも属性マッピングを変更または削除できます。

マッピングを変更する方法

1. 変更する行内のいずれかのフィールドにカーソルを置いて、新しいテキストを入力します。式ビルダを使用して現在の式の最後に値を追加することもできます。
2. ［次へ］をクリックして変更を保存し、ウィザードを終了します。

マッピングを削除する方法

1. 削除するエントリの［削除］列でごみ箱をクリックします。
2. ［次へ］をクリックして変更を保存し、ウィザードを終了します。

適切な構文の使用による属性マッピング ルールの作成

属性マッピングは、アサーション属性をアプリケーション属性に変換するマッピング ルールを使用します。 属性マッピングを有効にすると、SiteMinder はデフォルトのマッピング ルールを生成します。このルールは、リモート プロデューサまたは IdP エンティティに対して指定されたアサーション属性に基づいています。 このすべての設定はローカル依存パーティで行われます。 属性マッピングを無効にすると、アサーション属性は「現状のまま」ターゲット アプリケーションに渡されます。

SiteMinder は、JSP および JSF に類似したマッピング用の統一表現言語（UEL）構文を使用します。 各アサーション属性はハッシュマップに入れられ、**attr** キーワードを割り当てられます。 UEL 式エバリュエータはマッピング ルールのリストを検証して、アサーション属性のハッシュマップに適用します。 その後、式エバリュエータは、結果のアプリケーション属性を含む別のハッシュマップを生成します。 送信アプリケーション属性のハッシュマップは Cookie コンテンツまたはヘッダ変数に変換され、ターゲット アプリケーションに渡されます。

式を作成するためには、SiteMinder が式に使用する構文を理解することが重要です。

単一属性表記

単一のアサーション属性を表記するには、以下の構文を使用します。

#{attr["*attribute_name*"]}

例： #{attr["Name"]} は、名前アサーション属性の値を表します。

複合属性表記

複合値（区切り文字を含む場合もある）を形成するために値式を連結できます。 複合のアサーション属性を表記するには、以下の構文を使用します。

#{attr["*first_attribute*"]}*optional_character* #{attr["*second_attribute*"]}

マッピングの例

以下は一連のマッピング ルールの例です。 これらの例は以下の形式で表されます。

application_attribute=assertion_attributes_expression

名前の例

構文

ID = #{attr["Name"]}

サンプル結果

BobSmith

簡単な連結例

構文

FullName = #{attr["FirstName"]},#{attr["LastName"]}

サンプル結果

Bob,Smith

構文

FullName = #{attr["LastName"]},#{attr["FirstName"]}

サンプル結果

Smith,Bob

スペースは特殊文字とみなされます。式の属性間にスペースが必要な場合は、スペースを入力します。 例：

構文

FullName = #{attr["LastName"]}, #{attr["FirstName"]}

サンプル結果

Smith, Bob

日付の例

構文

Date = #{attr["month"]}/#{attr["dateOfMonth"]}/#{attr["year"]}

サンプル結果

01/05/2010

構文

Date = #{attr["monthSymbol"]} #{attr["dateOfMonth"]}, #{attr["year"]}

サンプル結果

2012/01/05

金額の例

構文

Price = #{attr["amount"]}#{attr["currency"]}

サンプル結果

2.50EUR

電子メール アドレスの例

構文

EmailAddress = #{attr["userName"]}@#{attr["domainName"]}

サンプル結果

JaneDoe@company.com

構文

AcmeEmailAddress = #{attr["AcmeIDKey"]}@acme.com

サンプル結果

bsmith@acme.com

依存パーティでの属性マッピングの設定

SiteMinder がアサーション属性に適用できるマッピング ルールのセットを定義します。 SiteMinder では、特定のアサーション属性または複数のアプリケーション属性の組み合わせをマッピングできます。 マッピングの結果は単一のアプリケーション属性または複数の属性です。

次の手順に従ってください:

1. パートナーシップ ウィザードの［アプリケーション統合］手順に移動します。
2. ［アプリケーション属性へのマップ］セクションで［属性マッピングの有効化］チェック ボックスをオンにします。

   ［アプリケーション属性定義］テーブルが表示されます。
3. テーブル内で既存のアプリケーション属性を変更する、または新しく定義します。すべてのアプリケーション属性はターゲット アプリケーションに渡されます。

   ［アサーション属性］列の値の構文は統一表現言語（UEL）に準拠する必要があります。

   スライダ ボタン (<<) を選択して式ビルダを開き、使用できるオプションを表示します。 属性値にリストから項目を追加するには、アサーションまたは特殊文字を選択して［追加］をクリックします。

   注: アプリケーション属性テーブルで Cookie データおよび特殊文字を指定する場合は、［URL エンコード属性 Cookie データ］オプションを選択します。 チェック ボックスはダイアログ ボックスの［ターゲット アプリケーション］セクションにあります。 特殊文字は、ドロップダウン リストから追加したり、手動で入力することができます。また、ターゲット アプリケーションでは、受け取ったアプリケーション属性の名前および値を URL デコードする必要があります。
4. （オプション）デフォルト マッピングが十分でない場合は、必要なだけ行を追加します。

   デフォルトでは、リモート プロデューサまたは IdP エンティティで定義されたすべてのアサーション属性は、デフォルト（ストレート）マッピングによってテーブルに含まれます。 元のアサーション属性は変更されません。 これらのマッピングを変更できます。
5. アプリケーション属性をターゲット アプリケーションに送信する方法を設定します。 ［アプリケーション統合］ダイアログ ボックスの［ターゲット アプリケーション］セクションで方法を設定します。

属性マッピング設定が完了しました。

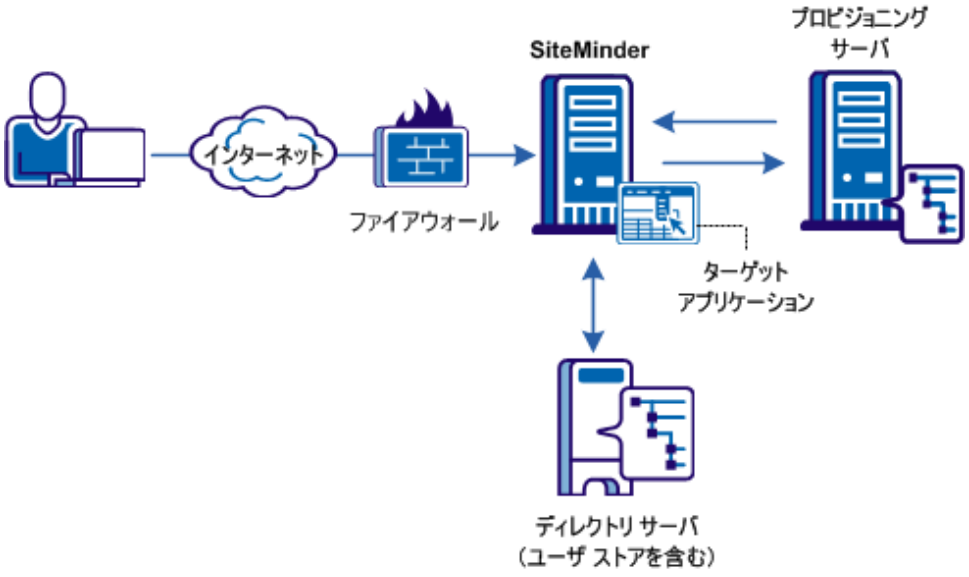
依存側でのユーザ プロビジョニング

フェデレーション ネットワークで、依存側は別のアサーティング パーティから連携するユーザのアカウントを作成できます。 動的プロビジョニングは、データおよびアプリケーションにアクセスするために必要なアカウント権限およびアクセス権限を持つクライアント アカウントを作成するプロセスをサポートします。

リモート プロビジョニング

リモート プロビジョニングは、ユーザ アカウントを作成するためにサードパーティ プロビジョニング アプリケーションを使用します。 アプリケーションは次に CA SiteMinder® Federation を使用してフェデレーション システムでポリシー サーバに必要な情報を渡します。ポリシー サーバは、ユーザ認証情報を作成するためにこのデータを使用します。

リモート プロビジョニングは依存パーティで実行されます。 以下の図ではリモート プロビジョニングのセットアップを示します。



高レベルのプロビジョニングのプロセスは以下のとおりです。

1. 依存パーティのポリシー サーバは、アサーションと共にリソースの要求を受信します。 しかし、ユーザがユーザ ディレクトリに見つかりません。
2. プロビジョニングが有効なまま、ポリシー サーバはアサーション データを含むアクティブ レスポンスを処理し、アサーション データを使用して Cookie を生成します。 さらに、状態を維持する Cookie が生成されてプロビジョニング リクエストが適切であることを示します。
3. ブラウザは、オープン形式の Cookie またはヘッダと一緒に、プロビジョニング アプリケーションにリダイレクトされます。
4. プロビジョニング アプリケーションは通常、ユーザにログインを要求します。 ユーザがログインしたら、Cookie またはヘッダが読み取られます。アプリケーションでは、ユーザ アカウントを確立するためにこのアサーション データおよびログイン認証情報を使用します。

   プロビジョニング アプリケーションは、CA SiteMinder® Federation Java または .NET SDK を使用してオープン形式の cookie を消費します。
5. アカウントがプロビジョニングされた後、ブラウザは依存パーティで再度アサーション コンシューマ サービスにユーザをリダイレクトします。プロビジョニングに関する状態情報を保持する Cookie は、ユーザがプロビジョニングされたことを確認するために検証されます。 認証情報が作成され、認証方式に渡されます。

   注: プロビジョニング アプリケーションは、依存パーティでのアサーション コンシューマ サービスの URI を知っている必要があります。たとえば、依存パーティにおける SiteMinder 用の SAML 2.0 URI は、https://*sp_server:port*/affwebservices/public/saml2assertionconsumer です。
6. ポリシー サーバは、ユーザの特定を 2 度試みます。プロビジョニングが成功した場合、ユーザは認証され、Cookie またはヘッダがターゲット アプリケーションに送信されます。

   ターゲット アプリケーションに対して選択したリダイレクト モードは、ターゲット アプリケーションへのデータ配信方法を決定します。
7. ユーザはターゲット リソースにリダイレクトされます。

プロビジョニング アプリケーションへのアサーション データの配信

リモート プロビジョニングを実行するために、SiteMinder は、アサーション データを含むブラウザをプロビジョニング アプリケーションにリダイレクトします。

SiteMinder これらのいずれかの方法を使用して、アサーション データを渡すことができます。

オープン形式の Cookie

オープン形式の Cookie で SAML アサーション情報を渡します。 Cookie には、アサーション データに基づくログイン ID が含まれます。

注: オープン形式の Cookie を使用する場合、SiteMinder システムおよびリモート プロビジョニング システムは同じドメインにある必要があります。

以下の 2 つの方法のいずれかで Cookie を作成できます。

- CA SiteMinder® Federation SDK によって Cookie を作成します。

  FIPS アルゴリズム（AES アルゴリズム）のいずれかを選択する場合は、CA SiteMinder® Federation SDK を使用して Cookie を生成します。 .NET SDK を使用する予定がある場合は、AES128/CBC/PKCS5Padding 暗号化アルゴリズムのみを使用します。プロビジョニング アプリケーションが .NET を使用する場合、プロビジョニング サーバ上の .NET SDK はオープン形式の Cookie を読み取ります。

  プロビジョニング アプリケーションは、Cookie の作成に使用している SDK と同じ言語を使用する必要があります。 CA SiteMinder® Federation Java SDK を使用している場合、アプリケーションは Java 内にある必要があります。 .NET SDK を使用している場合、アプリケーションは .NET をサポートしている必要があります。

- 手動でオープン形式の Cookie を作成します。

  CA SiteMinder® Federation SDK を使用せずにオープン形式の Cookie を作成するために、プログラミング言語を使用します。 オープン形式の Cookie のコンテンツについての詳細を確認します。

Cookie を書き込むための言語は UTF-8 エンコーディング、および管理 UI で選択できる **PBE** 暗号化アルゴリズムのいずれかをサポートしている必要があります。

Cookie を暗号化するために FIPS 準拠の（AES）アルゴリズムを選択する場合、プロビジョニング アプリケーションは、SDK を使用してオープン形式の Cookie を読み取る必要があります。

オープン形式の Cookie がブラウザで設定されていることを確認します。

オープン形式の Cookie ポスト

オープン形式の Cookie ポストはオープン形式の Cookie に似ていますが、このポストは HTTP-POST リクエストの形でデータを送信します。Cookie データ制限によってデータが失われる可能性がある場合は、このオプションを使用します。

HTTP ヘッダ

SiteMinder はアサーション情報を HTTP ヘッダとして渡すこともできます。 HTTP ヘッダを使用する場合、SiteMinder システムおよびリモート プロビジョニング システムは別のドメインにある場合があります。

アサーション データを渡すための HTTP ヘッダの使用 (P. 295)、およびヘッダを保護する方法について把握してください。

配信オプションはパートナーシップ ウィザードの［アプリケーション統合］手順で設定できます。

ユーザがプロビジョニング アプリケーションにリダイレクトされた後は、SiteMinder はプロセスを制御しなくなります。ユーザ アカウントをプロビジョニングするのに時間がかかる場合は、プロビジョニング アプリケーションがこの状況を処理します。 たとえば、プロビジョニングが進行中であることを説明するメッセージが、アプリケーションによってユーザに送信される場合があります。 この情報により、ユーザ アカウントが使用可能になる前にログインを試行してはいけないことをユーザに知らせます。

リモート プロビジョニング設定

リモート プロビジョニングを設定するには、アサーション データ用の配信オプションを決定してプロビジョニング サーバの URL を指定します。

リモート プロビジョニングの設定に加えて、［IDP にユーザ識別子の作成を許可する］オプションを選択できます。このオプションによって、ユーザの識別子が存在しない場合に IdP が永続識別子を作成できるようになります。 この許可/作成機能は、ローカル メソッドに必要ですが、ローカル アカウント リンクを使用するプロビジョニング専用ではありません。

他の属性と共に送信されるユーザ識別子を IdP に生成させる場合、リモート プロビジョニングと共に許可/作成機能を有効にできます。リモート プロビジョニング サーバのアプリケーションは、生成された識別子の使用方法を決定します。 アプリケーションはローカル アカウント リンクを実行できますが、SiteMinder ローカル アカウント リンクを実行することはできません。

リモート プロビジョニングを設定する方法

1. パートナーシップ ウィザードの［アプリケーション統合］手順から始めます。
2. ［ユーザ プロビジョニング］セクションでプロビジョニング タイプを選択します。
3. プロビジョニング タイプとして［リモート］を選択する場合は、表示される追加フィールドに入力します。

   フィールドの説明については、［ヘルプ］をクリックしてください。
4. ［確認］手順を選択して［完了］をクリックし、変更を保存します。

リモート プロビジョニング設定が完了しました。

リダイレクト URL の使用による失敗した認証の処理（依存パーティ）

アサーション ベースの認証は、アサーションを消費するサイトで失敗する場合があります。 認証が失敗する場合、以降の処理のためにユーザを別のアプリケーション（URL）にリダイレクトするようにポリシー サーバを設定できます。 たとえば、ユーザの特定に失敗した場合、プロビジョニング システムにユーザを送信するために SiteMinder を設定できます。 リダイレクト URL のセットアップはオプションであり、依存パーティでのみ設定できます。

次の手順に従ってください:

1. パートナーシップ ウィザードの［アプリケーション統合］手順から始めます。

   ダイアログ ボックスの［ステータス リダイレクト URL］セクションで、特定の失敗状態専用のリダイレクトを指定します。 SAML 2.0 の場合は、特定の HTTP エラー状態用のリダイレクトを設定することもできます。

   フィールドの説明については、［ヘルプ］をクリックしてください。

2. 設定する各リダイレクト オプションには、SiteMinder がユーザをリダイレクトする方法を指定します。 オプションを以下に示します。

   302 データなし（デフォルト）

   HTTP 302 リダイレクトによってデータなしでユーザをリダイレクトします。

   HTTP POST

   HTTP Post プロトコルによってユーザをリダイレクトします。

リダイレクト URL の設定が完了しました。

第 21 章: パートナーシップ設定に使用できるメタデータのエクスポート

このセクションには、以下のトピックが含まれています。

メタデータ エクスポートの概要

ローカル エンティティは、リモート エンティティがそのエンティティを作成し、パートナーシップを形成するために役立つメタデータを生成します。 パートナーシップの多くの特徴がメタデータ ファイルで定義されているので、メタデータによってパートナーシップ設定の効率は向上します。リモート パートナーは、メタデータをインポートできます。また、メタデータ ドキュメントの情報に基づいてパートナーシップまたはリモート エンティティを作成できます。

既存のローカル アサーティング エンティティまたはローカル依存エンティティからメタデータをエクスポートできます。

管理 UI は、メタデータのエクスポートに対するいくつかのオプションを提供します。

- ローカル エンティティからのエクスポート。
- ローカル パートナーシップからのエクスポート。
- ローカル WSFED パートナーシップのメタデータ交換。

ファイルを使用してメタデータを送信するか、メタデータ交換プロファイルを使用してメタデータを送信するかにかかわらず、最終目的はメタデータを取得することです。

注: SAML 1.1 の場合、メタデータ ファイルの用語は、SAML 2.0 の用語です。この規則は SAML 仕様に準拠しています。SAML 1.1 データをインポートする場合、用語は SAML 1.1 の用語を使用して正確にインポートされます。

エンティティ レベルのメタデータ エクスポート

ローカル エンティティからデータをエクスポートできます。 エンティティ レベルでメタデータをエクスポートする場合は、エクスポートするデータのパートナーシップ名を指定します。 このレベルのエクスポートでは、基本的なパートナーシップ データが定義されます。

次の手順に従ってください:

1. 管理 UI へのログイン
2. ［フェデレーション］-［パートナーシップ フェデレーション］-［エンティティ］を選択します。
3. リスト内のローカル エンティティの横の［アクション］プルダウン メニューをクリックして、［メタデータのエクスポート］を選択します。

   ［メタデータのエクスポート］ダイアログ ボックスが開きます。
4. 新しいパートナーシップ名を指定します。エクスポートによって作成されたメタデータ ファイルには、基本的なパートナーシップを確立するための情報が含まれています。
5. ダイアログ ボックスの残りのフィールドに入力します。 ダイアログ ボックスの［メタデータ エクスポート オプション］セクションの設定は必ず入力してください。

   注:各フィールドの説明については、［ヘルプ］をクリックしてください。
6. ［Export］をクリックします。
7. メタデータ ファイルを開く、または保存することを要求するダイアログ ボックスが表示されます。

   表示するためにのみメタデータ ファイルを開きます。
8. ローカル システム上の XML ファイルにデータを保存します。

メタデータは指定された XML ファイルにエクスポートされます。 このファイルをどのパートナーにでも送信できます。

パートナーシップ レベルのメタデータ エクスポート

ローカル パートナーシップからデータをエクスポートできます。 このレベルのエクスポートでは、基本的なパートナーシップ データが定義されます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. ［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］を選択します。
3. リスト内のパートナーシップの横の［アクション］プルダウン メニューを選択します。
4. ［メタデータのエクスポート］を選択します。

   ［メタデータのエクスポート］ダイアログ ボックスが開きます。
5. 情報を確認します。エクスポートによって作成されたメタデータ ファイルには、基本的なパートナーシップを確立するための情報が含まれています。
6. メタデータ ドキュメントを署名し、それを検証するための［メタデータ エクスポート オプション］セクションの設定に入力します。

   注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。
7. ［エクスポート］をクリックします。
8. メタデータ ファイルを開く、または保存することを要求するダイアログ ボックスが表示されます。

   表示するためにのみメタデータ ファイルを開きます。
9. ローカル システム上の XML ファイルにデータを保存します。

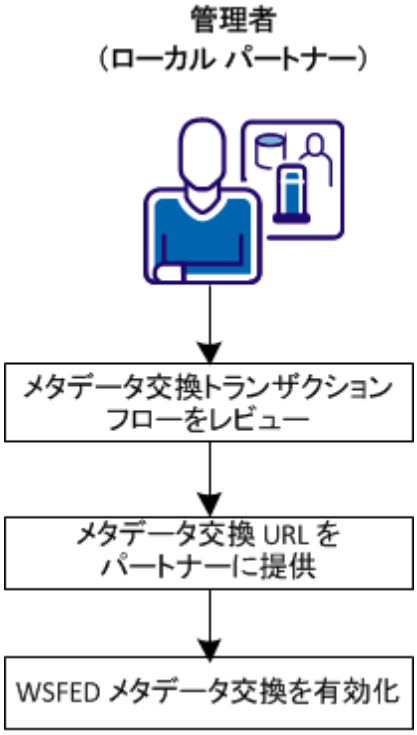
メタデータは指定された XML ファイルにエクスポートされます。 このファイルをどのパートナーにでも送信できます。

WS-フェデレーション メタデータ交換を有効にする方法

ポリシー サーバは、WS-フェデレーション パートナーシップに関して Web サービス メタデータ交換プロファイルをサポートしています。 この Web サービスは、SiteMinder のローカル パートナーを有効にして、メタデータのリモート パートナーからのリクエストに応答します。 HTTP リクエストおよびレスポンスとして、交換が発生します。

HTTP プロトコルを使用すると、リモート エンティティによってフェデレーションをプログラムで設定できます。アプリケーションは、URL を使用して必要な情報を収集できます。

以下の図は、メタデータ交換の設定手順を示しています。



メタデータ交換を実行するには、以下の設定を完了します。

1. メタデータ交換トランザクション フローを確認します。 (P. 315)
2. メタデータ交換 URL をパートナーに提供します。 (P. 315)
3. WSFED メタデータ交換を有効にします (P. 316)。

メタデータ交換トランザクション フロー

メタデータ交換トランザクションのプロセス フローは、以下のとおりです。

1. リモート パートナーは、ローカル パートナーによって提供されたメタデータ交換 URL にリクエストを送信します。
2. ローカル パートナーは、HTTP レスポンスでリモート パートナーにメタデータを送ります。ポリシー サーバは、レスポンスに署名することによってメタデータを保護します。リモート パートナーがレスポンスを確認できる証明書は、レスポンス内にあります。

   ポリシー サーバは、リクエスト時にメタデータ ドキュメントを生成します。このドキュメントは、ローカル パートナーでは格納されません。
3. リモート パートナーは、レスポンスの署名を確認します。署名が有効であると見なして、メタデータ ドキュメントを解析して情報を使用し、エンティティとパートナーシップを確立します。

パートナーへのメタデータ交換 URL の提供

メタデータ トランザクションが発生する前に、メタデータ交換リクエストの URL をリモート パートナーに提供します。 フェデレーション パートナーは、以下の URL にリクエストを送信する必要があります。

https://*server:port*/affwebservices/public/FederationMetadata/*partnership_name*

server:port

メタデータ交換サービスをホストするシステムの名前。

partnership_name

設定されたパートナーシップの名前。

WSFED メタデータ交換の有効化

ローカル WS フェデレーション パートナーでメタデータ交換機能を有効にします。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 変更する WSFED パートナーシップを選択します。
3. パートナーシップ ウィザードの［パートナーシップの設定］手順で、［メタデータ交換の有効化］チェック ボックスをオンにします。
4. ［確認］手順に移動し、［完了］をクリックします。
5. メインの［パートナーシップ フェデレーション］タブに戻ります（［フェデレーション］、［パートナーシップ フェデレーション］）。
6. 左ペインで、［メタデータ交換設定］を選択します。

   ［メタデータ交換設定］画面が表示されます。
7. レスポンスに署名するための値を指定します。
8. ［保存］をクリックします。

メタデータ交換がパートナーシップに対して設定されます。

第 22 章: トラブルシューティングに役立つログ ファイル

このセクションには、以下のトピックが含まれています。

フェデレーション トレース ロギング

フェデレーション Web サービス（FWS）のトレース ログ機能およびポリシー サーバ プロファイラは、フェデレーション サービスのパフォーマンスを監視します。 これらのロギング メカニズムは、フェデレーション操作に関する情報を提供するため、システム パフォーマンスの分析と問題のトラブルシュートが可能になります。

Web エージェント オプション パックとポリシー サーバがインストールされている場合にトレース ログ記録を有効にすると、フェデレーション プロセスに関する詳細な情報を抽出できます。 たとえば、FWSTrace.log を調べて、生成された SAML アサーションを参照する、または現在のユーザの名前を収集することができます。

注: パフォーマンスに影響を与えるおそれがあるため、トレース メッセージは通常、通常動作中にオフにされます。

収集されたトレース メッセージは 2 つのトレース ログに書き込まれます。

FWSTrace.log

FWSTrace.log は、Web エージェント オプション パックがインストールまたは展開されているアプリケーション サーバまたは Web サーバの /log ディレクトリにあります。

Web サーバ

webagent/log

webagent_optionpack/log

アプリケーション サーバ

default_deployment_directory/log

SPS フェデレーション ゲートウェイ

sps_home/secure-proxy/proxy-engine/logs

smtracedefault.log

smtracedefault.log はディレクトリ *siteminder_home*/log にあります。

siteminder_home は、製品のインストール ディレクトリを表します。

FWSTrace.log と smtracedefault.log には、トランザクション中に何が発生しているか示すチェックポイント ログ メッセージがあります。 例：

```
[13/07/30][11:34:44][4260][5824][1181adbb-993f775c-33ba08f3-76b52f
3b-3d2280cd-4ae][SSO.java][processRequest][Reading SAML 2.0 SP
Configuration [CHECKPOINT = SSOSAML2_SPCONFREAD_REQ]
```

これらのチェックポイント メッセージで検索し、トランザクション中に発生するプロセスの一部を追跡できます。

チェックポイント メッセージに加えて、ログのトランザクション ID を追跡し、トランザクションを追跡できます。 トランザクションが失敗した場合、チェックポイント メッセージとトランザクション ID を参照し、特定の問題を判断できます。

フェデレーションのトラブルシューティングに役立つトランザクション ID

多くのフェデレーション トランザクションが 1 つのログ ファイルに記録されると、それらのトランザクションのトラブルシューティングが難しくなります。 トレース ログのトランザクションを追跡するには、SAML トランザクション ID を使用します。 フェデレーション コールが発生すると、FWS アプリケーションはまず SAML トランザクション ID を生成します。 SAML トランザクション ID は、1 回のみ生成されます。この一意の SAML トランザクション ID は複数のトランザクション ID にマップできます。

たとえば、SAML 2.0 POST トランザクションにして fwstrace.log で以下のメッセージを参照できます。 太字の行が 2 つのトランザクション ID のマッピングを示していることに注意してください。

```
[13/08/01][17:33:54][2292][1884][1c2d7650-b006e46a-ed071f41-bbbede
33-fe78e2dd-38d][SSO.java][processAuthentication][SAMLTransactionI
D 2aaf90ec-fdef4897-0ef49d91-63d4031d-f508a3e9-12 maps to
TransactionID: 1c2d7650-b006e46a-ed071f41-bbbede33-fe78e2dd-38d.]
```

CA SiteMinder® Federation システムは、アサーティング パーティとして機能している場合にのみ、新しい SAMLTransactionID を生成します。 該当する場合は以下のとおりです。

- フェデレーション Web サービスがセッションを確立するために認証 URL にブラウザをリダイレクトする場合。
- 以下の HTTP-Artifact シングル サインオン トランザクションの場合。
  - アサーティング パーティが依存パーティに Artifact を送信するとき。
  - アサーティング パーティがアーティファクトを解決するとき。
- ユーザが Identity Discovery プロファイル URL にリダイレクトされる場合。
- アサーティング パーティのシングル ログアウト中。

依存パーティでは、ログ ファイルによって簡単にトレース可能なリクエスト ID が存在します。 リクエスト ID があれば、CA SiteMinder® Federation システムは、依存パーティで SAMLTransactionID を生成する必要がありません。

一意の SAML トランザクション ID ごとに、複数のトランザクション ID を生成できます。 新しい HTTP トランザクションが発生すると、新しいトランザクション ID が生成されます。 このトランザクション ID は、単一の SAML トランザクション ID にマップされます。 たとえば、トレース ログで以下のエントリを参照できます。

```
SamlTransactionID ["xyz"] maps to TransationID["123"]
["123"] HTTP operation
["123"] HTTP operation
```

新しいトランザクション ID "456" が生成されます。

```
SamlTransactionID["xyz"] Maps to Transactionid["456"]
["456"] <some operation>
["456"] <some operation>
```

トランザクション ID は fwstrace.log および smtracedefault.log に記録されます。 1 つのトランザクションに対するトランザクション ID の同じセットは、これらのログのそれぞれに書き込まれます。これらのログ内の ID を使用してトランザクションを追跡できるようになります。 失敗した場合は、ID を参照すると、トランザクションに対してどのイベントが失敗したのかを判断するのに役立ちます。

ログで単一トランザクションを追跡する方法

トランザクションを監視するには、FWSTrace.log または smtracedefault.log 内の 2 種類のトランザクション ID を追跡できます。失敗がある場合、ID を確認することにより、失敗した個所を確定するのに役立つ可能性があります。

ログ内のトランザクションを追跡するには、以下の方法を使用します。

- トレース ファイルをテキスト エディタで開き、文字列 SAMLTransactionID （スペースなし）を検索するか、特定の SAMLTransactionID を検索します。ログのエントリのこのコレクションは、エンドツーエンド トランザクション全体についての見方を提供します。 トランザクションの進行状況が分かります。

- ログ ファイル内のトランザクション ID を追跡します。 トランザクション ID は HTTP トランザクションを表します。 複数のトランザクション ID を 1 つの SAML トランザクション ID に関連付けることができます。 失敗したトランザクションについては、ブラウザにトランザクション ID が表示されます。 FWSTrace.log および smtracedefault ログでチェックポイント エラー メッセージを検索するには、表示されたトランザクション ID を使用します。
- ファイルを検索するツールでログ ファイルを解析します。 UNIX および Windows プラットフォームで、grep コマンドのようなツールを使用できます。grep コマンドを使用すると、大容量のテキスト ファイルをテキスト エディタにロードしなくても、生データを 1 行ずつ検索することができます。

  例：

  ```
  [usr@rhel632 etc]# more fwstrace.log| grep checkpoint
  [CHECKPOINT = SSOSAML2_SPCONFFROMPS_REQ]]
  [CHECKPOINT = SSOSAML2_SPCONFREAD_REQ]]
  [CHECKPOINT = SSOSAML2_SPCONFFROMCACHE_REQ]]
  [CHECKPOINT = SSOSAML2_SESSIONCOOKIEVALIDATE_REQ]]
  ```

フェデレーション サービス トレース ログ（smtracedefault.log）

プロファイラは、ログを記録するためのポリシー サーバ機能です。 プロファイラを使用してフェデレーション サービスのトレース メッセージを収集し、smtracedefault.log ファイルに書き込むことができます。

ポリシー サーバにおいてフェデレーション サービスに関するトレース メッセージを制御するコンポーネントは、Fed_Server コンポーネントです。

ポリシー サーバ プロファイラを使用すると、ポリシー サーバの内部診断と処理機能をトレースできます。

次の手順に従ってください:

1. ポリシーサーバ管理コンソールを起動します。

   重要: Windows Server 2008 上でこのグラフィカル ユーザ インターフェースにアクセスする場合は、管理者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. ［プロファイラ］タブをクリックします。

3. ［プロファイリングの有効化］オプションを設定して、プロファイリングを有効にします。
4. プロファイラの設定を選択するには、以下のいずれかを実行します。
   - ［設定ファイル］ドロップダウン リストに示されるデフォルトの smtracedefault.txt ファイルによって指定されるプロファイラ設定を受け入れます。
   - この管理セッションですでに選択されている別の設定ファイルを［設定ファイル］ドロップダウン リストから選択します。
   - ［参照］ボタンをクリックして、別の設定ファイルを選択します。
5. プロファイラの設定ファイルに格納されているプロファイラ設定を変更し、その変更内容を同じファイルまたは新しいファイルに保存するには、［環境設定］ボタンをクリックして［ポリシー サーバ プロファイラ］ダイアログ ボックスを開きます。
6. ［出力］グループ ボックスに示されている設定を調整して、ポリシー サーバ プロファイラによって生成される情報の出力形式を指定します。
7. ［適用］をクリックして、変更内容を保存します。

注:

プロファイラ設定に対する変更は自動的に有効になります。 ただし、ポリシー サーバを再起動すると、新しい出力ファイル（プロファイラでファイル出力が設定されている場合）が作成されます。 既存のプロファイラ出力ファイルは、バージョン番号と共に自動的に保存されます。 例：

```
smtracedefault.log.1
```

ロギング機能またはトレース機能の設定に対する変更がプロファイラ出力ファイルに関係がない場合（Windows でのコンソール ロギングの有効化または無効化など）、既存のファイルには新しい出力が追加され、そのバージョンは保存されません。

ポリシー サーバはデフォルトで、最大 10 個の出力ファイルを保持します（現在のファイルと 9 個のバックアップ ファイル）。 10 個のファイル制限を超えると、古いファイルは新しいファイルに自動的に置き換えられます。保持するファイルの数を変更するには、TraceFilesToKeep DWORD レジストリ設定で希望する 10 進数を指定します。TraceFilesToKeep レジストリ設定は、以下の場所で作成される必要があります。

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥
LogConfig¥TraceFilesToKeep
```

［プロファイラ］タブには［バッファト レーシング］オプションがあります。このオプションは、デフォルトでポリシー サーバのパフォーマンスを向上させるように設定されています。。 このオプションは、Solaris システムのみにあります。

フェデレーション Web サービス トレース ログ（FWSTrace.log）

トレース データの収集タスクを簡単にするために、事前設定済みの一連のテンプレートが Web エージェント オプション パックと共にインストールされます。 独自のトレース設定ファイルを作成する代わりにこれらのテンプレートを使用して、データを収集することができます。

使用可能なテンプレートは以下のとおりです。

テンプレート	収集されるトレース メッセージ
FWSTrace.conf	デフォルトのテンプレートです。ユーザ指定のデータを収集します。
FWS_SSOTrace.conf	シングル サインオン メッセージを収集します
FWS_SLOTrace.conf	シングル ログアウト メッセージを収集します
FWS_IPDTrace.conf	アイデンティティ プロバイダ ディスカバリ プロファイル メッセージの収集

すべての FWS テンプレートには、追跡される特定のデータに対応した Fed_Client コンポーネントおよびサブコンポーネントが含まれます。 正確なコンテンツを参照するには、各テンプレートを開いてください。

次の手順に従ってください:

1. web_agent または web_agent_option_pack_home/config 内のテンプレート ディレクトリに移動します。
2. テンプレートのコピーを作成し、名前を変更します。
3. （オプション）監視するデータのみが含まれるように、テンプレートを変更します。

   注: テンプレートを直接編集しないでください。
4. 新しいテンプレートを保存します。

テンプレートは、フェデレーション システムが監視するフェデレーション コンポーネントを決定します。 トレース ログ記録を有効にし、データがログ ファイルに記録される形式を指定するには、Logger.Config プロパティ ファイルを変更します。

次の手順に従ってください:

1. web_agent または webagent_optionpack_home/affwebservices/WEB-INF/classes に移動します。
2. LoggerConfig.properties ファイルを開きます。 LoggerConfig.properties ファイルには、これらすべての設定の説明が含まれています。
3. TracingOn を Yes に設定します。 このオプションは、メッセージをファイルに書き込むようにトレース機能に指示します。
4. TraceFileName 設定に、ログ ファイルの完全パスを設定します。デフォルトの場所は、*web_agent または webagent_optionpack_home*/config/FWSTrace.log です。

   注: ログ ファイルの名前は変更できます。 FWSTrace.log はデフォルトの名前です。

5. TraceConfigFile 設定に、トレース設定ファイルの完全パスを設定します。このファイルには、デフォルトのテンプレート、他のあらかじめ設定されたテンプレートの 1 つ、または独自の設定ファイルを使用できます。 どのテンプレートを指定するかにかかわらず、すべての出力は、TraceFileName 設定に指定したログ ファイルに書き込まれます。

   1 つのテンプレートのみ指定してください。すべてのテンプレートは、ディレクトリ *web_agent* または *web_agent_option_pack_home*/config に存在します。

6. 必要に応じて、トレース ログ出力ファイル内の情報の表示方法を変更できます。 以下の設定が、ログ ファイルの形式を指定します。

   - TraceRollover
   - TraceSize
   - TraceCount
   - TraceFormat
   - TraceDelim

FWS テンプレートのサンプル

以下のテキストは、FWS_SLOTrace.conf テンプレートからの抜粋です。 ほとんどのファイルにはコメントが含まれるほか、ファイル、コマンド構文、および Fed_Client コンポーネントに対して利用可能なサブコンポーネントの使用方法についての指示が含まれています。

この抜粋では、Fed_Client コンポーネントおよび監視されるサブコンポーネント（Single_Logout および Configuration）が示されています。 また、各メッセージの必要なコンテンツ（日付、時刻、Pid、Tid、TransactionId、SrcFile、機能、メッセージ）を示すデータ フィールドも示されています。

```
components: Fed_Client/Single_Logout, Fed_Client/Configuration
data: Date, Time, Pid, Tid, TransactionID, SrcFile, Function, Message
```

第 23 章: オープン フォーマット Cookie の詳細

フェデレーション オープン形式の Cookie によって、アプリケーションは SiteMinder に対してユーザ属性を保証し、SiteMinder がカプセル化するユーザ属性を消費します。 オープン形式 Cookie には以下の一般的特性があります。

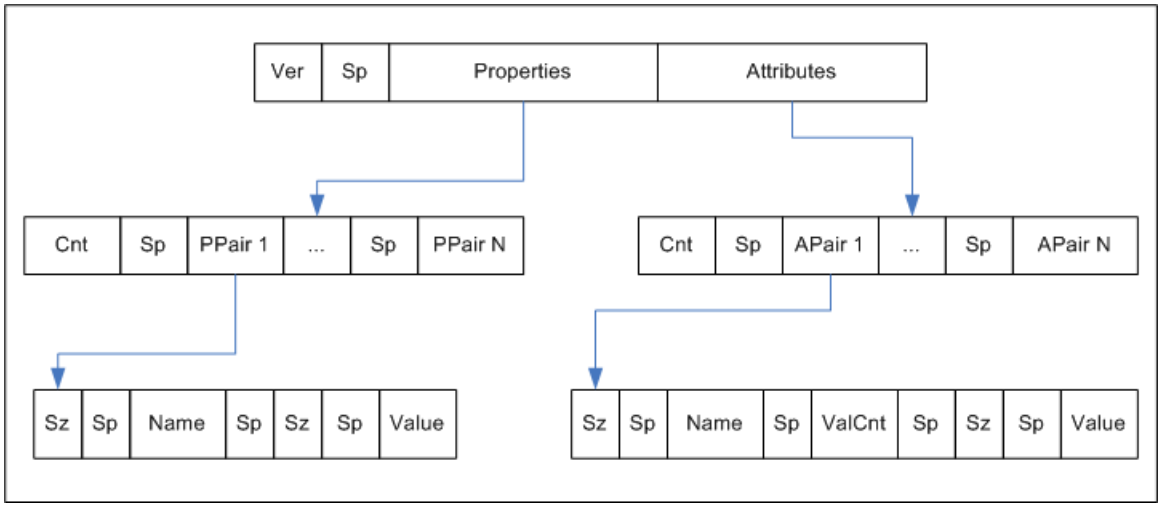
- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

重要: Cookie に「=」などのような安全でない文字が含まれる場合は、二重引用符でその値を囲んでください。 ユーザ インターフェース、または SDK によってこのオプションを指定できます。

オープン形式 Cookie には以下のプロパティ情報が含まれます。

- Cookie バージョン
- 名前 ID
- 名前 ID 形式
- セッション ID
- AuthnContext
- UserDN （ユーザ ID と同じ）

以下の図はオープン形式を表しています。



キー：

- Ver -- Cookie フォーマット バージョン。CA SiteMinder® Federation r12.1 の場合、この値は 1 です。
- Sp - 読みやすくするためにのみ使用される ASCII スペース文字。
- プロパティ - プリンシパルに関する情報。
- 属性 -- アサーションからの SAML 属性
- Cnt - ASCII で表される後続の名前値ペアの数。
- Sz -- 次に続く名前または値の長さ
- ValCnt -- 次に続く属性値の数。CA SiteMinder® Federation r12.1 については、属性に対する複数の値はサポートされていません。 この値を 1 に設定します。

このフォーマットのバッカス・ナウア記法（BNF）は以下の通りです（0* が 0 以上、1* が少なくとも 1 を意味します。）

- DIGIT = ASCII 数字（0 ～ 9）
- CHAR　= UTF-8 文字
- Sp = ASCII スペース（文字 32）
- トークン = 1*CHAR
- Cookie = バージョン Sp プロパティ属性
- バージョン = 1*DIGIT

- Cnt = 1*DIGIT
- プロパティ = Cnt 1*PPair
- 属性 = Cnt 0*APair
- ValCnt = 1*DIGIT
- PPair = Sz Sp 名前 Sp Sz Sp 値
- APair = Sz Sp 名前 Sp ValCnt Sp Sz Sp 値
- Sz = 1*DIGIT
- 名前 = トークン

値 = トークン

オープン形式の Cookie のコンテンツ

フェデレーション オープン形式の Cookie によって、アプリケーションは SiteMinder に対してユーザ属性を保証し、SiteMinder がカプセル化するユーザ属性を消費します。 オープン形式 Cookie には以下の一般的特性があります。

- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

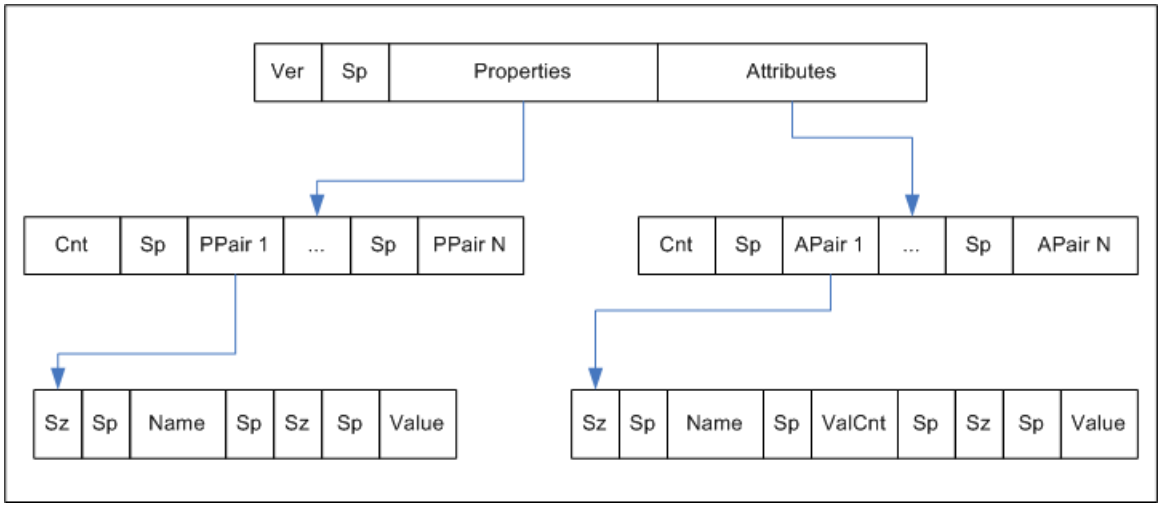
重要: Cookie に「=」などのような安全でない文字が含まれる場合は、二重引用符でその値を囲んでください。 ユーザ インターフェース、または SDK によってこのオプションを指定できます。

オープン形式 Cookie には以下のプロパティ情報が含まれます。

- Cookie バージョン
- 名前 ID
- 名前 ID 形式

- セッション ID
- AuthnContext
- UserDN （ユーザ ID と同じ）

以下の図はオープン形式を表しています。



キー：

- Ver -- Cookie フォーマット バージョン。CA SiteMinder® Federation r12.1 の場合、この値は 1 です。
- Sp - 読みやすくするためにのみ使用される ASCII スペース文字。
- プロパティ - プリンシパルに関する情報。
- 属性 -- アサーションからの SAML 属性
- Cnt - ASCII で表される後続の名前値ペアの数。
- Sz -- 次に続く名前または値の長さ
- ValCnt -- 次に続く属性値の数。CA SiteMinder® Federation r12.1 については、属性に対する複数の値はサポートされていません。 この値を 1 に設定します。

このフォーマットのバッカス・ナウア記法（BNF）は以下の通りです（0* が 0 以上、1* が少なくとも 1 を意味します。）

- DIGIT = ASCII 数字（0 ～ 9）
- CHAR　= UTF-8 文字
- Sp = ASCII スペース（文字 32）

- トークン = 1*CHAR
- Cookie = バージョン Sp プロパティ属性
- バージョン = 1*DIGIT
- Cnt = 1*DIGIT
- プロパティ = Cnt 1*PPair
- 属性 = Cnt 0*APair
- ValCnt = 1*DIGIT
- PPair = Sz Sp 名前 Sp Sz Sp 値
- APair = Sz Sp 名前 Sp ValCnt Sp Sz Sp 値
- Sz = 1*DIGIT
- 名前 = トークン
- 値 = トークン

付録 A: 暗号化および復号アルゴリズム

このセクションには、以下のトピックが含まれています。

オープン形式の Cookie 暗号化アルゴリズム

オープン形式の Cookie は、パスワード ベースの暗号化に対して以下のオプションをサポートしています。

FIPS_Compat モードおよび FIPS_Migration モード

- PBE/SHA1/AES/CBC/PKCS12PBE-1000-128
- PBE/SHA1/AES/CBC/PKCS12PBE-1000-192
- PBE/SHA1/AES/CBC/PKCS12PBE-1000-256
- PBE/SHA256/AES/CBC/PKCS12PBE-1000-128
- PBE/SHA256/AES/CBC/PKCS12PBE-1000-192
- PBE/SHA256/AES/CBC/PKCS12PBE-1000-256
- PBE/SHA1/3DES_EDE/CBC/PKCS12PBE-1000-3
- PBE/SHA256/3DES_EDE/CBC/PKCS12PBE-1000-3

FIPS_Only モード

- AES128/CBC/PKCS5Padding
- AES192/CBC/PKCS5Padding
- AES256/CBC/PKCS5Padding
- 3DES_EDE/CBC/PKCS5Padding

デジタル署名および秘密キー アルゴリズム

SiteMinder は、パートナーシップ署名オプション用に以下のアルゴリズムを使用します。

暗号化キー アルゴリズム

RSA-V15、RSA-OEAP

暗号化ブロック アルゴリズム

3DES、AES-128、AES-256

SiteMinder は秘密キー（証明書/キー）の生成に以下のアルゴリズムを使用します。

キー アルゴリズム

RSA

署名アルゴリズム

MD5withRSA、SHA1withRSA、SHA256withRSA および SHA512withRSA

バック チャネル通信アルゴリズム

HTTP-Artifact シングル サインオンおよび SAML 2.0 シングル ログアウトに関連するバック チャネル通信の場合、SiteMinder は FIPS モードに応じて、以下の暗号をサポートします。

FIPS_Compat モードおよび FIPS_Migration モード - RC4 および AES

RSA_With_RC4_SHA

RSA_With_RC4_MD5

RSA_With_AES_128_CBC_SHA

RSA_With_AES_256_CBC_SHA

FIPS_Only モード - AES のみ

RSA_With_AES_128_CBC_SHA

RSA_With_AES_256_CBC_SHA

Java SDK 暗号化アルゴリズム

CA SiteMinder® Federation Java SDK は以下の暗号化アルゴリズムをサポートします。

パスワードなし

「AES/CBC/PKCS5Padding」

パスワードあり

「PBE/SHA1/AES/CBC/PKCS12PBE-5-128」

暗号アルゴリズム

FMCrypto 暗号化/復号アルゴリズム

AES_128