CA SiteMinder Federation Standalone

Federation Standalone ガイド

r12.52 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

以下の更新は、SiteMinder の以前のリリース内に見つかった問題の結果として、12.52 のドキュメントに対して行われました。

- 署名検証失敗の解決 (P. 495) -- 署名検証が失敗した場合に備えて、XML 署名検証の無効化に関する情報が追加されました。 CQ 168095 を解決（STAR イシュー番号： 21321479;1）
- フェデレーション アクティビティを監視するためのログ (P. 459) -- server.log ファイルでの log4j の使用を反映して、ログ記録に関する情報が更新されました。 また、すべてのログ情報が 1 つのセクションに統合されました。 CQ 171956（STAR イシュー番号： 21454409-1 および CQ 165412 （STAR イシュー番号： 21257428-1）を解決
- 既存ファイルからのキー/証明書ペアのインポート (P. 153) -- [CA として使用] オプションのアクションの説明が追加されました。CQ 173083 を解決
- FEDSESSION Cookie タイムアウト設定 (P. 229) -- Idle および Max Timeout 設定が修正されました。 CQ 173107 を解決
- 既存設定のバックアップ (P. 488) -- XPSExport コマンド オプション -xa は廃止されました。このコマンドは -xe -xp コマンドに置き換えられます。 CQ 173659 を解決（STAR イシュー番号： 21480783-2）
- 委任認証用のクエリ文字列方式 (P. 287) -- クエリ文字列方式が実稼働環境でのみ使用可能であることを示す注が追加されました。 CQ 165470 および CQ 165473 を解決。
- SSL を介して LDAP ユーザ ディレクトリに接続する方法 (P. 95) -- cert7.db ファイルから cert8.db ファイルに更新されました。 CQ 172315 を解決（STAR イシュー番号： 21454358-01）
- 負荷分散 (P. 417)およびフェールオーバ サポート (P. 403) -- これらの機能に対する設定およびプロセスが明確化され図が更新されました。 CQ 145146 を解決（STAR イシュー番号： 20533073;1）

目次

第 1 章: はじめに

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

エンタープライズ アプリケーションやサービスでは、あるドメインから別のドメインのサービスに安全かつシームレスにアクセスする必要があります。 CA SiteMinder® Federation Standalone は、ID 情報を柔軟かつポータブルにすることによって、信頼されたビジネス 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。 スタンドアロン製品として、ターゲット システム上でフェデレーション ソフトウェアを必要としません。

CA SiteMinder® Federation Standalone は、以下の機能をサポートします。

- SAML 1.1
- SAML 2.0
- WS-protocol サポート。
- FIPS 140-2 と互換性のある暗号化のサポート。

- 製品は、フェデレーションおよびアクセス制御のためにスタンドアロン エンティティとして、または SiteMinder Web アクセス マネージャと共に展開できます。
- サイトは、アサーティング パーティおよび依存パーティとして機能できます。
- 暗号化された Cookie またはヘッダとして ID データをターゲット アプリケーションに渡す機能。

以下のフローチャートでは、製品を持ったフェデレーションを設定する一般的なプロセスを示します。


Federation
Standalone
Oracle または SQL Server データベース JDK Federation Standalone 製品のインストール
ユーザ ディレクトリ接続の確立
SAML/WSFED プロファイルの決定
サインイン/検証、暗号化/復号化用キーおよび証明書のインポート
ローカルおよびリモート エンティティの設定
このサイトはアサーティング パーティ、依存パーティ、またはその両方ですか?
アサーティング
依存
アサーティング パーティ パートナーシップの設定
依存パーティ パートナーシップの設定
SSO 起動リンクの作成

製品コンポーネント

CA SiteMinder® Federation Standalone には以下のコンポーネントが含まれています。

- 安全なプロキシ エンジン

  バックエンド サーバにトラフィックを転送します。 このエンジンは、Web サーバ、サーブレット エンジン、プロキシ サーバ、およびフェデレーション Web サービス機能を使用します。

  セキュア プロキシ エンジンには、以下のコンポーネントが含まれます。

  - Apache Web サーバ

    HTTP リスナとして働き、適切に設定すれば、受信要求用 HTTP トラフィックを処理します。HTTPS トラフィックを処理することもできます。

  - Tomcat サーバ

    Administrative UI の操作のためのサーブレット コンテナを提供します。 Apache Web サーバは、mod_jk という名前の Tomcat コネクタを介して Tomcat サーバに通信します。

- フェデレーション サーバ

  ユーザ ディレクトリ接続性、認証機能、およびセッション ストア機能を有効にします。

- 拡張可能なポリシー ストア

  CA SiteMinder® Federation Standalone データ オブジェクトをすべて格納します。

- Web ベースのユーザ インターフェース

  フェデレーション エンティティとパートナーシップの設定、秘密キーと証明書、およびさまざまなサーバ設定を管理します。

CA SiteMinder® Federation Standalone によって提供される FIPS 140-2 サポート

CA SiteMinder® Federation Standalone は、米国連邦情報処理規格（FIPS）140-2 準拠と認定された暗号ライブラリを使用します。 環境が FIPS に準拠した Advanced Encryption Standard（AES）アルゴリズムのみを使用して機密データを暗号化する場合、これらのライブラリは FIPS 操作モードを提供します。

以下のいずれかの FIPS 操作モードで、製品をインストールすることができます。

FIPS_COMPAT

FIPS_COMPAT（互換性）モードは、インストール中のデフォルトの FIPS 操作モードです。 FIPS_COMPAT モードでは、システムは、サポートされている FIPS 準拠アルゴリズムと共に非 FIPS アルゴリズムの現在のセットも引き続きサポートします。

FIPS_COMPAT モードは、旧バージョンの製品と互換性があります。 この互換性により、12.0 SP1 よりも前のバージョンの環境が現在のバージョンと相互運用できるようになります。 また、FIPS_COMPAT は、現在の製品実装で使用可能なセキュリティの程度に満足しているすべてのライアントにとっても適切です。

組織が FIPS の使用を必要としない場合は、FIPS_COMPAT モードで製品をインストールします。 追加設定は必要ありません。

FIPS_ONLY

FIPS_ONLY モードでは、環境は FIPS 準拠アルゴリズムのみを使用して機密データを暗号化します。

新しいインストールで FIPS 準拠アルゴリズムのみを使用する場合は、FIPS_ONLY モードで製品をインストールします。

このガイドの付録 (P. 507)には、さまざまな FIPS モードで動作する際にシステムが使用する特定の暗号化および復号化アルゴリズムのリストが示されています。

重要: FIPS_ONLY モードで実行されている r12.52 SP1 インストールは、製品で表示される以前のすべてのバージョンの API を含め、以前のバージョンの製品と相互運用できないか、または後方互換性がありません。 完全な FIPS_ONLY モードのための必要なサポートを得るために、そのようなすべてのソフトウェアを各 SDK の r12.52 SP1 バージョンと再リンクします。

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、ユーザの明確化、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 CA SiteMinder® Federation Standalone 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。 データをセキュリティ保護にするために、CA SiteMinder® Federation Standalone 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 は、CA SiteMinder® Federation Standalone 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リクエストに対するレスポンスをセキュリティ保護にします。この Cookie は、Perl や Ruby など、オープン形式の Cookie が使用するのと同じ暗号化および復号化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。

また、CA SiteMinder® Federation Standalone SDK は オープン形式の Cookie をサポートし、アプリケーションの混在を許可します。

以下の CA SiteMinder® Federation Standalone 機能は、プログラマなしフェデレーション モデルを実装しています。

分散代行認証

分散代行認証によって、CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone にフェデレーション ユーザ ID を送信します。

委任された認証用の通信は、HTTP/HTTPS リクエストおよびレスポンスによって処理されます。

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。 CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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。

プロビジョニング用の通信は、HTTP/HTTPS リクエストおよびレスポンスによって処理されます。

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。

- Basic SAML の基礎
- SAML の POST バインディングおよび Artifact 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 ECP（機能強化クライアントまたはプロキシ）などの SAML 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

本書では以下の用語を使用します。

アサーティング パーティ

依存パーティが使用するアサーションを生成する SAML 認証局。 アサーティング パーティは、ユーザの識別情報の作成、維持、および管理を行い、他の依存パーティにユーザ認証を提供します。 SAML 1.1 では、アサーティング パーティはプロデューサと呼ばれます。 SAML 2.0 および WS-フェデレーションでは、アサーティング パーティはアイデンティティ プロバイダと呼ばれます。

アサーション コンシューマ サービス（SAML 1.1 および 2.0）

SAML Artifact または埋め込み SAML レスポンスを含む HTTP フォームを受信し、対応する SAML アサーションを取得する サービス プロバイダ コンポーネント。 アサーション コンシューマ サービスは、フェデレーション セッション Cookie を発行します。また、SiteMinder と統合している場合は SiteMinder セッション Cookie を発行します。

アサーション検索サービス（SAML 1.1）

HTTP Artifact バインディングを使用して SAML 1.1 認証を処理するプロデューサ側のサービス。 このサービスは、プロデューサに格納されたアサーションを取得します。

Artifact **解決サービス**（SAML 2.0）

HTTP Artifact バインディングを使用して SAML 2.0 認証を実行するアイデンティティ プロバイダ側のサービス。 このサービスは、アイデンティティ プロバイダに格納されたアサーションを取得します。

認証リクエスト サービス（SAML 2.0）

サービス プロバイダがクロスドメイン シングル サインオン用の認証リクエスト メッセージを生成できるようにするサービス。 このメッセージには、CA SiteMinder® Federation Standalone がアイデンティティ プロバイダのシングル サインオン サービスにブラウザをリダイレクトできるようにする情報が含まれます。 認証リクエスト サービスは、POST および Artifact バインディングを使用するシングル サインオンに使用されます。

注: サービスによって発行される認証リクエスト メッセージのフォーマットは、OASIS の SAML（Security Assertion Markup Language） V2.0 用プロファイルで指定されます。

分散代行認証

サードパーティ Web アクセス管理システムの使用を可能にして、ユーザを認証してから CA SiteMinder® Federation Standalone にユーザをリダイレクトして、フェデレーション プロセスを続行する機能。

レガシー Cookie

（以前は FEDPROFILE Cookie）ユーザ ID 情報が含まれる Cookie。 この Cookie は PBE 暗号化アルゴリズムのみをサポートします。このアルゴリズムは FIPS に準拠していません。

アサーティング パーティで、Java SDK はレガシー Cookie を作成し、CA SiteMinder® Federation Standalone はそれを読み取ります。 依存パーティで、CA SiteMinder® Federation Standalone は、Java ベースのエンドユーザ アプリケーションで使用するためのレガシー Cookie を作成します。 これらのアプリケーションは、Java SDK を使用してこの Cookie を読み取ります。

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 Standalone 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、CA SiteMinder® Federation Standalone 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 Standalone 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 Standalone.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。

依存パーティ

SAML 認証局からの情報を使用してサービスへのアクセス権を提供する SAML エンティティ。 依存パーティは、アサーティング パーティから取得するアサーションを使用してユーザを認証します。 SAML 1.1 では、依存パーティはコンシューマと呼ばれます。 SAML 2.0 では、依存パーティはサービス プロバイダと呼ばれます。

重要: 本書では、「*依存パーティ*」という用語はコンシューマまたはサービス プロバイダを指して使用されます。

シングル ログアウト サービス（SAML 2.0）

このサービスにより、ユーザは単一のログアウト イベントでフェデレーション内のすべてのアプリケーションから同時にログアウトできます。 アイデンティティ プロバイダまたはサービス プロバイダでシングル ログアウトを開始できます。

シングル サインオン サービス（SAML 1.1 および SAML 2.0）

SAML 1.1 の場合、SSO サービスにより、プロデューサはフェデレーション リソースに対するプロデューサの要求を処理できます。

SAML 2.0 の場合、SSO サービスにより、アイデンティティ プロバイダはフェデレーション リソースに対する IdP または SP の要求を処理できます。

プロデューサ/IdP は、コンシューマ/SP から必要な情報を収集してアサーションを生成し、コンシューマ/SP にそれを渡します。 その後、コンシューマ/SP はアサーションを認証に使用します。

統一表現言語

統一表現言語（UEL）とは、主に Java Web アプリケーションで使用される特殊な 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。 CA SiteMinder® Federation Standalone 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。

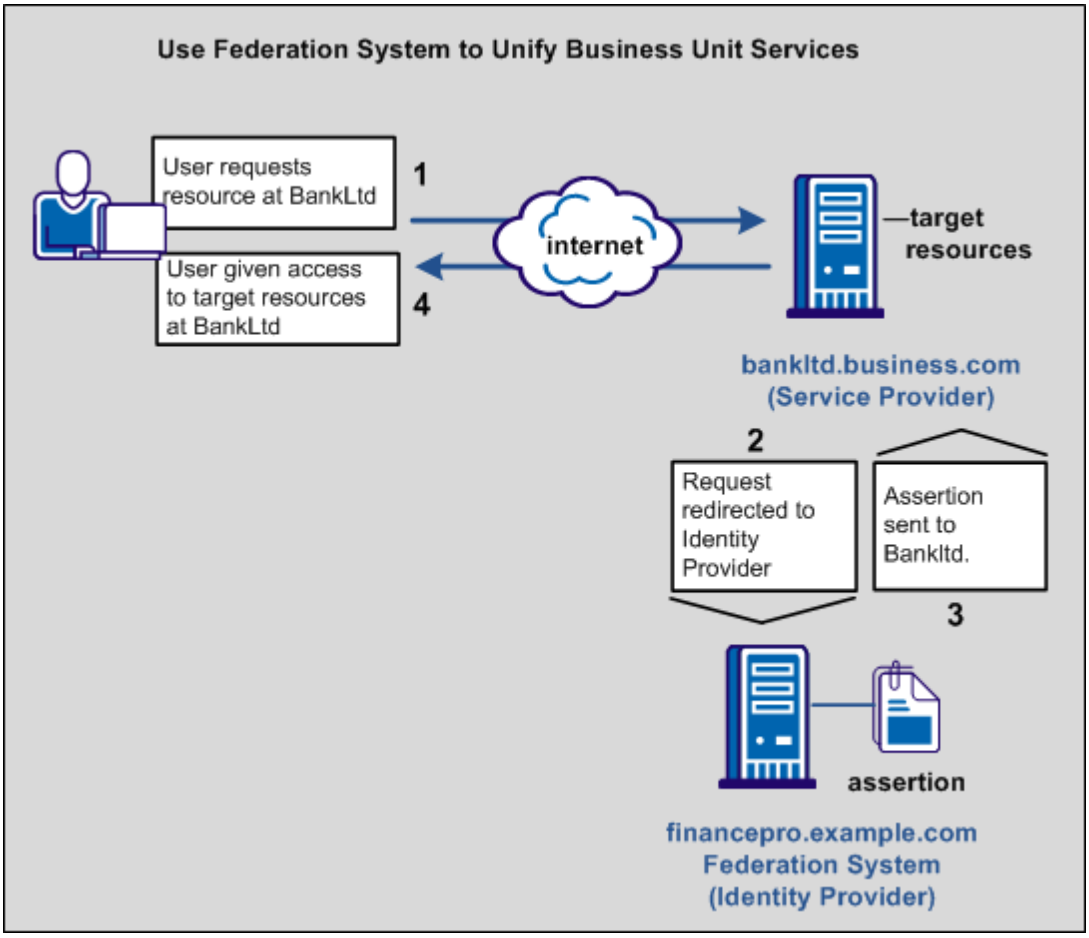
ユーザのエンタープライズでのフェデレーション

サンプル ビジネス ケースは、フェデレーション システムが一般的なビジネスの問題を解決できる方法を最適に示しています。

このビジネス ケースで、Financepro は、クライアントにプライベート バンキングを提供するために最近 BankLtd 銀行を買った投資コンサルタントです。 これらの 2 つの会社には異なる情報インフラがあります。しかし、これを顧客の目には 1 つの会社と映るようにしたいと考えています。この問題を解決するために、彼らはフェデレーション パートナーシップを築きました。

フェデレーション関係を確立することにより、2 つの会社はシングル サインオンを使用して、顧客にシームレスな操作性を提供できます。 顧客は何度も認証画面が表示されることなく、Financepro と BankLtd の間を行き来できます。 さらに、顧客 ID および顧客情報の共有はユーザの操作性をいっそうカスタマイズし、各パートナーの金融商品の販売促進を相乗的に行うことができます。

以下の図は、Financepro と BankLtd 間のフェデレーション パートナーシップを示しています。通信の流れは SAML 2.0 サービス プロバイダにより開始されるシングル サインオンに基づいています。



この図では、以下の情報の流れについて説明します。

1. ユーザが、BankLtd でフェデレーション リソースにアクセスしようとします。
2. このユーザは認証のために Financepro にリダイレクトされ、また、アサーションが生成されます。
3. アサーションは BankLtd に渡されます。
4. SAML HTTP-Artifact または HTTP-POST のいずれかに基づいてシングル サインオンが発生します。ユーザはターゲット リソースにアクセスします。

このパートナーシップが機能するには、CA SiteMinder® Federation Standalone を使用して関係を実装する前に、パートナーシップがどのように機能するかを決定します。

検討すべき問題には次のものがあります。

- ユーザがパートナーシップにおいて識別される方法。
- アサーションで送信する属性とその目的。
- 使用するフェデレーション バインディング（SAML または WS-フェデレーション）。

ユーザの決定は、ビジネス パートナーシップの構築を支援します。

パートナーシップにおけるユーザ識別

取引先企業にはそれぞれのユーザ ストアでユーザ ID を定義する独自の方法があります。 ユーザがどのように識別されるかで、ある提携先が別の提携先にそのユーザをマップできる方法が決まります。

次のようなシナリオを考慮する必要があります。

- ユーザ ID が各サイトのユーザ ストアで同じである。

  アカウント リンクがユーザの識別法です。

- ユーザ ID が各サイトのユーザ ストアで一意である。

  ID マッピングがユーザの識別法です。 FinancePro で顧客は JohnDoe と識別されますが、BankLtd ではこの同じ顧客が DoeJ と識別されます。パートナーは、ID マッピングに使用するユーザ属性プロファイルに同意する必要があります。

- ユーザ ID が依存側に存在しない。

  アカウント プロビジョニングがユーザの識別法です。アカウントのプロビジョニングには、ユーザ アカウントの作成が必要な場合や、単純に既存のユーザ アカウントへ SAML アサーションの情報を入力することが必要な場合があります。

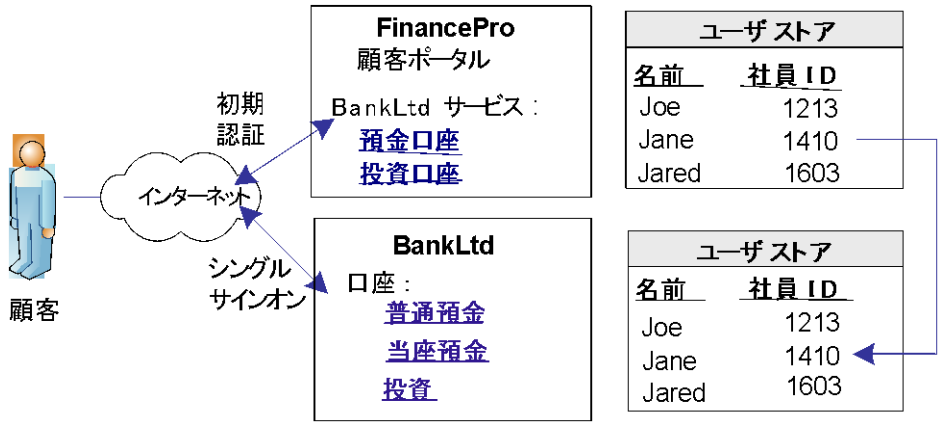
ユーザの識別法を決定することで、アサーションでどんな情報がユーザ ID として送信されるかが決まります。

フェデレーション ID を確立するアカウント リンク

FinancePro の顧客が BankLtd のリソースにアクセスする場合は、アサーションに必ず名前 ID があります。 この識別子によって、BankLtd はその顧客が誰か、また、その顧客に対して許可するアクセス レベルを決定できます。

各提携先のユーザ ストアが、同じ ID を使用する同じ方式でユーザを識別したときに、名前 ID はフェデレーション ID を確立できます。

次の図は同じ社員 ID を使用した各サイトのユーザ ストアを示しています。



CA SiteMinder® Federation Standalone では、パートナーシップ設定プロセスの一部としてアカウント リンクを設定できます。 ユーザは名前 ID の形式および名前 ID のタイプを指定します。 これにより、名前を定義する値のタイプが決まります。 特定の名前 ID タイプを、静的属性、ユーザ属性、またはユーザ ディレクトリの DN 属性と関連付けます。 CA SiteMinder® Federation Standalone によりアサーションに組み込まれる名前 ID は、ユーザが定義する設定に一致します。

依存パーティがアサーションを受信すると、BankLtd ではユーザの特定プロセスが発生します。 このプロセスは、アサーションの名前 ID 値をそのユーザ ストアのレコードにリンクします。

フェデレーション ID を確立する ID マッピング

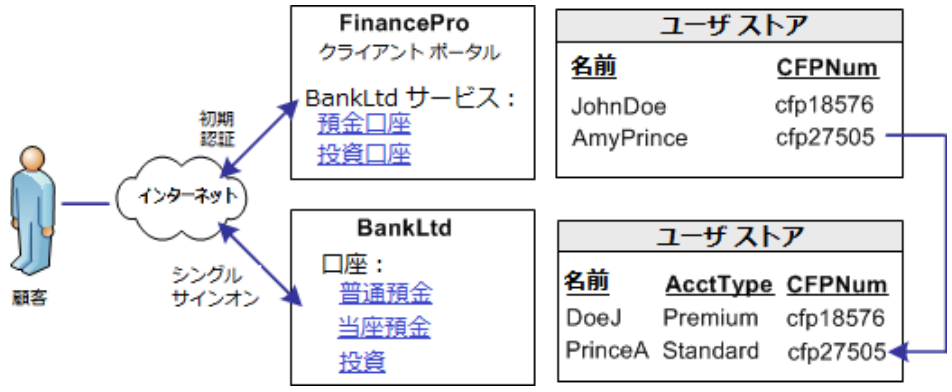
Financepro の投資者が認証を行い、BankLtd のアクセス情報へのリンクを選択します。この投資者はサインオンしなくても BankLtd Web サイトのアカウント領域に直接移動します。

BankLtd は、Financepro のすべての顧客に対してユーザ ID を保守しますが、BankLtd の ID は FinancePro での ID と異なります。 たとえば、FinancePro では JohnDoe は顧客です。 BankLtd では、この同じ顧客は DoeJ として識別されます。 いずれにせよ、BankLtd は会社の Web サイトの機密部分に対するアクセスを制御する必要があります。 フェデレーション ID を確立するために、両社はどちらのサイトでも 1 人の顧客に対して適切な ID にマップする属性に合意します。

両社は、帯域外の情報交換中に使用する属性に関して合意します。これは、この合意がチャネルを介した任意のメッセージの任意の通信の一部ではないことを意味します。 この例の場合、両社が合意した属性は、公認投資コンサルタント認可番号（各ユーザ ストアの CFPNum）です。

顧客が BankLtd でフェデレーション リソースへのアクセスを試行すると、その要求がシングル サインオン プロセスのトリガになります。FinancePro で生成されるアサーションには CFPNum 属性が含まれます。 BankLtd がアサーションを受信するときに、そのサイトのアプリケーションはユーザ明確化プロセスを実行する必要があります。 どのプロファイル ID を要求に使用するかをプロセスが決定するのは、属性に依存します。

次の図は、同じユーザが各社でどのように違って識別されるかを示しています。



SiteMinder フェデレーションでは、パートナーシップ設定プロセスの一部として ID マッピングを設定できます。 名前 ID および属性の設定について、CFPID と呼ばれる属性を定義します。 この属性をユーザ属性 CFPNum （各社のユーザ ストアの属性の名前）と関連付けます。

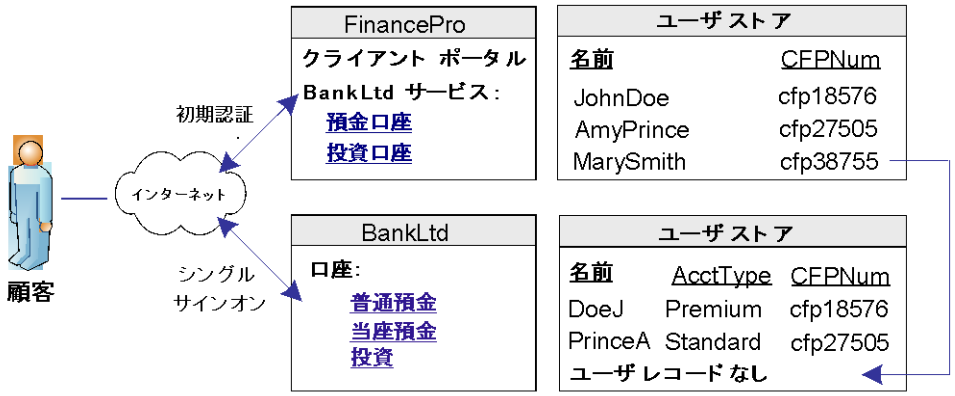
SiteMinder Federation はアサーションに属性を組み込みます。BankLtd がアサーションを受信すると、ユーザ明確化プロセスはアサーションの属性をそのユーザ ストアの適切なレコードにリンクします。

フェデレーション ID を確立するためのユーザ プロビジョニング

Financepro の投資家、メアリー スミスが認証を行い、リンクをクリックして BankLtd の情報にアクセスします。BankLtd では最初、メアリー スミスのユーザ アカウントを見つけられません。 BankLtd は、新しい顧客を許可する一方で、Web サイトの機密部分は保護したいと考えます。

BankLtd は、メアリー スミスの新しいフェデレーション ID を確立するプロビジョニングを実装するように CA SiteMinder® Federation Standalone を設定しました。CA SiteMinder® Federation Standalone は BankLtd のプロビジョニング サーバにメアリー スミスをリダイレクトします。プロビジョニング アプリケーションは、CA SiteMinder® Federation Standalone からの ID 情報を使用して、ユーザ ストアにユーザ アカウントを作成します。

次の図は、FinancePro と BankLtd のユーザ ストアを示しています。



CA SiteMinder® Federation Standalone では、依存パーティでパートナーシップ設定の一部としてプロビジョニングを設定できます。 この例で、ユーザはリモート プロビジョニングを選択し、アサーション データを BankLtd のプロビジョニング サーバに届ける方法を決定します。この設定により、ユーザ ストアでユーザ エントリを動的に作成できるようになります。

アプリケーションをカスタマイズするための属性

CA SiteMinder® Federation Standalone は、ターゲット アプリケーションをカスタマイズするために属性を使用する 2 つの方法を提供しています。

アサーティング パーティのアサーションに追加された属性

アプリケーションをカスタマイズする目的でユーザを識別するために、アサーションにユーザ ストア レコードの属性を含めることができます。

サーブレット、Web アプリケーションおよび他のカスタム アプリケーションは、カスタマイズされたコンテンツを表示したり、他のカスタム機能を有効あるいは無効にするために、属性を使用できます。 属性を Web アプリケーションと共に使用すると、ターゲット サイトでのユーザ アクティビティを制限することにより、きめの細かいアクセス制御を実装できます。 たとえば、Account Balance という名前の属性変数を送信し、これに、BankLtd のユーザの口座保有高を反映させるように設定します。

属性の形式は、名前/値のペアになっています。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。

依存パーティでの属性マッピング

依存パーティは一連のアサーション属性を受信します。この属性を、ターゲット アプリケーションに配信される一連のアプリケーション属性にマップできます。

たとえば、FinancePro にはアサーション属性 CellNo=5555555555 が含まれます。 BankLtd で、この属性名がアプリケーション属性 Mobile=5555555555 に変換されます。 属性名は変換されますが、値は同じままです。

複数のアサーション属性も単一のアプリケーション属性に変換できます。たとえば、FinancePro は、属性 Acct=Savings および Type=Retirement を持つ受信アサーションを送信し、BankLtd で FundType= Retirement Savings へ変換しました。

詳細情報:

パートナーシップの作成およびアクティブ化 (P. 179)

シングル サインオン用のフェデレーション プロファイル

パートナーシップのプロファイルの決定は、各サイドがサポートできるバインディングに左右されます。

新しい連携では、どちらの会社にもレガシー要件がありません。したがって、シングル サインオンに使用する推奨プロファイルは SAML 2.0 POST プロファイルです。 SAML 2.0 POST プロファイルは、アサーション データの安全な転送を提供します。また、設定プロセスは SAML Artifact プロファイルより単純です。 ただし、2 社間の契約により SAML Artifact が必要な場合は、このバインディングも実装できます。

パートナーシップ モデル

CA SiteMinder® Federation Standalone パートナーシップ モデルは、各会社のサイト間を移動する際の操作性を緩和し、それらが 1 つの会社のように見えるように、Financepro と BankLtd の間にフェデレーションを確立できます。

Administrative UI は、シングル サインオンを実行するために、パートナーシップの作成およびパートナーシップの両関係者を識別することに焦点を当てます。

これらの手順には次のものがあります。

1. パートナーシップの設定 -- パートナーシップに名前を付け、そのパートナーシップを構成する 2 つのエンティティを識別します。
2. フェデレーション ユーザ/ユーザ識別の確立 -- アサーティング パーティがアサーションを生成し、依存パーティが認証するユーザを指定します。
3. 名前 ID および属性 -- フェデレーション ID を確立する方法を決定し、識別する属性の追加とアサーション内容のカスタマイズを可能にします。

   名前 ID および属性を使用すると、依存パーティで適切な情報がアプリケーションに利用可能かどうかを確認できます。 これは、アカウント リンクおよび ID マッピングが設定される場所です。

4. SSO -- 名前 ID でアサーションを消費するサービスの場所をはじめとする、シングル サインオン（Artifact または POST バインディング）を定義します。 SAML 2.0 については、シングル ログアウト（SLO）、機能強化クライアントまたはプロキシ（ECP）プロファイル、およびアイデンティティ プロバイダ ディスカバリ プロファイルなどの追加機能を設定できます。
5. 署名および暗号化 -- 安全なアサーション交換、認証リクエスト、および SAML 2.0 シングル ログアウト リクエストおよびレスポンスのための署名および暗号化オプションを定義します。
6. アプリケーション統合 -- ユーザによるターゲット アプリケーションへのリダイレクト設定を可能にし、ユーザ レコードのプロビジョニング設定と依存パーティの属性マッピングを定義できるようにします。また、ユーザ認証失敗時のリダイレクトを設定できます。

第 2 章: Administrative UI

このセクションには、以下のトピックが含まれています。

Administrative UI の概要

設定は Administrative UI によって管理されます。Administrative UI は、製品のすべてのシステム管理機能へのアクセス権を管理者に提供する Web アプリケーションです。

管理者は、フェデレーション ソリューションを管理する権限および責任を持つユーザです。 複数の管理者がシステムの管理に責任を負うことができます。 複数の管理者を設定するための手順を確認します。

Administrative UI は、設定タブ、サブカテゴリ、リスト、およびタスク ボタンに編成されます。

UI を移動する際には、以下の点に注意してください。

- 主な設定カテゴリは、以下のタブによって反映されます。
  - フェデレーション
  - 証明書 & キー
  - ユーザ ディレクトリ
  - インフラストラクチャ

  オブジェクトを設定する場合は、まずこれらのタブの 1 つに移動します。

- 主な各設定タブのサブカテゴリを選択すると、フェデレーション セットアップの特定の特徴を設定できます。
- ほとんどのサブカテゴリについて、オブジェクトのリストが表示されます。 これらのリストには、既存のオブジェクトにアクセスするためのコンテキスト リンク、およびオブジェクトを作成または変更するためのタスク ボタンが含まれます。

オブジェクト管理

Administrative UI では、オブジェクトを作成、表示、変更、および削除できます。 各タスクの詳細はオブジェクトによって異なりますが、全般的な方法は似ています。 たとえば、フェデレーション エンティティを削除する手順は、ユーザ ディレクトリ接続を削除する手順に似ています。

各サブカテゴリ内のリストを使用すると、オブジェクトを操作できます。以下のいずれかを実行できます。

- オブジェクトを新規作成します。
- オブジェクトのリストから既存のオブジェクトを選択し、それを変更します。
- ［アクション］ボタンを使用して、オブジェクトでタスクを実行します。

新しいオブジェクト作成

設定タブの 1 つからサブ カテゴリを選択すると、オブジェクト リストが表示されます。 オブジェクト リストからオブジェクトを作成することができます。

たとえば、新しいパートナーシップを確立するには、［フェデレーション］タブを選択し、［フェデレーション パートナーシップの表示］ウィンドウを表示します。［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。

ボタンをクリックして新しいオブジェクトを作成した後、適切なダイアログ ボックスまたは設定ウィザードが表示され、手順が示されます。

オブジェクト リスト

UI 内の設定タブまたはサブカテゴリを表示すると、関連するオブジェクトのリストも表示されます。 リスト内の任意のオブジェクトのリンクをクリックして、そのオブジェクトに関する詳細情報を参照できます。

たとえば、［フェデレーション］タブから［エンティティ］を選択すると、［フェデレーション エンティティの表示］ページにフェデレーション エンティティのリストが表示されます。

オブジェクト リストの［アクション］ボタン

［アクション］ボタンはオブジェクト リスト内のすべてのオブジェクトの左側にあります。 このボタンをクリックすると、オブジェクトに対して実行できるタスクのメニューが表示されます。

［アクション］ボタンは、オブジェクトによって、異なるタスクを提供します。

フィルタリング オブジェクト リスト

オブジェクト タイプに設定されたエントリの特に長いリストがある場合、どのエントリを表示するかフィルタをかけることができ、リストをより簡単に読み取ることができます。 たとえば、ユーザは、そのステータスがアクティブなすべての設定されたパートナーシップを検索できます。そして、これらが［フェデレーション パートナーシップ リスト］に表示されます。

検索フィルタを指定する方法

1. 主な設定タブをクリックします。

   ［フィルタ］および［リスト］グループ ボックスが表示されます。

2. 以下のガイドラインを使用して、**フィルタ** *オブジェクト* グループ ボックス内の検索を設定します。

   a. ［検索対象］フィールドで検索内容を選択します。

      例： 証明書については、オペランドとして［エイリアス］を選択できます。 フェデレーション エンティティについては、オペランドとして［エンティティ タイプ］を選択できます。

b. 中央のフィールドのプルダウン メニューから［演算子］を選択します。

例： =、始まる、含まれている、終了する、その日またはその前に（［失効日］用の唯一のオプション）

c. 以下のいずれかを実行します。

- ［失効日］以外のオペランドについては、最後のフィールドに文字列を入力します（引用符は使用しないでください）。 これは検索フィルタの値です。
- ［失効日］については、フィールドの横のカレンダ アイコンを選択し、日付を選択します。 日付は手動で入力できますが、カレンダから日付を選択すると、確実に正しい日付表示形式となります。

注:

- すべてのリストを取得する場合は、3 番目のフィールドを空欄のままにしてください。 <ANY> またはアスタリスク（*）を入力することもできます。
- アスタリスクを埋め込みワイルドカード文字として使用することはできません。たとえば、アスタリスクは単独で入力できますが、**partner*** のような値として入力することはできません。

3. ［実行］をクリックして検索を開始します。

ページの表示

ユーザが UI 内の任意のオブジェクト用のリストを表示する場合、10 のレコードのみがデフォルトで表示されます。 リストの右下で、大なり記号（>）を選択し、次のページに移動します。 二重大なり記号（>>）を選択し、リストの最後に移動します。

注: 1 ページ当たり表示されるレコード数のデフォルトは 10 で、変更はできません。

リストの右上隅のすべて表示リンクを選択し、1 つのウィンドウ内にすべてのエントリを表示できます。

オブジェクト設定ウィザード

UI には、さまざまなオブジェクトの設定ウィザードが存在します。 エンティティまたはパートナーシップの作成や編集、証明書のインポート、およびメタデータ ファイルのインポートを行う際には、ウィザードが表示されます。

UI ウィザードは、特定のオブジェクトを設定する手順を案内します。 指定の手順で必要な設定に入力しない場合、ダイアログ ボックスの上部で、欠落した情報に入力するように通知するメッセージを受信します。 必須フィールドにすべて入力するまで、次の手順に進むことができません。

Administrative UI へのログイン

管理者は Administrative UI からフェデレーション エンティティを設定します。 Administrative UI にアクセスするためにさまざまな権限レベルを持った複数の管理者を設定できます。 複数の管理者を設定する場合の手順を確認してください。

次の手順に従ってください:

1. ブラウザの Java Script を有効にします。 JavaScript は Administrative UI を開くために必要です。
2. 使用するプラットフォームの手順に従います。

   Windows

   ［スタート］-［すべてのプログラム］-［CA］-［Federation Standalone］-［CA SiteMinder® Federation Standalone 管理 UI］を選択します。

   UNIX

   Web ブラウザを開き、次の URL を入力します：
   http://*fed_server:ui_port*/ca/federation/adminui

   fed_server:ui_port

   UI のポートを含む、CA SiteMinder® Federation Standalone がインストールされているサーバの完全修飾ドメイン名を指定します。 デフォルトのポートは 8888 です。

   例：

   http://fed1.example.com:8888/ca/federation/adminui

   ログイン ウィンドウが表示されます。

3. ユーザ名とパスワードを入力し、［ログイン］をクリックします。

   重要: デフォルトの管理者のユーザ名は常に **admin** です。 これは変更できません。デフォルト管理者パスワードはインストール時に設定されます。

   Administrative UI が起動します。

Active Directory の場合の UI ログイン パスワード条件

管理認証のユーザ ストアとして Active Directory を設定する場合、Administrative UI にログインするときに、以下のパスワード条件に注意します。

- ユーザが［無効］の場合、Administrative UI はユーザのログインを許可しません。 システムは、「エラー: ユーザー名またはパスワードが無効です。」というメッセージを表示します。
- ユーザが［期限切れ］の場合、Administrative UI はユーザのログインを許可しません。 システムは、「エラー: ユーザー名またはパスワードが無効です。」というメッセージを表示します。
- ユーザの属性が「次回のログオン時にパスワードの変更が必要」に設定されている場合、Administrative UI はユーザのログインを許可します。Administrative UI はパスワード管理を処理しません。

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

CA SiteMinder® Federation Standalone 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り、最小数の手順で製品がどのように動作するかを確認できます。

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、製品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 CA SiteMinder® Federation Standalone、およびもう一方のサイトで別の SAML 準拠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 CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone の設定元サイトのエンティティです。 これは、必ずしも CA SiteMinder® Federation Standalone がインストールされているシステム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、CA SiteMinder® Federation Standalone の設定元とは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、CA SiteMinder® Federation Standalone が両方のサイトにある場合に基本的な CA SiteMinder® Federation Standalone パートナーシップを作成する手順を示します。

1. ユーザ ディレクトリ接続を確立します。
2. 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。
3. IdP 側でローカル IdP から SP へのパートナーシップ定義を設定します。
4. SP 側でローカル SP から IdP へのパートナーシップ定義を設定します。
5. パートナーシップをアクティブにします。
6. 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。

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

CA SiteMinder® Federation Standalone 展開モード

スタンドアロン -- ［SiteMinder コネクタ］はありません

IdP の SSO サービス URL

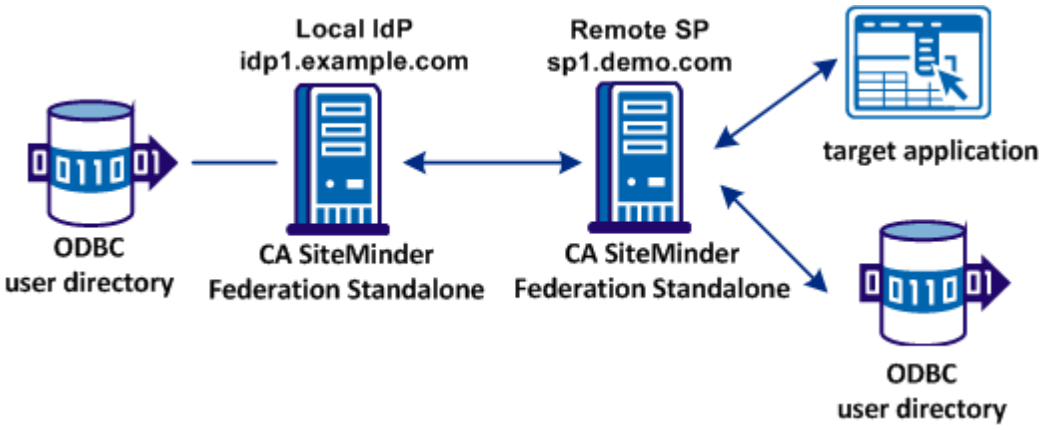
http://idp1.example.com:9090/affwebservices/public/saml2sso

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、CA SiteMinder® Federation Standalone がインストールされた 2 つのシステムが必要です。

以下の図は、サンプル パートナーシップを示しています。



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. Administrative UI にログオンします。
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. ［パートナーシップの作成］-［SAML2 IdP->SP］をクリックします。
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。

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。

続く手順では、製品と共にインストールされるデフォルト データ ソースを使用した ODBC ユーザ ディレクトリへの接続について説明します。

重要: CA FedManager データ ソースは、フェデレーション システム ポリシーが格納される場所です。 この例では、ユーザ ディレクトリとしてこのデータ ソースを使用します。しかし、実稼働環境では別のデータ ソースを使用します。

このデータ ソースの使用方法

- データ ソースのサンプル ユーザ用のスキーマをセットアップします。
- ディレクトリへの接続を確立します。

データ ソースのサンプル ユーザのセットアップ

ODBC スキーマおよびサンプル データをインポートすることにより、データ ストアのサンプル ユーザをセットアップします。

製品は、CA FedManager データ ソースにサンプル ユーザを格納するスキーマおよびデータを作成するためのスクリプト ファイルを提供します。 CA SiteMinder® Federation Standalone のインストール時に指定したのと同じ SQL Server または Oracle データベースにこのデータを格納できます。

次の手順に従ってください:

1. 以下のディレクトリに移動します。

   Windows （デフォルトの場所） :
   federation_install_dir¥siteminder¥db¥SQL

   UNIX : *federation_install_dir*/siteminder/db/sql

2. サンプル ユーザをデータベースに入力するために必要なスキーマ ファイルをインポートします。データベースでインポートを実行するためのツールを使用します。

   以下のファイルをインポートします。

   - smsampleusers_sqlserver.sql

     SQL Server データベース内にサンプル ユーザ用のスキーマを作成し、サンプル ユーザをデータベースに読み込みます。

   - smsampleusers_oracle.sql

     Oracle データベース内にサンプル ユーザ用のスキーマを作成し、サンプル ユーザをデータベースに読み込みます。

   たとえば、このスクリプトで検索すると、siteminder のパスワードを持つ GeorgeC という名前のサンプル ユーザを表示できます。

3. スキーマがインポートされたら、ディレクトリに接続します。

ODBC ディレクトリへの接続

ODBC ユーザ ディレクトリに入力するために適切なスキーマをインポートした後、ユーザ ディレクトリへの接続を確立します。

次の手順に従ってください:

1. Web ブラウザを開き、以下の URL を入力することにより、Administrative UI にログインします。

   http://idp1.example.com:8888/ca/federation/adminui

   CA SiteMinder® Federation Standalone は、サーバ名 idp1.example.com でインストールされています。 ブラウザで、CA SiteMinder® Federation Standalone がインストールされている IP アドレスにこのホスト名をマップします。

   注: Administrative UI を開くには、ブラウザで JavaScript が有効になっていることを確認します。

2. Administrative UI から［ユーザ ディレクトリ］タブを選択します。

   ［ユーザ ディレクトリの表示］ダイアログ ボックスが表示されます。

3. ［ODBC に接続］をクリックします。

   ［ODBC に接続］ダイアログ ボックスが表示されます。

4. ［ODBC ユーザ ディレクトリの設定］グループ セクションの以下の必須フィールドに入力します。

   ディレクトリ名

   FedSQL

   データソース

   CA FedManager データ ソース

5. ［接続認証情報］グループ セクションの以下のフィールドに入力します。

   接続に認証情報が必要

   チェック ボックスをオンにします。

   ユーザ名

   データベースへのアクセスに使用される名前を入力します。

パスワード

データベースへのアクセスに使用されるパスワードを入力します。

パスワードの確認入力

データベース パスワードを再度入力します。

6. ［ディレクトリ フィールド］グループ セクションの以下のフィールドに入力します。

［ユニバーサル ID］列

ユニバーサル ID として使用される ODBC ディレクトリ属性の名前を入力します。 ユーザの ID を維持するために、CA SiteMinder® Federation Standalone と通信する他のアプリケーションにこの値を渡すことができます。 SiteMinder コネクタが有効になっている場合、このフィールドは必須です。

7. ［保存］をクリックします。

［ユーザ ディレクトリの表示］ダイアログ ボックスに戻ります。

8. ［アクション］-［接続のテスト］を選択して、CA SiteMinder® Federation Standalone がユーザ ディレクトリに接続できることを確認します。

接続が成功するかどうかを示すメッセージを受信します。

IdP および SP のエンティティを設定することにより、続行します。

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ローカル側とリモート側を識別する必要があります。 Administrative 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。 ただし、実際のネットワーク設定では、両方がリモート エンティティを定義できるように、両方がローカル エンティティを作成し、メタデータ ファイルにローカル エンティティをエクスポートしてから、ファイルを交換することがよくあります。

次の手順に従ってください：

1. ［フェデレーション］タブで［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

   Local

   新しいエンティティ タイプ

   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 CA SiteMinder® Federation Standalone 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が、この値をサイトの他のエンティティと共有することはできません。

5. 最後の手順で設定を確認し、［完了］をクリックします。

［フェデレーション エンティティの表示］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 SP エンティティを作成する方法

1. まず、［フェデレーション エンティティの表示］ウィンドウに移動します。
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。

   ［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

   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 CA SiteMinder® Federation Standalone 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］グループ ボックス

   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

   エントリ行に対して、この列のチェック 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ら、パートナーシップを作成できます。

IdP から SP 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 IdP から SP へのパートナーシップを設定します。最初は、パートナーシップ用の名前およびその他の基本情報を提供することです。

次の手順に従ってください:

1. ［フェデレーション］タブを選択します。
2. ［パートナーシップの作成］（SAML2 IdP -> SP）をクリックします。

   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。
3. 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

   この値はデフォルトで提供されます。

   スキュー時間（秒）

   デフォルトを受け入れる
4. ODBC ディレクトリ（FedSQL）を［使用可能なディレクトリ］ボックスから［選択されたディレクトリ］ボックスへ移動します。
5. ［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。

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. ［ローカル認証タイプ］フィールドおよび［認証クラス］フィールド用のデフォルト（ベーシック）をそのまま使用します。
3. ［SSO バインディング］フィールドの［HTTP-POST］を選択します。
4. ユーザがリモート SP エンティティをすでに作成したと仮定して、［アサーション コンシューマ URL］の値が書き入れられます。
5. ［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. Administrative UI にログオンします。
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. ［パートナーシップの作成］（SAML2 SP->IdP）をクリックします。
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。

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。

続く手順では、CA SiteMinder® Federation Standalone でインストールされるデフォルト データ ソースを使用して、ODBC ユーザ ディレクトリへの接続について説明します。

重要: CA FedManager データ ソースは、CA SiteMinder® Federation Standalone ポリシーが格納される場所です。 この例では、ユーザ ディレクトリとしてこのデータ ソースを使用します。しかし、実稼働環境では別のデータ ソースを使用します。

このデータ ソースの使用方法

- データ ソースのサンプル ユーザ用のスキーマをセットアップします。
- ディレクトリへの接続を確立します。

データ ソースのサンプル ユーザのセットアップ

ODBC スキーマおよびサンプル データをインポートすることにより、データ ストアのサンプル ユーザをセットアップします。

製品は、CA FedManager データ ソースにサンプル ユーザを格納するスキーマおよびデータを作成するためのスクリプト ファイルを提供します。 CA SiteMinder® Federation Standalone のインストール時に指定したのと同じ SQL Server または Oracle データベースにこのデータを格納できます。

次の手順に従ってください:

1. 以下のディレクトリに移動します。

   Windows （デフォルトの場所）：
   federation_install_dir¥siteminder¥db¥SQL

   UNIX： *federation_install_dir*/siteminder/db/sql

2. サンプル ユーザをデータベースに入力するために必要なスキーマ ファイルをインポートします。データベースでインポートを実行するためのツールを使用します。

   以下のファイルをインポートします。

   - smsampleusers_sqlserver.sql

     SQL Server データベース内にサンプル ユーザ用のスキーマを作成し、サンプル ユーザをデータベースに読み込みます。

   - smsampleusers_oracle.sql

     Oracle データベース内にサンプル ユーザ用のスキーマを作成し、サンプル ユーザをデータベースに読み込みます。

   たとえば、このスクリプトで検索すると、siteminder のパスワードを持つ GeorgeC という名前のサンプル ユーザを表示できます。

3. スキーマがインポートされたら、ディレクトリに接続します。

ODBC ディレクトリへの接続

ODBC ユーザ ディレクトリに入力するために適切なスキーマをインポートした後、ユーザ ディレクトリへの接続を確立します。

次の手順に従ってください:

1. Web ブラウザを開き、以下の URL を入力することにより、Administrative UI にログインします。

   http://idp1.example.com:8888/ca/federation/adminui

   CA SiteMinder® Federation Standalone は、サーバ名 idp1.example.com でインストールされています。 ブラウザで、CA SiteMinder® Federation Standalone がインストールされている IP アドレスにこのホスト名をマップします。

   注: Administrative UI を開くには、ブラウザで JavaScript が有効になっていることを確認します。

2. Administrative UI から［ユーザ ディレクトリ］タブを選択します。

   ［ユーザ ディレクトリの表示］ダイアログ ボックスが表示されます。

3. ［ODBC に接続］をクリックします。

   ［ODBC に接続］ダイアログ ボックスが表示されます。

4. ［ODBC ユーザ ディレクトリの設定］グループ セクションの以下の必須フィールドに入力します。

   ディレクトリ名

   FedSQL

   データソース

   CA FedManager データ ソース

5. ［接続認証情報］グループ セクションの以下のフィールドに入力します。

   接続に認証情報が必要

   チェック ボックスをオンにします。

   ユーザ名

   データベースへのアクセスに使用される名前を入力します。

パスワード

データベースへのアクセスに使用されるパスワードを入力します。

パスワードの確認入力

データベース パスワードを再度入力します。

6. ［ディレクトリ フィールド］グループ セクションの以下のフィールドに入力します。

［ユニバーサル ID］列

ユニバーサル ID として使用される ODBC ディレクトリ属性の名前を入力します。 ユーザの ID を維持するために、CA SiteMinder® Federation Standalone と通信する他のアプリケーションにこの値を渡すことができます。 SiteMinder コネクタが有効になっている場合、このフィールドは必須です。

7. ［保存］をクリックします。

［ユーザ ディレクトリの表示］ダイアログ ボックスに戻ります。

8. ［アクション］-［接続のテスト］を選択して、CA SiteMinder® Federation Standalone がユーザ ディレクトリに接続できることを確認します。

接続が成功するかどうかを示すメッセージを受信します。

IdP および SP のエンティティを設定することにより、続行します。

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とリモート側を識別する必要があります。 Administrative 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。 ただし、実際のネットワーク設定では、両方がリモート エンティティを定義できるように、両方がローカル エンティティを作成し、メタデータ ファイルにローカル エンティティをエクスポートしてから、ファイルを交換することがよくあります。

次の手順に従ってください：

1. ［フェデレーション］タブで［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。

   ［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

   ローカル

   新しいエンティティ タイプ

   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 CA SiteMinder® Federation Standalone 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。

［フェデレーション エンティティの表示］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. まず、［フェデレーション パートナーシップの表示］ウィンドウに移動します。
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。

   ［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

   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 CA SiteMinder® Federation Standalone 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 ボックス

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後に、SP -> IdP パートナーシップの必要なコンポーネントを設定するためにパートナーシップ ウィザードに従います。

次の手順に従ってください:

1. ［フェデレーション］タブを選択します。
2. ［パートナーシップの作成］-［SAML2 SP->IdP］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を始めます。
3. 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

   スキュー時間（秒）

   デフォルトを受け入れる
4. ODBC ディレクトリ（FedSQL）を［使用可能なディレクトリ］ボックスから［選択されたディレクトリ］ボックスへ移動します。
5. ［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

ユーザを識別するためにアサーションのどの属性を使用すべきか指定します。 このアイデンティティ属性値は、ユーザ特定プロセスで使用されます。つまり SP のユーザ ディレクトリにユーザ レコードを置くプロセスで使用されます。

次の手順に従ってください:

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］グループ ボックスで、デフォルト（［名前 ID を使用］）を受け入れます。
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］グループ ボックスで、以下を入力します。

   ODBC 検索仕様

   Name=%s

   このエントリは、変数（%s）をアサーションからの［名前 ID］属性の値に置換し、かつサンプル ユーザ データベースで名前列と一致させるように CA SiteMinder® Federation Standalone に指示します。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. ［次へ］をクリックしてシングル サインオンを設定します。

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 バインディング］フィールドの［HTTP-POST］を選択します。
3. ［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、このターゲットは http://spapp.demo.com です:80/spsample/welcome.html です
4. ［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。

5. ユーザがリモート IdP を作成したと仮定し、［SSO サービス URL］の値が書き入れられます。
6. ［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。

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、パートナーシップのアクティブ化が可能になりました。

CA SiteMinder® Federation Standalone 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。

次の手順に従ってください:

1. ［フェデレーション］タブから、［パートナーシップ］を選択します

   ［フェデレーション パートナーシップの表示］ウィンドウが表示されます。

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、パートナーシップを編集し、必要な設定がすべて設定されていることを確認する必要があります。

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. ［アクティブ化の確認］ダイアログ ボックスで［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］になります。

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 Administrative 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

   注: このサンプル ネットワークでは、CA SiteMinder® Federation Standalone はスタンドアロン モードで展開されているので、ターゲット Web サーバは、CA SiteMinder® Federation Standalone が存在するサーバとは異なります。

   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、SP の AuthnRequest サービスからアイデンティティ プロバイダのシングル サインオン サービスにリダイレクトされます。

アイデンティティ プロバイダでは、ユーザを認証し、セッションを確立した後、サービス プロバイダのターゲット リソース（welcome.html）にユーザを戻します。 SP で作成したサンプル ウェルカム ページは、シングル サインオンが成功したことをユーザに知らせます。

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では、秘密キー/証明書ペアを使用します。

任意のトランザクションまたはランタイムのアクションの前に、IdP1 の管理者は、SP1 に証明書データを含むファイルを送信します。 このファイルには、IdP1 がアサーションを署名するために使用する秘密キーに関連付けられた証明書（公開キー）が含まれます。 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検証します。

以下の手順に、各サイトで署名をセットアップする方法を説明します。

IdP での署名処理の設定

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。 Idp1 は、証明書データ ストアの秘密キーを使用して、アサーションに署名します。

注: 例では、キーと証明書をインポートするファイルがあるか、署名および検証タスク用の秘密キーと証明書をすでに持っていることを前提としています。

次の手順に従ってください:

1. UI から、［フェデレーション］タブをクリックし、［パートナーシップ］を選択します。

   ［フェデレーション パートナーシップの表示］ウィンドウが表示されます。

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。

   編集する前にパートナーシップを非アクティブ化します。

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 ウィザードの最初の手順のダイアログ ボックスが開きます。

4. パートナーシップ ウィザードの［署名および暗号化］手順をクリックします。

5. ［署名］グループ ボックスで、次の操作を実行します。

   a. ［署名の処理を無効にする］を選択解除します。

   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. ファイルが pkcs#12 ファイルである場合は、ファイルを暗号化するためにパスワードを提供します。

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 パートナーシップの表示］ウィンドウに戻ります。

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ります。

9. エイリアス［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。

［フェデレーション パートナーシップの表示］ウィンドウに戻ります。

11. ［フェデレーション パートナーシップ リスト］で、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。

12. 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。

フェデレーション サービスを再起動して、システムに署名の変更を認識させます。

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 では IdP1 からの証明書（公開キー）が必要です。 これは、IdP1 がアサーションを署名するために使用した秘密キーに関連付けられた証明書です。

この証明書を SP1 証明書データ ストアにインポートする必要があります。

次の手順に従ってください:

1. Administrative UI から［フェデレーション］タブをクリックし、［パートナーシップ］を選択します。

   ［フェデレーション パートナーシップの表示］ウィンドウが表示されます。

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. パートナーシップ ウィザードの［署名および暗号化］手順をクリックします。

5. ［署名］グループ ボックスで、次の操作を実行します。
   a. ［署名の処理を無効にする］を選択解除します。
   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 パートナーシップの表示］ウィンドウに戻ります。
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ります。
9. 証明書のエイリアス［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。

    ［フェデレーション パートナーシップの表示］ウィンドウに戻ります。
11. ［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。
12. 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。

  フェデレーション サービスを再起動して、システムに署名の変更を認識させます。

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。

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. Administrative 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］をクリックします。

   ［フェデレーション パートナーシップの表示］ウィンドウが表示されます。
3. 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。

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

   パートナーシップの最初の手順のダイアログ ボックスが表示されます。

5. ［SSO と SLO］手順をクリックします。

6. ［SLO］セクションで、［SLO バインディング］に対して HTTP リダイレクトを選択して、シングル ログアウトを有効にします。

7. ［SLO サービス URL］テーブルの［行の追加］をクリックし、以下を入力します。

   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。

   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。

8. ［選択］列のオプション ボタンをクリックして設定した行を選択します。

9. 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。

10. ［完了］をクリックします。

    ［フェデレーション パートナーシップの表示］ウィンドウに戻ります。

11. ［フェデレーション パートナーシップ リスト］で、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。

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。

次の手順に従ってください:

1. Administrative 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］をクリックします。
3. 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。

   パ編集する前にートナーシップを非アクティブ化する必要があります。
4. 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。
5. ［SSO と SLO］手順をクリックします。
6. ［SLO］グループ ボックスで、［SLO バインディング］に対して HTTP リダイレクトを選択して、シングル ログアウトを有効にします。
7. ［SLO サービス URL］テーブルの［行の追加］をクリックし、使用可能な行がない場合は以下を入力します。

   SLO ロケーション URL

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

   これは、シングル ログアウト リクエストが送信されるリンクです。

   SLO 確認 URL

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

   これは、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。
8. ［選択］列のラジオ ボタンをクリックして設定した行を選択します。
9. 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。

10. ［完了］をクリックします。

    ［フェデレーション パートナーシップの表示］ウィンドウに戻ります。

11. ［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 Administrative 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. Administrative UI から［フェデレーション］タブをクリックし、［パートナーシップ］を選択します。
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. ［認証］グループ ボックスの既存の設定を保持します。
6. ［SSO］グループ ボックスで、以下の操作を実行します。
   a. ［SSO バインディング］フィールドで［HTTP-Artifact］をオンにします。
   b. ［アサーション コンシューマ サービス URL］テーブル内のバインディングを［HTTP-Artifact］に変更します。 URL は、POST プロファイルに使用されたものと同じままにできます。
7. ［バック チャネル］グループ ボックスで、以下を選択します。

   認証方法

   認証なし

8. ［SLO］および［IdP ディスカバリ］グループ ボックスをスキップします。
9. ［確認］手順をクリックし、設定を確認します。
10. ［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. Administrative UI から［フェデレーション］タブをクリックし、［パートナーシップ］を選択します。

   ［フェデレーション パートナーシップの表示］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. ［SSO］グループ ボックスで、以下のタスクを実行します。

   a. ［SSO バインディング］フィールドで［HTTP-Artifact］をオンにします。

   b. ［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。 URL は、POST プロファイルに使用されたものと同じままにできます。

   c. ［SSO サービス URL］の設定を変更しないでください。

6. ［SOAP Artifact 解決 URL］グループ ボックスで［行の追加］をクリックし、以下の URL を入力して、バックチャネルに認証が必要でないことを示します。

   http://idp1.example.com:9090/affwebservices/saml2artifactresolutionnoauth

   必ずテーブルの［選択］列のラジオ ボタンをクリックして、このエントリを選択します。

7. ［バック チャネル］グループ ボックスで、以下のオプションを選択します。

   認証方法

   認証なし

8. ［SLO］および［ステータス リダイレクト URL］グループ ボックスをスキップします。
9. ［確認］手順をクリックし、設定を確認します。
10. ［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 SP1 で設定されました。

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。

   Link for ARTIFACT Single Sign-on

   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: このサンプル ネットワークでは、CA SiteMinder® Federation Standalone はスタンドアロン モードで展開されているので、ターゲット Web サーバは、CA SiteMinder® Federation Standalone が存在す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 の AuthnRequest サービスからアイデンティティ プロバイダのシングル サインオン サービス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。ユーザは、SP で作成したサンプル歓迎ページを見ることで、シングル サインオンに成功したことがわかります。

簡単なパートナーシップ以外の設定手順

この章で説明されている単純なパートナーシップは、CA SiteMinder® Federation Standalone を使用したフェデレーション パートナーシップの設定の概要を示しています。

ガイドの残りの章では、CA SiteMinder® Federation Standalone で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、Administrative 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. 133)
パートナーシップの作成およびアクティブ化 (P. 179)
認証用のユーザ ディレクトリ接続 (P. 91)

第 4 章: ユーザ セッション、アサーション、および失効データの格納

このセクションには、以下のトピックが含まれています。

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。

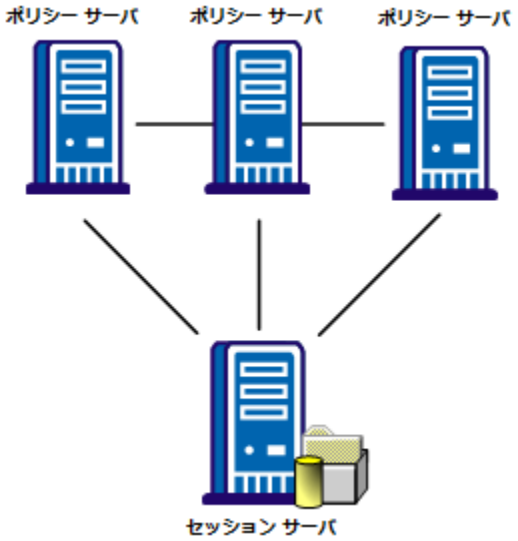
- SAML 2.0 シングル ログアウトおよび WS-フェデレーション サインアウトの場合、アサーティング パーティおよび依存パーティのセッション ストアを、クラスタ内のすべてのポリシー サーバで共有します。

  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: 認証用のユーザ ディレクトリ接続

このセクションには、以下のトピックが含まれています。

ユーザ ディレクトリ接続管理の概要

ディレクトリ接続により、CA SiteMinder® Federation Standalone がユーザ ID のコンテキストを確立する方法が解決されます。 システムでは、これらの接続を使用して、ユーザの識別情報を確認し、ユーザ ストアに含まれているユーザ属性を取得します。

アサーティング パーティは、ユーザ ディレクトリに照らして各ユーザを認証することにより、アサーションの作成可能な対象ユーザを判定します。依存パーティは、認証時にユーザのアサーションが提示されると、ユーザ ディレクトリを調べてユーザ レコードを確認します。

Administrative UI の［ユーザ ディレクトリ］タブを使用して、既存のユーザ ディレクトリへの接続を設定します。 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。

複数のディレクトリへの接続を設定できます。また、ディレクトリは同じタイプ（LDAP または ODBC）である必要はありません。

重要: SiteMinder コネクタを使用している場合は、SiteMinder がポイントしているのと同じディレクトリに接続するようにユーザ ディレクトリを設定する必要があります。また、SiteMinder がそのディレクトリに使用するのと同じ名前を使用して、これを設定する必要があります。

LDAP ディレクトリ接続

LDAP ディレクトリへの接続を確立して、CA SiteMinder® Federation Standalone でそれを認証用のユーザ ユーザに使用できるようにすることができます。

次の手順に従ってください:

1. ［ユーザ ディレクトリ］タブをクリックします。
2. ［ユーザ ディレクトリ リスト］セクションの［LDAP に接続］をクリックします。
3. 各セクションを設定します。赤いドットでマークされたパラメータは必須です。

   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. これらの機能のどちらかをセットアップする場合は、［フェイルオーバー］または［負荷分散］をクリックします。
5. ディレクトリ接続が有効であることを確認するには、［接続のテスト］をクリックします。

   ［内容の表示］をクリックして、ユーザ ディレクトリの内容を一覧表示できます。

   注: ［内容の表示］ボタンは、［検索ルート］、［ユーザ DN 検索の開始］、［ユーザ DN 検索の終了］、［ユニバーサル ID 属性］の値が設定されている場合のみ表示されます
6. ［保存］をクリックします。

   設定が有効な場合、［ユーザ ディレクトリの表示］ダイアログ ボックスにリダイレクトされます。

   LDAP ディレクトリへの接続が設定されます。

LDAP ユーザ ディレクトリのロードバランシングおよびフェールオーバ

CA SiteMinder® Federation Standalone は LDAP ユーザ ディレクトリ リクエストを複数の LDAP サーバ分散させ、フェールオーバおよびロードバランシングを行うことができます。

ロードバランシングでは、指定された LDAP サーバにわたってリクエストが均等に分散されます。 フェールオーバとロードバランシングを組み合わせると、より素早く効率的に LDAP ユーザディレクトリ情報にアクセスできるようになります。

フェールオーバでは、1 つの LDAP サーバを使用してリクエストに対応し、そのサーバが応答に失敗するまで続けます。 デフォルト サーバが応答しない場合、フェイルオーバー用に指定された次のサーバにリクエストがルーティングされます。 この処理を複数のサーバで繰り返すことができます。 デフォルト サーバが再度リクエストに応じることができるようになったら、リクエストは元のサーバに戻されます。

次の手順に従ってください:

1. UI の［ユーザ ディレクトリ］タブを選択します。
2. 以下のいずれかを実行します。
   - ［LDAP に接続］を選択して LDAP ディレクトリ接続を作成します。
   - 編集する既存の LDAP エントリの横の［アクション］-［変更］を選択します。

   ［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。
3. ダイアログ ボックスの［LDAP ユーザ ディレクトリの設定］で、［負荷分散およびフェールオーバの設定］をクリックします。

   ［LDAP サーバ負荷分散およびフェールオーバ］テーブルが表示されます。

4. 最初の［フェールオーバ ノード］フィールドに、*ip_address*:*port* の形式で IP アドレスとポート番号を入力します。フェールオーバの残りのフィールドに後続のディレクトリ サーバのアドレスを追加します。

   注: フェールオーバ用にサーバを追加する場合、そのフェールオーバ ディレクトリは、プライマリ ディレクトリと同じ通信タイプ（SSL または SSL 以外）を使用する必要があります。 両方のディレクトリは同じポート番号を共有しています。

   テーブル内に 1 つのエントリしかない場合、フェールオーバのみがサポートされます。

5. ロードバランシング用に別のグループを設定するには、［行の追加］をクリックして、前の手順で入力したようにフィールドに入力します。

   ロード バランシングに同じサーバを複数回追加して、単一システムで処理されるリクエストを強制的に増やすことができます。 たとえば、グループ内に 2 つのサーバ、Server1 と Server2 があるとします。Server1 は高性能サーバで、Server2 は性能が低いシステムです。Server1 をロード バランシング リストに 2 回追加すると、Server2 でリクエストが 1 つ処理される間に Server1 で 2 つ処理させることができます。

例: ロード バランシングとフェールオーバ

この例では、SiteMinder 環境に、A と B の 2 つのユーザ ディレクトリがあります。これらの 2 つのディレクトリは、以下の要件を満たしている必要があります。

- ユーザ ディレクトリ A は、ユーザ ディレクトリ B にフェールオーバし、ユーザ ディレクトリ B とロード バランシングする必要があります。
- ユーザ ディレクトリ B は、ユーザ ディレクトリ A にフェールオーバし、ユーザ ディレクトリ A とロード バランシングする必要があります。

この設定には、2 つのロードバランシング グループが必要です。

1. 最初のロードバランシング グループおよび最初のフェールオーバ ノード用のユーザ ディレクトリ B のアドレスを指定します。
2. ［行の追加］のクリックによりロードバランシング グループを追加します。
3. 新しいロード バランシング グループの最初のサーバとして、ユーザ ディレクトリ B をリストします。
4. ロード バランシング グループの 2 つ目のサーバとして、ユーザ ディレクトリ A をリストします。

結果は、フェールオーバ "A B" および "B A" に対して各々 1 台のサーバを持った 2 つのロードバランシング グループです。それはお互いにロードバランシングを行います。 両方のディレクトリが使用可能な場合は、各グループ内の最初のディレクトリ間、つまり A と B の間でロード バランシングが発生します。 ユーザ ディレクトリ A が使用不能になると、ユーザ ディレクトリ B に対するフェイルオーバーが発生します。 この結果、ユーザ ディレクトリ B は、ユーザ ディレクトリ A が使用可能になるまで、すべてのリクエストを処理します。

SSL を使用して LDAP ユーザ ディレクトリに接続する方法

SSL を使用して LDAP ユーザ ディレクトリに接続するには、証明書データベース ファイルを使用するようシステムを設定する必要があります。

次のセクションの手順に従い、SSL を使った接続を設定します。

注: CA Directory は、SSL を設定するこの方法をサポートしません。

SSL を使った LDAP 接続を設定する前に

SSL による LDAP ユーザ ディレクトリ接続を設定する前に、以下の点を確認してください。

- ディレクトリ サーバが SSL 対応であること。
- データベース ファイルが Netscape データベース バージョン ファイル形式（cert8.db）であること。 ポリシー サーバは、Mozilla LDAP SDK を使用して、LDAP ディレクトリと通信します。

  重要: cert8.db データベース ファイルへの証明書のインストールに Microsoft Internet Explorer を使用しないでください。

- （Active Directory）以下の点を考慮してください。
  - ユーザ ディレクトリ接続が AD ネームスペースで設定される場合、後続のトピックで説明する SSL プロセスは適用されません。 SSL 接続を確立するときに、AD ネームスペースは、ネイティブの Windows 証明書リポジトリを使用します。SSL を介して通信するように AD ネームスペースを設定する場合は、以下の点を確認してください。
    - ユーザ ディレクトリ接続が、安全な接続になるように設定されていること。

Active Directory インスタンスをホストしているコンピュータで、ルート CA 証明書およびサーバ証明書が、サービスの証明書ストアに追加されるていること。

証明書データベース ファイルの作成

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

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、NSS プロジェクト ページにある Mozilla マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行する前に、管理者権限でコマンド ライン ウィンドウを開きます。 アカウントに管理者権限がある場合でも、このようにコマンド ライン ウィンドウを開きます。

次の手順に従ってください:

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

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

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

2. 以下のコマンドを入力します。

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

   -N

   cert8.db、key3.db、および secmod.db の証明書データベース ファイルを作成します。

   -d *certificate_database_directory*

   certutil ツールが証明書データベース ファイルを作成するディレクトリを指定します。

   注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

   このユーティリティは、データベース キーを暗号化するためにパスワードの入力を求めます。

3. パスワードを入力および確認します。

   NSS は、必要な証明書データベース ファイルを作成します。

   - cert8.db
   - key3.db
   - secmod.db

例: 証明書データベース ファイルの作成

```
certutil -N -d C:¥certdatabase
```

ルート認証機関の証明書データベースへの追加

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

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、NSS プロジェクト ページにある Mozilla マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行する前に、管理者権限でコマンド ライン ウィンドウを開きます。 アカウントに管理者権限がある場合でも、このようにコマンド ライン ウィンドウを開きます。

次の手順に従ってください:

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

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

   注: Windows には固有の certutil ユーティリティがあります。 NSS ユーティリティの bin ディレクトリから作業するようにしてください。 誤って Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行します。

   ```
   certutil -A -n alias -t trust_arguments -i root_CA_path -d
   certificate_database_directory
   ```

   -A

   証明書データベースに証明書を追加します。

   -n *alias*

   証明書の別名を指定します。

   注: 別名にスペースがある場合は、その別名を引用符で囲んでください。

-t *trust_arguments*

証明書に適用する信頼属性を指定します。 使用可能な 3 つの信頼カテゴリは、「SSL、電子メール、オブジェクト署名」の順番で表記されます。 それぞれのカテゴリ位置に、以下の属性引数を 0 個以上使用することができます。

p

有効なピア。

P

信頼されたピア。 この引数は p を意味します。

c

有効な CA。

T

クライアント証明書を発行する信頼された CA。 この引数は c を意味します。

C

サーバ証明書を発行する信頼された CA（SSL のみ）。 この引数は c を意味します。

重要: これは SSL 信頼カテゴリに必須の引数です。

u

証明書は認証または署名に使用できます。

-i *root_CA_path*

ルート CA ファイルのパスを指定します。パスには証明書名も含める必要があります。 証明書の有効な拡張子には、.cert、.cer、.pem などがあります。

注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

-d *certificate_database_directory*

証明書データベースが入っているディレクトリのパスを指定します。

注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

例: 証明書データベースへのルート CA の追加

```
certutil -A -n "My Root CA"  -t "C,," -i C:¥certificates¥cacert.cer -d C:¥certdatabase
```

証明書データベースへのサーバ証明書の追加

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

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、NSS プロジェクト ページにある Mozilla マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行する前に、管理者権限でコマンド ライン ウィンドウを開きます。 アカウントに管理者権限がある場合でも、このようにコマンド ライン ウィンドウを開きます。

次の手順に従ってください:

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

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

   注: Windows には固有の certutil ユーティリティがあります。 NSS ユーティリティの bin ディレクトリから作業するようにしてください。 誤って Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行します。

   ```
   certutil -A -n alias -t trust_arguments -i server_certificate_path -d
   certificate_database_directory
   ```

   -A

   証明書データベースに証明書を追加します。

   -n *alias*

   証明書の別名を指定します。

   注: 別名にスペースがある場合は、その別名を引用符で囲んでください。

-t *trust_arguments*

信頼引数を指定します。 各証明書には、3 つの使用可能な信頼カテゴリがあります。これらのカテゴリを表記する順番は、「SSL、電子メール、オブジェクト署名」です。 それぞれのカテゴリ位置に、以下の属性引数を 0 個以上使用することができます。

p

有効なピア。

P

信頼されたピア。 この引数は p を意味します。

重要: これは SSL 信頼カテゴリに必須の引数です。

-i *server_certificate_path*

サーバ証明書のパスを指定します。 パスには証明書名も含める必要があります。 証明書の有効な拡張子には、.cert、.cer、.pem などがあります。

注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

-d *certificate_database_directory*

証明書データベースが入っているディレクトリのパスを指定します。

注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

NSS によって、証明書データベースにサーバ証明書が追加されます。

例: 証明書データベースへのサーバ証明書の追加

```
certutil -A -n "My Server Certificate" -t "P,," -i C:¥certificates¥servercert.cer -d
C:¥certdatabase
```

証明書がデータベースにあることの確認

証明書が証明書データベースにあることを確認するには、Mozilla Network Security Services（NSS） certutil アプリケーションを使用します。 ポリシー サーバにはこのツールが含まれています。

注: 以下の手順では、タスクを実行するための具体的なオプションおよび引数について詳しく説明します。NSS ユーティリティのオプションおよび引数の全リストについては、NSS プロジェクト ページにある Mozilla マニュアルを参照してください。

重要: Windows Server 2008 上で SiteMinder ユーティリティまたは実行可能ファイルを実行する前に、管理者権限でコマンド ライン ウィンドウを開きます。 アカウントに管理者権限がある場合でも、このようにコマンド ライン ウィンドウを開きます。

次の手順に従ってください:

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

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

   注: Windows には固有の certutil ユーティリティがあります。 NSS ユーティリティの bin ディレクトリから作業するようにしてください。 誤って Windows certutil ユーティリティを実行していることがあります。

2. 以下のコマンドを実行します。

   `certutil -L -d certificate_database_directory`

   -L

   証明書データベース内のすべての証明書を一覧表示します。

   -d *certificate_database_directory*

   証明書データベースが入っているディレクトリのパスを指定します。

   注: ファイル パスにスペースがある場合は、そのパスを引用符で囲んでください。

   このコマンドによって、証明書を証明書データベースに追加する際に指定した、ルート CA の別名、サーバ証明書の別名、および信頼属性が表示されます。

例: 証明書データベース内の証明書の一覧表示

```
certutil -L -d C:¥certdatabase
```

LDAP ユーザ ディレクトリ接続の SSL 対応化

システムに正しい証明書データベースをポイントさせてから、LDAP ユーザ ディレクトリへの SSL セキュア接続を有効にします。 SSL により、ポリシー サーバとユーザ ディレクトリの間の通信がさらに安全になります。

注: 以下の手順では、LDAP 接続が正しく動作していることが前提となります。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［ユーザ ディレクトリ］タブを選択します。

   ユーザ ディレクトリ リストが表示されます。
3. SSL 対応化する LDAP エントリの横の［アクション］-［変更］をクリックします。
4. ［LDAP ユーザ ディレクトリの設定］セクションの［サーバ］フィールドに SSL 接続の正しいサーバおよびポート値が含まれることを確認します。SSL は多くの場合、非 SSL 接続とは異なるポートを使用します。
5. ［接続認証情報］セクションの［安全な接続］チェック ボックスをオンにします。
6. ［保存］をクリックします。

   ［ユーザ ディレクトリ］ダイアログ ボックスに戻ります。
7. ［ユーザ ディレクトリ］リストで、SSL 対応化される LDAP エントリの横の［アクション］-［接続のテスト］を選択します。

   ダイアログ ボックスの一番上のメッセージにより、SSL が正しく設定されたことが確認されるか、またはエラーがレポートされます。

ユーザ ディレクトリ接続が、SSL を介して通信するように設定されます。

証明書データベースへの接続の確立

SSL を使用して LDAP ユーザ ディレクトリに接続するには、システムが適切な証明書データベース ファイルを参照している必要があります。 このデータベースには、cert8.db および key3.db ファイルが含まれている必要があります。

製品に付属の XPSConfig ツールにより、LdapObjCertDbPath 設定を使用して、証明書データベースへのパスを指定できます。

次の手順に従ってください:

1. コマンド ウィンドウを開きます。
2. federation_install_dir に移動します。
3. 「XPSConfig」と入力します。 UNIX プラットフォームのコマンドは大文字と小文字が区別されます。
4. 「SM」と入力します。
5. LdapObjCertDbPath に数値を入力します。
6. 値を変更するために「C」と入力します。
7. ［新しい値の入力］プロンプトに、証明書データベースへのパスを指定します。

   例：

   ```
   C:¥Program Files¥CA¥Federation Standalone¥ldaps¥certdb
   ```

8. XPSConfig を終了するまで、「Q」を入力します。

   新しい値が保存されます。

正しい証明書データベースが使用されます。

LDAP ディレクトリへの SSL 接続の確認

SSL 接続を確認して、ユーザ ディレクトリ接続がセキュリティ保護されていることを確信します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［ユーザ ディレクトリ］を選択します。

   ［ユーザ ディレクトリ］画面が表示されます。そのテーブルでは、既存のユーザ ディレクトリ接続の名前がリスト表示されます。
3. テストするユーザ ディレクトリの名前の横の［アクション］-［変更］を選択します。

   ディレクトリ設定が表示されます。
4. ［内容の表示］をクリックします。

   SSL が正しく設定されている場合は、［ディレクトリのコンテンツ］画面が表示され、ユーザ ディレクトリの内容がリスト表示されます。

LDAP ユーザ ディレクトリへの SSL 接続のトラブルシューティング

以下のリストは、SSL を使用した LDAP ユーザ ディレクトリへの接続時に問題が発生した場合に実行できるアクションを指定しています。

- 安全な接続を使用せずにユーザ ディレクトリに接続できることを確認します。
- 使用している LDAP サーバに対して SSL が有効であることを確認します。
- LDAP サーバの SSL ポートが CA SiteMinder® Federation Standalone ホストからアクセス可能であることを確認します。
- システムが証明書データベース ファイルが含まれるディレクトリを指していることを確認します。
- 証明書データベース ディレクトリに cert8.db および key3.db ファイルが含まれていることを確認します。
- LDAP サーバ設定（ポートを含む）、接続認証情報、および検索ルートが Administrative UI で正しく設定されることを確認します。

ODBC ディレクトリ接続

既存の ODBC ユーザ ストア（SQL または Oracle）へのディレクトリ接続を設定して、CA SiteMinder® Federation Standalone でそれを認証に使用できるようにすることができます。

注: Solaris 上の ODBC データ ソースに接続する予定がある場合は、データ ソースのワイヤ プロトコル ドライバを設定します。 詳細については、ワイヤ プロトコル ドライバの手順を参照してください。

次の手順に従ってください:

1. ［ユーザ ディレクトリ］タブをクリックします。
2. ［ユーザ ディレクトリ リスト］セクションの［ODBC に接続］をクリックします。
3. このダイアログ ボックスを設定します。赤いドットでマークされたパラメータは必須です。

   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. 冗長性のために追加の ODBC ディレクトリをセットアップする場合は、［フェールオーバ］をクリックします。
5. ［テスト接続］をクリックして、接続を検証します。

   ［内容の表示］をクリックして、ユーザ ディレクトリの内容を一覧表示できます。

   注: ［内容の表示］ボタンは、［ユニバーサル ID 列］の値が設定されている場合にのみ表示されます。
6. ［保存］をクリックします。

   設定が有効な場合、［ユーザ ディレクトリの表示］ダイアログ ボックスにリダイレクトされます。

   ODBC ディレクトリへの接続が設定されます。

詳細情報:

Solaris 設定要件上の［ODBC データ ソース］ (P. 108)
Oracle ワイヤ プロトコル ドライバの設定 (P. 109)
SQL Server ワイヤ プロトコル ドライバの設定 (P. 110)

ODBC ディレクトリ フェールオーバ設定

CA SiteMinder® Federation Standalone はフェールオーバ用の複数のデータ ソース サーバ上に ODBC ユーザ ディレクトリ リクエストを分散できます。

注: CA SiteMinder® Federation Standalone は、ODBC ユーザ ディレクトリ用のロードバランシングをサポートしません。

フェールオーバの場合、CA SiteMinder® Federation Standalone は、そのストアが存在するそのサーバが応答に失敗するまで、1 台の ODBC ディレクトリを使用してリクエストに応答し続けます。 デフォルト ディレクトリが応答しない場合、CA SiteMinder® Federation Standalone はフェールオーバ用に設定された次のストアにリクエストがルーティングされます。 この処理を複数のサーバで繰り返すことができます。 デフォルト サーバが再度リクエストに応じることができるようになったら、CA SiteMinder® Federation Standalone はリクエストを元のサーバに戻します。

ODBC フェールオーバの設定

1. UI の［ユーザ ディレクトリ］タブを選択します。
2. 以下のいずれかを実行します。
   - ［ODBC に接続］を選択して ODBC ユーザ ディレクトリ接続を作成します。
   - 編集する既存の ODBC エントリの横の［アクション］-［変更］を選択します。

   ［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。
3. ダイアログ ボックスの［ODBC ユーザ ディレクトリの設定］セクション内の［フェールオーバの設定］をクリックします。

   ［ODBC データ ソース フェールオーバ］テーブルが表示されます。
4. 最初の［フェールオーバ ノード］フィールドにデータ ソース名を入力します。フェールオーバ用の残りのフィールドに他のデータ ソースの名前を追加します。

   注: フェールオーバ用にサーバを追加する場合、そのフェールオーバ ディレクトリは、プライマリ ディレクトリと同じ通信タイプ（SSL または SSL 以外）を使用する必要があります。 両方のディレクトリは同じポート番号を共有しています。

   テーブル内に 1 つのエントリしかない場合、CA SiteMinder® Federation Standalone はフェールオーバのみをサポートします。

例: ODBC フェールオーバ

この例では、SiteMinder 環境に、A と B の 2 つのユーザ ディレクトリがあります。これらの 2 つのディレクトリは、以下の要件を満たしている必要があります。

- ユーザ ディレクトリ A は、ユーザ ディレクトリ B にフェールオーバする必要がある
- ユーザ ディレクトリ B は、ユーザ ディレクトリ A にフェールオーバする必要がある

この設定では、ユーザ ディレクトリ A 用のデータ ソース名と、ユーザ ディレクトリ B 用のデータ ソース名の 2 つのフェールオーバ ノードを必要とします。

Solaris 設定要件上の［ODBC データ ソース］

ユーザ ディレクトリとして UNIX システムで ODBC データ ソースを使用している場合は、system_odbc.ini ファイル内のデータ ソースを設定します。

federation_install_dir/siteminder/db フォルダにある system_odbc.ini ファイルには、使用可能なデータ ソースの名前がすべて含まれます。 さらに、このファイルには、これらのデータ ソースと関連付けられる属性が含まれます。 最初の属性は CA SiteMinder® Federation Standalone に割り当てられた ODBC ドライバです。 残りの属性は、そのドライバに固有です。

新しいデータ ソースを設定するためにファイルを更新している場合、データ ソースについて説明する新しいセクションを追加します。 ［CA FedManager データ ソース］を読み取るセクションの後に SQL Server または Oracle ドライバへのエントリを配置します。 オリジナルテキストを変更しないでください。

Oracle ワイヤ プロトコル ドライバの設定

Oracle ワイヤ プロトコル ドライバを設定し、データベースに接続するために CA SiteMinder® Federation Standalone が使用する設定を指定します。

Oracle ワイヤ プロトコル ドライバの設定方法

1. ディレクトリ *federation_install_dir*/siteminder/db に移動します。
2. テキスト エディタで system_odbc.ini ファイルを開きます。
3. ［CA FedManager データ ソース］を選択し、現在の場所のすぐ下にコピーを作成します。
4. テンプレートとして作成したコピーを使用して、ユーザのデータ ソースに適切なものへと角かっこ内の見出しの名前を変更します。
5. LogonID、Password、HostName および Service Name エントリ内の値を変更します。

   Oracle データ ソース用に変更したテキストは、以下のように表記されます。

   ```
   Driver=federation_install_dir/siteminder/odbc/lib/NSora23.so
   Description=DataDirect 5.3 Oracle Wire Protocol
   LogonID=uid
   Password=pwd
   HostName=servername
   PortNumber=1521
   ServiceName=servicename
   CatalogOptions=0
   ProcedureRetResults=0
   EnableDescribeParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   ```

6. ファイルを保存します。

   Oracle ワイヤ プロトコル ドライバが設定されます。

重要: ファイル内の他のエントリを変更しないでください。特に［CA FedManager データ ソース］でリスト表示されたものはすべて変更しないでください。

SQL Server ワイヤ プロトコル ドライバの設定

SQL ワイヤ プロトコル ドライバを設定し、データベースに接続するために CA SiteMinder® Federation Standalone が使用する設定を指定します。

SQL Server ワイヤ プロトコル ドライバの設定方法

1. ディレクトリ *federation_install_dir*/siteminder/db に移動します。
2. テキスト エディタで system_odbc.ini ファイルを開きます。
3. ［CA FedManager データ ソース］を選択し、現在の場所のすぐ下にコピーを作成します。
4. テンプレートとして作成したコピーを使用して、ユーザのデータ ソースに適切なものへと角かっこ内の見出しの名前を変更します。
5. SQL Server データ ソース用の変更されたテキストが以下のように表示されるように、値を変更し、新しいエントリを追加します。

```
Driver=federation_install_dir/siteminder/odbc/lib/NSmass23.so
Description=DataDirect 5.0 SQL Server Wire Protocol
Database=database_instance
Address=host_IP_address, port_number （デフォルト： 1433）
QuotedId=No
AnsiNPW=No
```

6. ファイルを保存します。

   ワイヤ プロトコル ドライバが設定されます。

重要: このファイル内の他の設定を変更しないでください。特に［CA FedManager データ ソース］でリスト表示されたものはすべて変更しないでください。

ディレクトリ リストからユーザ ディレクトリ接続をテストします。

ユーザ ディレクトリへの接続をテストできます。

ユーザ ディレクトリ接続をテストする方法

1. ［ユーザ ディレクトリ］タブをクリックします。

   ［ユーザ ディレクトリの表示］リストが表示されます。

2. テストするリスト内のエントリの隣の［アクション］ドロップダウン メニューから［接続のテスト］を選択します。

ダイアログ ボックスの一番上のメッセージは接続を確認するか、またはエラーを表示します。

注: ［ユーザ ディレクトリの作成］または［ユーザ ディレクトリの変更］ダイアログ ボックスの［接続認証情報］内の［接続のテスト］をクリックして、接続をテストすることもできます。

ディレクトリ全体の同じユーザ情報の共通のビューの作成

ディレクトリ接続により、CA SiteMinder® Federation Standalone がユーザ ID のコンテキストを確立する方法が解決されます。 アサーティング パーティは、ユーザ ディレクトリに照らして各ユーザを認証することにより、アサーションの作成可能な対象ユーザを判定します。

フェデレーション環境内の複数のユーザ ディレクトリでは、格納するユーザ情報のタイプは同じでも、その情報を識別するために使用する基礎スキーマとユーザ属性名が異なることがよくあります。 そのため、CA SiteMinder® Federation Standalone は、同じユーザ情報の異なるビューを受け取ります。 たとえば、LDAP ディレクトリは、属性の **uid** を使用して、ユーザ名を表すことができますが、ODBC ディレクトリは同じ情報に属性の**名前**を使用できます。

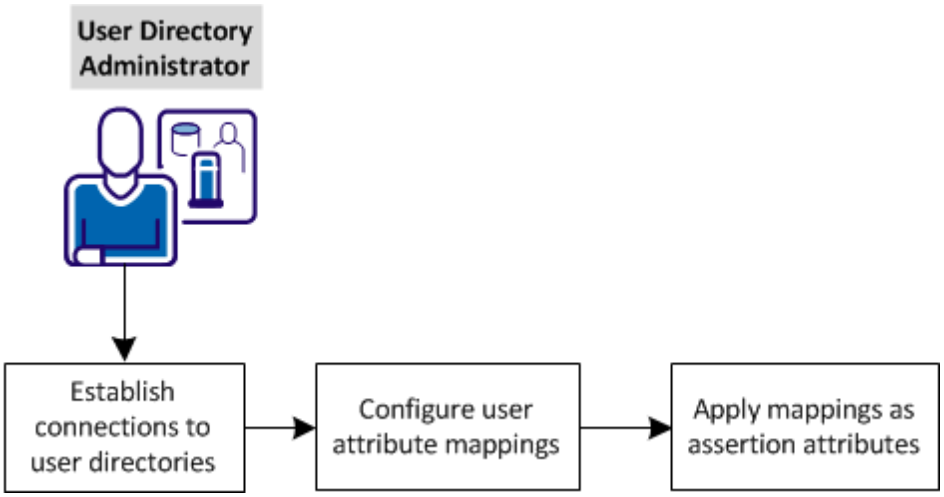
ユーザ属性マッピングの目的は、ユニバーサルなスキーマを定義することによって、同じ情報の共通のビューを作成することです。 ユニバーサルなスキーマは、複数のユーザ ディレクトリ間でユーザ情報を解決できます。 システムは、ディレクトリ タイプに関係なく、ユーザ属性を参照できるため、複数のユーザ ディレクトリに必要な設定オブジェクトの数を大幅に減らすことができます。

ユーザ属性マッピングはそれぞれ、それが定義されているユーザ ディレクトリに固有です。

ユーザ ディレクトリへの接続を設定したら、1 つの共通名を使用して、異なるユーザ ディレクトリ内の同じユーザ情報を参照できるようになります。

ユニバーサル スキーマを作成するために使用する機能は*ユーザ属性マッピング*と呼ばれます。 Administrative UI のユーザ ディレクトリ設定内で、この機能を設定します。

次の画像は、アサーティング パーティでユーザ属性マッピングを設定するプロセスを示します。



アサーティング パーティで、ユーザ属性マッピングの以下のタスクを実行します。

1. ユーザ ディレクトリへの接続を確立します (P. 113)。
2. ユーザ属性マッピングを設定します (P. 114)。
3. アサーション属性としてマッピングを適用します (P. 131)。

ユーザ ディレクトリへの接続の確立

ユーザ属性マッピングを確立する前に、ユーザ レコードを格納するユーザ ディレクトリへの接続を確立します。

LDAP または ODBC は、製品が接続できる 2 つのタイプのディレクトリです。

次の手順に従ってください:

1. ［ユーザ ディレクトリ］タブをクリックします。
2. ［LDAP に接続］または［ODBC に接続］をクリックします。
3. 各セクションを設定します。必須パラメータは赤いドットでマークされています。

   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. これらの機能のどちらかをセットアップする場合は、［フェイルオーバー］または［負荷分散］をクリックします。
5. 接続が有効であることを確認するには［接続のテスト］をクリックします。

   ［内容の表示］をクリックして、ユーザ ディレクトリの内容を一覧表示できます。

   注:

   - LDAP ディレクトリ接続については、［内容の表示］ボタンは、［検索ルート］、［ユーザ DN 検索の開始］、［ユーザ DN 検索の終了］、［ユニバーサル ID 属性］の値が設定されている場合にのみ表示されます。
   - ODBC ディレクトリ接続については、［ユニバーサル ID 列］の値が設定されている場合にのみ［内容の表示］ボタンが表示されます。
6. ［保存］をクリックします。

   設定が有効な場合、［ユーザ ディレクトリの表示］ダイアログ ボックスにリダイレクトされます。

   ディレクトリへの接続が設定されます。

ユーザ属性マッピングの設定

次のマッピング タイプの 1 つまたは複数を使用して、属性マッピングを定義します。

- エイリアス
- グループ名
- マスク
- 定数
- 式

次の表に、マッピング定義に入力できるデータのタイプのリストを示します。 展開の各ユーザ ディレクトリの個別のマッピングを定義します。

マッピング タイプ	共通名のマッピング先	データ型	アクセス
エイリアス	ディレクトリのユーザ属性名。	文字列、数値、ブール	読み取り/書き込み
グループ名	ユーザが特定のグループに属するかどうか識別する属性。	ブール値	読み取り/書き込み
マスク	ビット パターンを格納するユーザ属性。	ブール値	読み取り/書き込み
定数	ディレクトリ内のすべてのユーザに同じか*定数*の値。	文字列、数値、ブール	読み取り
式	式。 完全な構文情報については、「*SiteMinder ポリシー サーバ設定ガイド*」の「属性および式のリファレンス」付録を参照してください。このガイドは SiteMinder マニュアル選択メニューに含まれています。	文字列、数値、ブール	読み取り

各マッピング タイプの設定手順は基本的に同じです。実装例については、各マッピング タイプのユース ケースを参照してください。

次の手順に従ってください:

1. Administrative UI で、［ユーザ ディレクトリ］タブに移動します。
2. ［ユーザ ディレクトリ リスト］の［*接続先*］オプションの 1 つを選択します。
3. ユーザ ディレクトリ接続が設定されていることを確認するか、または接続を設定します。
4. ［ディレクトリ マッピング属性］セクションまでスクロールし、［マッピングの作成］を選択します。
5. ［一般］のフィールドに値を入力します。

   名前

   このマッピングの共通名を指定します。 共通名はユーザ属性名と同じルールに従う必要があります。

   説明

   属性マッピングの説明を入力します。
6. ［プロパティ］フィールドに値を入力します。

   マッピング タイプ

   設定するマッピング タイプを選択します。

   定義

   適切な構文を使用して、マッピング定義を入力します。 先述の表を参照してください。
7. （オプション）［無効］を選択して、この属性マッピングを無効にします。
8. ［保存］をクリックします。

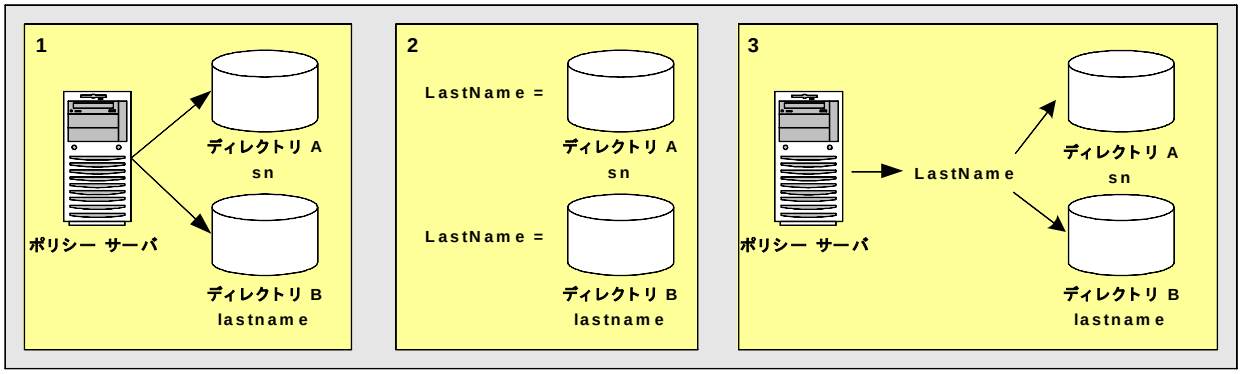
新しい属性マッピングがサブミットされ、［属性マッピング リスト］テーブルのリストに追加されます。

エイリアス属性のユース ケース

このユース ケースでは、ユーザのラスト ネーム（姓）を識別するが、基礎スキーマが異なる 2 つの LDAP ユーザ ディレクトリの基本的なシナリオを表します。

注: 上級ユーザの属性マッピングの例を確認してください。異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法が詳しく説明されています。

以下の図に、2 つのエイリアス属性マッピングによって、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



1. 2 つのユーザ ディレクトリは、別々の方法でユーザの姓を識別します。
   - ディレクトリ A は、ユーザの姓を sn で識別します。
   - ディレクトリ B は、ユーザの姓を lastname で識別します。

   この結果、同じユーザ情報の 2 つの異なるビューが生成されます。
2. LastName が、基礎となるディレクトリ スキーマにマッピングする共通名、つまり「エイリアス」です。
   - LastName は、ディレクトリ A では sn にマッピングされます。
   - LastName は、ディレクトリ B では lastname にマッピングされます。

LastName により、同じユーザ情報の共通のビューが生成されます。LastName は、姓を使用するアサーション属性または名前 ID 属性を定義するときに使用します。 ディレクトリは操作上は同一であるため、システムは、ディレクトリ固有のスキーマを考慮しません。

詳細情報:

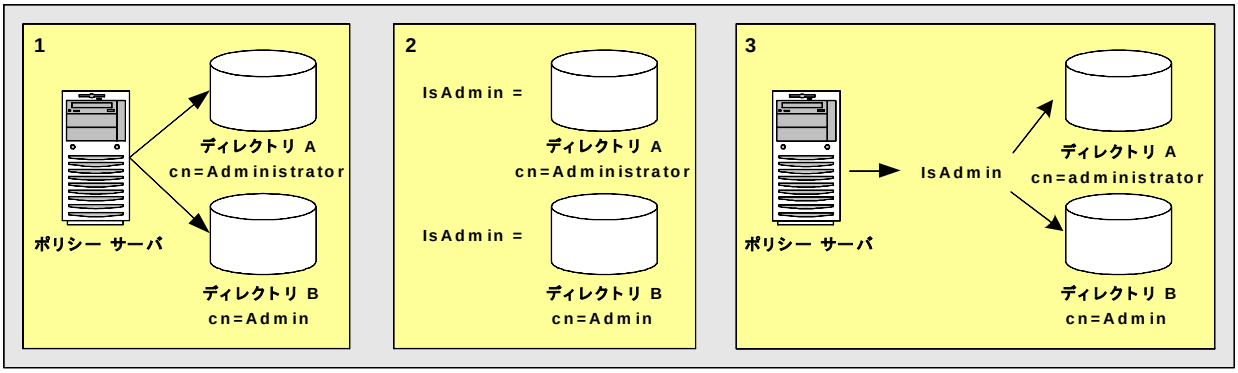
上級ユーザ属性マッピングの例 (P. 125)

グループ名のユース ケース

このユース ケースでは、別々の基本スキーマを使用して管理者グループに属するユーザを識別する 2 つの LDAP ユーザ ディレクトリを示します。

注: 上級ユーザの属性マッピングの例を確認してください。異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法が詳しく説明されています。

以下の図に、2 つのグループ名属性マッピングによって、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



1. 2 つのユーザ ディレクトリは、別々の方法で管理者グループのメンバシップを識別します。
   - ディレクトリ A は、管理者グループのメンバシップを cn=Administrators,ou=groups,o=acme.com として識別します。
   - ディレクトリ B は、管理者グループのメンバシップを cn=Admin,ou=groups,o=acme.com として識別します。

   この結果、同じユーザ情報の 2 つの異なるビューが生成されます。

2. IsAdmin が、基礎となるディレクトリ スキーマにマッピングする共通名です。
   - IsAdmin は、ディレクトリ A では cn=Administrators,ou=groups,o=acme.com にマッピングされます。
   - IsAdmin は、ディレクトリ B では cn=Admin,ou=group,o=acme.com にマッピングされます。

IsAdmin により、管理者グループの共通のビューが生成されます。 管理者グループに適用するアサーション属性または名前 ID 属性を定義するときに IsAdmin を参照できます。ディレクトリは操作上は同一であるため、システムは、ディレクトリ固有のスキーマを考慮しません。

詳細情報:

上級ユーザ属性マッピングの例 (P. 125)

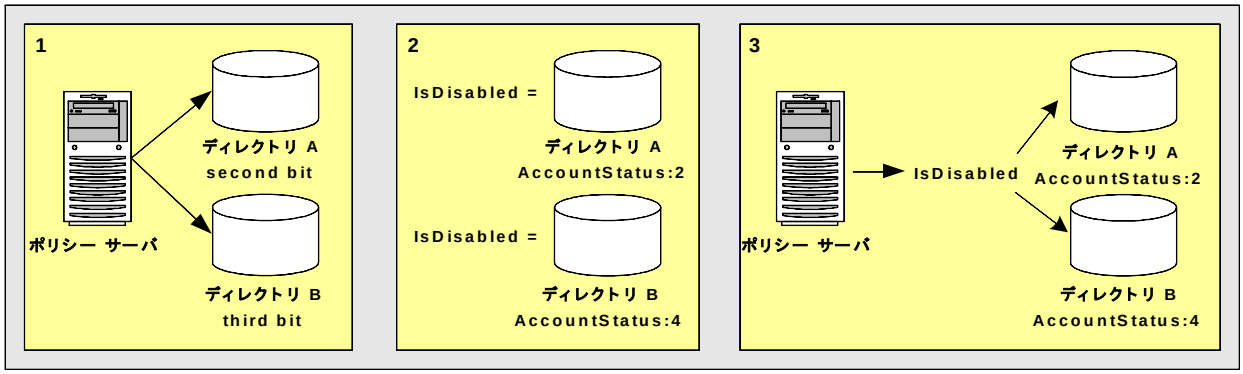
マスクのユース ケース

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

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

注: 上級ユーザの属性マッピングの例を確認してください。異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法が詳しく説明されています。

以下の図に、2 つのマスク属性マッピングによって、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



1. 2 つのユーザ ディレクトリには、AccountStatus というユーザ属性が含まれています。AccountStatus で、ユーザ情報はビット パターンで格納されます。ここで、各ビットはフラグです。

   - ディレクトリ A では、2 番目のビットが無効なアカウントにフラグを立てます。2 番目のビットが 1 に等しい場合、アカウントは無効です。
   - ディレクトリ B では、3 番目のビットが無効なアカウントにフラグを立てます。3 番目のビットが 1 に等しい場合、アカウントは無効です。

   この結果、同じユーザ情報の 2 つの異なるビューが生成されます。

2. IsDisabled が、基礎となるディレクトリ スキーマにマッピングする共通名です。両方のディレクトリとも、IsDisabled は AccountStatus にマッピングされます。

   - ディレクトリ A では、AccountStatus の 2 番目のビットが設定されて、アカウントが無効であるかどうかを、ビット マスク 2 （10 進数）によって判断します。
   - ディレクトリ B では、AccountStatus の 3 番目のビットが設定されて、アカウントが無効であるかどうかを、ビット マスク 4 （10 進数）によって判断します。

IsDisabled により、無効なユーザ アカウントの共通のビューが生成されます。 ユーザのアカウント ステータスを必要とするアサーション属性または名前 ID 属性を定義するときに IsDisabled を参照できます。ディレクトリは操作上は同一であるため、システムは、ディレクトリ固有のスキーマを考慮しません。

詳細情報:

上級ユーザ属性マッピングの例 (P. 125)

マスク属性マッピングのビット マスク

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

マスク属性マッピングは、以下のように定義されます。

```
user_attribute_name:bit_mask
```

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

ビット パターン	フラグ
00?	アカウントは無効ですか?
0?0	パスワードは期限切れですか?
?00	ゴールド メンバですか?

ビットが 1 に等しいとき、フラグは TRUE になります。 表に結果を説明します。

ビット パターン	アカウント ステータス
000 (0)	TRUE のフラグなし
001 (1)	アカント無効
010 (2)	パスワード期限切れ
100 (4)	ゴールド メンバー
011 (3)	パスワード期限切れ、アカウント無効
101 (5)	ゴールド メンバ、アカウント無効
110 (6)	ゴールド メンバ、パスワード期限切れ
111 (7)	ゴールド メンバ、パスワード期限切れ、アカウント無効

注: 同等の 10 進数値をかっこ内に示します。

ユーザがゴールド メンバかどうかのみをテストする場合を考えます。 このビットをテストするには、ゴールド メンバに対応するビット パターンをビット マスクとして選択します。たとえば、2 進数では 100、10 進数では 4 を指定します。結果として作成されるマスク属性マッピングは以下のように定義されます。

```
AccountStatus:4
```

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

アカウント ステータス	ビット マスク	ビット単位の AND の結果	ゴールド メンバですか?
000 (0)	100 (4)	000 (0)	FALSE
001 (1)	100 (4)	000 (0)	FALSE
010 (2)	100 (4)	000 (0)	FALSE
011 (3)	100 (4)	000 (0)	FALSE
100 (4)	100 (4)	100 (4)	TRUE
101 (5)	100 (4)	100 (4)	TRUE
110 (6)	100 (4)	100 (4)	TRUE
111 (7)	100 (4)	100 (4)	TRUE

注: 同等の 10 進数値をかっこ内に示します。

また、ビット マスクを使用して、ビット セットの値や一度に複数のビットをテストできます。 アカウントが無効かどうか、パスワードの期限が切れているかどうかを知りたいとします。 これらのビットをテストするには、011 （2 進数）または 3 （10 進数）のビット マスクを指定します。結果として作成されるマスク属性マッピングは以下のように定義されます。

```
AccountStatus:3
```

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

アカウント ステータス	ビット マスク	ビット単位の AND の結果	両方のフラグが設定されていますか?
000 (0)	011 (3)	000 (0)	FALSE
001 (1)	011 (3)	001 (1)	FALSE
010 (2)	011 (3)	010 (2)	FALSE
011 (3)	011 (3)	011 (3)	TRUE
100 (4)	011 (3)	000 (0)	FALSE
101 (5)	011 (3)	001 (1)	FALSE
110 (6)	011 (3)	010 (2)	FALSE
111 (7)	011 (3)	011 (3)	TRUE

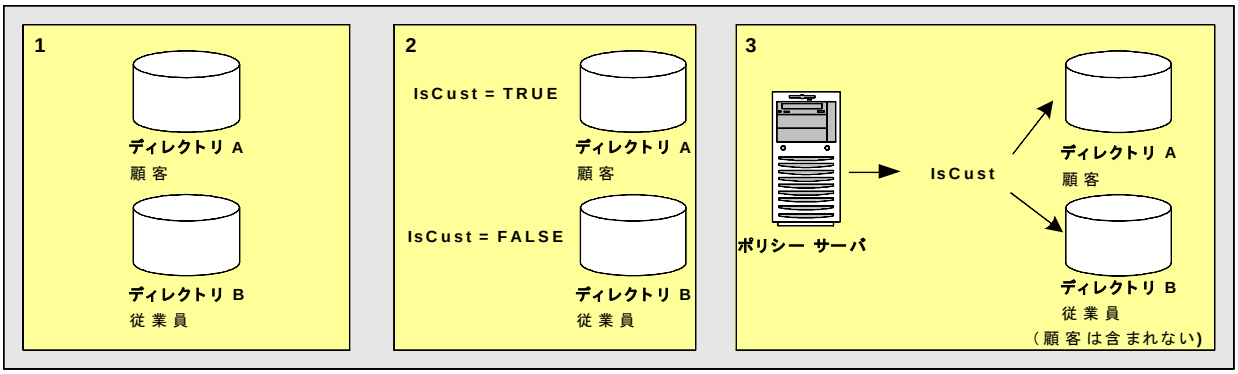
注: 同等の 10 進数値をかっこ内に示します。

定数のユース ケース

このユース ケースでは、一方のユーザ ディレクトリに顧客のみを格納し、もう一方のディレクトリに従業員のみを格納する場合のシナリオを表します。

注: 上級ユーザの属性マッピングの例を確認してください。異なるタイプの属性マッピングを使用して、異なるタイプのディレクトリ間で同じユーザ属性を識別する方法が詳しく説明されています。

以下の図に、2 つの定数属性マッピングによって異なるユーザ ディレクトリの異なる値をどのように表すかを詳しく示します。



1. ディレクトリ A には、顧客のみが格納されます。ディレクトリ B には、従業員のみが格納されます。
2. IsCust が、異なるディレクトリ内の異なる値にマッピングする共通名です。
   - IsCust は、ディレクトリ A では TRUE にマッピングされます。
   - IsCust は、ディレクトリ B では FALSE にマッピングされます。
3. IsCust は、アサーション属性または名前 ID 属性を定義するときに参照します。 共通名を使用することで、システムは、ユーザが格納される特定のディレクトリに関係なく、ユーザが顧客かどうかを判断できます。 このマッピングは、ディレクトリ A 内のユーザはすべて顧客であり、ディレクトリ B 内のユーザはすべて顧客ではないことを示します。

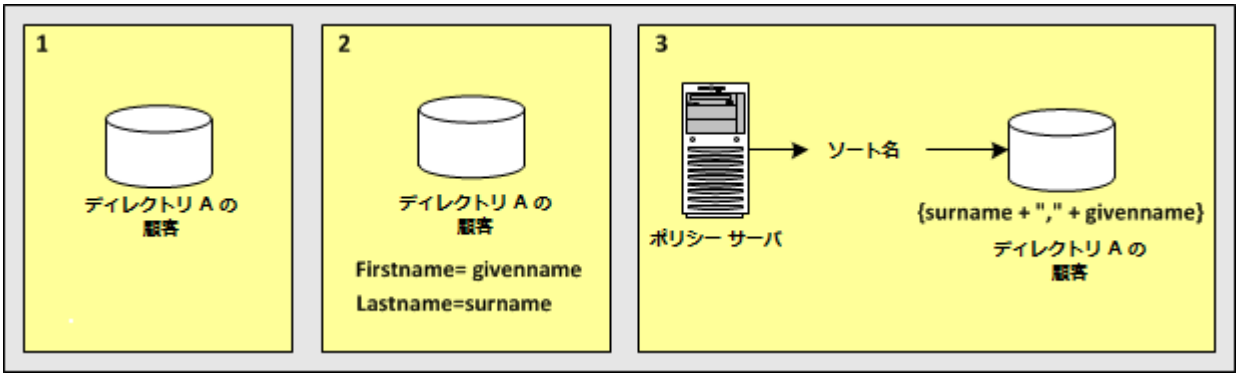
詳細情報:

上級ユーザ属性マッピングの例 (P. 125)

式のユース ケース

このユース ケースでは、式属性マッピングを使用して、1 つのディレクトリ内の複数のユーザ属性への参照を簡略化する方法を示します。 保護されているリソースは、各ユーザの *ソート名*（last name, first name）を必要とします。 ユーザ ディレクトリは一意にこの属性を参照しません。 代わりに、ディレクトリは、各ユーザのラスト ネーム（姓）を surname として、ファースト ネーム（名）を givenname として格納します。

以下の図に、式属性マッピングによって、同じユーザ情報の共通のビューを作成する方法を詳しく示します。



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

- ディレクトリ A には、すべてのユーザ レコードが含まれます。
- マッピングの名前は **SortName** です。
- SortName を定義する式は次のとおりです。

  ```
  {surname + "," + givenname}
  ```

  注: この式は、SiteMinder 式の構文ルールに従います。 完全な構文情報については、SiteMinder マニュアル選択メニューの「*SiteMinder ポリシー サーバ設定ガイド*」の「属性および式のリファレンス」付録を参照してください。
- SortName は、surname および givenname 属性が含まれる式にマッピングされる共通名です。

ディレクトリ固有のスキーマに関係なく、ユーザのソート名を必要とアサーション属性または名前 ID 属性を定義するときに SortName を参照します。

詳細情報:

上級ユーザ属性マッピングの例 (P. 125)

上級ユーザ属性マッピングの例

以下の例では、より複雑なユーザ属性マッピングの設定を示します。

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

ディレクトリ A

従業員専用の内部 LDAP ユーザ ディレクトリです。

ディレクトリ B

カスタマのみ用の ODBC ユーザ ディレクトリです。

ユーザ属性マッピングはそれぞれ、それが定義されているユーザ ディレクトリに固有です。

以下の表に、ディレクトリ A とディレクトリ B が同じユーザ情報を識別する方法を詳しく示します。これに伴うユース ケースでは、異なる属性マッピングを使用して同じユーザ情報の共通のビューを定義する方法について説明します。 共通のビューはユニバーサルなスキーマとして役立ちます。 これによってディレクトリは操作上、同一となります。

属性の説明	ディレクトリ A 属性（LDAP）	ディレクトリ B 属性（ODBC）
各ユーザのファースト ネーム（名）	givenname	u_first_name
各ユーザのラスト ネーム（姓）	surname	u_last_name
各ユーザのソート名（姓、名）	ユーザ ディレクトリは、ユーザ属性を一意に格納しません。	sort_name
カスタマとしてのユーザ	group:cn=customer,ou=groups,o=acme.com	ユーザは常に顧客です。
ユーザ アカウントのステータス	AccountStatus 属性（1 セットのフラグ）。2 番目のビットは無効なアカウントです。	u_disabled

エイリアス マッピング タイプを持つ名の属性をマップする

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

展開

ユーザ ディレクトリ A は、ユーザの名を givenname で識別します。 ユーザ ディレクトリ B は、ユーザの名を u_first_name で識別します。

解決方法

1. ディレクトリ A 用の別名属性マッピングを作成します。

   名前

   FirstName

   マッピング タイプ

   エイリアス

   定義

   givenname

2. ディレクトリ B 用の別名属性マッピングを作成します。

   名前

   FirstName

   マッピング タイプ

   エイリアス

   定義

   u_first_name

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

エイリアス マッピング タイプを持つ姓の属性をマップする

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

展開

ユーザ ディレクトリ A は、ユーザの姓を surname で識別します。 ディレクトリ B は、ユーザの姓を u_last_name で識別します。

解決方法

1. ディレクトリ A 用の別名属性マッピングを作成します。

   名前

   LastName

   マッピング タイプ

   エイリアス

   定義

   surname

2. ディレクトリ B 用の別名属性マッピングを作成します。

   名前

   LastName

   マッピング タイプ

   エイリアス

   定義

   u_last_name

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

式およびエイリアス マッピングのタイプを持つソート名属性をマップする

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

展開

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

解決方法

1. ディレクトリ A の式属性マッピングを以下のように作成します。

   名前

   ソート名

   マッピング タイプ

   式

   定義

   ```
   (surname + "," + givenname)
   ```

   注: この式は、式の構文ルールに従う必要があります。

2. ディレクトリ B 用の別名属性マッピングを作成します。

   名前

   ソート名

   マッピング タイプ

   エイリアス

   定義

   sort_name

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

グループおよび定数マッピングのタイプを持つカスタマをマップする

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

展開

- ディレクトリ A には、従業員が格納されます。 会社の従業員は顧客にもなれるため、ディレクトリ A では、以下に属す従業員を顧客として識別します。

  `cn=Customers、ou=Groups、o=acme.com`

- ディレクトリ B は顧客のみ格納します。 ディレクトリ B には、顧客を識別するユーザ属性はありません。ディレクトリ B に格納されている以上、そのユーザは顧客です。

解決方法

1. ディレクトリ A のグループ属性マッピングを以下のように作成します。

   名前

   IsCustomer

   マッピング タイプ

   グループ

   定義

   `cn=Customers、ou=Groups、o=acme.com`

2. ディレクトリ B の定数属性マッピングを作成します。

   名前

   IsCustomer

   マッピング タイプ

   定数

   定義

   TRUE

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

マスクおよび式マッピングのタイプを持つアカウント ステータスをマップする

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

展開

- ディレクトリ A は一連のフラグである、AccountStatus という名前のユーザ属性で無効なアカウントを識別します。 2 番目のビットは無効なアカウントを示します。
- ディレクトリ B は無効なアカウントを u_disabled という名前のユーザ属性で識別します。 u_disabled が「y」と等しいとき、アカウントは無効になります。u_disabled が「n」と等しいとき、アカウントはアクティブです。

解決方法

1. ディレクトリ A のマスク属性マッピングを作成します。

   名前

   IsDisabled

   マッピング タイプ

   マスク

   定義

   AccountStatus： 2

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

2. ディレクトリ B の式属性マッピングを以下のように作成します。

   名前

   IsDisabled

   マッピング タイプ

   式

   定義

   ```
   (u_disabled = "y")
   ```

   u_disabled はブール式です。

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

アサーション属性へのマッピングの適用

ユーザ ディレクトリに対するユーザ属性マッピングを定義した後、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のアサーション設定にユーザ属性マッピングを追加します。 マッピングは、ディレクトリ タイプごとにさまざまな属性が存在するのにかかわらず、アサーティング パーティがアサーションに正しい属性を含めるのに役立ちます。

名前 ID タイプは、アサーション設定内のユーザ属性にすることができます。

次の手順に従ってください:

1. Administrative 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］をクリックします。
3. アサーティング パーティのパートナーシップに対して［アクション］-［変更］を選択します。
4. ［アサーションの設定］タブに移動します。
5. ［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。

6. フィールドにユーザ マッピングのデータを以下のように入力します。

   アサーション属性

   アサーション属性の名前/値ペアの任意の名前を指定します。

   形式

   属性名を解釈する方法を示す形式を選択します。

   タイプ

   ユーザ属性

   必ずこのフィールドの値としてユーザ属性タイプを選択します。

   値

   ［ユーザ ディレクトリ］ダイアログ ボックスのユーザ マッピング セクションの［名前］フィールドの値を入力します。

   例： マッピングに割り当てた名前が FullName である場合は、このフィールドに FullName を入力します。

7. （オプション）。名前 ID タイプはユーザ属性にすることができるので、［名前 ID］エントリの［値］フィールドをアサーション属性エントリの［値］フィールドと一致させます。 その後、そのアサーションは、名前 ID の同じユーザ属性、およびユーザを識別するアサーション属性を使用します。
8. すべてのアサーション属性に対して、前のステップの手順を繰り返します。
9. ［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。

第 6 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. 133)。
- 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。

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。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［フェデレーション］タブで［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。

   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. エンティティの場所に関して以下のいずれかのオプションを選択します。

   Local

   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. ［新規エンティティ タイプ］フィールドでの特定のエンティティ タイプを特定します。 ドロップダウン リストには、すべてのオプションが表示されます。
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 および属性情報

以下の概念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。

エンティティ名によって、システムのデータベースのエンティティ 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、この値を共有することはできません。

署名と暗号化機能

署名と暗号化機能については、データベースに適切なキー/証明書エントリが存在する必要があります。 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。

アサーション属性の設定

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。

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 エンティティのタイプに関連付けられた機能およびサービスの必須フィールドに入力します。

   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。
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。

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 タイプの場合は、以下のオプションを定義します。

- 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

以下の概念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。

エンティティ名によって、システムのデータベースのエンティティ 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、この値を共有することはできません。

署名と暗号化機能

署名と暗号化機能については、データベースに適切なキー/証明書エントリが存在する必要があります。 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。

アサーション属性の設定

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。

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. 設定するリモート エンティティのタイプに関連付けられた機能およびサービスの必須フィールドに入力します。

   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。
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。

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. 183)」を参照してください。

メタデータのインポートによりエンティティを作成する方法

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。 SAML メタデータのインポートは、パートナーシップを形成するのに必要な設定の量を減らします。

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 CA SiteMinder® Federation Standalone への移行促進に役立ちます。

注: 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。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します。

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ベースとなるメタデータ 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）エンティティを設定するには、インポートするメタデータ ファイル内の証明書を選択します。

   これらの証明書は、署名、検証、またはシングル ログアウトなどのさまざまなフェデレーション機能向けです。
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

メタデータに基づいてフェデレーション 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. Administrative UI にログインします。
2. ［フェデレーション］タブで［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。

   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. 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: 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する必要があります。

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 7 章: キーおよび証明書管理

このセクションには、以下のトピックが含まれています。

証明書および秘密キーの使用

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概要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。

参照証明書データ ストア コンテンツのエイリアス

証明書データ ストア内のキー/証明書ペア、クライアント証明書、および信頼された証明書にはそれぞれ、一意のエイリアスが必要です。 エイリアスは、証明書ストア内のあらゆる秘密キー/証明書ペアまたは単一の証明書への参照です。 証明書データ ストアは複数のキー/証明書ペアおよび単一の証明書を格納します。 フェデレーション環境には、複数のパートナーがいます。 複数のパートナーに対して、各パートナー用に別のペアを使用できます。

アサーションの署名に、署名エイリアスが設定されている場合、アサーション ジェネレータは、エイリアスと関連付けられたキーを使用してアサーションを署名します。 署名するエイリアスが設定されない場合、アサーション ジェネレータでは、以下のエイリアスによるキーを使用してアサーションを署名します。

```
defaultenterpriseprivatekey
```

アサーション ジェネレータでは、デフォルトのエンタープライズ秘密キーが見つからない場合、アサーションを署名するためにストア内の最初の秘密キーが使われます。

重要: 複数のキーを格納する場合は、後のキーを追加する前に、以下のエイリアスで追加する最初のキーを定義します。

```
defaultenterpriseprivatekey
```

指定されたポリシー サーバはレスポンスに署名するか、署名して検証します。署名および検証に使用されるキーと証明書を、同じ証明書データ ストアに追加します。

以下のタイプの秘密キー/証明書ペアおよび単一の証明書が、証明書データ ストアに格納されます。

機能	秘密キー/証明書ペア	証明書（公開キー）	CA 証明書	クライアント証明書
アサーション、認証リクエスト、SLO リクエスト、レスポンスを署名する	X			
アサーション、認証リクエスト、SLO リクエスト/レスポンスを検証する		X		
アサーション、名前 ID、属性を暗号化する（SAML 2.0 のみ）		X		
アサーション、名前 ID、属性を復号化する（SAML 2.0）	X			
Artifact バック チャネルのクライアント証明書認証用の認証情報として機能する				X
他の証明書および証明書廃棄リストを検証する			X	
SSL 接続を使用して、Web サービス変数を処理する			X	

署名および検証操作

システムは、署名および検証タスクに秘密キー/証明書ペアを使用します。秘密キー/証明書ペアは、実行されるトランザクションに応じて、アサーション、アサーション レスポンス、または認証リクエストを署名します。トランザクションの署名前に、アサーションを署名するパートナーは、パートナーに対して秘密キー/証明書ペアと関連付けられている証明書（公開キー）を送信します。 この通信は、帯域外の通信として行われます。 パートナーは証明書を使用して、署名を検証します。

トランザクションが発生すると、アサーティング パーティにはデフォルトでアサーションの証明書が含まれます。 ただし検証時に、パートナーはそのサイトに格納される証明書を使用して、署名を検証します。

SAML 2.0 の単一のログアウトについては、ログアウトを開始する側はリクエストに署名し、リクエストを受信する側は署名を検証します。反対に、受信する側は SLO レスポンスを署名し、ログアウト開始側はレスポンスを検証します。

暗号化と復号化の操作

SAML 2.0 では、CA SiteMinder® Federation Standalone を設定して、全アサーション、名前 ID、または他の属性を暗号化できます。 暗号化を有効にすると、アサーティング パーティでは、依存パーティがデータを暗号化するために送信する証明書（公開キー）を使用します。 トランザクションの前に、依存パーティは、帯域外の通信でアサーティング パーティに証明書を送信します。依存パーティは秘密キー/証明書ペアを使用して、データを復号します。

注: SAML 1.1 および WS フェデレーションは、アサーション データの暗号化をサポートしていません。

SSL 接続用の証明書

Artifact バック チャネルに対して SSL を有効にして、SSL 接続をセキュリティ保護し、バックチャネル通信をセキュリティ保護することができます。

SSL 接続を確立するには、依存パーティが署名済みの SSL サーバ証明書に CA 証明書を関連付ける必要があります。SSL サーバ証明書は SSL 接続をセキュリティ保護します。CA 証明書は、SSL サーバ証明書が信頼できることを検証します。

Artifact バック チャネルをセキュリティ保護する証明書

Artifact のバインドを使用して、シングル サインオンを実装するには、依存パーティがアサーションに対するリクエストをアサーティング パーティの CA SiteMinder® Federation Standalone に送信します。 アサーション リクエストは アサーション検索サービス（SAML 1.1）または Artifact 解決サービス（SAML 2.0）に送られます。検索サービスは、依存パーティによって提供された Artifact を取得し、それを使用してアサーションを検索します。 CA SiteMinder® Federation Standalone ではバック チャネルを介して依存パーティにレスポンスを送信します。 バック チャネルはアサーティング パーティと依存パーティとの間の安全性が確保されている接続です。一方、Web ブラウザ通信はフロント チャネル上で発生します。

以下のいずれかの認証方式を使用して、不正なアクセスからバック チャネルと検索サービスを保護します。

- 基本
- SSL を介した基本
- X.509 クライアント証明書

  認証方式として X.509 クライアント証明書を使用する場合、依存パーティはその認証情報としてクライアント証明書を提供する必要があります。 この認証情報を使用して、依存パーティはアサーションを検索するアサーティング パーティのサービスにアクセスが可能になります。

認証方式を選択する際には、以下の点を考慮します。

- バック チャネルに対して SSL 接続の使用を考慮します。信頼された CA によって署名された SSL サーバ証明書で、SSL 接続の安全性を確保します。

  共通ルートおよび中間 CA 証明書のデフォルトのセットは、証明書データ ストアに付属しています。 CA によって署名された別のサーバ証明書を使用するには、信頼済みの CA 証明書として CA 証明書をストアにインポートします。

  バック チャネル リクエストを処理する場合、フェデレーションは SSL クライアントを使用します。 アサーティング パーティの Web サーバを設定して、以下の暗号によって、SSL バージョン TLSV1_1 および TSLV1_2 を使用できます。

  - RSA_With_AES_128_CBC_SHA256
  - RSA_With_AES_256_CBC_SHA256

  これらの暗号は FIPS および非 FIPS モードの両方でサポートされています。SHA256 の使用の可否については、SP サーバ側で決定されます。フェデレーションには、アルゴリズムを選択するための設定がありません。管理者は、アサーティング パーティのサーバが適切に設定されていることを確認する必要があります。

- X.509 クライアント証明書が、接続の確立に必須である場合、依存パーティにはキー/証明書ペアが必要であり、これがない場合にはクライアント証明書の認証が失敗します。 クライアント証明書がアサーティング パーティの証明書データ ストアに存在することを確認します。 依存パーティがアサーションのリクエストを送信するとき、クライアント証明書が依存パーティの認証情報として機能して検索サービスにアクセスします。

フェデレーショントランザクションのキー/証明書ペアを取得します

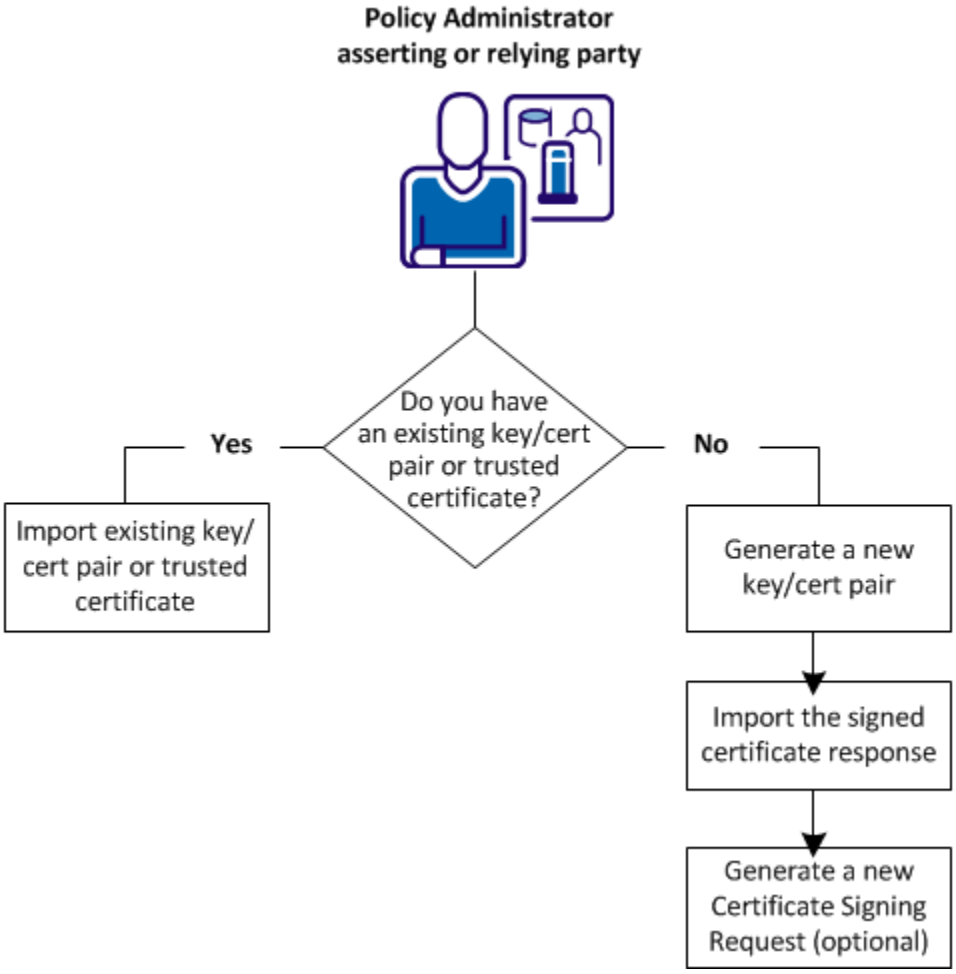
CA SiteMinder® Federation Standalone では、キー/証明書ペアと、多くの機能に対する信頼済み証明書を使用します。 CA SiteMinder® Federation Standalone がキーと証明書を使用するタスクを実行するには、これらのアイテムが証明書データ ストアにある必要があります。

証明書データ ストアにキー/証明書のペアがない場合、処理として以下の 2 つの選択肢があります。

- 既存のファイル（.p12 または .pfx）からキー/証明書のペアをインポートする。
- キー/証明書のペアを生成する。

  新規のキー/証明書ペアを生成するには、信頼された認証機関からの証明書をリクエストし、機関から戻される署名された証明書レスポンスをインポートします。

次の図では、キー/証明書ペアまたは信頼される証明書を取得する各方法の手順を示します。



既存のファイルからキー/証明書のペアをインポートする

証明書データ ストアにキー/証明書のペアがない場合、既存の .p12 または .pfx ファイルからそれをインポートします。

CA SiteMinder® Federation Standalone では、インポートする証明書を信頼される証明書として処理します。 例外は、以下のような自己署名証明書です。

- システムが V3 自己署名証明書を非 CA 証明書として識別する場合、証明書は CA 証明書として処理されます。 この処理は、［証明書/秘密キー］ダイアログ ボックスからインポートを開始した場合も同様です。

- CA SiteMinder® Federation Standalone では以下の場合に、信頼される証明書として証明書を処理します。
  - CA SiteMinder® Federation Standalone が V3 自己署名証明書を CA として識別しない場合。
  - 証明書が V1 自己署名証明書である場合。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ダイアログ ボックスが表示されます。
3. ［新規インポート］をクリックし、ウィザードに従います。

   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。

   ウィザードを実行する際に、以下の点に注意します。

   - 中に含むキーと証明書、または個別のキーと証明書ファイルを使用して、単一のファイルをインポートできます。使用しているファイルに該当するオプション ボタンを選択します。
   - 認証機関証明書として自己署名証明書をインポートするには、［CA として使用］オプション ボタンを［はい］に設定します。 証明書は CA 証明書としてインポートされ、パートナーシップを設定する際には使用できません（たとえば、署名または暗号化）。

     それ以外の場合、デフォルトの［いいえ］設定を受け入れ、パートナーシップを設定する場合に使用可能な信頼された証明書として証明書をインポートします。
   - DER（バイナリ）形式の信頼済み証明書ファイルについては、ファイルに 1 つ以上の証明書エントリを含めることが可能です。 PEM（base 64）形式の信頼済み証明書ファイルについては、CA SiteMinder® Federation Standalone では 1 ファイルにつき 1 つの証明書を前提とします。

     DER または PEM 形式のファイルの標準の拡張子は、*.crt または *.cer です。
   - .p12 ファイルを使用している場合、パスワードの入力が必要です。CA SiteMinder® Federation Standalone は、.p12 または .pfx ファイルを、キー/証明書のペアを格納するファイルとして処理します。

- 証明書データ ストアに追加する予定の各エントリに対しては、そのエントリと関連付けるエイリアスを入力します。 複数のエントリを選択する場合、各エントリごとに一意のエイリアスが必要です。

4. 「確認」の手順では、情報を確認し、［完了］をクリックします。

キー/証明書のペアが証明書データ ストアにインポートされます。

キー/証明書ペアの作成方法

証明書データ ストアにキー/証明書のペアがない場合、新規のキー/証明書のペアを作成できます。

以下の手順に従います。

1. 証明書リクエストを生成し、信頼された認証機関にリクエストを送信します。
2. 機関から署名済み証明書レスポンスをインポートします。

証明書リクエストの生成

証明書データ ストアにキー/証明書のペアがない場合、信頼された認証機関からそれをリクエストします。 CA が署名済みの証明書レスポンスを返すとき、それを証明書データ ストアにインポートします。

証明書リクエストを作成すると、CA SiteMinder® Federation Standalone では秘密キーと自己署名証明書のペアを生成します。 CA SiteMinder® Federation Standalone は、このペアを証明書データ ストアに格納します。作成したリクエストを使用して、認証機関に問い合わせます。作成したリクエストの内容を CA 証明書リクエスト フォームに貼り付けて CA 証明書リクエスト フォームに入力します。

CA は通常、署名された証明書レスポンスを PKCS #7 形式で発行します。署名済み証明書レスポンスを証明書データ ストアにインポートできます。署名された証明書レスポンスがインポートされると、同じエイリアスの既存の自己署名証明書のエントリが置き換えられます。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ダイアログ ボックスが表示されます。
3. ［証明書のリクエスト］をクリックします。

   ［証明書のリクエスト］ダイアログ ボックスが表示されます。
4. 必要なフィールドを入力します。

   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。
5. ［保存］をクリックします。

PKCS #10 の仕様に一致するファイルが生成されます。

証明書リクエストが含まれるファイルを保存または開くように、ブラウザのメッセージが表示されます。 このファイルを保存しない場合（開いてテキストを抽出する場合）であっても、CA SiteMinder® Federation Standalone は秘密キーと自己署名証明書のペアを生成します。 ［CSR の生成］機能を使用して、新規の証明書署名リクエストを生成し、秘密キーの新規リクエスト ファイルを取得します。

署名済み証明書レスポンスのインポート

証明書リクエストを入力し、認証機関にそれを送信すると、認証機関は署名された証明書レスポンスを発行します。

署名された証明書を証明書データ ストアにインポートして、同じエイリアスの既存の自己署名証明書エントリを置き換えます。

次の手順に従ってください：

1. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ダイアログ ボックスが表示されます。
2. 同じエイリアスを持つ自己署名エントリを検索します。

3. 自己署名証明書を含むエントリの隣の［アクション］-［証明書の更新］を選択します。

   証明書とキーをインポートするためのウィザードが表示されます。

   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. 必要なファイルを参照します。 以下を使用できます。
   - 署名された証明書および対応する証明書チェーンが含まれる .p7 または .p7b ファイル。
   - 証明書チェーンのない署名済み証明書を含む .cer または .crt ファイル（base64 PEM ファイル）。
5. 該当のエントリを選択します。
6. 「確認」の手順では、証明書を確認し、［完了］をクリックします。

署名された証明書が、証明書データ ストアにインポートされ、自己署名証明書が置き換えられます。

新規の証明書署名リクエストの生成

証明書署名リクエスト（CSR）は、認証機関に ID 証明書を申請するために送るメッセージです。 CSR を生成する前に、CA SiteMinder® Federation Standalone はキー/証明書ペアを生成する必要があります。 その後、証明書は CSR に配置されます。

以下の理由で、既存の秘密キーに対して新規リクエストを生成します。

- CA SiteMinder® Federation Standalone によって、秘密キー/自己署名証明書ペア用に生成された元のリクエストがもうない。
- 期限切れになるため新規の証明書が必要であり、それは認証機関にサブミットするために CSR の新規コピーを必要とする。

自己署名または、CA 署名された秘密キー/証明書のペアに対して、新規の CSR を生成できます。 秘密キーは既存の秘密キーを変更せずに、常に同一の CSR を生成します。

次の手順に従ってください:

1. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   証明書および秘密キー リストが表示されます。

2. 必要としている新規 CSR の秘密キーのエントリに対して、［アクション］、［CSR の作成］を選択します。

   PKCS #10 仕様に一致するファイルが生成され、CA SiteMinder® Federation Standalone で CSR の保存を求めるメッセージが表示されます。

3. ［保存］をクリックします。

4. （オプション）CA 署名済み証明書を必要とする場合、認証機関に問い合わせて、認証機関で要求される手順に従ってリクエストをサブミットします。 リクエストについて前の手順で保存した PKCS#10 ファイルを使用します。

証明書リクエストのプロセスを完了した後、認証機関が署名済みの証明書レスポンスを発行します。このレスポンスを証明書データ ストアにインポートします。 CA SiteMinder® Federation Standalone で、同じエイリアスの既存の証明書エントリを新しくインポートされた証明書で置き換えます。

CRL を使用して、証明書が有効であることを確認する方法

証明書失効リスト（CRL）は、証明機関によってそのサブスクライバへ発行されます。 リストには、無効のまたは取り消しされた証明書のシリアル番号が含まれます。 サーバにアクセスする要求が受信されると、サーバは CRL に基づいてアクセスを許可または拒否します。

CA SiteMinder® Federation Standalone は、その証明書機能の CRL を利用できます。CA SiteMinder® Federation Standalone が CRL を使用するには、証明書データ ストアが現在の CRL を参照している必要があります。 CA SiteMinder® Federation Standalone が取り消されたパートナー証明書を使用しようとすると、エラー メッセージが表示されます。レガシー フェデレーションでは、エラー メッセージは SAML アサーション内に表示されます。メッセージは、認証が失敗したことを示します。

CA SiteMinder® Federation Standalone は、以下の CRL 機能をサポートします。

- ファイル ベースの CRL または LDAP CRL

  CA SiteMinder® Federation Standalone は、証明書データ ストアに CRL を格納します。 ファイル ベースの CRL は、Base64 またはバイナリ エンコーディングにある必要があります。 LDAP CRL はバイナリ エンコーディングにある必要があります。 さらに、LDAP CRL には以下のいずれかの属性に CRL データが含まれる必要があります。

  - certificateRevocationList;binary
  - authorityRevocationList;binary

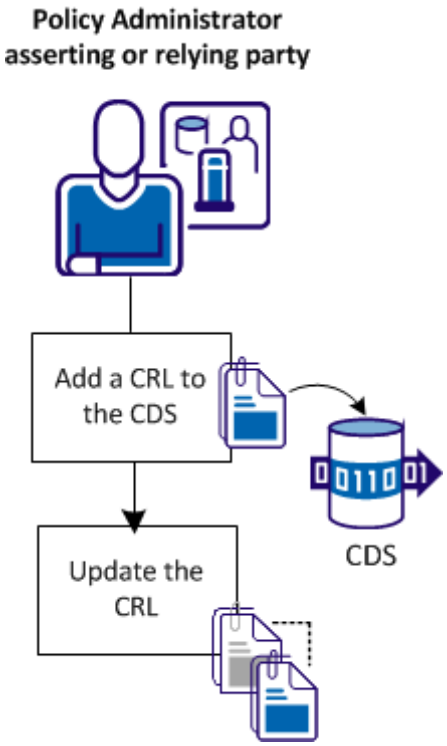
  証明機関が LDAP CRL を発行する場合、RFC4522 および RFC4523 に従ってバイナリ フォーマットで CRL データを返す必要があります。そうしないと、CA SiteMinder® Federation Standalone は CRL を使用できません。

- ファイル CRL 用の PEM および DER のエンコード形式
- LDAP CRL 用の DER エンコード形式

CA SiteMinder® Federation Standalone は、CRL に照らして SSL サーバ証明書を検証しません。 CA SiteMinder® Federation Standalone がインストールされている Web サーバは、SSL サーバ証明書を管理します。

システム内の各ルート CA の CRL を保持している必要はありません。 ルート CA の CRL がない場合、CA SiteMinder® Federation Standalone はその CA によって署名されたすべての証明書が信頼された証明書であると仮定します。

次の図では、CRL を管理するための手順を示します。



CRL 設定手順を以下に示します。

1. CDS に CRL を追加します。 (P. 161)
2. CRL を更新します (P. 162)。

CDS への CRL の追加

証明書を照合できる CRL の使用により、有効な証明書のみがフェデレーション関連の PKI 機能に使用されていることを確認します。

重要: CA SiteMinder® Federation Standalone は、certificateRevocationList;binary LDAP 属性を使用して、バイナリ転送エンコーディングで LDAP CRL を明示的にリクエストします。これは、CRL データをこの属性に格納する必要があることを意味します。 認証機関（CA）が LDAP プロトコルを使用して、CRL を発行する場合、RFC4522 および RFC4523 に従って、バイナリ フォーマットで CRL データを返す必要があります。

CA SiteMinder® Federation Standalone で CRL を使用するには、CRL の場所を指定します。

次の手順に従ってください:

1. ［証明書 & キー］タブに移動します。
2. ［破棄リスト（CRL）］を選択します。

   使用可能な CRL の場所のリストが表示されます。
3. ［追加］をクリックします。

   ［Add Certificate Revocation List］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. CRL の発行者のエイリアスおよび、証明書廃棄リストの場所（URL）を指定します。

   場所は、CRL ファイルのファイル パスおよび LDAP CRL の LDAP 検索パスである必要があります。
5. ［保存］をクリックします。

CRL が証明書データ ストアに追加されました。

CRL の更新

CRL を更新して、使用中の証明書データが最新であることを確認します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブを選択します。
3. ［CDS 設定］を選択します

   ［証明書の設定］ダイアログ ボックスが表示されます。
4. 以下のいずれかの操作を実行します。
   - 格納された CRL ファイルに NextUpdate 値が含まれない場合は、［CRL 更新期間］を設定して、次の CRL が発行される頻度を指定します。
   - アップデータが更新を確認する頻度を変更するには、［CRL Updater スリープ期間］を変更します。

   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。
5. ［CRL Updater］セクションで、［CRL Updater 状態］フィールドの［有効］を選択します。
6. ［保存］をクリックします。

証明書キャッシュ リフレッシュおよび猶予期間の管理

証明書の有効性チェック（CRL または OCSP）を管理するために 2 つの他のタスクを実行できます。

- パフォーマンス向上のための証明書キャッシュ リフレッシュの変更

  証明書のキャッシュ リフレッシュ間隔は、証明書データ ストアがポリシー ストアの証明書データを更新する頻度を示します。証明書データはメモリにキャッシュされ、SiteMinder のパフォーマンスを向上させます。 データが現在のものになるようにメモリ内の情報をリフレッシュします。

- デフォルトの取り消し猶予期間の変更

  デフォルトの取り消し猶予期間は、証明書が取り消されたときから、証明書が無効になるときまでの遅延です。猶予期間中、システムでは、取り消された証明書をそれが無効になるまで使用できます。 証明書が無効になった後は、それはアクティブではなくなり、CA SiteMinder® Federation Standalone はそれを使用できません。

  これらのコンポーネントを追加する際に、CRL または OCSP レスポンダの猶予期間の値を指定しない場合、CA SiteMinder® Federation Standalone はデフォルトの猶予期間を使用します。CRL または OCSP 用にそれぞれ猶予期間を設定した場合、それがこのデフォルト猶予期間値より優先されます。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブを選択します。
3. ［CDS 設定］を選択します

   ［証明書の設定］ダイアログ ボックスが表示されます。
4. 次の設定を変更できます。
   - 証明書のキャッシュ リフレッシュ間隔
   - 破棄猶予期間
   - LDAP アクセス タイムアウト

   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。
5. ［保存］をクリックします。

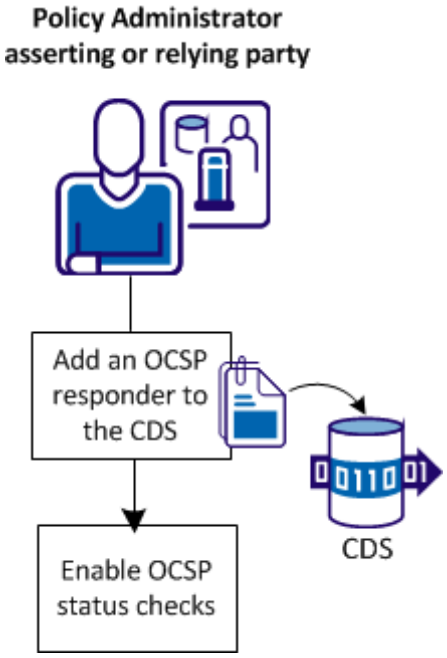
OCSP を使用して、証明書が有効であることを検証する方法

特定のフェデレーション タスクは、証明書データ ストア内の証明書の検証を必要とします。 これらのタスクには HTTP-Artifact バック チャネルの保護、SAML メッセージの確認、および SAML メッセージの暗号化が含まれます。

証明書の有効性を確認するために、証明書データ ストアは OCSP サービスを使用できます。 OCSP は、証明機関（CA）が提供する HTTP サービスを使用して、要求に応じて証明書の取り消しステータスを提供します。

デフォルトでは、CA SiteMinder® Federation Standalone は、証明書データ ストア内の証明書の取り消しステータスを確認しません。 OCSP レスポンダによって取り消しステータスを確認するには、Administrative UI から OCSP を有効にします。 有効にすると、OCSP サービスは設定された OCSP レスポンダの取り消しステータスを 5 分おきに確認します。このデフォルトの頻度は設定可能です。

次の図は、OCSP 設定手順を示しています。



設定プロセスは以下のとおりです。

1. CDS に OCSP レスポンダを追加します。（P. 165）
2. OCSP 状態チェックを有効にします。（P. 166）

OCSP に関する要件

証明書検証に OCSP を使用するには、以下のコンポーネントをセットアップする必要があります。

- OCSP レスポンダをセットアップします。

- 証明書データ ストアに OCSP の信頼されたレスポンダ証明書を格納します。 レスポンダ証明書は、OCSP レスポンスの署名が CA SiteMinder® Federation Standalone に返されたことを検証します。 この証明書は、1 つの信頼された検証証明書または証明書のコレクションになります。

  これらの証明書は、OCSP トランザクションとは別の通信で CA から取得します。

  CA SiteMinder® Federation Standalone は、SHA-1 および SHA-2 ファミリのアルゴリズム（SHA224、SHA256、SHA384、SHA512）を使用して署名されたすべての OCSP レスポンスを使用することができます。

  OCSP レスポンダには、レスポンスと共に署名検証の証明書を含めることができます。CA SiteMinder® Federation Standalone は証明書データ ストア内の信頼された証明書によってレスポンス署名を検証します。

  署名検証の証明書がレスポンスにない場合、CA SiteMinder® Federation Standalone は証明書データ ストア内の証明書または証明書のコレクションによって、署名を検証します。

  Administrative UI 内の OCSP を設定し、証明書または証明書のコレクションの場所を指定する必要があります。
- ユーザ証明書を発行した CA 証明書を証明書データ ストアに格納します。 この CA 証明書はユーザ証明書を検証します。
- （オプション）CA SiteMinder® Federation Standalone が OCSP リクエストに署名するために使用する秘密キー/証明書ペアを証明書データ ストアに格納します。

CDS への OCSP レスポンダの追加

CA SiteMinder® Federation Standalone が対話する各レスポンダの証明書データ ストアに OCSP レスポンダ レコードを追加します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブに移動します。
3. ［OCSP 設定］オプションを選択します。

   ［OCSP 設定リスト］が表示されます。

4. ［追加］をクリックします。
5. フィールドに入力して、OCSP レスポンダ設定を追加します。

   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。
6. ［保存］をクリックします。
7. 設定する各 OCSP レスポンダに対し、このプロセスを繰り返します。

OCSP レスポンダ レコードが証明書データ ストアに含まれました。

OCSP 状態チェックの有効化

CA SiteMinder® Federation Standalone が対話する各レスポンダの証明書データ ストアに OCSP レスポンダ レコードを追加します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブに移動します。
3. ［CDS 設定］オプションを選択します。

   ［CDS 設定リスト］が表示されます。
4. ［OCSP Updater］セクションの［OCSP Updater 状態］フィールドで［有効］を選択します。
5. ［保存］をクリックします。

OCSP ステータスの確認が有効になります。

証明書キャッシュ リフレッシュおよび猶予期間の管理

証明書の有効性チェック（CRL または OCSP）を管理するために 2 つの他のタスクを実行できます。

- パフォーマンス向上のための証明書キャッシュ リフレッシュの変更

  証明書のキャッシュ リフレッシュ間隔は、証明書データ ストアがポリシー ストアの証明書データを更新する頻度を示します。証明書データはメモリにキャッシュされ、SiteMinder のパフォーマンスを向上させます。 データが現在のものになるようにメモリ内の情報をリフレッシュします。

- デフォルトの取り消し猶予期間の変更

  デフォルトの取り消し猶予期間は、証明書が取り消されたときから、証明書が無効になるときまでの遅延です。猶予期間中、システムでは、取り消された証明書をそれが無効になるまで使用できます。 証明書が無効になった後は、それはアクティブではなくなり、CA SiteMinder® Federation Standalone はそれを使用できません。

  これらのコンポーネントを追加する際に、CRL または OCSP レスポンダの猶予期間の値を指定しない場合、CA SiteMinder® Federation Standalone はデフォルトの猶予期間を使用します。CRL または OCSP 用にそれぞれ猶予期間を設定した場合、それがこのデフォルト猶予期間値より優先されます。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブを選択します。
3. ［CDS 設定］を選択します

   ［証明書の設定］ダイアログ ボックスが表示されます。
4. 次の設定を変更できます。
   - 証明書のキャッシュ リフレッシュ間隔
   - 破棄猶予期間
   - LDAP アクセス タイムアウト

   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。
5. ［保存］をクリックします。

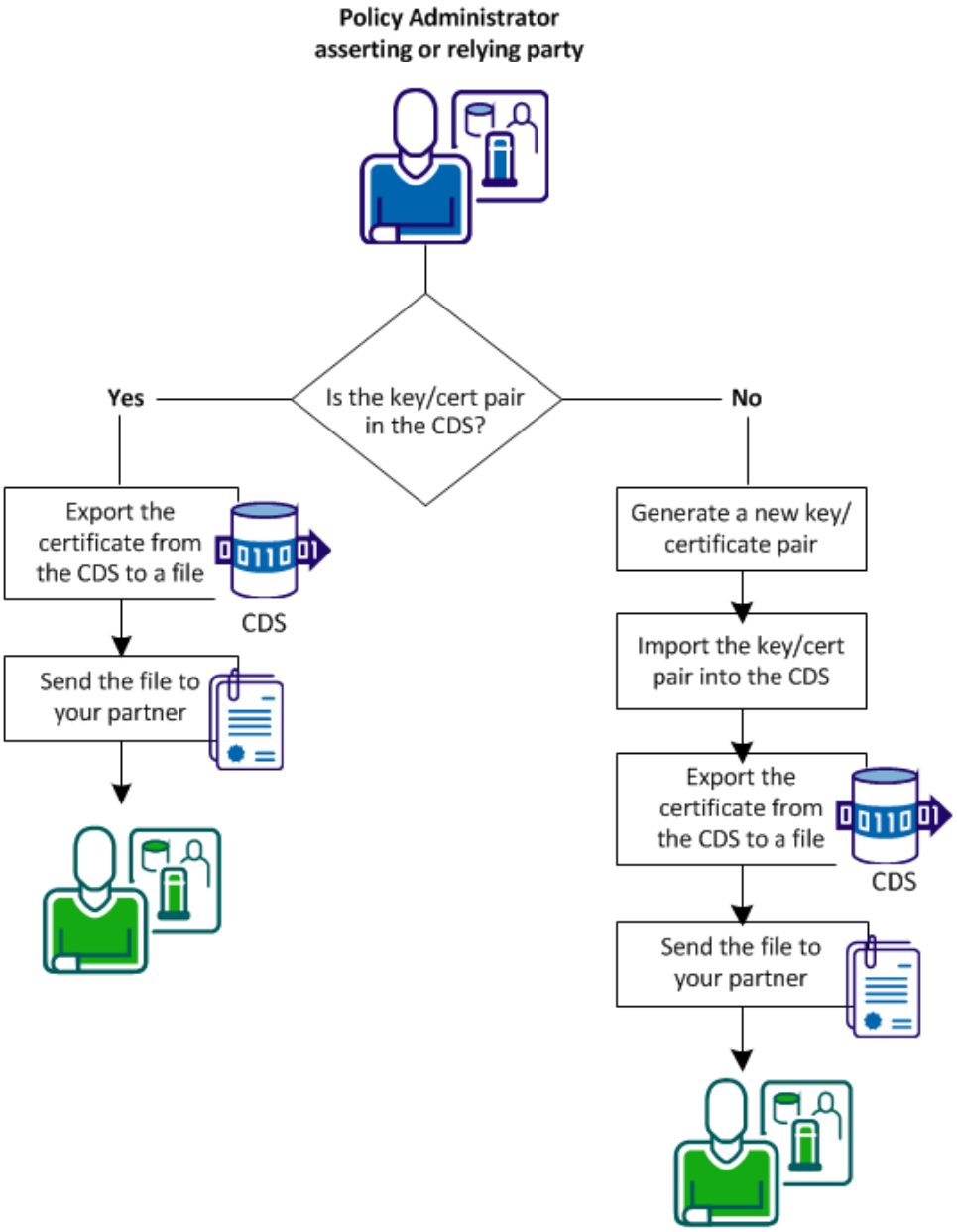
パートナーに証明書を送信する方法

パートナーがメッセージを検証できるように、メッセージに署名するパートナーは、関連付けられた証明書（公開キー）を他のパートナーに送信する必要があります。

メッセージを暗号化するパートナーは、メッセージを復号化することが予想されるパートナーから証明書（公開キー）を受け取る必要があります。

パートナーに必要な証明書ファイルを送信する手順は、キー/証明書ペアがすでに CDS にあるかどうかによって異なります。

次の図に、証明書ファイルを共有する手順を示します。



次の手順に従ってください:

1. 新しいキー/証明書のペアを生成します。
2. CDS にキー/証明書ペアをインポートします (P. 170)。
3. CDS からファイルに証明書をエクスポートします (P. 173)。
4. パートナーに証明書ファイルを送信します。 (P. 173)

UI またはサードパーティ ツールを使用した新しいキー/証明書ペアの生成

証明書データ ストアにキー/証明書のペアがない場合、信頼された認証機関からそれをリクエストします。 CA が署名済みの証明書レスポンスを返すとき、それを証明書データ ストアにインポートします。

Administrative UI を使用するか、またはサードパーティ ツールを使用して、証明書リクエストを生成します。

Administrative UI を使用してリクエストを作成すると、CA SiteMinder® Federation Standalone は秘密キーと自己署名証明書のペアを生成します。 CA SiteMinder® Federation Standalone は、このペアを証明書データ ストアに格納します。 作成したリクエストを使用して、認証機関に問い合わせます。作成したリクエストの内容を CA 証明書リクエスト フォームに貼り付けて CA 証明書リクエスト フォームに入力します。

CA は通常、署名された証明書レスポンスを PKCS #7 形式で発行します。署名済み証明書レスポンスを証明書データ ストアにインポートできます。署名された証明書レスポンスがインポートされると、同じエイリアスの既存の自己署名証明書のエントリが置き換えられます。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。
3. ［証明書のリクエスト］をクリックします。
4. 必要なフィールドを入力します。

   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。
5. ［保存］をクリックします。

PKCS #10 の仕様に一致するファイルが生成されます。

証明書リクエストが含まれるファイルを保存または開くように、ブラウザのメッセージが表示されます。 このファイルを保存しない場合（開いてテキストを抽出する場合）であっても、CA SiteMinder® Federation Standalone は秘密キーと自己署名証明書のペアを生成します。 ［CSR の生成］機能を使用して、新規の証明書署名リクエストを生成し、秘密キーの新規リクエスト ファイルを取得します。

CDS へのキー/証明書ペアのインポート

キー/証明書ペアをインポートする手順はさまざまです。 適切な手順を参照してください。

- 既存のファイルからキー/証明書のペアをインポートします。

  システムにはローカル ファイルがあります。

- 署名済み証明書レスポンスをインポートします (P. 156)。

  Administrative UI からキー/証明書ペアを生成した場合は、認証機関によって送信された、署名されたレスポンスから証明書をインポートします。

既存のファイルからキー/証明書のペアをインポートする

証明書データ ストアにキー/証明書のペアがない場合、既存の .p12 または .pfx ファイルからそれをインポートします。

CA SiteMinder® Federation Standalone では、インポートする証明書を信頼される証明書として処理します。 例外は、以下のような自己署名証明書です。

- システムが V3 自己署名証明書を非 CA 証明書として識別する場合、証明書は CA 証明書として処理されます。 この処理は、［証明書/秘密キー］ダイアログ ボックスからインポートを開始した場合も同様です。
- CA SiteMinder® Federation Standalone では以下の場合に、信頼される証明書として証明書を処理します。
  - CA SiteMinder® Federation Standalone が V3 自己署名証明書を CA として識別しない場合。
  - 証明書が V1 自己署名証明書である場合。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ダイアログ ボックスが表示されます。
3. ［新規インポート］をクリックし、ウィザードに従います。

   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。

   ウィザードを実行する際に、以下の点に注意します。

   - 中に含むキーと証明書、または個別のキーと証明書ファイルを使用して、単一のファイルをインポートできます。使用しているファイルに該当するオプション ボタンを選択します。
   - 認証機関証明書として自己署名証明書をインポートするには、［CA として使用］オプション ボタンを［はい］に設定します。 証明書は CA 証明書としてインポートされ、パートナーシップを設定する際には使用できません（たとえば、署名または暗号化）。

     それ以外の場合、デフォルトの［いいえ］設定を受け入れ、パートナーシップを設定する場合に使用可能な信頼された証明書として証明書をインポートします。
   - DER（バイナリ）形式の信頼済み証明書ファイルについては、ファイルに 1 つ以上の証明書エントリを含めることが可能です。 PEM（base 64）形式の信頼済み証明書ファイルについては、CA SiteMinder® Federation Standalone では 1 ファイルにつき 1 つの証明書を前提とします。

     DER または PEM 形式のファイルの標準の拡張子は、*.crt または *.cer です。
   - .p12 ファイルを使用している場合、パスワードの入力が必要です。CA SiteMinder® Federation Standalone は、.p12 または .pfx ファイルを、キー/証明書のペアを格納するファイルとして処理します。
   - 証明書データ ストアに追加する予定の各エントリに対しては、そのエントリと関連付けるエイリアスを入力します。 複数のエントリを選択する場合、各エントリごとに一意のエイリアスが必要です。
4. 「確認」の手順では、情報を確認し、［完了］をクリックします。

キー/証明書のペアが証明書データ ストアにインポートされます。

署名済み証明書レスポンスのインポート

証明書リクエストを入力し、認証機関にそれを送信すると、認証機関は署名された証明書レスポンスを発行します。

署名された証明書を証明書データ ストアにインポートして、同じエイリアスの既存の自己署名証明書エントリを置き換えます。

次の手順に従ってください：

1. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ダイアログ ボックスが表示されます。
2. 同じエイリアスを持つ自己署名エントリを検索します。
3. 自己署名証明書を含むエントリの隣の［アクション］-［証明書の更新］を選択します。

   証明書とキーをインポートするためのウィザードが表示されます。

   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. 必要なファイルを参照します。 以下を使用できます。
   - 署名された証明書および対応する証明書チェーンが含まれる .p7 または .p7b ファイル。
   - 証明書チェーンのない署名済み証明書を含む .cer または .crt ファイル（base64 PEM ファイル）。
5. 該当のエントリを選択します。
6. 「確認」の手順では、証明書を確認し、［完了］をクリックします。

署名された証明書が、証明書データ ストアにインポートされ、自己署名証明書が置き換えられます。

Administrative UI を使用した CDS からの証明書のエクスポート

ファイルに秘密キー/証明書ペアをエクスポートし、フェデレーション パートナーに証明書ファイル（公開キー）を送信できます。 パートナーは、証明書を使用して、関連付けられた秘密キーで作成されたアサーション レスポンスの署名を検証したり、関連付けられた秘密キーで復号化されるレスポンスを暗号化したりすることができます。

重要: バックアップの一部として秘密キーをエクスポートする場合は、他の誰ともそれを共有しないでください。

次の手順に従ってください：

1. Administrative UI にログインします。
2. ［証明書 & キー］タブから、［証明書および秘密キー］を選択します。

   ［証明書および秘密キーの表示］ウィンドウが表示されます。
3. エクスポートする［証明書および秘密キーリスト］内のエントリに対し、［アクション］-［エクスポート］を選択します。

   ［キー ストア エントリのエクスポート］ダイアログ ボックスが表示されます。
4. エクスポートされたデータから作成するファイルの形式を選択します。

   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。
5. ファイル形式を選択します。
6. ［Export］をクリックします。

   ローカル システムでファイルを開くか保存するかを問われます。

   CA SiteMinder® Federation Standalone によって、キーまたは証明書を示す、エンコードされたファイル コンテンツが生成されます。
7. パートナーにファイルを送信します (P. 173)。

パートナーへの証明書ファイルの送信

証明書を使用してエンコードされたファイルをエクスポートしたら、フェデレーション パートナーにこのファイルを送信します。 パートナーは、この証明書をインポートして、フェデレーション メッセージの検証または暗号化を処理する必要があります。

証明書データ ストアの証明書の更新

以下の方法で、キー/証明書ペアおよびスタンドアロンの証明書を更新できます。

- 期限切れになる信頼済み証明書の更新は、既存の証明書を削除して信頼される新規の証明書をインポートすることにより対応します。 新規証明書は、証明書データ ストア内の期限切れになる証明書に一致する必要があります。
- 信頼される署名済み証明書または PKCS7 署名済みレスポンスをインポートすることにより、証明書を更新します。 新規証明書は、証明書データ ストア内の期限切れになる証明書に一致する必要があります。
- PKCS#12 ファイルからの証明書で証明書を更新します。 新規の秘密キーと証明書のペアは、証明書データ ストア内の期限切れになる秘密キー/証明書ペアに一致する必要があります。

CA SiteMinder® Federation Standalone で期限切れになる証明書が更新される前に、新規証明書が有効になっている必要があります。 証明書はインポートされると直ちに、更新され利用可能になります。 有効期間に定められているとおりに、新規証明書が有効になっていないと CA SiteMinder® Federation Standalone は新規証明書を使用できません。

信頼された証明書のみをインポートするには、PEM または DER のエンコーディングがある証明書ファイルを使用します。 これらのタイプのファイルの標準的な拡張子は *.crt または *.cer です。ファイルが .p12 または .pfx で終わる場合、それはキー/証明書ペアを含む証明書データ ストア ファイルとして処理されます。 また、ファイルが .p7 または .p7b で終わる場合は、署名されたレスポンス ファイルとして処理されます。それ以外のファイルはすべて証明書ファイルとして扱われ、CA SiteMinder® Federation Standalone は、そのファイルから証明書をロードしようとします。

注: フェデレーション環境の証明書を更新する場合、期限切れになる証明書を使用してフェデレーション オブジェクトを更新する必要はありません。

認証機関(CA)証明書の使用

フェデレーション システムは認証機関の証明書を使用して、以下の項目を検証します。

- SSL 接続の SSL サーバ証明書が SAML HTTP Artifact バック チャネルが信頼できることを保証するものであるかどうか。
- HTTP Artifact シングル サインオンの場合、SSL 接続を使用してバック チャネルをセキュリティで保護します。フェデレーション システムに組み込まれた Web サーバで、認証機関の証明書を検証することによって信頼される証明書によって SSL 接続がセキュリティ保護されていることを確認できます。この証明書は、証明書データ ストアに格納されている必要があります。
- 証明書廃棄リストが有効かどうか。

  CRL は認証機関から取得されます。対応する CA の証明書は、信頼されるまで CRL を検証する必要があります。 CRL はデータ ストアに格納され、ランタイムで使用されます。

共通ルートおよび中間 CA 証明書のデフォルトのセットは、その目的のために製品に付属しています。

CA 証明書のインポート

共通ルートおよび中間 CA のセットが、製品に含まれています。 証明書データ ストアにない CA 証明書を使用するには、それらをインポートします。

インポートする証明書は CA 証明書として扱われます。 例外は、以下のような自己署名証明書です。

- システムが V3 自己署名証明書を非 CA 証明書として識別する場合、証明書は信頼された証明書として処理されます。この処理は、［CA 証明書のインポート］ダイアログ ボックスからインポートを開始した場合も同様です。
- システムが V1 自己署名証明書を識別する場合、証明書は CA 証明書として扱われます。

注: ルート CA 証明書をインポートする場合、それらが信頼チェーンの一部である場合は、チェーン内のすべてのルート CA 証明書をインポートします。

CA 証明書をインポートする方法

1. Administrative UI にログインします。
2. ［証明書 & キー］-［権威者］の順に選択します。

   認証機関リストが表示されます。
3. ［新規インポート］をクリックします。

   ［CA 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。

   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. ウィザードに従って、新規エントリをインポートします。
5. 「確認」の手順では、証明書を確認し、［完了］をクリックします。

CA 証明書が証明書データ ストアにインポートされます。 変更は、インポートの終了直後に有効になります。

重要: CA 証明書がシステムで使用中の他の証明書の信頼チェーンの一部である場合、それを削除することはできません。 使用中の CA 証明書を削除しようとすると、証明書を削除できないことを知らせるエラー メッセージが表示されます。

バック チャネル通信の証明書署名検証のトラブルシューティング

症状:

HTTP-Artifact がシングル サインオンに使用されているプロファイルです。アサーティング パーティは、SSL バック チャネルを介して、依存パーティと通信しています。 依存パーティは、SSL バック チャネルを介して通信するために、アサーティング パーティでサーバ証明書の署名を検証する必要があります。

サーバ証明書の署名の検証の失敗として、以下のエラーがログに記録されます。
［リクエスト メッセージの処理中に不明な例外をスローされたディスパッチャ オブジェクト。 メッセージ： 証明書は検証されませんでした..］

解決方法:

依存パーティはルート証明書を証明書データ ストアにインポートする必要があります。 この証明書は、アサーティング パーティでサーバ証明書の署名を検証するために必要です。 検証のため、サーバ証明書を署名したルート CA をインポートします。

検証のためにインポートされた CA 証明書に関する以下の情報を確認します。

- ルート CA 証明書の発行者とサブジェクト DN は同じですか。そうでない場合、証明書は中間ルート CA です。信頼チェーン内のすべてのルート CA 証明書をインポートします。
- アサーティング パーティ サーバ証明書の発行者がインポートしたルート CA のサブジェクトおよび発行者に一致することを確認します。

第 8 章: パートナーシップの作成およびアクティブ化

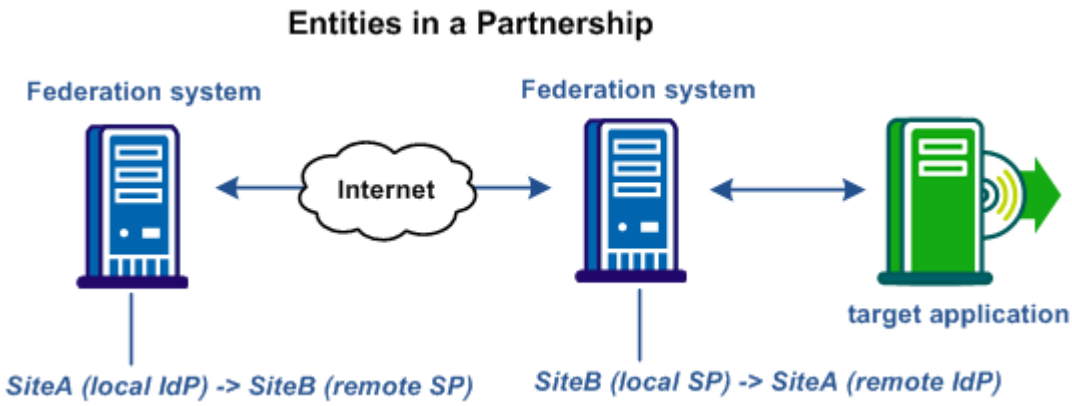
このセクションには、以下のトピックが含まれています。

パートナーシップ作成

CA SiteMinder® Federation Standalone の主な目的は、2 つの組織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パーティの役割を担うことができます。

CA SiteMinder® Federation Standalone 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。 2 つの定義によって単一のパートナーシップを定義します。

以下の図では、SiteA がローカル SAML 2.0 IdP として設定されると、SiteB がリモート SAML 2.0 SP として指定されます。 SiteB がローカル SAML 2.0 SP として設定されれば、SiteA はそのリモート SAML 2.0 IdP です。



注: アサーティング パーティは複数の依存パーティとのパートナーシップを持つことができ、依存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。

フェデレーション パートナーシップの作成は、以下の手順で構成されます。

1. パートナーシップ タイプを指定します。
2. 以下のパートナーシップの詳細を設定します。
    a. パートナーシップ名および参加するエンティティ
    b. フェデレーション ユーザ（ローカルのアサーション パーティのみ）
    c. 名前 ID 形式およびその他のアサーション属性（ローカルのアサーティング パーティのみ）
    d. ユーザ ID（ローカルの依存パーティのみ）
    e. SSO（Single Sign-On）
    f. シングル ログアウト（SLO） - SAML 2.0 のみ
    g. 署名
    h. 暗号化 - SAML 2.0 のみ

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によって、ローカルのフェデレーション ロールおよびリモートのフェデレーション ロールを指定します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［フェデレーション］タブから、［パートナーシップ］を選択します。
3. ［フェデレーション パートナーシップ リスト］セクションの［パートナーシップの作成］をクリックします。

   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。
4. 以下のいずれかのオプションを選択します。
   - 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 IP->RP（リソース パートナーはローカルです）。

［フェデレーション パートナーシップの作成］ダイアログ ボックスに、パートナーシップ設定の最初の手順が表示されます。

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）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。

6. ［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。

7. 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、Administrative 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

パートナーシップの必要なすべての設定を設定したら、それを使用するためにアクティブにします。 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］タブから、［パートナーシップ］を選択します。
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。

   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、［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。 新しい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。

Administrative 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: Administrative 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。システムは、選択されたパートナーシップから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］タブから、［パートナーシップ］を選択します。
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。
3. ダイアログ ボックスのフィールドに入力します。

   ACTIVE ステータスでパートナーシップをエクスポートする場合、ほとんどのフィールドは読み取り専用になりますが、［有効期間］フィールドおよびエイリアス ドロップダウン リストのみ編集できます。

   注: フィールド、コントロール、およびそれぞれの要件については、［ヘルプ］をクリックしてください。
4. ［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。 さらに、SiteMinder との CA SiteMinder® Federation Standalone 統合用の SiteMinder コネクタを有効にできます。

SiteMinder コネクタは、展開された SiteMinder システムが CA SiteMinder® Federation Standalone と統合することを可能にするソフトウェア コンポーネントです。 CA SiteMinder® Federation Standalone がアサーティング パーティにある場合、SiteMinder コネクタは SiteMinder セッションから CA SiteMinder® Federation Standalone セッションを作成できます。 SiteMinder セッションを確立するには、まず SiteMinder がユーザを認証してから、ユーザがアサーティング パーティにアクセスします。

パートナーシップ単位で SiteMinder コネクタを有効にできますが、ただ 1 つのグローバル コネクタ設定がすべてのパートナーシップに適用されます。 ［展開設定］内のチェック ボックスがオンになっており、設定が定義されている場合に限り、コネクタは使用可能になります。 UI 内の［インフラストラクチャ］タブから［展開設定］にアクセスします。 コネクタをグローバルに有効にした後で、CA SiteMinder® Federation Standalone はパートナーシップ設定を評価して、コネクタが有効かどうかを判断します。パートナーシップはグローバル コネクタ設定を使用します。

パートナーシップのコネクタを無効にするには、パートナーシップ レベルでチェック ボックスをオフにします。 コネクタをグローバルに無効にするには、［展開設定］で無効にします。

重要: コネクタがグローバル レベルで無効になっている場合、CA SiteMinder® Federation Standalone はパートナーシップ レベルでチェック ボックスを無視します。

フェデレーション ユーザの設定

フェデレーション ユーザは保護されているフェデレーション リソースへのアクセスが許可されているユーザです。

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。

6. （オプション） SiteMinder コネクタを設定します。

   a. CA SiteMinder® Federation Standalone が既存の SiteMinder 展開と統合している場合は、チェック ボックスをオンにして SiteMinder コネクタを有効にします。

b. （オプション） CA SiteMinder® Federation Standalone または SiteMinder がユニバーサル ID を使用してユーザ レコードを取得するように、［UserDN とディレクトリ名の比較の実行］をオフにします。 ユニバーサル ID を使用すると、物理的に異なるさまざまなタイプのユーザ ディレクトリが可能になります。ユニバーサル ID を使用することにより、取得されたユーザ レコードを正しいレコードと見なすことができます。

注: ユニバーサル ID に基づいている場合、各ユーザは一意のユニバーサル ID を持っている必要があります。ユニバーサル ID が一意でない場合、ユーザ レコードにアクセスするシステムは不正な記録を取得する可能性があります。

チェック ボックスをオン（デフォルト）にしておく場合、CA SiteMinder® Federation Standalone と SiteMinder は同じ物理ディレクトリを使用する必要があります。 これらの両方のディレクトリの名前は、ユーザ ストア検索に対して同じである必要があります。ユーザを認証するエンティティは、ユーザが指定した情報をユーザ レコードの UserDN およびディレクトリ名と比較します。

ユーザの選択が完了します。

7. ［次へ］をクリックします。

［アサーションの設定］ダイアログ ボックスが表示されます。

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 クラス	有効なエントリ
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))
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。

ユーザ クラス	有効なエントリ
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、パートナーはローカル ユーザ ディレクトリ内のユーザを見つけられる必要があります。 ユーザ ディレクトリでユーザを特定することは明確化のプロセスです。 ［ユーザ識別］ダイアログ ボックスでユーザ明確化用の ID 属性を設定します。

CA SiteMinder® Federation Standalone は明確化プロセスのために以下のいずれかの方法を使用できます。

- 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 クエリによって取得された値を使用します。

  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を特定したら、CA SiteMinder® Federation Standalone がユーザ ストア内のユーザを見つけるために使用する検索仕様にこの属性を含めます。明確化プロセスが成功した後で、CA SiteMinder® Federation Standalone 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 許可/作成機能 (P. 197)を設定することもできます。

依存パーティはシングル サインオンを開始して、アサーティング パーティに認証リクエスト（AuthnRequest）を送信できます。 このリクエストで、依存パーティはアサーティング パーティに、アサーションに特定のユーザ属性を含めるように求めることができます。 ただし、必要な属性の値がアサーティング パーティ ユーザ レコードで使用できない場合があります。

依存パーティからの認証リクエストに許可/作成属性が含まれ、アサーティング パーティが新しい ID を作成するように設定されている場合、アサーティング パーティは名前 ID として一意の値を生成します。 この値はアサーションに配置され、依存パーティに送信されます。

［ユーザ識別］ダイアログ ボックスで、SiteMinder コネクタを有効にすることもできます。

SiteMinder コネクタは、CA SiteMinder® Federation Standalone に含まれるソフトウェア コンポーネントです。このコネクタは、展開された SiteMinder システムが CA SiteMinder® Federation Standalone と統合することを可能にします。依存パーティで SiteMinder と CA SiteMinder® Federation Standalone を統合した場合、SiteMinder は、ユーザが SiteMinder 保護リソースをリクエストするときに、CA SiteMinder® Federation Standalone によって認証されたユーザの認証情報を再要求することはありません。 コネクタ、およびポリシー サーバのカスタム SiteMinder 認証方式により、CA SiteMinder® Federation Standalone によって認証されたユーザの SiteMinder セッションの設立が可能になるので、ユーザ認証情報は再要求されません。

パートナーシップ単位で SiteMinder コネクタを有効にできますが、ただ 1 つのグローバル SiteMinder コネクタ設定がすべてのパートナーシップに適用されます。［展開設定］内のチェック ボックスがオンになっており、設定が定義されている場合に限り、コネクタは使用可能になります。UI 内の［インフラストラクチャ］タブから［展開設定］にアクセスします。 コネクタをグローバルに有効にした後で、CA SiteMinder® Federation Standalone はパートナーシップ設定を評価して、コネクタが有効かどうかを判断します。 パートナーシップはグローバル コネクタ設定を使用します。

パートナーシップのコネクタを無効にするには、パートナーシップ レベルでチェック ボックスをオフにします。 コネクタをグローバルに無効にするには、［展開設定］で無効にします。

重要: コネクタがグローバル レベルで無効になっている場合、CA SiteMinder® Federation Standalone はパートナーシップ レベルでチェック ボックスを無視します。

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:

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. 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
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 フォーマット］エントリは永続識別子である必要があります。 名前 ID のこの新しい値は、アサーティング パーティが依存パーティに返すアサーションに含まれます。

3. LDAP または ODBC の検索仕様を指定します。両方のディレクトリが存在する場合は、両方の検索仕様を設定します。

   LDAP の例

   ou=%s,o-ca

   ODBC の例

   name=%s

   ODBC 検索指定フィールドで、検索文字列内の %s を置換するユーザ ストアの値には、等号（=）を含めることができます。 値に等号が含まれる場合は、エントリの先頭に値 **user=** を追加します。たとえば、ユーザ ストアの ElectronicMail の値が CN=catechnologies である場合は、ODBC 検索指定フィールドに「**user=ElectronicMail=%s**」を入力します。user= を追加すると、ポリシー エンジンは文字列を正しく解釈できます。

4. （オプション） SiteMinder コネクタを設定します。

   a. CA SiteMinder® Federation Standalone が既存の SiteMinder 展開と統合している場合は、チェック ボックスをオンにして SiteMinder コネクタを有効にします。

   b. （オプション） CA SiteMinder® Federation Standalone または SiteMinder がユニバーサル ID を使用してユーザ レコードを取得するように、［UserDN とディレクトリ名の比較の実行］をオフにします。 ユニバーサル ID を使用すると、物理的に異なるさまざまなタイプのユーザ ディレクトリが可能になります。ユニバーサル ID を使用することにより、取得されたユーザ レコードを正しいレコードと見なすことができます。

      注: ユニバーサル ID に基づいている場合、各ユーザは一意のユニバーサル ID を持っている必要があります。ユニバーサル ID が一意でない場合、ユーザ レコードにアクセスするシステムは不正な記録を取得する可能性があります。

      チェック ボックスをオン（デフォルト）にしておく場合、CA SiteMinder® Federation Standalone と SiteMinder は同じ物理ディレクトリを使用する必要があります。 これらの両方のディレクトリの名前は、ユーザ ストア検索に対して同じである必要があります。ユーザを認証するエンティティは、ユーザが指定した情報をユーザ レコードの UserDN およびディレクトリ名と比較します。

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=example.com です。

   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. （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 Standalone 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。

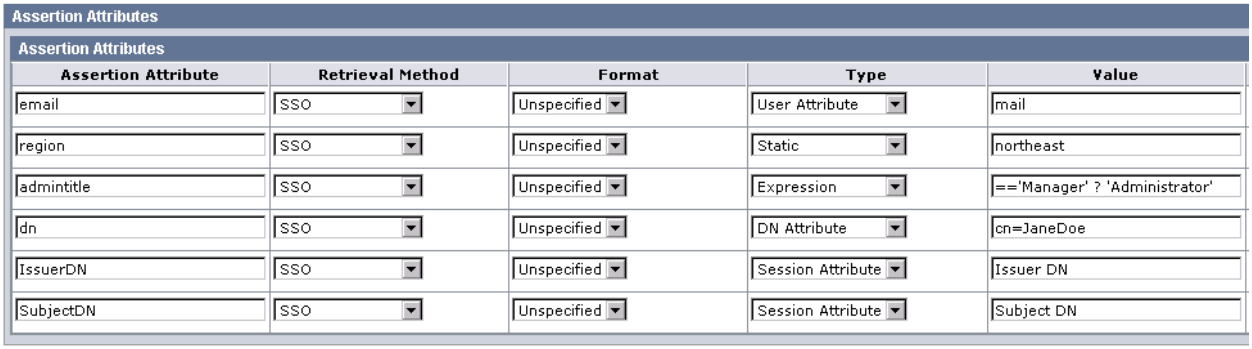
   プラグインの作成については、「*Programming Guide for the Federation Java SDK*」を参照してください。

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=example.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。

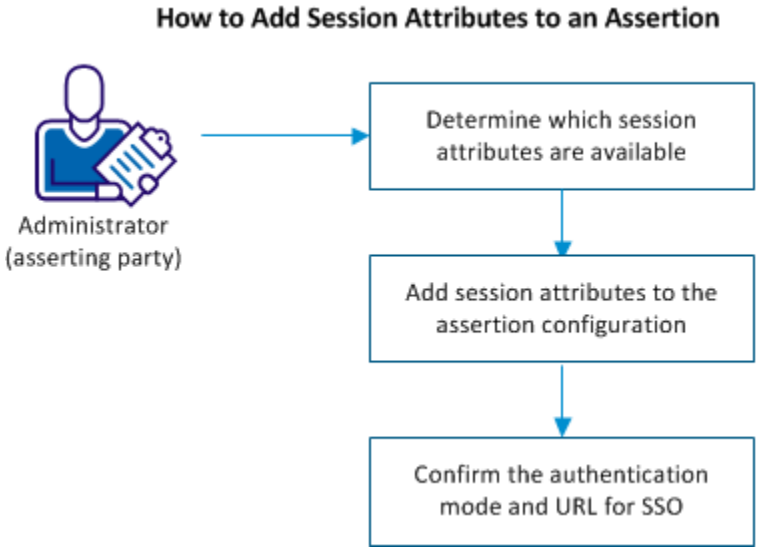
- 委任認証以外の展開。

  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. 204）
2. セッション属性をアサーション設定に追加します。（P. 205）
3. SSO の認証モードと URL を確認します。

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. Administrative 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

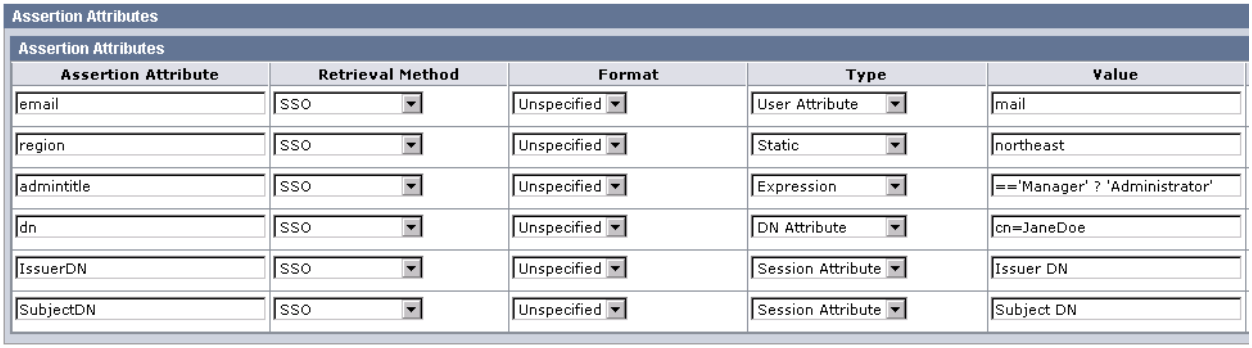
   値

   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。

Administrative 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。

3. ［確認］手順に移動し、［完了］をクリックします。

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。

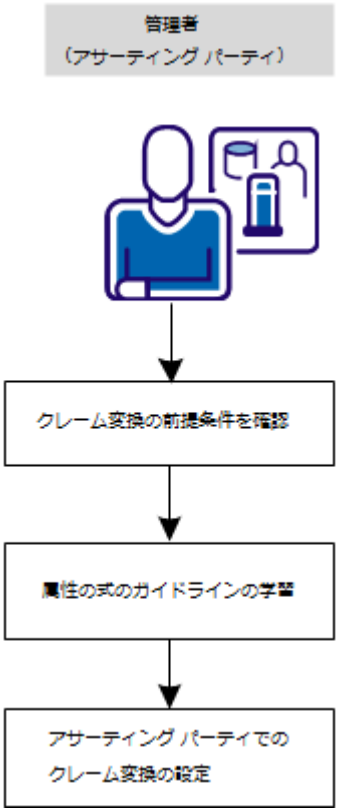
アサーション属性を変更すると、依存パーティがユーザ情報を適用し、ターゲット アプリケーションで使用できるようになります。 たとえば、クレーム変換によって別のドメインにある別のパートナーでロールを関連付けることができます。あるドメインで、ユーザはエンジニアリング マネージャであり、EngineerAdmins という名前のグループに属しているとします。 ただし、依存パーティでは同じロールを DevelAdmins として識別します。 アサーティング パーティは、アサーションを発行する前にロール属性を変更します。 このユーザは、依存パーティのアプリケーションで認識できる DevelAdmins ロ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. 209)
2. 属性式のガイドラインについて学習します (P. 209)。
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. 211)。

クレーム変換の前提条件

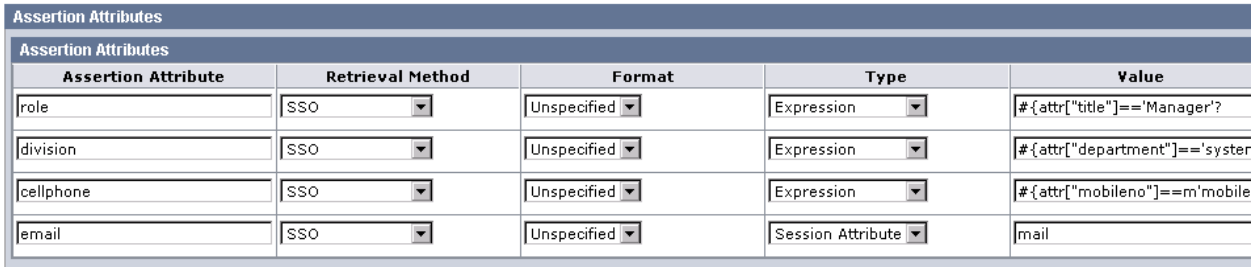
クレーム変換を設定する前に、以下の前提条件を確認してください。

- 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。

Administrative 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。

オペレータ	意味
!=	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」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ら、アサーションが変更され、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. Administrative 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］をクリックします。
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存在する属性はアサーション内に残りますが、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. 213)
   - アサーションへのクレームの追加 (P. 215)。
   - アサーションからのクレームの削除 (P. 216)。

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。

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

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

アサーション ジェネレータ プラグインを使用して、アサーションの内容を変更できます。 プラグインでは、パートナーとの間の業務契約およびベンダーとの間の業務契約に基づいて、アサーションの内容をカスタマイズできます。 パートナーごとに、1 つのプラグインが許可されます。

アサーション ジェネレータ プラグインを設定する手順は、以下のとおりです。

1. CA SiteMinder® Federation Standalone SDK をインストールします（未インストールの場合）。
2. AssertionGeneratorPlugin.java インターフェースを実装します（CA SiteMinder® Federation Standalone SDK に含まれています）。
3. アサーション ジェネレータ プラグイン実装クラスを展開します。
4. Administrative UI 内でアサーション ジェネレータ プラグイン パラメータを設定します。

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 Standalone 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. 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、プロトコル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。 これらのバイナリはフェデレーション 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 Standalone 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 Standalone 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、Administrative UI 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 Standalone がプラグインの検索場所を認識できます。

プラグインを展開するまで、プラグイン設定を実行しないでください。

次の手順に従ってください:

1. Administrative 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 Standalone 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 Standalone 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 Standalone 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: 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 Standalone 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 Standalone 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 Standalone 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. Administrative 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。

アサーション処理のカスタマイズ（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 Standalone 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 Standalone 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 Standalone 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. Administrative 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¥Federation Standalone¥sdk¥java¥sample

パッケージ名は com¥ca¥federation¥sdk¥plugin¥sample です。

UNIX

/FederationStandalone/sdk/java/sample

パッケージ名は com/ca/federation/sdk/plugin/sample です。

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、Administrative 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 Standalone にプラグインの検索場所が指定されます。

プラグインを展開するまで、プラグイン設定を実行しないでください。

次の手順に従ってください:

1. Administrative 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。

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 Standalone 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 Standalone 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_install_dir*¥siteminder¥config 内の JVMOptions.txt ファイルを見つけます。

   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。

第 12 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）

アサーティング パーティでシングル サインオンを設定する場合、アサーティング パーティがアサーションを依存パーティに送信する方法を指定します。

ブラウザでは、1 つのシングル サインオン セッションのみが維持されます。セッション情報は FEDSESSION Cookie に格納されます。 同じブラウザで別のパートナーシップにアクセスする場合、同じブラウザ セッション中に、基礎となるユーザ ディレクトリが、以前にアクセスされたパートナーシップと同じでないかぎり、FEDSESSION Cookie は有効ではありません。

FEDSESSION Cookie は以下のタイムアウト設定を使用します。

- アイドル タイムアウト： 3600 秒（1 時間）
- 最大タイムアウト： 7200 秒（2 時間）

UI でこれらのタイムアウト設定を変更することはできません。

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

   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。

2. ［認証］グループ ボックスの［認証モード］のオプションを選択します。

   認証モード

   ［ローカル］または［委任］を選択します

   - フェデレーション システムがユーザ認証を処理している場合は、［ローカル］をクリックします。
   - サードパーティ Web アクセス管理（WAM）システムがユーザ認証を処理している場合は、［委任］をクリックします。

3. 選択した認証モードの［認証タイプ］を選択します。オプションはローカル認証を使用しているか委任認証を使用しているかによって異なります。

ローカル認証タイプ（ローカル モードのみ）

ベーシックまたはフォーム ベースの選択

日本語ユーザまたはフランス語ユーザ向けにローカライズされる CA SiteMinder® Federation Standalone を使用している場合は、フォーム ベースの認証方式を選択します。 ベーシック認証は、ローカライズされたユーザに対してはサポートされていません。

フォーム認証では、日本語およびフランス語に対してサンプル ログイン フォームが使用可能です。 このフォームは、ディレクトリ *federation_install_dir*/secure-proxy/proxy-engine/examples 内のフォルダ formsja（日本語）および formsfr（フランス語）にあります。

ローカライズされたフォームを使用する方法

a. *federation_install_dir*/secure-proxy/proxy-engine/examples に移動します。

b. フォーム フォルダのバックアップ コピーを作成します。

c. 言語用のフォルダの名前（日本語では formsja、フランス語では formsfr）を **forms** に変更します。

委任認証タイプ

［レガシー Cookie］、［クエリ文字列］、［オープン形式の Cookie］を選択する

注: オープン形式の Cookie は委任認証の唯一の FIPS 互換オプションです。

4. 委任認証の場合のみ、選択した委任認証のタイプに必要なパラメータを設定します。

レガシー Cookie

ユーザ ID 情報が Cookie でサードパーティ WAM から渡されている場合は、［委任された認証 URL］を設定します。ユーザが最初に CA SiteMinder® Federation Standalone にアクセスした場合、この URL は、WAM システムにリクエストをリダイレクトします。ユーザが初めて WAM にアクセスする場合、URL は適用されません。

クエリ文字列

ユーザ ID 情報がクエリ文字列内のサードパーティ WAM から渡されている場合は、以下を設定します。

- 委任認証 URL

  ユーザが最初に CA SiteMinder® Federation Standalone にアクセスした場合、この URL は、WAM システムにリクエストをリダイレクトします。 ユーザが最初に WAM にアクセスした場合、この URL は適用されません。

- ハッシュ秘密キー
- ハッシュ秘密キーの確認

オープン形式の Cookie

ユーザ ID 情報が FIPS 暗号化 Cookie 内のサードパーティ WAM から渡されている場合は、［委任された認証 URL］を設定します。オープン形式の Cookie は委任認証用の唯一の FIPS 互換オプションです。 ユーザが最初に CA SiteMinder® Federation Standalone にアクセスした場合、この URL は、WAM システムにリクエストをリダイレクトします。 ユーザが最初に WAM にアクセスした場合、この URL は適用されません。

注:［委任された認証タイプ］として［レガシー Cookie］または［オープン形式の Cookie］を選択する場合は、必要なグローバル Cookie を設定します。［インフラストラクチャ］-［展開設定］に移動して、展開設定を特定します。

5. ［認証クラス］フィールドに、使用するユーザ認証方法の URI を入力します。 この 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。

   ガイドライン：

   - ユーザがローカルに認証する場合、［パスワード］に対してデフォルトの URI を受け入れます。
   - ユーザがリモート サードパーティで認証する場合は、このフィールドを編集して認証方式を反映するようにします。

6. ［SSO］グループ ボックス内の必須フィールドに入力して、シングル サインオンの動作方法を設定します。

   以下の点に注意してください。

   - Artifact バインディングを選択する場合、Artifact エンコーディング（URL または FORM）を選択します。 エンコーディングは、Artifact がどのように依存パーティに戻るかを定義します。 URL オプションを選択する場合、Artifact が URL 内のクエリ パラメータとして送り返されます。 FORM を選択する場合、Artifact がフォーム データとしてポストされます。
   - SAML 2.0 に対して両方のバインドを選択できます。 バインドの試行シーケンスはローカル エンティティによって決定されます。

     注：Artifact バインディングに対しては、アサーションは安全なバックチャネルを介して送信されます。 そのため、［バックチャネル］グループ ボックスを設定します。

- SSO バインディングを選択する場合、一致するバインディングを持つアサーション コンシューマ サービスを少なくとも 1 つ設定します。 拡張クライアントおよびプロキシ プロファイルを選択する場合、PAOS バインディングを持つアサーション コンシューマ サービスが必要です。
- ［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ためにアサーション有効期間 (P. 236)に関する情報を参照してください。

7. アサーション コンシューマ サービスの URL を指定します。 このサービスは、受け取ったアサーションを処理する依存パーティでのサービスです。

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、すでに入力されています。

この手順により、アサーティング パーティの SSO 設定が完了します。

詳細情報:

シングル サインオンのアサーション有効期間 (P. 236)
分散代行認証 (P. 285)

HTTP-POST SSO 用の AutoPOST フォームのカスタマイズ

ユーザ操作性を高めるために、SAML レスポンスで依存パーティに送信された自動 POST フォームをカスタマイズできます。

カスタマイズされたフォームを使用するには、ウィザードの［SSO と SLO］手順の［SSO］セクションにある［カスタム Post フォーム］フィールドに、フォームの名前を入力します。システムは、レスポンスで指定したフォームを使用します。 製品には、defaultpostform.html という名前のフォームが含まれます。

注: フォームへのパスではなくフォームの名前のみを入力します。

物理的なページがディレクトリ *federation_install_dir*¥customization に存在する必要があります。ここで、*federation_install_dir* は製品のインストール場所です。

パートナーシップ フェデレーションを使用する認証オプション

スタンドアロン 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選択できます。 アサーティング パーティでシングル サインオン設定の一部としてモードを選択します。

- **ローカル認証モード**

  ローカル認証は、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。

- **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 は、ユーザを認証した後、ユーザの認証を最初に求めたエンティティに、フェデレーション ユーザ ID を返送します。

シングル サインオン設定（依存パーティ）

依存パーティでシングル サインオンを設定するには、依存パーティによってサポートされる SAML バインディング、および依存パーティによるシングル サインオン通信の処理方法の関連する特徴を指定します。

CA SiteMinder® Federation Standalone が依存パーティにある場合は、パートナーシップに対して設定された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 CA SiteMinder® Federation Standalone が使用する方法を理解するために、アサーション有効期間 (P. 236)についての詳細を参照してください。

次の手順に従ってください:

1. パートナーシップ 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

2. 使用しているプロファイルに対して、［SSO］グループ ボックスを設定します。

   SAML 2.0 では、Artifact および POST の両方を選択できます。バインドの試行シーケンスはローカル エンティティによって決定されます。

   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。

3. HTTP-Artifact を選択する場合は、送信バックチャネルの認証方式も設定します。

この手順により、依存パーティの SSO 設定が完了します。

シングル サインオンのアサーション有効期間

シングル サインオンの場合、スキュー時間および SSO 有効期間の値が、CA SiteMinder® Federation Standalone によるアサーションの合計有効時間の計算方法を決定します。 CA SiteMinder® Federation Standalone 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、有効間隔の開始および終了は NotBefore および NotOnOrAfter の値で表します。

アサーティング パーティでは、CA SiteMinder® Federation Standalone がアサーション有効期間を設定します。 CA SiteMinder® Federation Standalone は、アサーション生成時のシステム時間を取得することで、有効間隔の開始を決定します。 CA SiteMinder® Federation Standalone は、この時間に基づいてアサーションの IssueInstant の値を設定します。 その後、CA SiteMinder® Federation Standalone は IssueInstant 値からスキュー時間値を引きます。 その結果の時間が NotBefore 値になります。

NotBefore=IssueInstant - スキュー時間

有効間隔の終了を決定するために、CA SiteMinder® Federation Standalone は、有効期間の値とスキュー時間を IssueInstant 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。

依存パーティにおいても、CA SiteMinder® Federation Standalone は、アサーティング パーティで実行する場合と同じ計算を実行し、受信したアサーションが有効かどうか判別します。

CA SiteMinder® Federation Standalone がパートナーシップの両側にある場合のアサーション有効期間の計算

CA SiteMinder® Federation Standalone がパートナーシップの両側にある場合、アサーションが有効な総時間は、SSO 有効期間とスキュー時間の 2 倍の合計です。 計算式は次のとおりです。

アサーション有効期間 = 2 x スキュー時間（アサーティング パーティ） + SSO 有効期間 + 2 x スキュー時間（依存パーティ）

式の前半部分（2 x スキュー時間 + SSO 有効期間）は、アサーティング パーティにおける有効期間ウィンドウの開始と終了を表します。 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: CA SiteMinder® Federation Standalone の場合、［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

依存パーティは NotBefore および NotOnOrAfter の値を取得し（つまり、アサーションで受け取り）、これらの値にスキュー時間を適用して新しい NotBefore および NotOnOrAfter 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. Administrative 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。

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. Administrative 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。

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

   Administrative 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。

HTTP Artifact バック チャネルの設定

アサーティング パーティがアサーションを依存パーティに送信する際に使用する HTTP Artifact バック チャネルを保護します。

以下の制限を考慮してください。

ServletExec を実行する以下の Web サーバではクライアント証明書認証を使用できません。

- SiteMinder プロデューサ/ID プロバイダの IIS Web サーバ（IIS の制限のため）。
- SiteMinder プロデューサ/ID プロバイダの SunOne/Sun Java サーバ Web サーバ（ServletExec に記載された制限のため）。

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［シングル サインオン］または［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. ［SSO］セクションで［HTTP-Artifact］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. 受信または送信バック チャネル、または両方に対して認証方法のタイプを選択します。

   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。

   - クライアント証明書認証方式を選択する場合は、秘密キー/証明書ペアを証明書データ ストアに追加します。秘密キー/証明書ペアは認証局から発行されます。

     重要: 証明書のサブジェクトの CN は、プロデューサで設定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内のパートナーシップ名と同じである必要があります。

     証明書を追加する手順については、「ポリシー サーバ設定ガイド」を参照してください。キー/証明書ペアがすでにデータ ストアにある場合は、この手順をスキップします。
   - 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。セキュリティ強化のために接続の両側で SSL を有効にすることができます。

SAML 2.0 属性クエリのサポートを有効にする方法

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. 247)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。

以下の手順を実行します。

- 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. 246)。
- ID プロバイダで、SAML 2.0 属性機関を設定します (P. 246)。

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. 133)。
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. 91)。
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. 179)。
4. SAML 2.0 属性機関を設定します (P. 246)。

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. Administrative 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］をクリックします。
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。

   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. （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。

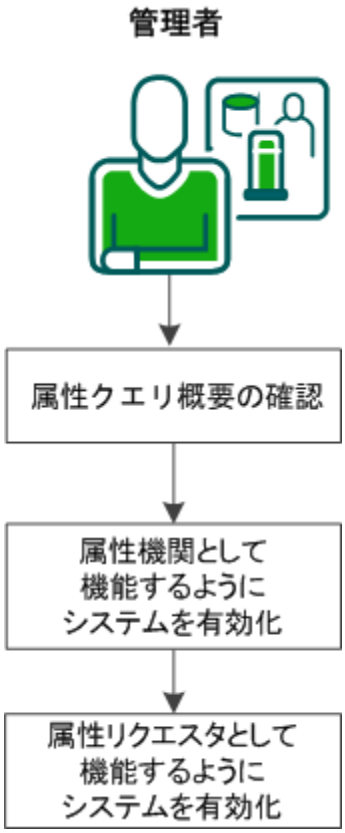
10. パートナーシップを保存してアクティブにします。

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。

サードパーティのソースからユーザ属性値を取得する方法

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。

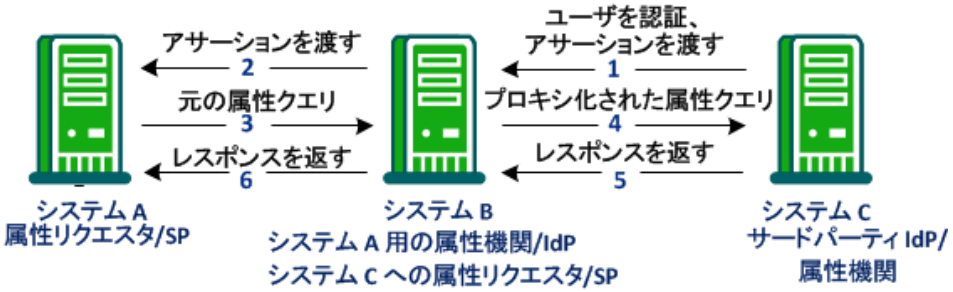
1. プロキシ化された属性クエリの概要を確認します。 (P. 248)
2. 属性機関として機能するシステムを有効にします (P. 250)。
3. 属性リクエスタとして機能するシステムを有効にします (P. 251)。

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. Administrative 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］をクリックします。
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。

   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. （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. Administrative 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］をクリックします。
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。

   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。
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。

アサーションを送信するためにユーザ許諾を取る方法

フェデレーション パートナーシップは、2 つのパーティ間の信頼に依存します。 信頼関係の一部には、依存パートナーに ID 情報を渡すユーザ権限を持つなどの契約上の要件があります。 さらに、リクエストされたサービスについて ID 情報を交換するかどうかを制御するユーザは信頼関係の強化に役立ちます。

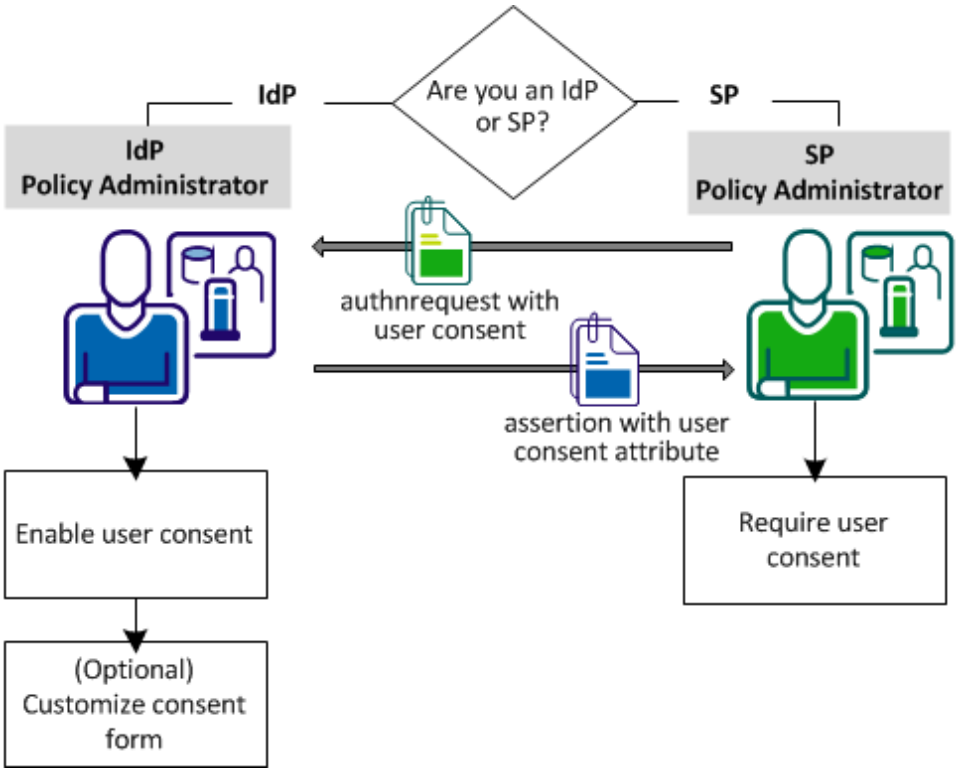
アイデンティティ プロバイダとして動作するフェデレーション システムは SAML 2.0 ユーザ許可機能をサポートします。 アイデンティティ プロバイダ サイトでのユーザ許可では、アイデンティティ プロバイダが、アサーションをパートナーに送信する前に、ユーザに許可を求める必要があります。 アイデンティティ プロバイダでユーザ許可を有効にしていると、アイデンティティ プロバイダがユーザに許可を求めます。 アイデンティティ プロバイダは、アサーション内に許可値を渡します。

許可の有効期間は 5 分間です。 アイデンティティ 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、ユーザは、許可を付与するまでに 5 分間与えられます。 その後、ユーザはアイデンティティ プロバイダにリダイレクトされます。 その後、アイデンティティ 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、アイデンティティ 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。 アイデンティティ 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 分の有効期間が切れる前に、同じユーザはアイデンティティ プロバイダに戻れますが、アイデンティティ プロバイダはユーザにさらに許可を求めます。

注: 有効期間は設定できません。

この図は、各パートナーでの設定タスクを示しています。



IdP での設定タスクは次のとおりです。

1. IdP でユーザ許諾を有効にします (P. 254)。
2. ユーザ許諾フォームをカスタマイズします（オプション） (P. 255)。

SP での設定タスクは次のとおりです。

1. SP でユーザ許可を必要とします。 (P. 256)

ユーザ許可の例

次のユース ケースに、ユーザ許可を示します。

ユーザ 1 は 2:00PM に MyWorkPlace.com にログインして認証します。MyWorkPlace はアイデンティティ 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。有効期間が切れたので、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 までにアイデンティティ プロバイダにリダイレクトされれば、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。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、有効期間はまだアクティブです。

IdP でのユーザ許諾の有効化

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。

- Administrative UI 内でユーザ許諾を有効にします。
- ユーザ許可フォームの名前を入力します。

  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。

Administrative UI を使用して、アイデンティティ プロバイダでユーザ許諾を設定します。UI を使用してこの機能を設定する場合は、アサーション レスポンスで以下の URI のみが使用されます。

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

また、CA SiteMinder® Federation Standalone Java または .NET SDK を使用してこの機能を有効にすることもできます。SDK は、委任された認証を実行しているサードパーティから受信するすべてのユーザ許諾値を渡します。

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アイデンティティ プロバイダに要求できます。

1. Administrative 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］に移動します。
3. 変更する 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. ［SSO］セクションで、以下の操作を実行します。
   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。
6. 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。

ユーザ許可フォームのカスタマイズ（オプション）

SiteMinder には、ca_defaultconsentform.html という名前の *フェデレーションの許可*フォームが付属しています。 アイデンティティ プロバイダは、ユーザにカスタム フォームを送信して、そのユーザのアサーションを送信する権限を取得します。 デフォルトの許可フォームは次の場所にあります。

Windows： %FEDROOT%¥customization

UNIX： $FEDROOT/customization

FEDROOT はシステム環境変数です。

デフォルト許可フォームを使用する代わりにカスタム フォームに書き込むことができます。

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. フォームをカスタム ディレクトリに配置します。
3. Administrative UI で［ユーザ許諾 Post フォーム］フォームの場所を指定します。

SP でユーザ許可を必要とする

SP は、IdP によって返されたアサーション レスポンスに、ユーザ許可属性が含まれていることを必要とすることがあります。 認証リクエストにこの属性を含めるには、Administrative UI で設定を有効にします。

次の手順に従ってください:

1. Administrative UI にログインします。
2. 該当する SP から IdP へのパートナーシップを変更します。
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. ダイアログ ボックスの［SSO］セクションの［ユーザ許可が必要］設定を選択します。

   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。

5. 他の変更がない場合は、［確認］手順を選択し、［完了］をクリックして、変更を保存します。

ユーザ許可属性が IdP に送信される認証リクエストに配置されます。

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。

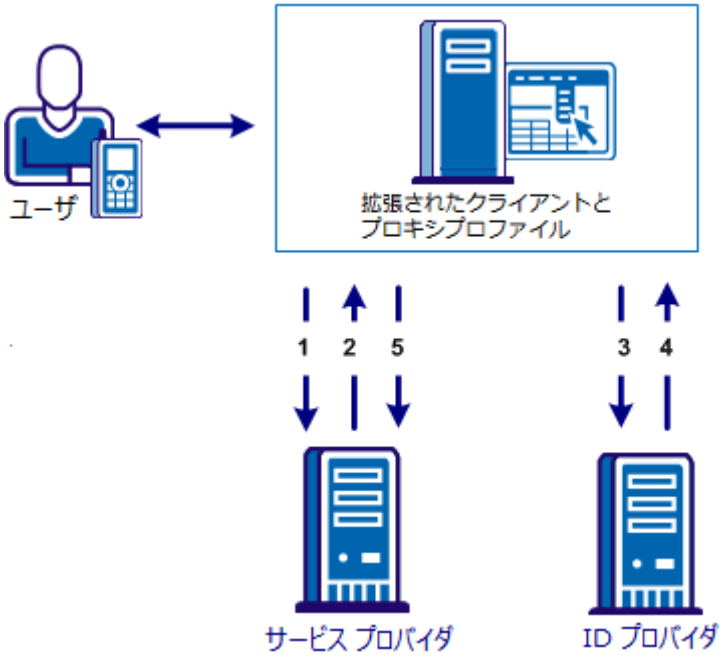
ECP プロファイルは、アイデンティティ プロバイダとサービス プロバイダが直接通信できない場合に、シングル サインオンを有効にします。ECP は、サービス プロバイダとアイデンティティ プロバイダの間で仲介する機能を果たします。

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。 CA SiteMinder® Federation Standalone 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。 CA SiteMinder® Federation Standalone アイデンティティ 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. Administrative 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. Administrative 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。

アイデンティティ プロバイダ ディスカバリ プロファイルを有効にする方法

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。 CA SiteMinder® Federation Standalone の場合、この URL は以下のようになります。

   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

   CA SiteMinder® Federation Standalone 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。

   CA SiteMinder® Federation Standalone 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。

   federation_install_dir/secure-proxy/Tomcat/webapps/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。

IdP ディスカバリ ターゲットの攻撃からの保護

CA SiteMinder® Federation Standalone アイデンティティ プロバイダ ディスカバリ サービスが共通ドメイン Cookie のリクエストを受け取る場合、リクエストには IPDTarget という名前のクエリ パラメータが含まれています。このクエリ パラメータは、Discovery サービスでリクエストを処理した後にリダイレクトする URL をリスト表示します。

IdP の場合、IPDTarget は SAML 2.0 シングル サインオン サービスです。 SP の場合、ターゲットは共通ドメイン Cookie を使用するリクエスト アプリケーションです。

PDTarget クエリ パラメータをセキュリティ攻撃から保護することが推奨されます。不正なユーザはこのクエリ パラメータに任意の URL を配置し、悪意のあるサイトにリダイレクトさせる可能性があります。

クエリ パラメータを攻撃から保護するには、エージェント設定オブジェクトの設定項目である **ValidFedTargetDomain** を設定します。ValidFedTargetDomain パラメータ は、フェデレーション環境の有効なドメインのすべてをリストします。

注: ValidFedTargetDomain 設定は、Web エージェントが使用する ValidTargetDomain 設定に類似していますが、この設定は特にフェデレーション用に定義されます。

IPD サービスが IPDTarget クエリ パラメータを検査する場合、このサービスはクエリ パラメータによって指定された URL のドメインを取得します。IPD サービスでは、このドメインを、ValidFedTargetDomain パラメータで指定されるドメインのリストと比較します。 URL ドメインが ValidFedTargetDomain に設定されたドメインの 1 つと一致する場合、IPD サービスは指定された URL にユーザをリダイレクトします。

ドメインが一致しない場合、IPD サービスはユーザ リクエストを拒否し、ブラウザに 403 Forbidden が返されます。 また、FWS トレース ログおよび affwebservices ログにエラーが報告されます。 これらのメッセージは、IPDTarget のドメインが有効なフェデレーション ターゲット ドメインとして定義されないことを示します。

ValidFedTargetDomain を設定しない場合、検証は行われず、ユーザはターゲット URL にリダイレクトされます。

SAML 2.0 HTTP-POST バインディング設定

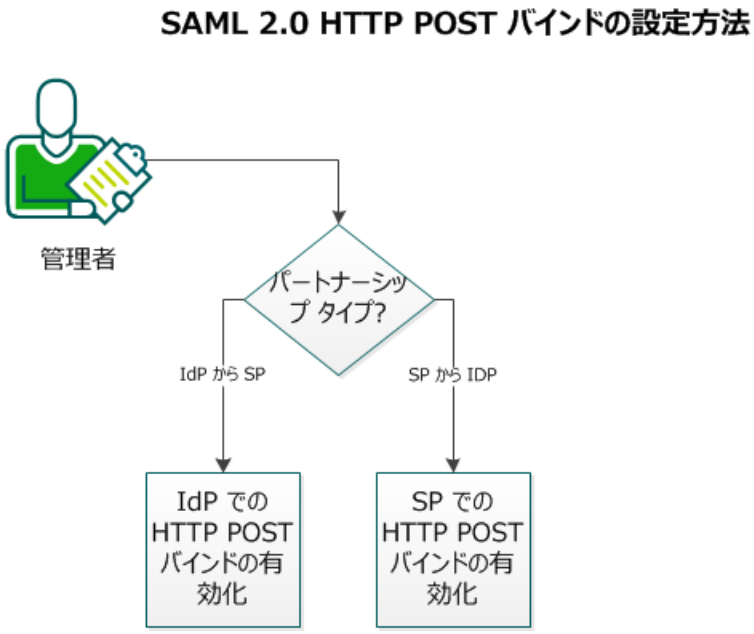
シングル サインオンおよびシングル ログアウトのリクエストの場合、リクエストとレスポンスを交換する方法として SAML 2.0 HTTP-POST バインディングを有効にできます。このバインディングは SAML プロトコルを標準メッセージ形式および通信プロトコルにマップします。

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. 267)。
   - SP において HTTP POST バインディングを有効にします (P. 268)。

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。

Administrative 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. Administrative 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. Administrative 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。

第 13 章: ソーシャル サインオンの設定

CA SiteMinder® Federation Standalone （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

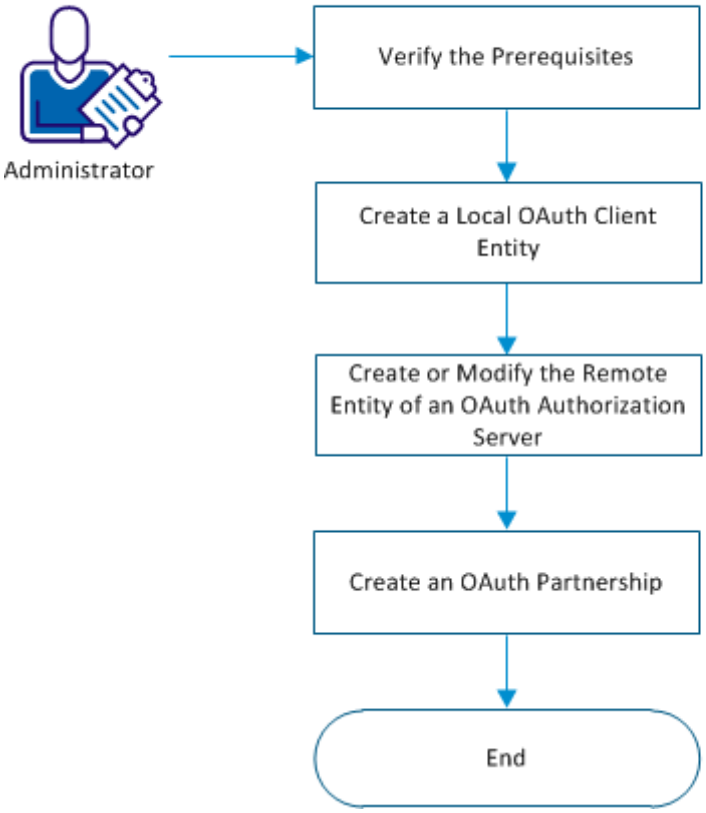
- Facebook
- Google
- LinkedIn
- Windows Live

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. 273)。
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. 274)。
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. 274)。
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. 276)。

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。

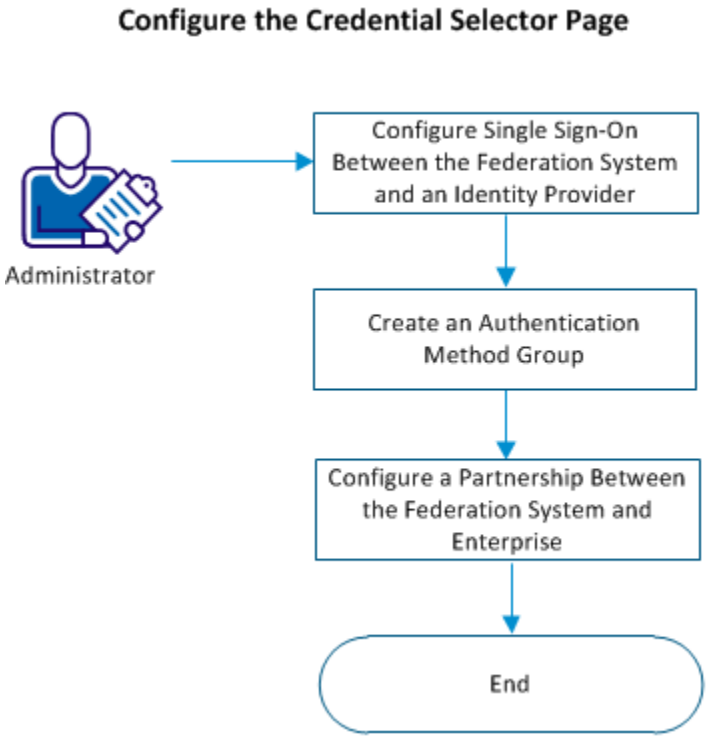
   h. フェデレーション システム（依存パーティ）は、セッションを生成し、ユーザをフェデレーション システム（アサーティング パーティ）にリダイレクトします。

   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. 281)
2. 認証方式グループを作成します (P. 282)。
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. 283)。

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、ユーザの企業の要件にあ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¥Federation Standalone¥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 14 章: 分散代行認証

分散代行認証の概要

ユーザがフェデレーション パートナーシップ用のシングル サインオンを設定する場合、ユーザが決定した設定の 1 つが、ユーザの認証方法を決めます。

CA SiteMinder® Federation Standalone では、認証の選択肢が 2 つあります。

- ローカル認証
- 分散代行認証

CA SiteMinder® Federation Standalone はローカル認証を実行できます。ただし、使用可能な認証方式はベーシックおよび HTML フォームのみです。

分散代行認証によって、CA SiteMinder® Federation Standalone 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 システムは認証を実行し、続いて CA SiteMinder® Federation Standalone にフェデレーション ユーザ ID を転送します。 CA SiteMinder® Federation Standalone はユーザ ID 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 CA SiteMinder® Federation Standalone 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、これ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 CA SiteMinder® Federation Standalone によって開始される認証

CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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、ID 情報が CA SiteMinder® Federation Standalone 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対して直接行われ、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 CA SiteMinder® Federation Standalone に送信します。 CA SiteMinder® Federation Standalone 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、ID 情報が CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone に渡します。

- レガシー Cookie またはオープン形式の Cookie の使用。

  オープン形式の Cookie はデータのセキュリティを保証するために暗号化できます。

- ブラウザを CA SiteMinder® Federation Standalone 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、FIPS 準拠のパートナーシップを作成しません。

  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 CA SiteMinder® Federation Standalone 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

CA SiteMinder® Federation Standalone は、ユーザ ID を渡すためにレガシーまたは オープン形式の Cookie を使用できます。 Cookie には値の 1 つとしてユーザ ログイン ID が含まれます。

注: Windows 認証用の CA SiteMinder® Federation Standalone エージェントを使用する委任認証を設定した場合、エージェントは オープン形式の Cookie を使用することが必要となります。 ただし、SiteMinder コネクタも設定されている場合、委任認証用の オープン形式の Cookie オプションは使用可能ではありません。 CA SiteMinder® Federation Standalone Windows エージェントおよび SiteMinder コネクタは、1 つの展開で共存できません。

WAM システムまたは CA SiteMinder® Federation Standalone で認証を開始できます。認証が CA SiteMinder® Federation Standalone で開始される場合、ユーザは WAM システムへリダイレクトされます。WAM システムでの認証プロセスは、まるで WAM システムで認証が開始されたかのように CA SiteMinder® Federation Standalone と同じです。

委任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、レガシー Cookie または オープン形式の Cookie を作成するために CA SiteMinder® Federation Standalone SDK を使用します。 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 Standalone 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 と同じ言語を使用する必要があります。 CA SiteMinder® Federation Standalone 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 Standalone SDK を使用せずに、オープン形式の Cookie を作成できます。オープン形式の Cookie Cookie を手動で作成するには、UTF-8 エンコーディング、および CA SiteMinder® Federation Standalone がパスワード ベースの暗号化に使用する以下の PBE 暗号化アルゴリズムのいずれかをサポートするあらゆるプログラミング言語を使用します。
     - 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

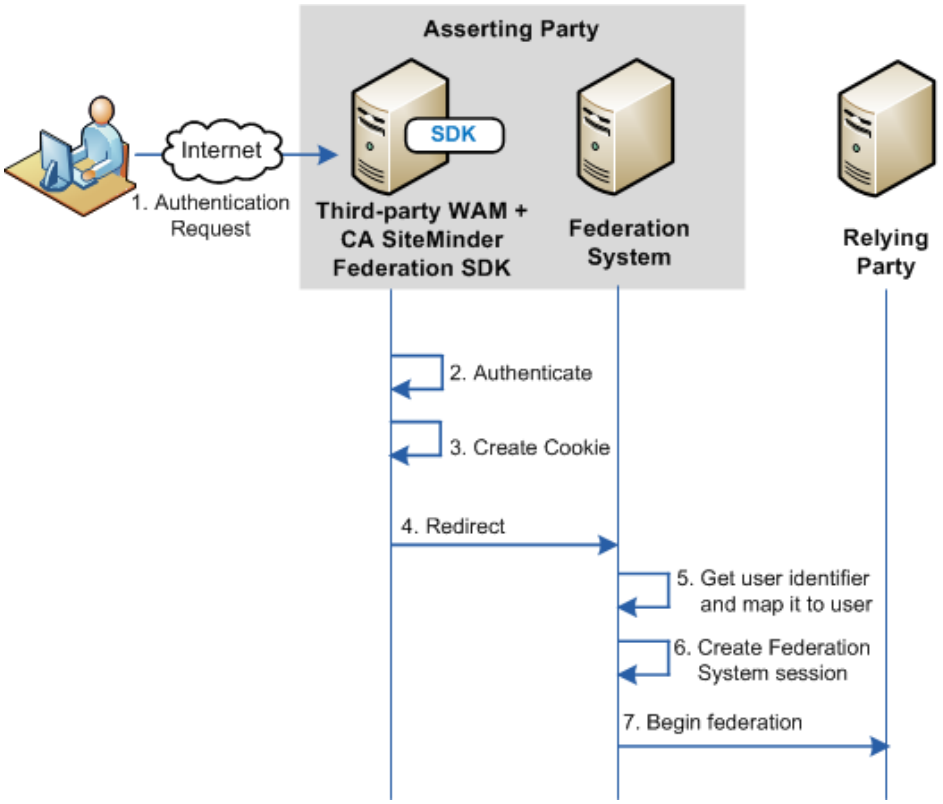
また、ユーザのブラウザで オープン形式の Cookie が設定されると確信する必要があります。

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 (P. 503)」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 CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone にリダイレクトします。
5. CA SiteMinder® Federation Standalone 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. CA SiteMinder® Federation Standalone は CA SiteMinder® Federation Standalone 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。



重要: レガシー Cookie または SDK が作成した 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 Standalone SDK をインストールする必要があります。 SDK は CA SiteMinder® Federation Standalone 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、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。

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、WAM システムから CA SiteMinder® Federation Standalone にユーザを送信するリダイレクト URL 上のクエリ文字列の追加により、ユーザ ID を CA SiteMinder® Federation Standalone へ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 CA SiteMinder® Federation Standalone 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。

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。
- CA SiteMinder® Federation Standalone または依存パーティで認証を開始することもできます。

WAM システムで認証が開始された場合、クエリ文字列を使用する委任認証用のトランザクション フローは以下のとおりです。

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、CA SiteMinder® Federation Standalone 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。

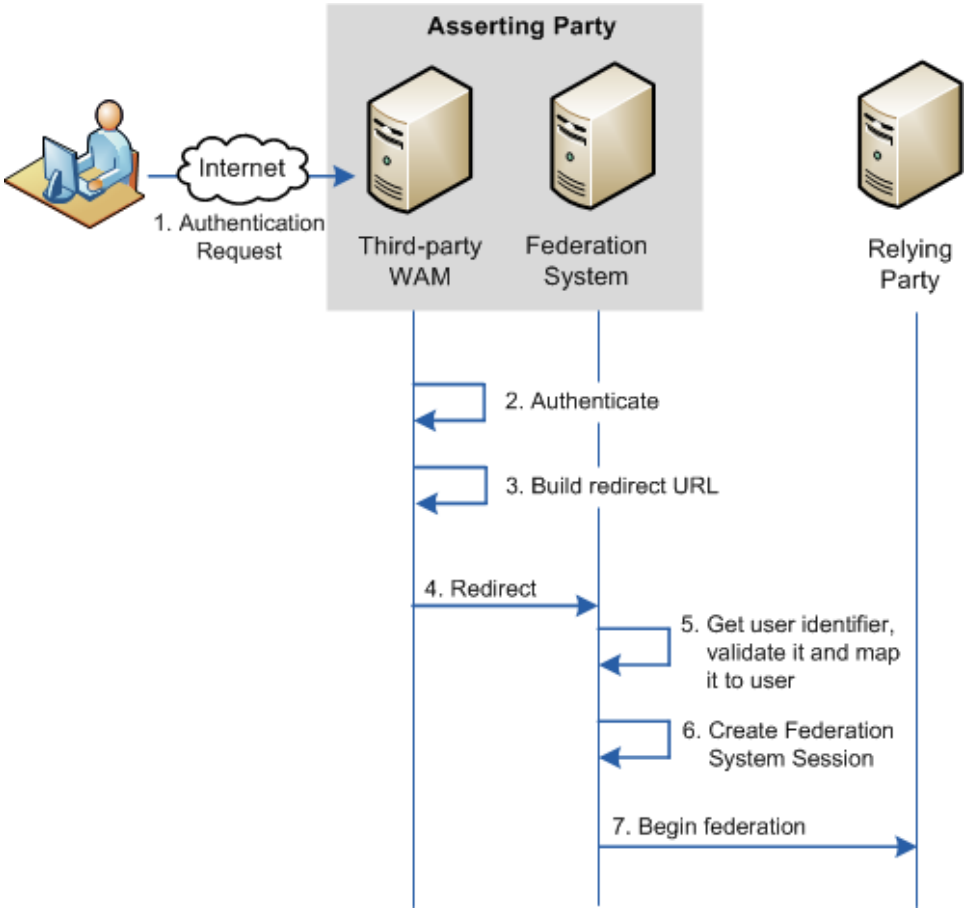
   リダイレクト URL の例

   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 CA SiteMinder® Federation Standalone にリダイレクトします。

5. CA SiteMinder® Federation Standalone 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、ハッシュされた値を使用して識別子を検証し、ユーザ ディレクトリでユーザを特定します。
6. CA SiteMinder® Federation Standalone 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 Standalone 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。

詳細情報

展開設定 (P. 426)

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 Standalone 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 Standalone 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 Standalone 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 Standalone 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 Standalone .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 Standalone Java SDK ガイド」または「CA SiteMinder® Federation Standalone .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 (P. 503)に関する詳細を確認してください。

- サードパーティは、CA SiteMinder® Federation Standalone アサーティング パーティで設定されている以下の Administrative UI 設定（［Cookie ゾーン］および［暗号化パスワード］パラメータ）の値を認識している必要があります。
  - グローバル Cookie ゾーン
  - 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

  これらの値は、Cookie の作成時に使用されます。
- サードパーティ WAM システムは、ユーザを CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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。 CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone から認証リクエストを受信した後、受信認証リクエストの一部として受信する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 CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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

値が、かつてはサードパーティ WAM システムへのユーザを識別したことを指定します。

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 CA SiteMinder® Federation Standalone 設定で指定されます。

CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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: 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。

サードパーティ WAM システムは、ユーザを CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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 システムが CA SiteMinder® Federation Standalone から認証リクエストを受信した後、受信認証リクエストの一部として受信するあらゆる既存のクエリ文字列を忘れずにキャプチャし再び送る必要があります。 受信リクエストにクエリ文字列内の CA SiteMinder® Federation Standalone リクエスト情報がある場合、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 CA SiteMinder® Federation Standalone 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を開始する必要があります。 未承認応答を開始するには、CA SiteMinder® Federation Standalone 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

CA SiteMinder® Federation Standalone 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。

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

シングル サインオンがサービス プロバイダによって開始される場合、このサービス プロバイダは、認証リクエスト メッセージ内に ForceAuthn または IsPassive クエリ パラメータを含めることができます。

注: CA SiteMinder® Federation Standalone アイデンティティ プロバイダは IsPassive クエリ パラメータをサポートしません。ただし、IsPassive パラメータは、サードパーティのサービス プロバイダによって送信された認証リクエスト メッセージに含まれている可能性があります。

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む場合、CA SiteMinder® Federation Standalone 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、認証リクエスト メッセージで ForceAuthn=True を含む場合、CA SiteMinder® Federation Standalone アイデンティティ プロバイダはセッションが存在する場合でも、ユーザに認証情報を要求します。

IsPassive の処理

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付けることができない場合、以下のいずれかの SAML レスポンスがサービス プロバイダに返送されます。

- 認証リクエスト メッセージに IsPassive=True が含まれ、CA SiteMinder® Federation Standalone セッションがない場合、CA SiteMinder® Federation Standalone アイデンティティ プロバイダは CA SiteMinder® Federation Standalone がセッションを必要とするのでエラー メッセージが含まれる SAML レスポンスを返します。
- 認証リクエスト メッセージに IsPassive=True が含まれ、CA SiteMinder® Federation Standalone セッションがある場合、CA SiteMinder® Federation Standalone 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、これは無効なリクエストなので CA SiteMinder® Federation Standalone 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 Standalone 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

CA SiteMinder® Federation Standalone SP が AuthnRequest 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。

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 Standalone 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情報を要求します。 ユーザが正常に認証する場合、IdP はアサーションで既存のセッションからアイデンティティ情報を送信し、認証に対して生成された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 はユーザに認証情報を要求します。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。両者は互いに排他的だからです。 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、パートナーシップ用に定義された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したい場合、ProtocolBinding パラメータが必要です。

- 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。

ProtocolBinding の任意使用

ProtocolBinding クエリ パラメータを使用 *しない* 場合、以下の情報が当てはまります。

- 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。 これは ProviderID クエリ パラメータでアイデンティティ 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、CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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（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 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。

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。

CA SiteMinder® Federation Standalone が IdP にある場合の SLO 設定

CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone 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。

CA SiteMinder® Federation Standalone が SP にある場合の SLO 設定

CA SiteMinder® Federation Standalone 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。 開始 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。 CA SiteMinder® Federation Standalone 用のリンクは以下のとおりです。

http://s*p_host:port*/affwebservices/public/saml2slo

この埋め込みリンクによって、CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone 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 リクエストの期限切れがいつかを特定するために、CA SiteMinder® Federation Standalone 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場合は、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 リダイレクトおよび SOAP 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 CA SiteMinder® Federation Standalone 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。

- CA SiteMinder® Federation Standalone 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することをお勧めします。

  SOAP および HTTP リダイレクトをサポートする環境でシングル ログアウトの管理に関する設定ガイドライン (P. 316)を確認してください。

パートナーシップの一方の側でシングル ログアウトを設定する方法

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。このオプションの設定はパートナーシップ ウィザードの［署名および暗号化］手順にあります。

   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
   - 有効期間
   - リレー状態を使用して SLO 確認 URL をオーバーライドする

   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. ［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。

   SLO サービス URL はシングル ログアウトを開始し、その後、CA SiteMinder® Federation Standalone 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 CA SiteMinder® Federation Standalone にログアウト リクエスト メッセージを送信する場所を示します。

   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。

   テーブルにエントリを追加するには、[行の追加]をクリックします。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。

これらの手順が完了した後で、シングル ログアウトが設定されます。

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して有効化されたシングル 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します。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 により、サーバ上のさまざまな SAML サービスが特定されます。たとえば、シングル サインオン、シングル ログアウト、Artifact 解決サービス（SAML 2.0）、アサーション検索サービス（SAML 1.x）などが特定されます。

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］オプションはテスト目的の場合に使用し、実稼働環境では使用しないでください。ただし、CA SiteMinder® Federation Standalone が SSL 対応のフェールオーバ用に設定されており、かつ、プロキシ サーバの後ろに配置される場合は除きます。 この場合、クライアント証明書認証がバックチャネルを保護するために使用される場合、プロキシ サーバにサーバ証明書があるので、プロキシ サーバが認証を処理します。 この場合、すべての IdP->SP パートナーシップは認証タイプとして［認証なし］を使用できます。

重要: 受信バック チャネル用に選択された認証方法は、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。 認証方法の選択の合意は帯域外で処理されます。

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］グループ ボックスから始めます。
2. ［SLO］グループ ボックスで［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。

   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。

   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。

   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。

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。

詳細情報:

Apache Web Server および UI 用の SSL 管理 (P. 443)

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. Administrative 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?LocalLogout=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、Administrative 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 - SiteMinder コネクタが IdP から SP へのパートナーシップに対して有効な場合にのみ使用できます。

     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 テンプレートの設定

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 は特定のコンテキスト クラスを識別します。 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。

共通の機能に加えて、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 でテンプレートを必要とするのは、次の状況です。

- SiteMinder コネクタが有効にされている。
- IdP が SP リクエストから認証コンテキストを自動検出する。

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。

注: 保護レベルは SiteMinder コネクタを使用している場合のみ使用できます。

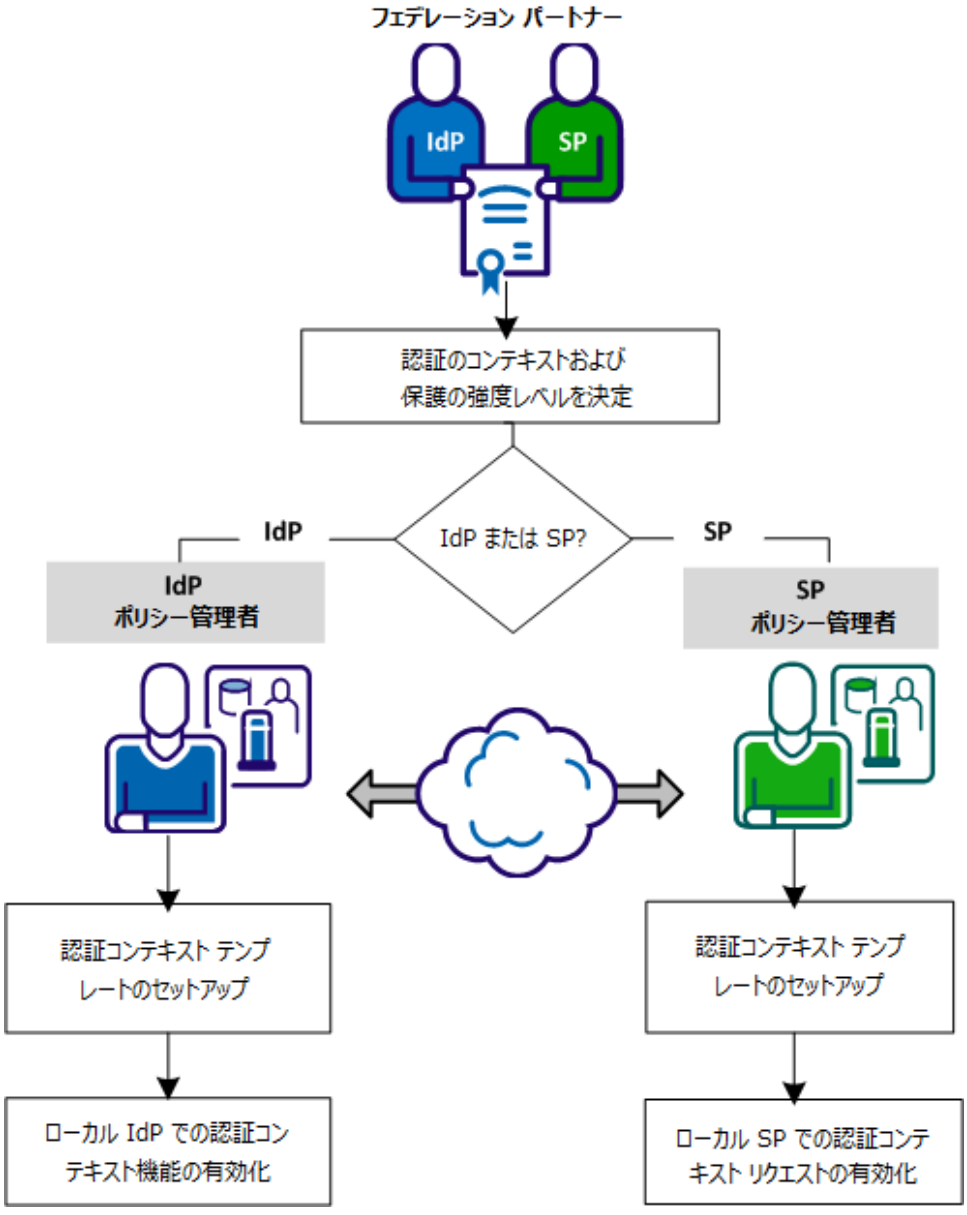
SP

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。

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。 各サイトに 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. 認証コンテキストと強度レベルを決定します。 (P. 333)
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 (P. 333)。
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機能を有効にします (P. 336)。
   - 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 (P. 339)。

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. Administrative UI にログインします。
2. ［フェデレーション］タブから、［AuthnContext 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. SiteMinder コネクタの展開の場合のみ

   a. ［保護レベルの有効化］をクリックします。

   b. 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、Administrative UI によって最小が計算されます。 Administrative 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 (P. 335)についての詳細を参照してください。

10. ［次へ］をクリックしてウィザードの最後の手順に移動します。

11. ［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

SiteMinder コネクタを委任認証に使用するフェデレーション展開では、保護レベルを各認証 URI と関連付ける必要があります。保護レベルは、認証の強度における保証レベルを示します。各保護レベルは URI の強度レベルにマッピングされます。保護レベルの割り当てが SiteMinder 認証方式の保護レベルを反映していることを確認します。

注: SiteMinder コネクタによる展開で、保護レベルは、コネクタ認証方式に指定されたレベルをオーバーライドします。

Administrative UI で保護レベルを割り当てる場合、範囲を指定します。 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計算されます。範囲は設定されている SiteMinder 認証方式を対象にする必要があります。たとえば、SiteMinder が X.509 認証方式を 20 の保護レベルに設定している場合は、CA SiteMinder® Federation Standalone に指定されている範囲に 20 が含まれることを確認します。

保護レベルの例

SiteMinder 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

各保護レベルは URI の強度レベルにマッピングされます。 表に、URI の元のリストを示します。

URI	保護レベル最大	URI 強度
urn:oasis:names:tc:SAML:2.0:ac:classes:X509	1000	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
urn:oasis:names:tc:SAML:2.0:ac:classes:Password	5	1

範囲は、SiteMinder 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

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。以下の変更された表に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機能の有効化

CA SiteMinder® Federation Standalone IdP 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。これは、SiteMinder コネクタが有効にされている場合のみ使用できます。

  システムは認証コンテキスト テンプレートを使用して、自動的に認証コンテキストを検出します。

  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、IdP が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のフローに関する詳細を参照できます。

認証コンテキストを取得する方法を設定します。

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。
   - ローカル認証の場合は、事前定義済み認証クラスを使用する必要があります。
   - SiteMinder コネクタによる委任認証の場合は、事前定義済み認証クラスを選択するか、または認証コンテキスト テンプレートでクラスを自動検出します。
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。

     注: このオプションは SiteMinder コネクタが有効な場合のみ使用できます。
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。
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. Administrative 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

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。［署名］手順（SAML 1.1/WS-フェデレーション）および［署名および暗号化］手順（SAML 2.0）によって、アサーションの署名および暗号化を設定できます。

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。

このセクションには、以下のトピックが含まれています。

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. Administrative 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. Administrative 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: サービス プロバイダでのセッション継続期間管理

このセクションには、以下のトピックが含まれています。

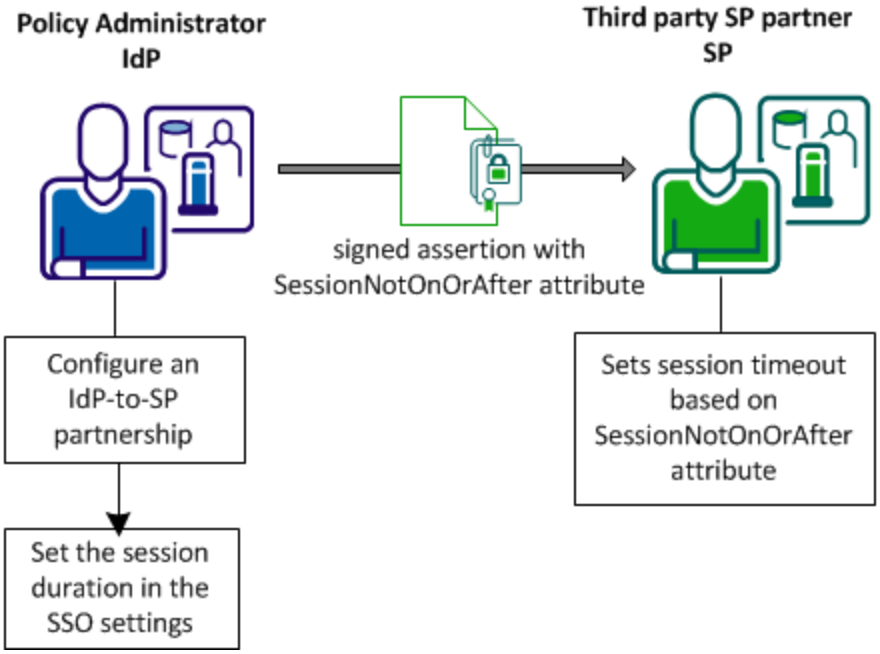
サービス プロバイダで認証セッションの継続期間を管理する方法

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。

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。

セッションの継続期間を指定する値により、SP でのセッションがきわめて短い場合に、ユーザは再度認証することがなくなります。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。 ユーザのシームレスな操作性を実現するため、それに応じて、SP でセッションを管理します。

次の画像は、IdP での設定手順とサードパーティ SP が実行する結果のアクションを示しています。



アサーションにセッション継続期間属性を含める

セッション継続期間の設定は IdP で実行します。 SP に送信されたアサーションには、SP が SP サイトのタイムアウト値を設定するために使用するセッション属性が含まれます。

重要: CA SiteMinder® Federation Standalone 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、SP は、レルム タイムアウトから、ターゲット リソースを保護する SAML 認証方式に対応するセッション 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. Administrative 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、［推奨される SP セッション期間］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、以下のいずれかのオプションを選択できます。
   - 属性を省略する
   - 属性を IdP セッション タイムアウトに設定する
   - 独自の継続期間を指定する

   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。
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。

設定に基づいて、セッション属性がアサーションに配置され、SP に送信されます。

第 20 章: SiteMinder と CA SiteMinder® Federation Standalone の統合

このセクションには、以下のトピックが含まれています。

CA SiteMinder® Federation Standalone および SiteMinder を統合する方法

展開した SiteMinder システムは、SiteMinder コネクタ（CA SiteMinder® Federation Standalone に含まれるソフトウェア コンポーネント）を使用して、CA SiteMinder® Federation Standalone と統合できます。 コネクタにより、フェデレーションと Web アクセス管理展開間の次の対話が可能になります。

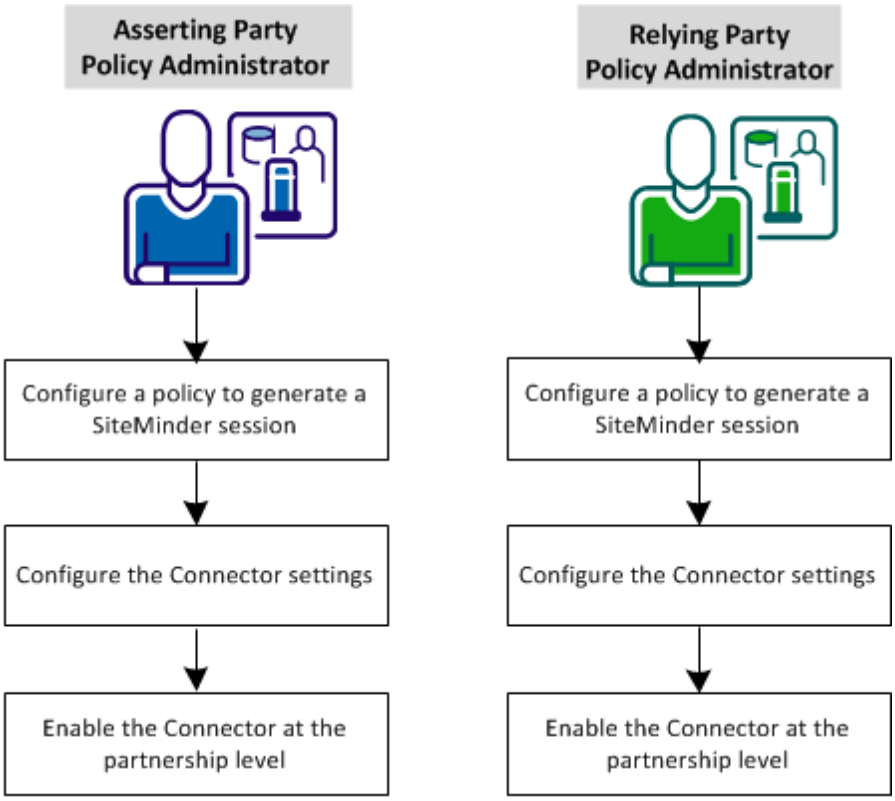
- アサーションを生成するための ID の設定。

  アサーティング パーティで、ユーザは CA SiteMinder® Federation Standalone に到達しますが、セッションがありません。SiteMinder セッションを確立するため、コネクタは SiteMinder と通信します。SiteMinder セッションのコンテンツに基づいて、コネクタはフェデレーション セッションを作成します。このセッション情報を使用して、ユーザーの SAML アサーションが生成されます。

- 認可権限を決定するための SiteMinder の ID の設定。

  依存パーティで、ユーザは CA SiteMinder® Federation Standalone で認証し、フェデレーション セッションが生成されます。 コネクタは、ユーザ名でフェデレーション セッションを SiteMinder に渡し、フェデレーション セッションから SiteMinder セッションを生成します。ユーザが識別されたので、認証情報のために再認証されません。SiteMinder は、依存パーティでリクエストされたリソースの許可権限を決定します。

次の図に、コネクタと統合する場合の設定プロセスを示します。



以下の設定手順を実行します。

1. SiteMinder セッションを生成するポリシーを設定します。
2. コネクタを設定します。
3. パートナーシップ レベルでコネクタを有効にします。

SiteMinder コネクタを使用した SiteMinder との統合

SiteMinder コネクタにより、次の統合が可能になります。

- アサーションを生成するための ID の設定。

  アサーティング パーティで、ユーザは CA SiteMinder® Federation Standalone に到達しますが、セッションがありません。SiteMinder セッションを確立するため、コネクタは SiteMinder と通信します。 セッション情報の使用によって、フェデレーション セッション、およびユーザの SAML アサーションが生成されます。 このアサーションにより、ユーザは依存パーティで保護されているフェデレーション リソースにアクセスできます。

- アサーションからの認可権限の決定。

  依存パーティで、ユーザは CA SiteMinder® Federation Standalone で認証し、フェデレーション セッションが生成されます。 コネクタは、ユーザ名でフェデレーション セッションを SiteMinder に渡し、SiteMinder セッションを生成します。 このセッションの確立によって、保護されているリソースにアクセスするときに、これらのユーザは再認証されません。 ユーザが識別されたので、依存パーティでのユーザのアクセス権限を特定できます。

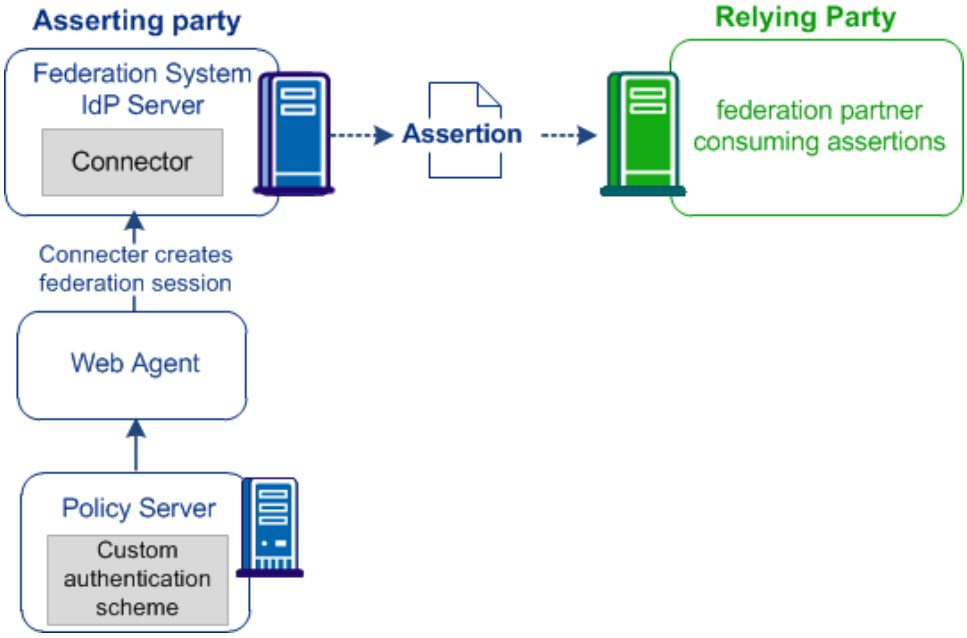
FEDSESSION Cookie は以下のタイムアウト設定を使用します。

- アイドル タイムアウト： 600 秒（10 分）
- 最大タイムアウト： 900 秒（15 分）

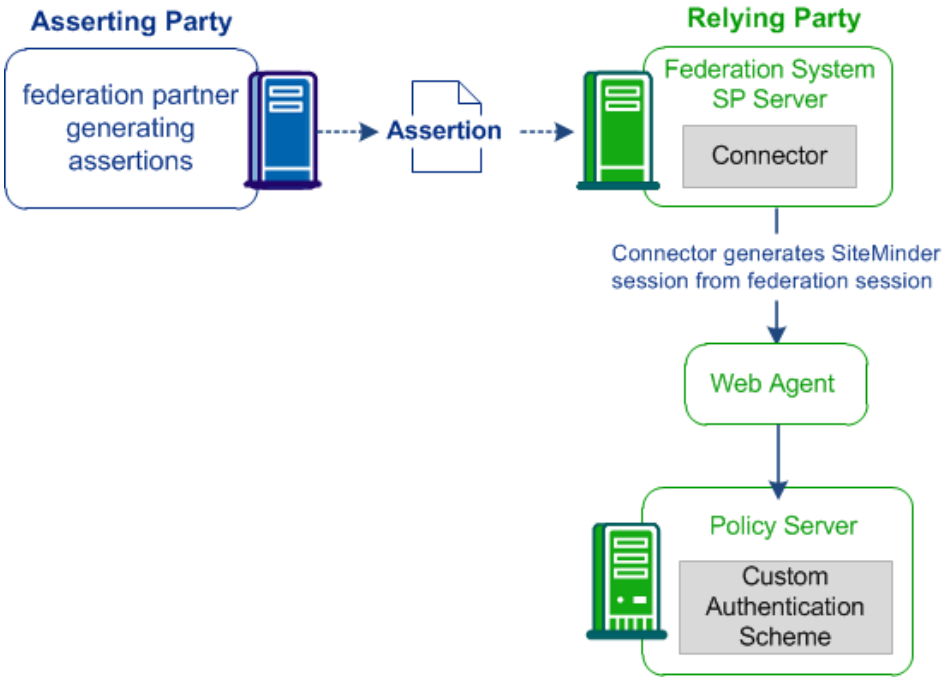
UI でこれらのタイムアウト設定を変更することはできません。

次の図に示すように、コネクタは、SiteMinder 環境、および CA SiteMinder® Federation Standalone 環境で設定を必要とします。

この図は、アサーティング パーティでのコネクタを示しています。



この図は、依存パーティでのコネクタを示しています。



各サイトでセッションを生成するポリシーの設定

SiteMinder コネクタは、CA SiteMinder® Federation Standalone が既存のポリシー サーバで動作することを可能にします。 最初の手順はポリシーを設定することです。アサーティング パーティで、ポリシーはフェデレーション セッションを生成します。依存パーティで、ポリシーは SiteMinder セッションを生成します。 このポリシーは他のポリシーとして機能しますが、その主な目的はリソースの保護ではなくセッションのトリガです。

注: アサーティング パーティおよび依存パーティでポリシーを設定します。

ポリシーでは、典型的なポリシー オブジェクトを設定する必要がありますが、カスタム SiteMinder コネクタ認証方式を適用します。このポリシーはコネクタ セットアップに固有です。

ポリシー サーバ オブジェクトを設定するには、「*ポリシー サーバ設定ガイド*」を参照してください。

重要: コネクタを設定する前に、ポリシー サーバで以下の設定手順に従います。

次の手順に従ってください:

1. フェデレーション システム上で smauthconnectors.zip アーカイブを解凍します。このアーカイブはフェデレーション製品キットに含まれています。
2. SiteMinder 動作環境に対して正しいカスタム認証方式ライブラリを選択します。
   - **Windows:** smauthsmconnector.dll
   - **Solaris/Linux:** libsmauthsmconnector.so

     注: UNIX プラットフォームでは、名前は大文字と小文字が区別されます。
3. SiteMinder システム上の適切なポリシー サーバ ディレクトリにライブラリをコピーします。
   - **Windows:** *policy_server_home*/siteminder/bin
   - **Solaris/Linux:** *policy_server_home*/siteminder/lib
4. SiteMinder Administrative UI にログオンします。
5. フェデレーション システムを表す Web エージェントを作成します。たとえば、Federation Agent という名前を付けます。

   重要: サポート 4.x エージェントに対してオプションを選択しないでください。
6. エージェント設定オブジェクト（エージェント設定を指定する）を作成し、DefaultAgentName 設定の値を指定します。オブジェクトにはこの設定だけで十分です。
7. ホスト設定オブジェクトを作成します。

   ホスト設定オブジェクトは、信頼されたホストおよびポリシー サーバの間の接続を定義します。フェデレーション システムとポリシー サーバを統合するには、ホスト設定オブジェクトにより、フェデレーション システムが接続するポリシー サーバを定義します。

   既存のホスト設定オブジェクトで、フェデレーション システムが 1 つ以上のポリシー サーバに接続するには、そのオブジェクトを使用します。それ以外の場合は、フェデレーションからポリシー サーバへの接続用にオブジェクトを作成します。

8. 以下の値を使用して、カスタムのコネクタ認証方式を作成します。

   ライブラリ

   smauthsmconnector

   この値は、大文字と小文字が区別されます。

   秘密キー

   英数字

   このフィールドの値は、Administrative UI でのコネクタ設定の［共有秘密キー］値に一致する必要があります。

9. フェデレーション製品用のポリシー ドメインを作成します。このドメインには、SiteMinder セッションを作成するポリシーに追加する、必要なレルムおよびリソースが含まれる必要があります。

10. フェデレーション システムおよびポリシー サーバによって使用されるユーザ ディレクトリを、設定したドメインに追加します。

11. 以下の値が含まれるレルムを作成します。

    エージェント

    前の手順から Web エージェントを指定します。

    リソース フィルタ

    ダミー ディレクトリを指定します（/federation/ など）。 このディレクトリは Web サーバに存在する必要はありません。

    認証方式

    以前に作成されたカスタム認証方式に指定した名前を入力します。

12. 以下の値が含まれるルールを作成します。

    リソース

    *

    アクション

    Web エージェント -- Get と Post

13. 以下の設定を使用してポリシーを作成します。

    ユーザ

    フェデレーション システムと SiteMinder が共有するユーザ ディレクトリからユーザを指定します。

    ルール

    コネクタに対して作成されたルールを追加します。

これで、CA SiteMinder® Federation Standalone と通信するときに SiteMinder セッションを生成するポリシーができました。

コネクタの設定

コネクタが SiteMinder と対話するには、CA SiteMinder® Federation Standalone Administrative UI 内でコネクタを設定します。 コネクタを使用するパートナーシップはすべて単一の設定を使用し、単一の SiteMinder 環境に接続します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［インフラストラクチャ］タブに移動します。
3. ［展開設定］を選択します。

   ［展開の設定］ダイアログ ボックスが表示されます。
4. ［SiteMinder コネクタ設定］セクションのすべてのフィールドに入力します。 以下の点に注意してください。
   - 指定されたパートナーシップのコネクタを有効または無効にするには、パートナーシップ レベルで有効にします。
   - コネクタをグローバルに有効または無効にするには、展開設定のチェック ボックスを使用します。

   重要: コネクタがグローバル レベルで無効になっている場合、CA SiteMinder® Federation Standalone はパートナーシップ レベルでチェック ボックスを無視します。

   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。

5. ［ホストの登録］を選択し、SiteMinder ポリシー サーバに対してレガシー管理者認証情報を指定します。 レガシー管理者のみがホスト登録を実行できます。

   この手順では、SiteMinder ポリシー サーバにエージェントとして CA SiteMinder® Federation Standalone を登録します。

   注: 複数のポリシー サーバを指定することにより、ホスト登録プロセスに対するフェールオーバ サポートを設定できます。 プライマリ ポリシー サーバへの登録が失敗した場合、登録プロセスは正常に完了するまで、指定された次のポリシー サーバで試行します。

6. ［保存］をクリックします。

   重要: ホストの登録後、特定的に［SiteMinder コネクタ設定］セクションで［保存］を選択します。

7. 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。

SiteMinder コネクタ設定が完了しました。

パートナーシップ レベルでのコネクタの有効化

コネクタを有効にする前に、以下の点を確認します。

- SiteMinder ポリシー管理者が、フェデレーション通信のポリシーを設定しました。
- CA SiteMinder® Federation Standalone 内のコネクタに固有の設定を行いました。

SiteMinder が展開されるパートナーシップのコネクタを有効にします。

- SiteMinder がアサーティング パーティにある場合は、IdP から SP、または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コネクタを有効にします。
- SiteMinder が依存パーティにある場合は、SP から IdP、またはコンシューマからプロデューサへのパートナーシップのコネクタを有効にします。

既存のパートナーシップを変更するか、または新しいパートナーシップを設定するかにかかわらず、標準のパートナーシップ設定手順が適用されます。固有の設定手順はありません。ただし、依存パーティでターゲット リソースを指定する際には、以下のガイドラインに従います。

- CA SiteMinder® Federation Standalone がスタンドアロン モードで展開される場合、ターゲット リソースは、SiteMinder Web エージェントが保護する Web サーバに存在します。
- CA SiteMinder® Federation Standalone がプロキシ モードで展開される場合、すべてのプロキシ リクエストが SiteMinder に戻るので、ターゲット リソースは CA SiteMinder® Federation Standalone サーバの URL となります。

次の手順に従ってください:

1. Administrative UI にログインします。
2. フェデレーション パートナーシップのリストからパートナーシップを選択するか、または新しいパートナーシップを作成します。

   ［パートナーシップ］ダイアログ ボックスが表示されます。

3. ウィザードの以下のいずれかの手順に移動します。
   a. 依存パーティで、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。
   b. アサーティング パーティで、パートナーシップ ウィザードの［フェデレーション ユーザ］手順に移動します。
4. ［SiteMinder コネクタの有効化］チェック ボックスをオンにします。

   設定フィールドが使用可能になります。
5. （オプション）［UserDN とディレクトリ名の比較の実行］チェック ボックスをオンにします。 このチェック ボックスをオンにすると、SiteMinder のユーザ ディレクトリと CA SiteMinder® Federation Standalone のディレクトリの間で UserDN およびユーザ ディレクトリ名エントリの比較が実行されます。

   このチェック ボックスをオンにする場合、CA SiteMinder® Federation Standalone 展開と SiteMinder 展開のユーザ ディレクトリは同じ物理ディレクトリである必要があります。 これらの両方のディレクトリの名前は、ユーザ ストア検索に対して同じである必要があります。 このチェック ボックスをオフにする場合、ユニバーサル ID がユーザ レコードの検索に使用される属性となります。ユニバーサル ID が使用される場合、ディレクトリは同じである必要はありません。 ユニバーサル ID に基づいている場合、各ユーザは一意のユニバーサル ID を持っている必要があります。ユニバーサル ID が一意でない場合、ユーザ レコードにアクセスするシステムは不正な記録を取得する可能性があります。
6. 変更内容を保存します。

コネクタを無効にするには、パートナーシップ レベルで、または［展開設定］でグローバルに実行できます。

第 21 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。

フェデレーション通信の保護

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。

CA SiteMinder® Federation Standalone でフェデレーション環境をセットアップするとき、ユーザの環境を保護するためのいくつかの推奨事項を以下に示します。

- 使い捨てのアサーションを生成します。
- フェデレーション環境間の接続を安全にします。
- クロス サイト スクリプティングに対して保護します。

これらについては、後述の項で説明します。

アサーションの使い捨ての適用

その有効期限を過ぎたアサーションを再利用すると、期限切れの ID 情報に基づく認証判断という結果になります。 再利用を防ぐために、CA SiteMinder® Federation Standalone 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関連する問題を防止します。

CA SiteMinder® Federation Standalone 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**］を選択できます。 SAML 2.0 IdP の場合、［**OneTimeUse 条件の設定**］を選択できます。 これらの環境設定によって、CA SiteMinder® Federation Standalone 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。 CA SiteMinder® Federation Standalone 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

クライアント サイトで依存パーティからターゲット アプリケーションへのデータ送信をセキュリティ保護することが重要です。 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。

Administrative UI の［展開設定］で［安全な Cookie の有効化］チェック ボックスをオンにすることにより、この脆弱性から保護します。 ［安全な Cookie の有効化］をオンにすると、「secure」フラグが付けられた Cookie を生成するように CA SiteMinder® Federation Standalone に指示されます。 このフラグは、CA SiteMinder® Federation Standalone が SSL 通信チャネルのみに Cookie を送信することを示します。

CA SiteMinder® Federation Standalone 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

セキュリティ アサーティング パーティでのユーザの初期認証では、潜在的な脆弱性が生じます。 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 接続上で送信される場合、攻撃者はインパーソネーションまたは個人情報盗難を目的として、Cookie を取得してユーザの機密情報を盗むことができます。

Administrative UI の［展開設定］で［安全な Cookie の有効化］チェック ボックスをオンにすることにより、この脆弱性から保護します。 ［安全な Cookie の有効化］をオンにすると、「secure」フラグが付けられた Cookie を生成するように CA SiteMinder® Federation Standalone に指示されます。 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）攻撃が発生するのは、ブラウザからの入力テキスト（通常は、ポストされたデータ、または URL 内のクエリ パラメータから得られたデータ）がアプリケーションによって表示される場合です。このとき、実行可能なスクリプトを形成することが可能な文字を、フィルタ処理なしでブラウザ内で表示してしまうことが問題です。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。

CA SiteMinder® Federation Standalone 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。

CA SiteMinder® Federation Standalone 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。 これは、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ために使用できることができるサンプル アプリケーションです。通常は独自のアプリケーションを使用します。

- signoutconfirmurl.jsp

  アカウント パートナーで WS フェデレーション サイン アウトに使用されます。

- 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

文字	説明
+	プラス記号

CA SiteMinder® Federation Standalone で提供される各 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 22 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

ターゲット アプリケーションへのユーザのリダイレクト

［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］グループ ボックスでは、ユーザを CA SiteMinder® Federation Standalone からターゲット 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。 そのためにはいくつかの方法があります。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。
   - ［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。
   - 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、フェデレーション システムはすべての値を Cookie でターゲット アプリケーションへ渡します。
   - FIPS 互換のアルゴリズム（AES アルゴリズム）のいずれかを選択する場合は、CA SiteMinder® Federation Standalone SDK を使用してオープン形式の Cookie を使用する必要があります。 .NET SDK を使用する場合は、AES128/CBC/PKCS5Padding 暗号化アルゴリズムを使用する必要があります。
   - CA SiteMinder® Federation Standalone をプロキシ モードに設定し、リダイレクト モードとして［HTTP ヘッダ］を選択する場合、CA SiteMinder® Federation Standalone は単一のヘッダ内に各値をカンマで区切って複数の属性値を提供できます。

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

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。

   CA SiteMinder® Federation Standalone がプロキシ モードで動作している場合、プロキシはフェデレーション要求をすべてローカルで処理するため、プロキシ ホストの URL を入力します。 CA SiteMinder® Federation Standalone 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、CA SiteMinder® Federation Standalone がインターネットから直接アクセスされる場合は、CA SiteMinder® Federation Standalone 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 モードで動作する場合、ターゲットとして指定する URL は CA SiteMinder® Federation Standalone を経由する必要があります。 たとえば、CA SiteMinder® Federation Standalone のベース URL が fed.demo.com:5555 で、バックエンド サーバ リソースが mytarget/target.jsp である場合、このターゲット フィールドの値は http://fed.demo.com:5555/mytarget/target.jsp となります。

   注: CA SiteMinder® Federation Standalone 設定ウィザードを実行する場合、プロキシ ホストの背後に位置するバックエンド サーバを指定します。 再度、設定ウィザードを実行することにより、バックエンド サーバ エントリを変更できます。

   SAML 2.0 の場合、リレー状態クエリ パラメータの値でオーバーライドする場合は、このフィールドを空白のままにできます。この値は、シングル サインオンをトリガする URL に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. (任意ですが推奨されます) HTTP ヘッダのプレフィックスとして任意の文字列を入力します。

   SiteMinder は、すべての HTTP ヘッダにプレフィックスを追加します。プレフィックスを設定することにより、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のを防ぐことができます。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. 375)についての詳細を参照してください。

   プレフィックスを HTTP ヘッダに追加するには、以下の手順に従います。

   a. Administrative UI にログインします。
   b. [インフラストラクチャ] - [展開設定] をクリックします。
   c. [HTTP ヘッダ プレフィックス] に文字列を指定します。

      注: このオプションは、プロキシ展開モードにのみ使用可能です。

   d. [保存] をクリックします。
3. 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。

アプリケーション属性へのアサーション属性のマッピング（SAML のみ）

依存パーティで、CA SiteMinder® Federation Standalone によって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ピングできます。 その後、CA SiteMinder® Federation Standalone は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渡します。 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。

CA SiteMinder® Federation Standalone は以下のタイプのマッピングを実行できます。

- 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。

［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 エンティティに対して指定されたアサーション属性に基づいて入力されます。 このローカル依存パーティ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 アプリケーションによって使用され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切な値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。

式ビルダの［アサーション属性］リストは、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あらかじめ入力されます。このローカル依存パーティでそれらの属性を設定します。 属性がアサーション内にあれば、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か、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 パーティでのアサーション属性の指定方法に対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。CA SiteMinder® Federation Standalone 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 および属性］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、CA SiteMinder® Federation Standalone 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、データをアプリケーションに送信します。 ［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。 属性マッピングが有効な場合、CA SiteMinder® Federation Standalone 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。 属性マッピングが無効な場合、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。

CA SiteMinder® Federation Standalone 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。

UEL の詳細については、Sun Developer Network http://developers.sun.com/ を参照してください。

式を作成するためには、CA SiteMinder® Federation Standalone 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

10/01/05

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

CA SiteMinder® Federation Standalone 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。 CA SiteMinder® Federation Standalone によって、特定のアサーション属性または複数の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。

属性マッピ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。

詳細情報:

適切な構文の使用による属性マッピング ルールの作成 (P. 381)

依存パーティでのユーザ ID の動的プロビジョニング

フェデレーション ネットワークでは、異なるアサーティング パーティからフェデレーション ユーザ用にアカウントを確立することは、依存パーティでは珍しいことではありません。 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。

CA SiteMinder® Federation Standalone は、プロビジョニングをサポートする 2 つのメソッドを提供します。

- ローカル アカウント リンク（SAML 2.0 のみ）
- リモート プロビジョニング

それぞれの手順は、以下のセクションで詳細に説明します。

プロビジョニングのためのローカル アカウント リンク

ローカル アカウント リンクを使用して、SAML 2.0 展開に対してのみプロビジョニングを実行できます。プロビジョニングは、IdP のユーザ アカウントを SP のアカウントにリンクすることにより行われます。

ローカル アカウント リンク プロセスを以下に示します。

1. ユーザが SP のフェデレーション ターゲット リソースへのアクセスを要求します。
2. SP が AllowCreate という名前の属性が含まれる認証リクエストを生成し、true に設定されます。 SP が IdP に認証リクエストを送信し、ユーザの ID を取得します。
3. 認証リクエストを受信すると、IdP はアサーションを生成します。 アサーション生成中に IdP は、アサーションで名前 ID として使用されるように設定された属性に適したユーザ レコードを検索します。
4. IdP が名前 ID として使用される属性の値を検出できない場合、IdP は、識別番号を作成する機能を有効にして、永続的な識別番号を生成します。

永続的な識別番号は、ランダムに生成された ID です。 IdP では、この識別番号を名前 ID 属性の値として使用し、アサーション内に配置します。 その後、IdP は SP にアサーションを返します。

注: 許可/作成機能を両方のサイトで設定する必要があります。 許可/作成機能属性が認証リクエスト メッセージ内にあるかどうかにかかわらず、IdP が識別番号を作成するように設定されていない場合、アサーション生成は失敗します。

5. SP の CA SiteMinder® Federation Standalone は、IdP からのアサーションを処理します。ただし、名前 ID 値が SP に存在しないので、ユーザ レコードを検索できません。 その結果、認証に失敗します。
6. 失敗した認証の試行によって、アサーション コンシューマ サービスから渡されたすべてのアサーション、および他のデータが linkaccount.jsp ページにリダイレクトされます。
7. linkaccount.jsp ページへのアクセスを取得する前に、ユーザはローカル認証情報で認証される必要があります。 ログインの成功により、ユーザが識別されます。 CA SiteMinder® Federation Standalone は、ローカル ユーザ ディレクトリの適切なユーザ レコードを参照し、そのユーザ用のセッションを作成します。 ユーザはまだ、最初に要求したフェデレーション リソースにアクセスできません。
8. linkaccount.jsp は、アサーションおよび他のすべてのデータをアサーション コンシューマ サービスに戻します。 CA SiteMinder® Federation Standalone は、再度アサーションを持つユーザを認証します。これで、ユーザを識別するセッションができたので、CA SiteMinder® Federation Standalone は、適切なユーザ レコードをアサーションからの永続的な識別番号を持ったローカル ユーザ ディレクトリに設定します。 CA SiteMinder® Federation Standalone は、この目的のために特別に設定した属性に永続的な識別番号を格納します。 IdP のアカウントはこれで、SP のアカウントとリンクされました。 認証に成功します。
9. 最後に、CA SiteMinder® Federation Standalone はユーザを要求されたリソースにリダイレクトします。

注: ローカル アカウント リンクは、［**IdP にユーザ ID の作成を許可**］機能を使用する必要があります。ただし、許可/作成機能はローカル アカウント リンク専用ではありません。 ローカル アカウント リンクを実装しない場合も、この機能を選択できます。

ローカル アカウント リンクの設定（SAML 2.0）

ローカル アカウント リンクのプロビジョニングの実行では、アイデンティティ プロバイダおよびサービス プロバイダでの設定を必要とします。

アイデンティティ プロバイダでローカル アカウント リンクを設定する方法

1. パートナーシップ ウィザードにアクセスし、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］グループ ボックス内の必須フィールドを設定します。

   これらのフィールドで、アサーションで名前 ID に使用される属性を決定します。

   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。
3. ［**ユーザ識別子の作成を許可**］チェック ボックスをオンにします。
4. パートナーシップ ウィザードの［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。

アイデンティティ プロバイダでの設定は完了しました。

サービス プロバイダでローカル アカウント リンクを設定する方法

1. パートナーシップ ウィザードにアクセスし、［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］グループ ボックスで次の処理を実行します。
   - 識別に使用されるアサーションからの属性として名前 ID を選択します。
   - ［IDP にユーザ識別子の作成を許可］を選択します。
3. ［検索仕様］フィールドに値を入力します。

   ［検索条件］値は CA SiteMinder® Federation Standalone がユーザを検索し、IdP から送信された永続的な識別番号を格納するために使用する属性です。 たとえば、buyerID に名前 ID の値を格納する場合、文字列を buyerID=%s に設定します。
4. ［アプリケーション統合］手順に移動します。

5. ダイアログ ボックスの[ユーザ プロビジョニング]セクション内の[プロビジョニング タイプ］フィールドで［ローカル アカウント リンク］を選択します。

   このオプションを選択すると、POST のメソッドを持った linkaccount.jsp ページに［ユーザ不明 URL］が自動的に設定されます。この URL は、CA SiteMinder® Federation Standalone が最初の失敗した認証の試行の後にユーザをリダイレクトする場所です。

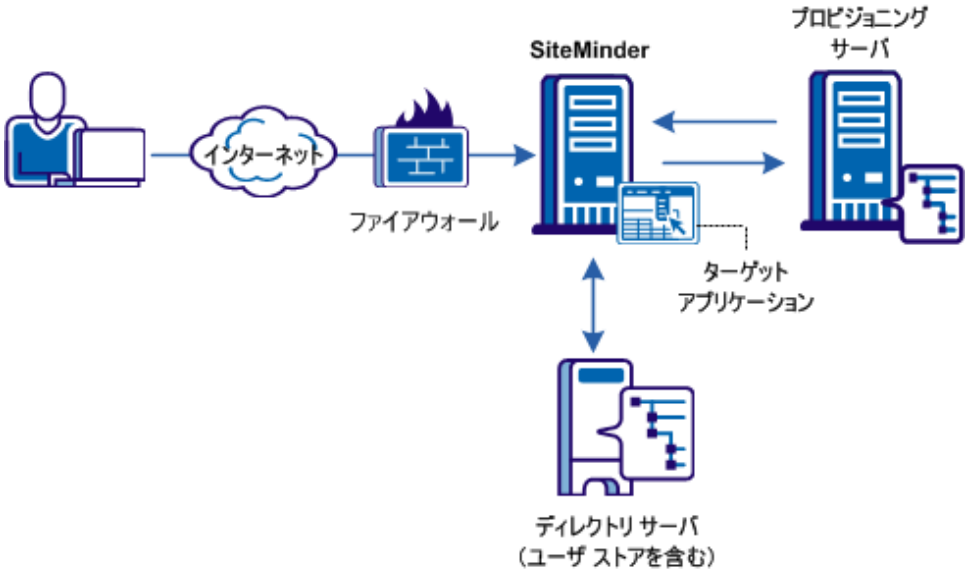
6. （オプション）linkaccount.jsp ファイルをカスタマイズして、ユーザが認証の試行に失敗した後にリダイレクトされるときに、カスタム ユーザ操作性を提供します。このファイルは、accountlinking パラメータおよび samlresponse パラメータをアサーション コンシューマ サービスに再度 POST する必要があります。 accountlinking を yes に設定する必要があります。 ページは、*federation_install_dir*/secure-proxy/Tomcat/webapps/affwebservices/public にあります。

7. パートナーシップ ウィザードの［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

リモート プロビジョニングは、新しいユーザ 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。 そのアプリケーションは必要な情報を CA SiteMinder® Federation Standalone システムに渡します。 フェデレーション システムは、そのデータを使用してユーザ認証情報を作成します。

以下の図では、リモート プロビジョニング セットアップを設定する方法を示します。



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 Standalone 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、フェデレーション システムは、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。

フェデレーション システムは以下のいずれかのメソッドを使用して、アサーション データを渡すことができます。

レガシー Cookie

フェデレーション システムによって生成されたレガシー 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。 レガシー Cookie を使用する場合、プロビジョニング アプリケーションがレガシー Cookie を読み取ることができるように、プロビジョニング アプリケーションをインストールしたシステムに CA SiteMinder® Federation Standalone Java SDK をインストールする必要があります。

注: レガシー Cookie を使用する場合、フェデレーション 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。

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、フェデレーション 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 Standalone 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 Standalone SDK を使用して Cookie を生成する必要があります。.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する必要があります。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、CA SiteMinder® Federation Standalone .NET SDK をプロビジョニング サーバ上にインストール可能で、オープン形式 Cookie の読み取りに使用されます。

  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 Standalone 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 Standalone SDK を使用せずに オープン形式の Cookie を作成するには、任意のプログラミング言語を使用して Cookie を作成します。 オープン形式の Cookie のコンテンツ (P. 503)についての詳細を確認します。

  Cookie を記述するために使用する言語は UTF-8 エンコーディング、および CA SiteMinder® Federation Standalone がパスワード ベースの暗号化に使用する PBE 暗号化アルゴリズムのいずれかをサポートする必要があります。それには以下のものが含まれます。

  - 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

Cookie を暗号化するために FIPS 互換の（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 のないオープン形式の Cookie を読み取ることができません。

また、オープン形式の Cookie がユーザのブラウザで設定されていることを確認する必要があります。

注: CA SiteMinder® Federation Standalone を FIPS 専用モードでインストールした場合、オープン形式の Cookie のみ使用可能です。

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

プロキシ モードを使用する場合、この情報も HTTP ヘッダとして渡すことができます。 HTTP ヘッダを使用する場合、CA SiteMinder® Federation Standalone 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。

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、CA SiteMinder® Federation Standalone 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 アプリケーションは、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をユーザに送信することによって、この状況に対処する責任があります。 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 を指定する必要があります。

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。

リモート プロビジョニング サーバに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、これは CA SiteMinder® Federation Standalone ローカル アカウント リンクではありません。

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］グループ ボックスでプロビジョニング タイプを選択します。

   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. 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、表示される次の追加フィールドに入力します。
   - 配信オプション
   - プロビジョニング サーバ URL

4. 配信オプションとして オープン形式の Cookie を指定する場合、［オープン形式の Cookie］グループ ボックス内の追加の設定に入力する必要があります。

   これらの設定には、Cookie の名前、Cookie を暗号化するアルゴリズムおよび暗号化パスワードが含まれます。 必要に応じて、Cookie の整合性を確認する、HMAC 関数を有効にできます。

5. ウィザードの［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。

詳細情報:

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 (P. 391)

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、さらなる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 CA SiteMinder® Federation Standalone を設定できます。 たとえば、ユーザの特定に失敗した場合、CA SiteMinder® Federation Standalone はプロビジョニング システムにユーザを行かせるように設定できます。プロビジョニング システムは SAML アサーション内にある情報に基づいてユーザ アカウントを作成できます。

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。

リダイレクト URL 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. ダイアログ ボックスの［ステータス リダイレクト URL］で、ユーザをリダイレクトしたい失敗条件用の設定のみを設定します。［ステータス リダイレクト URL］グループ ボックス内の設定は次のとおりです。
   - ユーザが見つかりません
   - 無効な SSO メッセージ
   - 承認されないユーザ認証情報（SSO メッセージ）

   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. ユーザが設定する各リダイレクト オプションに対し、CA SiteMinder® Federation Standalone 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 23 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。

Administrative 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. Administrative UI にログインします。
2. ［フェデレーション］-［エンティティ］をクリックします。
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: 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。
6. ［エクスポート］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. Administrative 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］をクリックします。
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。

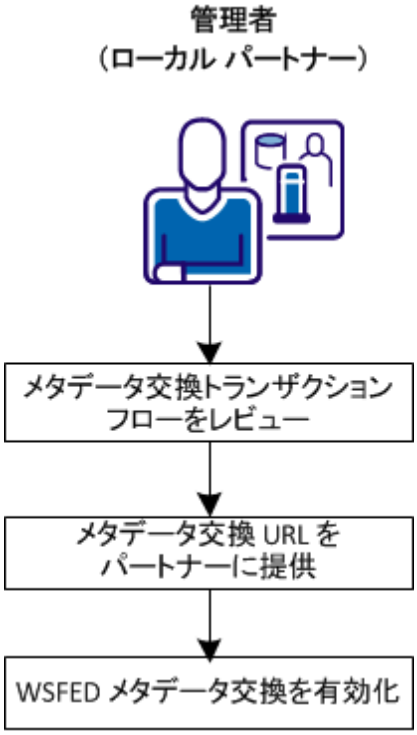
メタデータは指定された XML ファイルにエクスポートされます。 このファイルをどのパートナーにでも送信できます。

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. 401）
2. メタデータ交換 URL をパートナーに提供します。（P. 401）
3. WSFED メタデータ交換を有効にします（P. 402）。

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. Administrative 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 24 章: フェデレーション システムに対するフェールオーバのサポート

このセクションには、以下のトピックが含まれています。

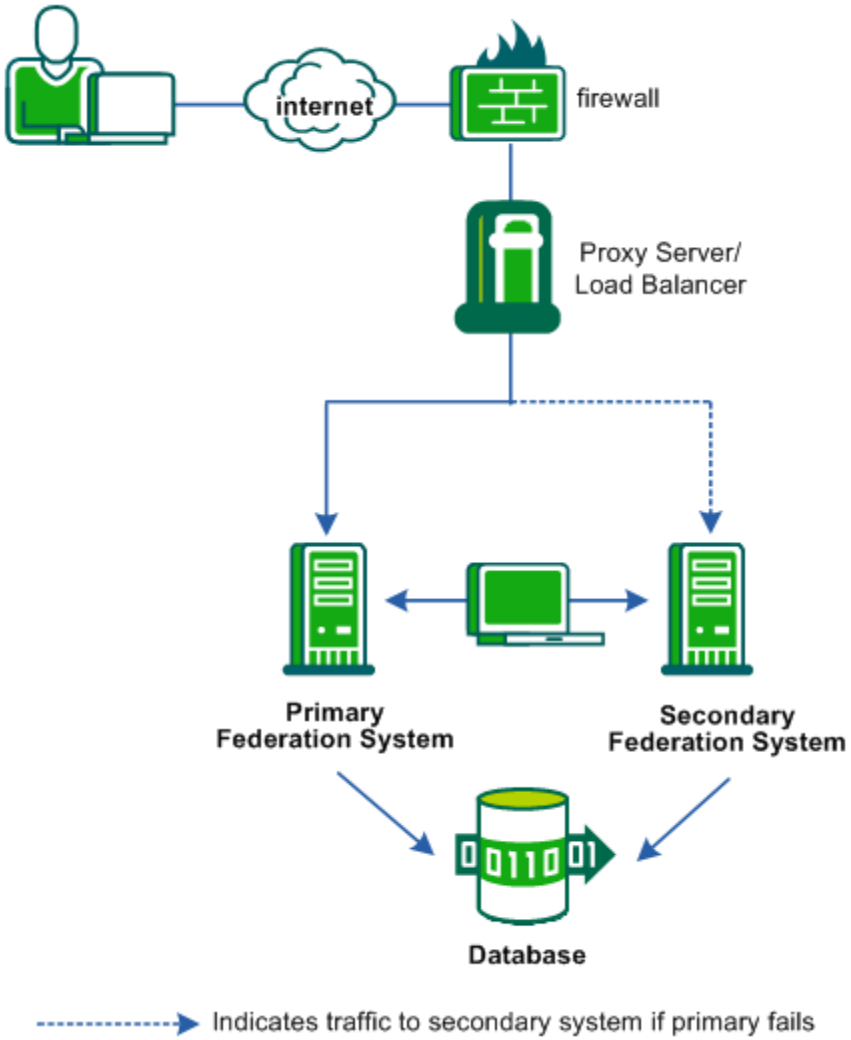
フェールオーバ概要

フェールオーバ サポートは、CA SiteMinder® Federation Standalone がユーザのフェデレーション ネットワーク内の単一障害点ではないことを確認します。 フェールオーバは、プライマリおよびセカンダリ CA SiteMinder® Federation Standalone システムの設定により、ユーザのネットワークに冗長性を構築します。 プライマリ CA SiteMinder® Federation Standalone システムが失敗する場合、バックアップ システムが必要なフェデレーション通信を実行できます。

フェールオーバは、アサーティング パーティおよび依存パーティとして動作する CA SiteMinder® Federation Standalone 用に設定できます。

注: ユーザが SiteMinder コネクタを有効にした場合、フェールオーバ サポートはコネクタ登録プロセスで使用可能です。 手順は「SiteMinder コネクタの設定」で説明します。

以下の図では、フェールオーバのある CA SiteMinder® Federation Standalone 展開を示します。 プライマリ システムが失敗する場合、セカンダリ システムにトランザクションは向けられます。



以前の図で表示されたように、同じデータベースを使用する 2 台のマシンに CA SiteMinder® Federation Standalone がインストールされます。

フェールオーバの設定方法

フェールオーバの設定には以下のタスクを必要とします。

- CA SiteMinder® Federation Standalone を少なくとも 2 つのシステムにインストールすること。
- 2 つのフェデレーション システムで設定ウィザードを実行すること。
- フェデレーション システムのフェールオーバを管理するプロキシ サーバまたはロード バランサをセットアップすること。

プロキシ サーバまたはロード バランサ用フェールオーバを設定する前に各フェデレーション システムを設定することをお勧めします。

重要: SSL をフェデレーション サービスに使用する計画の場合は、SSL が有効なフェールオーバ環境 (P. 408)用の手順に従います。

各フェデレーション システムでのフェールオーバのセットアップ

フェデレーション展開でフェールオーバを有効にするには、プライマリおよびセカンダリの CA SiteMinder® Federation Standalone システムがインストールされ設定されている必要があります。

SSL 対応フェールオーバ環境では、フェールオーバ環境に対して SSL を有効にする (P. 408)手順に従います。

重要: Solaris プラットフォームでは、Solaris ゾーンを物理マシンとして扱います。 各ゾーンに個別の CA SiteMinder® Federation Standalone インスタンスをインストールして設定します。各ゾーンのホスト ID が異なるので、CA SiteMinder® Federation Standalone は単一のインスタンスについて一方のゾーンからもう一方のゾーンへのフェールオーバをサポートしません。

次の手順に従ってください:

1. インストールごとに同じフェデレーション管理者パスワードを指定して、各システムに製品をインストールします。

   注: 製品はスタンドアロンまたはプロキシ モードで実行できますが、プライマリ サーバとセカンダリ サーバは同じモードを使用する必要があります。

2. 両方のシステムに対して同じデータベース情報を使用して、各システムでフェデレーション システムの設定ウィザードを実行します。

3. Administrative UI にログインします。
4. ［インフラストラクチャ］タブで［システム設定］を選択します。
5. フェデレーション ネットワークにプロキシ サーバまたはロード バランサのホストおよびポートを含めるように、グローバル ベース URL を変更します。 この URL を設定すると、パートナーシップ内のすべてのエンティティのデフォルト URL が正しいことを確認できます。

   CA SiteMinder® Federation Standalone が複数の仮想ホストまたはドメインを使用する場合は、エントリをすべて含めるように server.conf ファイルを変更します。

   server.conf ファイルを変更する方法

   a. *federation_install_dir*/secure-proxy/proxy-engine/conf に移動します。
   b. エディタで server.conf ファイルを開きます。
   c. ［# デフォルト仮想ホスト］セクションに移動します。
   d. 以下のように、完全修飾ホスト名を使用して**ホスト名**設定にベース URL を追加します。

      <VirtualHost name="default">

      hostnames="defaultbaseurl.example.com:80, newbaseurl.example.com:80"

      </VirtualHost>

      注: 各エントリをカンマで区切ることにより、ホスト名設定に複数の *host_name:port* エントリを指定します。

      例：

      <VirtualHost name="default"

      hostnames=lb5.example.com:80

      </VirtualHost>

両方の CA SiteMinder® Federation Standalone システムが同じデータベースをポイントしていています。 プライマリ システムからセカンダリにフェールオーバするように、プロキシ サーバまたはロード バランサをセットアップできます。

フェールオーバ用のプロキシ サーバまたはロード バランサのセットアップ

プロキシ サーバまたはロード バランサのフェールオーバ先を CA SiteMinder® Federation Standalone に設定できます。

注: プロキシ サーバまたはロード バランサの管理者は、展開でシステム用フェールオーバをセットアップする方法を知る必要があります。

次の手順に従ってください:

1. 1 つの CA SiteMinder® Federation Standalone システムをプライマリ ホストとして、もう 1 つのシステムをセカンダリ ホストとして識別します。

   システム用の負荷分散を設定しないでください。

2. 以下の URL を CA SiteMinder® Federation Standalone システムへ渡すことを確実にするため、CA SiteMinder® Federation Standalone 展開用のプロキシ サーバまたはロード バランサを設定します。

   - /affwebservices/*
   - /siteminderagent/*

   これらの URL は、プロキシ サーバまたはロード バランサが CA SiteMinder® Federation Standalone システム間のトラフィックの平衡を保つことを可能にします。

プロキシ サーバまたはロード バランサは設定されました。

有効な SSL を持ったフェールオーバを設定する方法

フェデレーション システムがロード バランサまたはプロキシ サーバの背後に位置していても、フェールオーバ環境で SSL を有効にできます。 このタイプのセットアップには特定の設定手順があります。

SSL が有効なフェールオーバの設定は以下のタスクを必要とします。

- CA SiteMinder® Federation Standalone を少なくとも 2 つのシステムにインストールすること。
- 2 つのフェデレーション システムで設定ウィザードを実行すること。
- 埋め込まれた Apache Web サーバ用に SSL を有効にすること（フェデレーション システムがロード バランサの後ろにある場合のみ）。
- プライマリからセカンダリ システムに SSL 設定を移行すること（フェデレーション システムがロード バランサの後ろにある場合のみ）。
- フェデレーション システムのフェールオーバを管理するプロキシ サーバまたはロード バランサをセットアップすること。

プロキシ サーバまたはロード バランサ用フェールオーバを設定する前に各フェデレーション システムを設定することをお勧めします。

ロード バランサの背後での SSL 対応フェールオーバの設定

TCP ロード バランサの背後に位置するようにシステムを設定できます。ロード バランサはリクエストをシステムに渡し、その後システムはサーバ側の SSL 処理を実行します。

次の手順に従ってください:

1. インストールごとに同じフェデレーション管理者パスワードを指定して、各システムに製品をインストールします。

   注: 製品はスタンドアロンまたはプロキシ モードで実行できますが、プライマリ サーバとセカンダリ サーバは同じモードを使用する必要があります。

2. 設定ウィザードを実行し、両方のシステムに対して同じデータベース接続情報を使用します。

3. 設定ウィザードは、Apache 設定情報の指定を促すメッセージを表示します。 プライマリおよびセカンダリ フェデレーションシステムの［サーバ名］設定で、同じ仮想ホスト名を指定します。 両方のシステムは同じ仮想ホスト名を使用する必要があります。

   製品が複数の仮想ホストまたはドメインを使用している場合は、プロキシ エンジンの server.conf ファイルを変更します。 server.conf ファイルには、ホスト名およびドメインのすべてがリストされる必要があります。 デフォルトの仮想ホストのホスト名フィールドに名前を追加します。

   server.conf を編集する方法

   a. 以下のディレクトリに移動します。

      Windows： *federation_install_dir*¥secure-proxy¥proxy-engine¥conf

      UNIX： *federation_install_dir*/secure-proxy/proxy-engine/conf

   b. エディタで server.conf ファイルを開きます。

   c. ［# Default Virtual Host］セクションに移動し、以下のように、完全修飾 URL を使用してホスト名設定に名前を追加します。

      <VirtualHost name="default">

      hostnames="virtualhost1.example.com, virtualhost2.example.com"

      </VirtualHost>

      注: 各エントリをカンマで区切ることにより、ホスト名設定に複数の URL を指定できます。

4. Administrative UI にログインします。

5. ［インフラストラクチャ］-［システム設定］をクリックします。

6. フェデレーション ネットワークにプロキシ サーバまたはロード バランサのホストおよびポートを含めるように、グローバル ベース URL を変更します。 この URL を設定すると、パートナーシップ内のすべてのエンティティのデフォルト URL が正しいことを確認できます。

   server.conf ファイルを変更する方法

   a. *federation_install_dir*/secure-proxy/proxy-engine/conf に移動します。

   b. エディタで server.conf ファイルを開きます。

c. ［# デフォルト仮想ホスト］セクションに移動します。

d. 以下のように、完全修飾ホスト名を使用して**ホスト名**設定にベース URL を追加します。

```
<VirtualHost name="default">
hostnames="defaultbaseurl.example.com:80,
newbaseurl.example.com:80"
</VirtualHost>
```

注: 各エントリをカンマで区切ることにより、ホスト名設定に複数の *host_name:port* エントリを指定します。

7. プライマリ フェデレーション システムで、埋め込み Apache Web サーバの SSL を有効にします。
8. フェールオーバ展開でセカンダリ システムに Apache SSL 設定を移行します。
9. ロード バランサで、同じホスト名に対して複数の IP アドレス（フェデレーション システムにマップされる）を設定します。

セカンダリ システムへの SSL セットアップの移行

Apache SSL をプライマリ CA SiteMinder® Federation Standalone マシンで設定した後、ロード バランサの背後にあるセカンダリ マシンに移行できます。

注: CA SiteMinder® Federation Standalone がプロキシ サーバの背後にある場合、この手順は適用されません。

以下の基準が満たされていることを確認します。

- 各 CA SiteMinder® Federation Standalone マシンで同じ証明書が使用されています。
- CA SiteMinder® Federation Standalone マシンをそれぞれ同じホスト名で設定する必要があります。
- CA SiteMinder® Federation Standalone がロード バランサを介してアクセスされます。
- すべてのマシンが同じプラットフォーム（Windows/Solaris/Linux）である必要があります。

セカンダリ マシンに SSL 設定をコピーする方法

1. プライマリ CA SiteMinder® Federation Standalone マシン上で Apache SSL を有効にします。 これを有効にすると、以下のコンポーネントが使用可能になります。

   - SSL サーバ証明書

     federation_install_dir/secure-proxy/SSL/certs/server.crt

   - CA バンドル

     federation_install_dir/secure-proxy/SSL/certs/ca-bundle.cert

   - SSL サーバ キー

     federation_install_dir/secure-proxy/SSL/keys/server.key

   - 証明書リクエスト ファイル

     federation_install_dir/secure-proxy/SSL/keys/fedmgrsslcertrequest.pem

   - SSL プロパティ ファイル

     federation_install_dir/config/fedmanager.properties

2. SSL サーバ証明書に署名した CA 証明書をセカンダリ マシンにインポートします。Administrative UI を使用して証明書をインポートします。

   この証明書は、プライマリ マシン上の SSL 設定プロセスの前に、またはそのプロセス中にインポートされる必要があります。プライマリ マシンでこの証明書に使用されたものと同じエイリアスを使用することをお勧めします。

3. セカンダリ マシン上の同じ場所に、手順 1 でリストされている各ファイルをコピーします。 フォルダはすでに存在しているはずです。

   以下の点に注意してください。

   - セカンダリ マシンには、ca-bundle.cert のコピーがすでに存在しているはずです。 そのコピーはバックアップまたは削除する必要があります。プライマリ マシンからの新しいコピーには、セカンダリ マシンが必要とする追加のデータが存在します。

   - 証明書リクエスト ファイル（fedmgrsslcertrequest.pem）は、セカンダリ マシンで Administrative UI を使用してこのファイルを取得する場合にのみコピーする必要があります。 それ例外の場合は、ファイルをコピーしないでください。

- SSL プロパティ ファイルには、少なくとも以下の 2 つのプロパティが含まれる必要があります。
  - fedmgr.ssl.enabled、Y に設定されています。
  - fedmgr.ssl.ca.alias、SSL サーバ証明書リクエストに署名した CA のエイリアスに設定されています。
  - セカンダリ マシン上でこの証明書をインポートするときに別のエイリアスを使用した場合は、実際に使用したエイリアス値でこのプロパティを更新します。

これで、設定が移行され、セカンダリ システム上で SSL をアクティブ化できます。

セカンダリ フェールオーバ システム上での SSL のアクティブ化

セカンダリ システムに Apache SSL 設定を移行した後で、SSL を有効にします。

セカンダリ マシン上で SSL をアクティブ化する方法（Windows）

1. セカンダリ マシン上でコマンド プロンプト ウィンドウを開きます。
2. *federation_install_dir*/secure-proxy/httpd/bin フォルダに移動します。
3. 以下のコマンドを実行します。

   configssl.bat -enable

4. 以下のショートカットを使用して、CA SiteMinder® Federation Standalone サービスを停止して再起動します。

   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。

   - ［スタート］、［すべてのプログラム］、［CA］、［Federation Standalone］、［サービスの停止］
   - ［スタート］、［すべてのプログラム］、［CA］、［Federation Standalone］、［サービスの開始］

セカンダリ マシン上で SSL をアクティブ化する方法（UNIX）

1. *federation_install_dir* に移動します。
2. コマンドの実行により、CA SiteMinder® Federation Standalone サービスをシャットダウンします。

   ./fedmanager.sh stop
3. 以下のコマンドを実行して、SSL 対応モードで CA SiteMinder® Federation Standalone サービスを再起動します。

   ./fedmanager.sh startssl
4. 入力を促されたら、Administrative UI パスワードを入力して、SSL モードでの Apache Web サーバの起動を有効にします。

プロキシ サーバの後ろの SSL で有効なフェールオーバの設定

CA SiteMinder® Federation Standalone がプロキシ サーバの後ろにある場合、プロキシ サーバが SSL 処理に対処します。 多くのプロキシ サーバが他のシステムに SSL リクエストの処理を委任できないので、CA SiteMinder® Federation Standalone は SSL を処理できません。 従って、サーバ証明書およびあらゆるリモート クライアント証明書を署名したサーバ証明書および CA 証明書を持つプロキシ サーバを設定します。

プロキシ サーバの後ろに位置する CA SiteMinder® Federation Standalone マシンでは、特定の SSL 設定は必要ありません。

フェールオーバ用のプロキシ サーバまたはロード バランサのセットアップ

プロキシ サーバまたはロード バランサのフェールオーバ先を CA SiteMinder® Federation Standalone に設定できます。

注: プロキシ サーバまたはロード バランサの管理者は、展開でシステム用フェールオーバをセットアップする方法を知る必要があります。

次の手順に従ってください:

1. 1 つの CA SiteMinder® Federation Standalone システムをプライマリ ホストとして、もう 1 つのシステムをセカンダリ ホストとして識別します。

   システム用の負荷分散を設定しないでください。

2. 以下の URL を CA SiteMinder® Federation Standalone システムへ渡すことを確実にするため、CA SiteMinder® Federation Standalone 展開用のプロキシ サーバまたはロード バランサを設定します。

   - /affwebservices/*
   - /siteminderagent/*

   これらの URL は、プロキシ サーバまたはロード バランサが CA SiteMinder® Federation Standalone システム間のトラフィックの平衡を保つことを可能にします。

プロキシ サーバまたはロード バランサは設定されました。

各システムでの同じ設定の保持

設定に何らかの変更を加える場合、常にプライマリ CA SiteMinder® Federation Standalone システムへのこれらの変更を管理し、セカンダリ マシンに設定をエクスポートします。

設定変更に関する次の情報に注意してください。

設定変更後の遅延

プライマリ システム UI を使用して行った変更は、必ずしも、セカンダリ システムですぐに使用できるとは限りません。 プライマリ システムとセカンダリ システムでは同じデータベースを共有しているため、データベースに保存されている UI で管理されるデータは、セカンダリ システムで使用できます。ただし、セカンダリ システムのポリシー エンジンが変更をピックアップするまで、遅延が発生することがあります。

一部の設定変更では再起動が必要

一部の設定変更では、システムの再起動を必要とします。 プライマリ システムの設定を変更し、この変更が再起動を必要とする場合は、セカンダリ システムも再起動します。

同じプライマリ システムでの管理の実行

常に同じプライマリ システムで管理を実行します。このプラクティスを適用するためにセカンダリ マシンでの UI 管理を無効にできます。

次の手順に従ってください：

1. Administrative UI にログオンします。
2. ［インフラストラクチャ］-［システム設定］を選択します。

   ［システムの設定］ダイアログ ボックスが表示されます。
3. ［UI 設定］グループ ボックスの［管理の無効化］をクリックします。

   アクションを確認するように求めるメッセージが表示されます。
4. ［はい］をクリックして UI 管理を無効にします。

   ［管理が無効化されました］ダイアログ ボックスが表示され、UI の他のすべての部分が使用不可能になります。 このダイアログ ボックスで、再度管理を有効にできます。

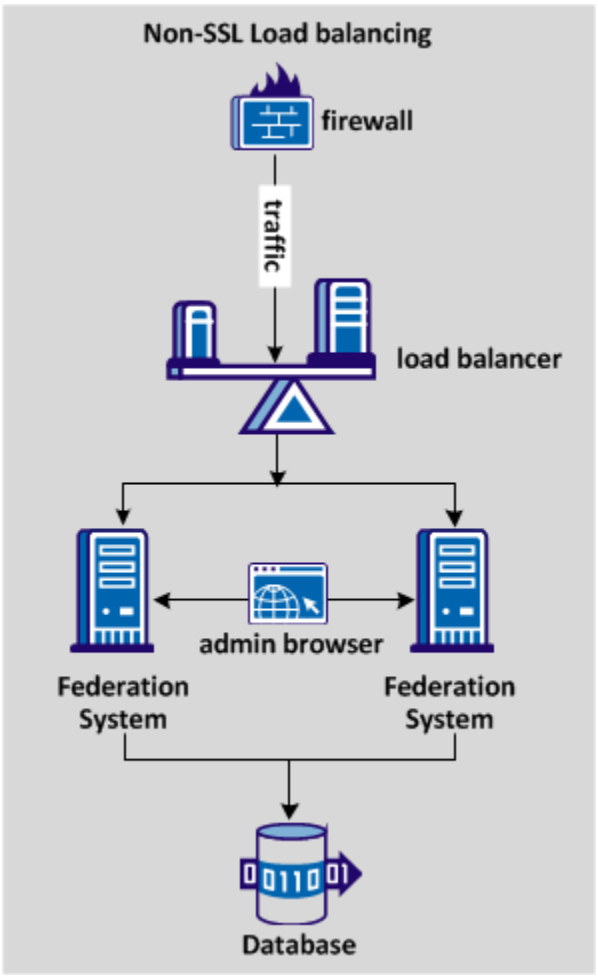
第 25 章: フェデレーション システムに対するロード バランシングのサポート

このセクションには、以下のトピックが含まれています。

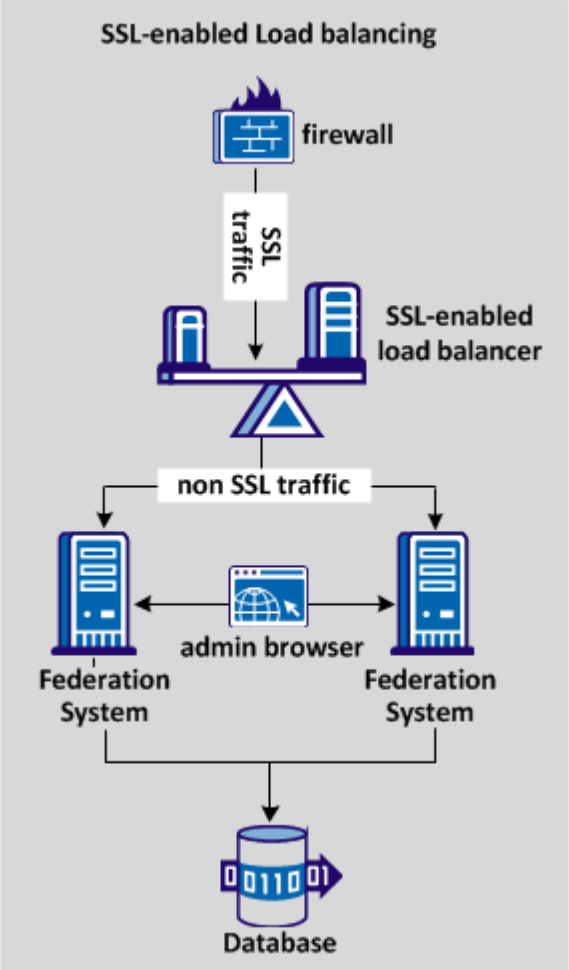
ロード バランシングを設定する方法

ロード バランシングは、単一のデバイスに過度な負荷がかからないように、ネットワーク全体に通信アクティビティを均等に分散します。 CA SiteMinder® Federation Standalone はスタンドアロン エンドポイントとなることを目的としているため、ロード バランシングのためのビルトインのサポートがありません。ただし、CA SiteMinder® Federation Standalone を使用して、ネットワーク内の単純なロード バランシング展開を有効にすることは可能です。 SSL 対応のロード バランシングはオプションですが、機密データの送信にはお勧めします。

以下の図では、非 SSL ロード バランシング展開を示しています。



以下の図では、同じデータベース ストアを使用する 2 つのシステム間でトラフィックを分散する SSL ロード バランシング展開を示しています。



ロード バランシング トラフィック用の以下の設定手順を実行します。

1. ロード バランサを設定します。
2. ロード バランサと連携する複数の CA SiteMinder® Federation Standalone システムをセットアップします。
3. （オプション）ロード バランサが SSL を使用している場合は、SSL リダイレクトを処理するように CA SiteMinder® Federation Standalone を設定します。

ロード バランサの設定

CA SiteMinder® Federation Standalone をインストールしたネットワークで動作するようにロード バランサを設定します。 CA SiteMinder® Federation Standalone システムに存在するリソースをリクエストするユーザをリダイレクトするには、ロード バランサ ホストおよびポートを使用します。ロード バランサ ホストおよびポートの使用は CA SiteMinder® Federation Standalone システム上のすべてのリソースに適用されます。

注: この手順は、ロード バランサ管理者が展開でシステムをセットアップする方法を知っていることを前提としています。

次の手順に従ってください:

1. ロード バランサを設定して、フェデレーション展開用の IP アドレスおよびホスト名をマップします。
2. 展開用にロード バランサを設定し、次の URL が CA SiteMinder® Federation Standalone システムに確実に渡されるようにします。
   - /affwebservices/*
   - /siteminderagent/*
   - /forms/*

   これらの URL により、ロード バランサはフェデレーション システム間のトラフィックのバランスを取ることができます。
3. （オプション）SSL トラフィックを処理するようにロード バランサを設定します。

   ロード バランサが SSL 対応の場合、すべてのフェデレーション トラフィックが SSL を介してロード バランサに送られます。 ただし、ロード バランサは、非 SSL （HTTP）接続を介して CA SiteMinder® Federation Standalone システムにトラフィックを送信します。

ロード バランサが CA SiteMinder® Federation Standalone システムと連動するように設定されました。

ロード バランサと連携するフェデレーション システムのセットアップ

フェデレーション展開でロード バランシングを使用するには、複数の CA SiteMinder® Federation Standalone システムをセットアップします。

注: 手順では、システムがすべてバージョン r12.52 SP1 であると仮定します。

次の手順に従ってください:

1. インストールごとに同じフェデレーション管理者パスワードを指定して、各システムに製品をインストールします。

   注: 製品がスタンドアロンかプロキシ モードかで実行しているかに関係なく、サーバでは同じモードを使用する必要があります。

2. 1 つのシステムで設定ウィザードを実行します。
3. Administrative UI にログインします。
4. ［インフラストラクチャ］-［システム設定］に移動します。
5. ［サーバ設定］セクションで、ユーザのネットワークにロード バランサのホストおよびポートを含めるためにグローバル ベース URL を変更します。すべてのパートナーシップ エンティティのデフォルト URL が正しくなるように、この URL を設定します。
6. 以下のタスクを実行して、フェデレーション パートナーシップをセットアップします。
   a. 証明書および秘密キーをインポートします。
   b. ユーザ ディレクトリ接続を確立します (P. 91)。
   c. ローカル エンティティを設定します (P. 133)。
   d. リモート エンティティを指定します (P. 133)。
   e. ローカル エンティティとリモート エンティティ間のパートナーシップを設定します。
   f. フェデレーションがリモート パートナーと連携することを確認します。
7. 最初のシステムに入力したロード バランサの同じ仮想ホスト名を使用して、セカンダリ システムで設定ウィザードを実行します。

   各フェデレーション システムで同じ仮想ホスト名を使用する必要があります。仮想ホスト名は、設定ウィザードを実行したときに、Apache の設定の［サーバ名］に指定したホストです。

製品で複数の仮想ホストまたはドメインを使用する場合は、server.conf ファイルを変更して、追加のエントリを含めます。

server.conf ファイルを変更する方法

a. *federation_install_dir*/secure-proxy/proxy-engine/conf に移動します。

b. エディタで server.conf ファイルを開きます。

c. ［# デフォルト仮想ホスト］セクションに移動します。

d. 以下のように、完全修飾ホスト名を使用して**ホスト名**設定にベース URL を追加します。

```
<VirtualHost name="default">
hostnames="defaultbaseurl.example.com:80,
newbaseurl.example.com:80"
</VirtualHost>
```

注: 各エントリをカンマで区切ることにより、ホスト名設定に複数の *host_name:port* エントリを指定します。

例：

```
<VirtualHost name="default"
hostnames=lb5.example.com:80
</VirtualHost>
```

8. 埋め込みの Apache および Tomcat Web サーバによって保存される SSL キーおよび証明書を移行します。

   - このタスクを実行するには、SSL 移行手順 (P. 453)に従います。 SSL データを移行することで、新しいキーまたは証明書を購入せずに済みます。
   - 新しいキー/証明書リクエストを生成してから、証明書に署名させます。SSL 証明書はインポートされた設定ファイルに含まれていません。

   注: 1 つのシステム上の証明書設定の変更はすべて、他のすべてのシステムにレプリケートします。 UI で［証明書 & キー］から設定を変更します。 変更には証明書、キー、または CRL データの追加や削除が含まれます。

9. パートナーシップが設定されていない他のシステムの Administrative UI にログインします。
10. ［インフラストラクチャ］-［システム設定］に移動します。 ［UI 設定］セクションで、［管理の無効化］をクリックします。

    ロード バランサに進まずに、Administrative UI にローカルにアクセスします。 他のシステムが稼働中の場合は、1 つのシステム上の管理のみを有効にします。 管理システムがいつでも無効にされている場合は、別のシステムにログインして、管理を再度有効にします。

すべてのフェデレーション システムが同じデータ ストアを指すようになったため、設定されたロード バランサはシステム間でトラフィックのバランスを取ることができるようになりました。

SSL ロード バランサへのリダイレクトの設定（オプション）

ロード バランサが SSL を使用している場合、SSL 接続を介してトラフィックをリダイレクトするようシステムを設定することをお勧めします。 トラフィックをリダイレクトするには、各フェデレーション システム上で以下の 2 つのファイルを変更します。

- LocalConfig.conf
- httpd.conf

注: トラフィックをリダイレクトするすべてのフェデレーション システムでこれらのファイルを変更します。

次の手順に従ってください:

1. *federation_install_dir*/secure-proxy/proxy-engine/conf/defaultagent に移動します。
2. テキスト エディタで WebAgent.conf ファイルを開きます。 **localconfigfile** で始まる行のコメントを外し、ファイルを保存します。
3. テキスト エディタで LocalConfig.conf ファイルを開きます。
4. LocalConfig.conf ファイルに以下の設定を追加して、ファイルを保存します。

   ```
   HttpsPorts="443"
   ```

   ロード バランサがリスンするポートを指定します。

   ```
   GetPortFromHeaders="YES"
   ```

5. *federation_install_dir*/secure-proxy/httpd/conf に移動します。
6. エディタで httpd.conf ファイルを開きます。
7. SeverName 設定を見つけて、ロード バランサの *hostname:port* を指定します。 フェデレーション システム サーバのホスト名を入力しないでください。

   例：

   ```
   ServerName lb5.example.com:443
   ```

8. ServerName 設定の後に、UseCanonicalName 設定を追加し、それを On に設定します。 例：

   ```
   UseCanonicalName on
   ```

フェデレーション システムが SSL 接続を介してトラフィックをリダイレクトするようになりました。

第 26 章: フェデレーション システム管理

このセクションには、以下のトピックが含まれています。

サーバ ステータス モニタリング

［サーバ ステータス］ダイアログ ボックスは、サーバの条件の確認に役立つ情報のスナップショットを提供します。 例にはシステム パフォーマンスの強化または予想どおりにインストールされたことの確認が含まれます。

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。

サーバ設定を表示する方法

1. Administrative UI にログインします。
2. ［インフラストラクチャ］、［サーバ ステータス］を選択します。
3. ステータス ページの情報を見直します。

いつでも［リフレッシュ］をクリックして更新されたサーバ情報を参照できます。

システム設定の変更

［システムの設定］ダイアログ ボックスでは、アクティブなサーバ スレッドの数、および許可されるアクティブなサーバ接続の数を指定でき、この設定はシステム パフォーマンスに影響を及ぼす可能性があります。また、ローカル ホストに対して UI 管理を無効にしたり再度有効にしたりできます。

次の手順に従ってください:

1. Administrative UI を起動します。
2. ［インフラストラクチャ］タブで［システム設定］を選択します。
3. 必要に応じて、任意の設定を変更します。

   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。
4. ［保存］をクリックします。
5. CA SiteMinder® Federation Standalone を再起動します。

   サーバの再起動後に変更が有効になります。ただし、［管理の無効化］機能は例外で、直ちに有効になります。

展開設定

［展開設定］により、ユーザは以下を実行できます。

- ［FIPS モード］設定を表示します。
- CA SiteMinder® Federation Standalone 展開モード（スタンドアロンまたはプロキシ）を表示します。
- 有効なフェデレーション ドメインを指定します。
- CA SiteMinder® Federation Standalone が有効な SiteMinder コネクタで作動している場合は、SiteMinder コネクタを設定します。
- CA SiteMinder® Federation Standalone セッション Cookie 名を変更します。

展開モードおよび FIPS 設定

［展開設定］は、CA SiteMinder® Federation Standalone をインストールして設定するときに選択した展開モードおよび FIPS モードのステータスを表示します。 さらに、有効なフェデレーション ドメインを指定でき、プレフィックスを設定してプロキシ モードで HTTP ヘッダを保護できます。

各設定の変更プロセスは異なります。

有効なフェデレーションドメインおよび HTTP ヘッダ プレフィックス

［有効なフェデレーション ドメイン］および［HTTP ヘッダ プレフィックス］エントリを UI から変更します。 これらの設定の変更は任意です。

次の手順に従ってください:

1. 必要に応じて、フィールドに値を入力します。
2. セクションの右隅の［保存］をクリックします。

FIPS モードの変更

FIPS モードを変更するには、再度インストール ウィザードを実行し、新しい設定を選択します。

重要: FIPS モードを変更するたびに CA SiteMinder® Federation Standalone を再起動します。

展開モードの変更

展開モードを変更するには、再度、設定ウィザードを実行し、モードを変更します。

詳細情報:

暗号化および復号アルゴリズム (P. 507)

依存パーティでのプロキシ モード展開の HTTP ヘッダ保護

依存パーティでのプロキシ モード展開で、CA SiteMinder® Federation Standalone は HTTP ヘッダを使用し、SAML アサーションからバックエンド アプリケーションに ID 属性を渡します。 ほとんどの場合、ヘッダは安全です。ただし、未許可のユーザがアサーション属性名を知っている場合、その名前をブラウザにヘッダとして設定し、ターゲット アプリケーションへのアクセスを取得できます。 ターゲット アプリケーションは、CA SiteMinder® Federation Standalone 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。

［HTTP ヘッダ プレフィックス］設定に値を指定し、次のシナリオに対して保護できます。

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、CA SiteMinder® Federation Standalone にヘッダを含めた受信リクエストを送信します。
3. CA SiteMinder® Federation Standalone 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. CA SiteMinder® Federation Standalone は自身の正規ヘッダをターゲット アプリケーションへ渡す前に、指定されたプレフィックスを各ヘッダに追加し、ヘッダをターゲット アプリケーションに渡します。

HTTP ヘッダ プレフィックスを設定するには

1. ［インフラストラクチャ］、［展開設定］に移動します。
2. ［HTTP ヘッダ プレフィックス］フィールドでプレフィックスとして任意の有効な文字列を入力します。

   CA SiteMinder® Federation Standalone をインストールするときにプロキシ モードを有効にした場合にのみ、このフィールドが表示されます。
3. 変更内容を保存します。

SiteMinder コネクタ設定

SiteMinder コネクタを使用すると、CA SiteMinder® Federation Standalone はフェデレーション通信のために SiteMinder 環境と統合できます。

アサーティング パーティで、SiteMinder コネクタは、委任された認証用のサードパーティ WAM として SiteMinder で動作できます。依存パーティで、SiteMinder は、ターゲット リソースが存在するサーバを保護できます。SiteMinder がアクセス制御を実行している場合、SiteMinder コネクタは、SiteMinder がターゲット リソースへのアクセス権をユーザに付与するように、SiteMinder セッションを確立するために［ポリシー サーバ］にアクセスします。

CA SiteMinder® Federation Standalone が SiteMinder と連携するには、Administrative UI 内で SiteMinder コネクタを設定します。

SiteMinder コネクタを使用するパートナーシップはすべて単一の設定を使用し、単一の SiteMinder 環境に接続されます。 Administrative UI の［展開設定］で、コネクタ設定を定義します。 指定されたパートナーシップのコネクタを有効にするには、パートナーシップ レベルで有効にします。パートナーシップ レベルで、または［展開設定］でグローバルに無効にすることにより、コネクタを無効にします。

重要: コネクタがグローバル レベルで無効になっている場合、CA SiteMinder® Federation Standalone はパートナーシップ レベルでチェック ボックスを無視します。

次の手順に従ってください:

1. Administrative UI にログインします。
2. フェデレーション パートナーシップのリストから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。
3. 以下のいずれかを実行します。
   a. 依存パーティで、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。
   b. アサーティング パーティで、パートナーシップ ウィザードの［フェデレーション ユーザ］手順に移動します。

4. ［SiteMinder コネクタの有効化］チェック ボックスをオンにします。

   設定フィールドが使用可能になります。

5. （オプション）［UserDN とディレクトリ名の比較の実行］チェック ボックスをオンにします。 このチェック ボックスをオンにすると、SiteMinder のユーザ ディレクトリと CA SiteMinder® Federation Standalone のディレクトリの間で UserDN およびユーザ ディレクトリ名エントリの比較が実行されます。

   このチェック ボックスをオンにする場合、CA SiteMinder® Federation Standalone 展開と SiteMinder 展開のユーザ ディレクトリは同じ物理ディレクトリである必要があります。 これらの両方のディレクトリの名前は、ユーザ ストア検索に対して同じである必要があります。 このチェック ボックスをオフにする場合、CA SiteMinder® Federation Standalone はユニバーサル ID を使用してユーザ レコードを検索するので、ディレクトリが同じである必要はありません。 ユニバーサル ID に基づいている場合、各ユーザは一意のユニバーサル ID を持っている必要があります。 ユニバーサル ID が一意でない場合、ユーザ レコードにアクセスするシステムは不正な記録を取得する可能性があります。

6. 変更内容を保存します。

7. ［インフラストラクチャ］タブに移動します。

8. ［インフラストラクチャ］タブで［展開設定］を選択します。

   ［展開の設定］ダイアログ ボックスが表示されます。

9. ［SiteMinder コネクタ設定］セクションのすべてのフィールドに入力します。

   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。

10. ［ホストの登録］を選択し、SiteMinder ポリシー サーバに対して管理者認証情報を指定します。

    この手順では、SiteMinder ポリシー サーバにエージェントとして CA SiteMinder® Federation Standalone を登録します。

    注: 複数のポリシー サーバを指定することにより、ホスト登録プロセスに対するフェールオーバ サポートを設定できます。 プライマリ ポリシー サーバへの登録が失敗した場合、CA SiteMinder® Federation Standalone は登録プロセスが正常に完了するまで、指定された次のポリシー サーバに移動します。

11. ダイアログ ボックスの［SiteMinder コネクタ設定］セクションで、［保存］を選択します。

    ホストの登録後、［SiteMinder コネクタ設定］セクションで［保存］を選択する必要があります。

12. 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。

SiteMinder コネクタ設定が完了しました。

セッション Cookie およびアイデンティティ Cookie の Cookie 設定

CA SiteMinder® Federation Standalone は、シングル サインオン セキュリティ ゾーンをサポートします。シングル サインオン セキュリティ ゾーンは、同じ Cookie ドメイン内のアプリケーションのグループ間に設定可能な信頼関係を提供します。

同じゾーン内ではシングル サインオンが適用されますが、ゾーン間で定義された信頼関係に応じて、別のゾーンを入力すると、ユーザに認証情報が再要求される場合があります。 信頼関係に含まれているセキュリティ ゾーンでは、グループ内のいずれかのゾーンで有効なセッションを持つユーザには認証情報が再要求されません。

セキュリティ ゾーン アフィリエーションは Cookie 名に反映されます。CA SiteMinder® Federation Standalone では、デフォルトのセッション Cookie とアイデンティティ Cookie には FEDSESSION と FEDPROFILE という名前が付けられます。

フェデレーション パートナーは、アプリケーション自体のセッション Cookie またはアイデンティティ Cookie を使用するアプリケーションを持っている可能性があります。 パートナー Cookie の名前は、CA SiteMinder® Federation Standalone の Cookie の名前と競合することがあります。 たとえば、SiteMinder サイトと通信している場合、SiteMinder はそれ自体のセッション Cookie およびアイデンティティ Cookie を生成するので、FEDSESSION および FEDPROFILE という名前のクッキーが存在する可能性があります。 この場合、CA SiteMinder® Federation Standalone のグローバル Cookie ゾーン プレフィックスを変更できるので、その Cookie の名前が変更されます。

注: CA SiteMinder® Federation Standalone SDK を使用しているアプリケーションを持っている場合、［グローバル Cookie ゾーン］および［暗号化パスワード］設定に対して設定された値は、SDK が使用する値に一致する必要があります。 必ずこれらの設定の値を組織の該当するパーティと共有してください。 アサーティング パーティで、SDK および Web アクセス管理システムはこれらの値を必要とします。 依存パーティで、CA SiteMinder® Federation Standalone、およびアプリケーションをホストするターゲット システムは、これらの値を認識している必要があります。

詳細については、「CA SiteMinder® Federation Standalone *Java SDK ガイド*」または「*.NET SDK ガイド*」を参照してください。

このグループ ボックス内の他の Cookie パラメータは オープン形式の Cookie 設定です。 オープン形式の Cookie 設定は、委任された認証の オープン形式の Cookie 方式にのみ使用され、パートナーシップ レベルではなくグローバル レベルで適用されます。

注: 依存パーティでは、この Cookie データの設定はグローバル レベルではなくパートナーシップ レベルで実行されます。

Cookie 設定を変更する方法

1. Administrative UI にログインします。
2. ［インフラストラクチャ］タブで［展開設定］を選択します。

   ［展開の設定］ダイアログ ボックスが表示されます。
3. （オプション）必要に応じて、［Cookie 設定］セクションの設定をすべて変更します。

   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。
4. セクションの右隅の［保存］をクリックします。

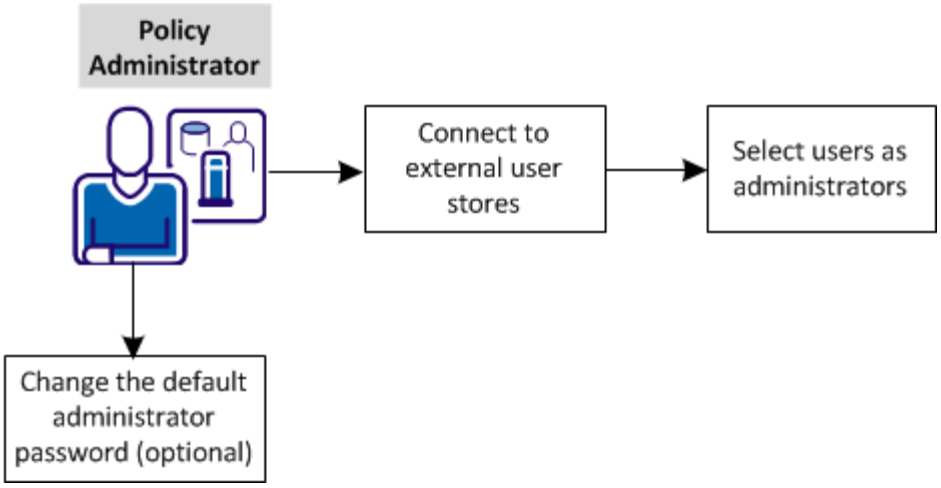
フェデレーション システム管理者を設定する方法

会社の複数の管理者が、フェデレーション管理のさまざまな側面を担当することができます。 組織の複数の人に CA SiteMinder® Federation Standalone の管理を割り当てて、責任と責任の分担を確立します。

デフォルトの管理者アカウントは CA SiteMinder® Federation Standalone を管理するために常に使用できます。 新しい管理者を追加した後、必要に応じて、デフォルト管理者アカウントを無効にします。

Administrative UI から新しい管理者ユーザを作成し、メンテナンスします。

次の画像に、管理者の設定の設定タスクを示します。



以下のタスクを実行します。

1. 外部ユーザ ディレクトリに接続します (P. 434)。
2. 管理者としてユーザを選択します (P. 436)。
3. デフォルトの管理者パスワードを変更します (オプション) (P. 437)。

外部ユーザ ストアへの接続

LDAP および ODBC の外部ユーザ ストアへの接続を作成します。 この手順は、複数の管理者を設定する前に必要です。

LDAP および ODBC はフェデレーション システムがサポートする 2 つのタイプのディレクトリです。

次の手順に従ってください:

1. ［ユーザ ディレクトリ］タブをクリックします。
2. ［LDAP に接続］または［ODBC に接続］をクリックします。

   ［アクション］-［変更］を選択して、既存のディレクトリ接続の設定を確認することができます。
3. 各セクションの必須設定を設定します。 赤いドットで、必須パラメータがマークされています。

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

4. ［ユニバーサル ID 属性］（LDAP）または［ユニバーサル ID 列］（OCBC）の値を入力します。 この値は複数の管理者を設定するために必要です。

   ユニバーサル ID 値はディレクトリの個別のユーザを識別するために一意である必要があります。 たとえば、各ユーザは uid を持つため、LDAP ディレクトリのユニバーサル ID として uid を入力します。 同じ職位のユーザが多くいるため、職位などの属性は使用しないでください。

5. LDAP ディレクトリのみの場合、［ユーザ DN 検索の開始］および［ユーザ DN 検索の終了］フィールドの値を指定します。 例：

   ユーザ DN 検索の開始

   (uid=

   ユーザ DN 検索の終了

)

6. 接続が有効であることを確認するには［接続のテスト］をクリックします。

   ［内容の表示］をクリックして、ユーザ ディレクトリの内容を一覧表示できます。

   注:

   - LDAP ディレクトリ接続については、［内容の表示］ボタンは、［検索ルート］、［ユーザ DN 検索の開始］、［ユーザ DN 検索の終了］、［ユニバーサル ID 属性］の値が設定されている場合にのみ表示されます。
   - ODBC ディレクトリ接続については、［ユニバーサル ID 列］の値が設定されている場合にのみ［内容の表示］ボタンが表示されます。

7. ［保存］をクリックします。

   設定が有効な場合、［ユーザ ディレクトリの表示］ダイアログ ボックスにリダイレクトされます。

   ディレクトリへの接続が設定されます。

管理者としてのユーザの選択

外部ユーザ ストアへの接続を確立した後、管理者とするユーザを選択します。

次の手順に従ってください:

1. Administrative UI にログオンします。
2. ［インフラストラクチャ］-［管理］の順に移動します。
3. ［管理認証の設定］を選択します。
4. これらのタスクを実行するには、設定ウィザードに従います。
   - 外部ユーザ ストアを選択します。
   - 管理者にするユーザを 1 人以上選択します。
   - 各管理者のアクセス権のタイプを決定します。 オプションを以下に示します。
     - スーパーユーザ
     - フェデレーション管理者
     - 読み取り専用ユーザ

     注:各権限の説明については、［ヘルプ］をクリックしてください。
   - デフォルト管理者を無効にするかどうかを決定します。 中央管理者アカウントの使用を防ぐには、デフォルト管理者を無効にします。 デフォルト管理者なしで、監査可能な個別の管理者を使用します。
5. Administrative UI からログアウトし、変更が有効になるまで数分待ちます。
6. 新しい管理者の認証情報を使用して、Administrative UI に再度ログインします。
7. ［管理者］ページに戻り、管理者のリストが表示されることを確認します。
8. （オプション）。［アクション］メニューから、エントリを変更するか表示します。

   管理者の権限を変更でき、管理者を有効/無効にできます。

複数の管理者を使用して、フェデレーション管理タスクを分割できるようになりました。

デフォルトのマスタ管理者パスワードの変更（オプション）

セキュリティ上の理由で、Administrative UI へのデフォルト管理者アクセス権を与えるパスワードを変更します。 このタスクはオプションです。

管理者パスワードを変更するには、2 つの方法を使用できます。

- UI からパスワードを変更します。
- コマンド ラインからパスワードを変更します。

  管理者ユーザ アカウントがロックされているか、管理者を使用できない場合は、コマンド ラインからパスワードをリセットします。

UI からのデフォルト管理者パスワードの変更

Administrative UI から管理者パスワードを変更できます。

次の手順に従ってください:

1. ［インフラストラクチャ］タブから［パスワード］を選択します。

   ［管理者パスワードの変更］ダイアログ ボックスが表示されます。
2. 古いパスワードと新しいパスワードのフィールドに入力します。
3. ［サブミット］をクリックします。
4. システムを再起動します。

デフォルト管理者パスワードが変更され、アクティブになります。

コマンド ラインからのデフォルト管理者パスワードの変更

管理者ユーザ アカウントがロックされているか、使用できない場合は、XPSConfig ユーティリティを使用して、コマンド ラインから管理者パスワードを変更します。

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、コマンド ウィンドウを開きます。
2. 「XPSConfig」と入力します。

   UNIX プラットフォームで、以下に示すようなユーティリティの名前を入力します。 名前では大文字と小文字が区別されます。

   製品メニューが表示されます。

3. 「FED」と入力して、フェデレーション製品を選択します。
4. 「1」と入力します。

   オプション 1 はパスワードのオプションです。
5. 「C」と入力して、パスワードの値を変更します。
6. プロンプトに新しい値を入力します。
7. 「Q」と入力し、保存して終了します。

新しいパスワードがアクティブになります。

管理者のセッション管理

一度にアクティブにできるのは、1 つの管理セッションのみです。 単一の管理セッションは、管理者が新しいセッションを確立しようとする場合、フェデレーション オブジェクトの同時編集を防ぎます。 管理セッションが確立された後で、新しいログイン試行が行われた場合、警告メッセージを受信します。

警告メッセージは、セッションが存在することを管理者に伝えます。 同じ認証情報を使用して、管理者がログインに進んだ場合、システムは既存のセッションを無効にし、保存されていないデータはすべて失われます。最初のセッションが無効になった後、最初のセッションの管理者がログアウトされます。 管理者が任意の設定アクティビティを試行する場合、システムは管理者をログイン ダイアログ ボックスにリダイレクトします。

次のセクションでは、ある管理者がすでに確立されている管理者セッションと同じ認証情報を使用してログインを試みた場合、何が起きるかについて説明します。

管理セッションのやり取り

以下のシナリオで管理セッションの競合が発生します。

同じ認証情報でログインが試行された

管理者が CA SiteMinder® Federation Standalone にログインします。別の管理者または同じ管理者が、最初のログインと同じ認証情報を使用して、別のブラウザ セッションからログインを試みます。

CA SiteMinder® Federation Standalone は警告ダイアログを表示しますが、2 番目のユーザがログインすることを決定したので、CA SiteMinder® Federation Standalone は最初のセッションを無効にします。 最初のセッションの管理者がオブジェクトを変更しようとする場合、CA SiteMinder® Federation Standalone は新しいセッションが既存のセッションを無効にしたことを管理者に知らせます。また、CA SiteMinder® Federation Standalone は最初の管理者をログアウトさせます。

2 番目のユーザがログインしないことを選択できる

ブラウザ セッションが終了しても、管理者がログアウトしない

管理者が CA SiteMinder® Federation Standalone にログインします。管理者がログアウトせずにブラウザ セッションを閉じたり、ブラウザ セッションが予期せず閉じると、管理者はログアウトする機会がありません。 別の管理者は、最初の管理者と同じ認証情報を使用して、別のブラウザ セッションからログインします。

CA SiteMinder® Federation Standalone は警告ダイアログを表示しますが、2 番目のユーザがログインすることを決定したので、最初のセッションを無効にします。

最初のセッションの管理者がブラウザ セッションを再開し、オブジェクトを変更しようとする場合、CA SiteMinder® Federation Standalone は新しいセッションが既存のセッションを無効にしたことを管理者に知らせます。 ブラウザを閉じたとき、最初のセッションが無効にされました。

注: すべてのブラウザで、予期せず閉じられたブラウザ セッションを再開することができるとは限りません。 それらのブラウザについては、CA SiteMinder® Federation Standalone は、既存のセッションが無効であることを示すアラートを表示しません。

管理が無効である

管理者が CA SiteMinder® Federation Standalone にログインします。［システム設定］から、管理者は管理を無効にします。 別の管理者は、最初の管理者と同じ認証情報を使用してログインを試みます。 CA SiteMinder® Federation Standalone は警告ダイアログを表示しますが、2 番目のユーザがログインすることを決定します。 CA SiteMinder® Federation Standalone は最初のセッションを無効にします。

最初の管理者（ログアウトしなかったか、ブラウザを閉じなかった）は、管理を再度有効にしようとします。 CA SiteMinder® Federation Standalone は、最初の管理者にセッションが無効で、管理者をログアウトさせたことを伝えるメッセージを表示します。

UI 管理の無効化

［UI 設定］グループ ボックスでは、ローカル ホストの CA SiteMinder® Federation Standalone 管理を無効にし、再度有効にできます。

2 つの CA SiteMinder® Federation Standalone システムがフェールオーバをサポートするようにセットアップされている場合、UI の管理を無効にする機能が役立ちます。 CA SiteMinder® Federation Standalone の管理がプライマリ CA SiteMinder® Federation Standalone システムでのみ行われるようにすることができます。 設定をセカンダリ CA SiteMinder® Federation Standalone システムにエクスポートできます。

管理がプライマリ システムでのみ行われるように、UI 機能の無効化を使用して、セカンダリ システムの管理を無効にします。

次の手順に従ってください:

1. Administrative UI にログインします。
2. ［インフラストラクチャ］-［システム設定］に移動します。
3. ［UI 設定］グループ ボックスの［管理の無効化］をクリックします。

   管理を無効にすると、すべての管理アクションが妨げられます。

重要: ユーザがアクションを確認するとすぐに変更が有効になります。

管理を無効にすると、［管理が無効化されました］ダイアログ ボックスが表示され、UI の他のすべての部分が使用できなくなります。 その後のログイン試行では、警告メッセージと管理を再度有効にするボタンのみが表示されます。

管理を再度有効にする方法

［管理が無効化されました］ダイアログ ボックスの［管理の有効化］をクリックします。

UI のすべての部分が再びアクティブになります。

第 27 章: フェデレーション システムに対する SSL 管理

このセクションには、以下のトピックが含まれています。

Apache Web Server および UI 用の SSL 管理

以下の目的に対して SSL を有効にします。

- SSL 接続全体でフェデレーション トラフィックを処理すること。
- HTTP-Artifact シングル サインオン用のバックチャネルの安全な通信を有効にすること。
- Administrative UI への安全なアクセスを有効にすること。

埋め込まれた Apache の Web サーバにより、フェデレーション システムは SSL フェデレーション トラフィックを処理し、HTTP-Artifact シングル サインオン用のバックチャネルを安全にします。 埋め込まれた Tomcat Web サーバは、UI への安全なアクセスを許可します。

Apache および Tomcat の Web サーバ用の SSL を有効にするには、以下のプロセスを完了します。

1. サーバ証明書の証明書リクエストを作成します。
2. 認証機関（CA）によって発行される証明書をインポートします。
3. Administrative UI 内の SSL をアクティブ化します。［インフラストラクチャ］、［SSL 設定］に設定を置きます。

注: FIPS Migrate または FIPS のみのモードで作動する CA SiteMinder® Federation Standalone インストールでは、証明書用に FIPS 互換の暗号化キー アルゴリズムの使用が可能です。

Apache Web サーバおよび UI に対して SSL を有効にする方法

埋め込み Apache Web サーバおよび Administrative UI に対して SSL を有効にする手順は同じです。

- 以下の操作を実行する場合は、埋め込み Apache Web サーバに対して SSL を有効にします。
  - SSL 接続全体でフェデレーション 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。

  設定ウィザードを実行するときに SSL ポート番号が指定されることに注意してください。
- UI への接続のセキュリティを保護するために、Administrative UI に対して SSL を有効にします。

  SSL を有効にすることで、CA SiteMinder® Federation Standalone はサーバ証明書用の FIPS 互換の秘密キーを生成します。

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。

SSL 通信を有効にする方法：

1. サーバ証明書をリクエストします。
2. サーバ証明書に署名する CA 証明書を指定します。
3. 署名済みの証明書をシステムにアップロードします。

   証明書が正常にアップロードされたら、CA SiteMinder® Federation Standalone は SSL 接続をアクティブ化します。

これらの必要な手順に加えて、以下の操作を実行できます。

- 証明書署名リクエストを取得します。
- SSL を無効にします。
- システムから SSL 設定を削除します。

SSL サーバ証明書のリクエスト

SSL 接続の確立の最初の手順は、サーバ証明書リクエストを完了することです。 信頼された認証機関（CA）に完了済みのリクエストを送信し、この機関は署名済みのサーバ証明書を返します。

重要: SSL サーバ証明書をリクエストします。

次の手順に従ってください:

1. Administrative UI で、［インフラストラクチャ］-［SSL 設定］を選択します。

   ［SSL 設定］ダイアログ ボックスが表示されます。［SSL 設定ステータス］フィールドで、ステータスに［**サーバ証明書はリクエストされませんでした**］と表示されます。

2. ［リクエスト］をクリックして、証明書リクエストを作成します。
3. ［証明書リクエスト］ダイアログ ボックスのフィールドに入力し、［保存］をクリックします。

   特定のフィールドには、必須の値がすでに割り当てられています。［リクエスタ名］フィールドにはデフォルト値が提示されていますが、この値は変更が可能です。 ［リクエスタ名］値は、CA SiteMinder® Federation Standalone が展開されているサーバと関連付けられた完全修飾ドメイン名である必要があります。

   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。

証明書リクエストの作成時に、CA SiteMinder® Federation Standalone は秘密キーを生成します。 秘密キーは内部のファイルの場所に格納されます。

リクエストが生成されたら、証明書に署名する指定の CA にサーバ証明書リクエストを送信します。

認証機関は、生成された証明書リクエストに基づいて証明書を発行します。証明書の有効期間は、以下のいずれかの値に等しくなります。

- 認証機関のデフォルト値。
- リクエスタと認証機関の間のビジネス契約に基づく値。

署名されたサーバ証明書のアップロード

ユーザが証明書リクエストを完了した後、［SSL 設定ステータス］フィールドは「**サーバ証明書はリクエストされましたが、署名されませんでした**」を読み取ります。これは、証明書リクエストが署名されるのを待っていることを示しています。 CA SiteMinder® Federation Standalone は、Base64 でエンコードされた PEM 証明書または完全な PKCS #7 証明書/チェーン レスポンスを受け入れます。

ユーザが CA から署名された証明書を受信した後、その証明書をストレージ ロケーションにアップロードする必要があります。

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。

署名されたサーバ証明書をアップロードする方法

1. リクエストの開始時と同じ［SSL 設定］で開始します。
2. ［署名された証明書レスポンス］フィールド内の署名された証明書レスポンスを選択します。 ファイルを検索するには、［参照］をクリックします。

   注: SSL が複数のペアをサポートしないので、1 つのキーおよび証明書のペアのみが SSL 機能に必要とされます。

3. ［CA 証明書］フィールド内のプルダウン メニューから SSL 証明書を署名した CA を識別します。

   CA 証明書がキー ストアにない場合は、SSL 証明書リクエストを署名するために使用した CA 証明書のコピーをインポートします。［インポート］をクリックし、インポート手順を完了することで、証明書をインポートします。

4. ［適用］をクリックし、CA SiteMinder® Federation Standalone にサーバ証明書をアップロードします。

   確認メッセージが表示されます。そして、［SSL 設定］は証明書が現在更新されたことを反映して変更されます。

5. 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 startssl

     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。

サーバ証明書がシステムにアップロードされた後、CA SiteMinder® Federation Standalone は証明書を更新し、SSL をアクティブ化します。 証明書アップロードが成功したと仮定して、［SSL 設定ステータス］は［**SSL アクティブ**］を読み取ります。 設定グループ ボックス内のボタンは［非アクティブ］に変わります。

UI は、アップロードされた証明書が FIPS 承認されるかどうかも示します。

SSL の非アクティブ化

SSL が必要でなくなった場合は、SSL 設定を非アクティブ化できます。 たとえば、バック チャネル認証が必要でなくなったか、または UI への SSL 接続が必要でなくなった場合、SSL を非アクティブ化できます。

注: SSL が有効になっている Windows システムを再設定する場合は、システムを再設定する前に SSL 設定を非アクティブ化します。再設定が完了したら、SSL を再度アクティブにします。

次の手順に従ってください:

1. ［SSL 設定］ダイアログ ボックスで開始します。
2. ［埋め込み Web サーバ］または［管理 UI］セクションで、［非アクティブ化］をクリックします。

   SSL を無効にするかどうかを尋ねる確認プロンプトが表示されます。
3. ［はい］をクリックして非アクティブ化を完了します。
4. Administrative UI の場合にのみ、tomcat.keystore ファイルを手動で削除します。 このファイルは、以下のディレクトリにあります。

   federation_install_dir/secure-proxy/SSL/keys

   Administrative UI に対して SSL を非アクティブ化しても、対応するキーストア ファイルは削除されません。何らかの理由で UI SSL 証明書を変更した場合、証明書は更新されず、CA SiteMinder® Federation Standalone は不正な証明書を使用することになります。 Tomcat キー ストアを削除すると、SSL 証明書に対して行ったすべての更新が反映されていることを確認できます。
5. 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。

SSL 接続がアクティブではなくなり、［SSL 設定ステータス］設定が［**サーバ証明書は CA によって署名され、SSL の準備ができています**］に変わります。 ユーザが SSL を再度有効にできるように、証明書およびキー ファイルは残ります。

SSL の再アクティブ化

任意の理由で SSL を非アクティブ化した場合は、再アクティブ化します。SSL を有効にすることで、CA SiteMinder® Federation Standalone はサーバ証明書用の FIPS 互換の秘密キーを生成します。

注: Status 設定が「**CA が Server 証明書を署名、SSL の準備完了**」と読み取れる場合、SSL 接続をアクティブ化します。

次の手順に従ってください:

1. ［SSL 設定］ダイアログ ボックスで開始します。
2. ［埋め込み Web サーバ SSL 設定］グループ ボックス内の［アクティブ化］をクリックします。

   ［SSL 設定ステータス］設定は［**SSL アクティブ**］に変わり、ダイアログ ボックスに確認メッセージが表示されます。
3. 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 startssl

     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。

SSL が有効になりました。 証明書が期限切れになるまで、SSL 設定を変更する必要はありません。

SSL の証明書署名リクエストの置換または再サブミット

Apache サーバまたは UI によって使用中の秘密キー/証明書のペアと関連付けられた証明書署名リクエストのコピーを取得できます。 ユーザがリクエスト ファイルを削除またはファイルを保存しない場合、証明書署名リクエストのコピー取得機能は役立ちます。 また、リクエストのコピーは後日、署名された証明書が期限切れになる前に、リクエストを再サブミットするときに役立ちます。

「取得」機能では、証明書署名リクエストのコピーを取得できます。

注:「取得」オプションは、証明書がリクエストされ、ユーザが「再起動」ボタンを使用しても SSL 設定が削除されていない場合のみ、使用可能です。

証明書署名リクエストを取得する方法

1. UI から、［インフラストラクチャ］、［SSL 設定］を選択します。

   ［SSL 設定］ダイアログ ボックスが表示されます。

2. ［取得］をクリックします。

   「ファイルのダウンロード」ダイアログ ボックスが開き、ファイルを開くか保存するようにユーザに促します。

3. ファイルを保存します。

署名リクエストが取得され、UI は［SSL 設定］ダイアログ ボックスに戻ります。

埋め込み Apache サーバおよび UI からの SSL の削除

- 以下の場合に、埋め込み Apache Web サーバから SSL を削除します。
  - Artifact シングル サインオン用のバック チャネル接続が必要でなくなった場合
  - SSL を使用しなくなった場合
- UI への SSL 接続が必要でなくなった場合は、UI 接続から SSL を削除します。

［再起動］機能を使用すると、既存の SSL 設定を無効にし、SSL 設定と関連付けられたファイルをすべて削除できます。 特に、秘密キーとサーバ証明書および元のサーバ リクエスト ファイルが削除されます。

SSL を無効にし、関連するファイルを削除する方法

1. Administrative UI にログインします。
2. ［インフラストラクチャ］-［SSL 設定］を選択します。

   ［SSL 設定］ダイアログ ボックスが表示されます。
3. SSL を必要としない機能のグループ ボックスで、［再起動］をクリックします。

   再起動を確認するプロンプトが表示されます。
4. ［はい］をクリックします。

   SSL 設定がシステムから削除されます。
5. 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。

   サービスを再起動すると、Apache Web サーバおよび UI は非 SSL 操作に戻ります。 CA SiteMinder® Federation Standalone に着信する送信される後続の HTTPS リクエストは失敗します。 SSL を削除したので、今後はすべてのサービス URL が http で始まる必要があります。

SSL キーと証明書を移行する方法

CA SiteMinder® Federation Standalone r12.52 SP1 については、埋め込み Apache と Tomcat サーバの SSL キーと証明書ファイルが暗号化されます。リリース 12.0 と 12.0 SP1 については、これらのファイルは暗号化されません。 暗号化ファイルの新しいキー/証明書ペアを購入しないようにするには、r12.0/r12.0 SP1 から r12.52 SP1 に既存のキーまたは証明書ファイルを移行します。また、それらを移行せず、バックアップ目的でこれらのファイルをエクスポートできます。

重要: r12.1 の前のフェデレーション システムについては、埋め込み Tomcat サーバは自己署名証明書を使用します。 この自己署名証明書を r12.52 SP1 への移行に使用することはできません。 署名付き証明書を購入し、署名付き証明書で Tomcat SSL 設定をアップグレードします。

Apache については、r12.0 以降、SSL 接続のファイルを移行できます。Tomcat については、r12.1 以降からのみファイルを移行できます。リリース 12.0 では、自己署名証明書が Tomcat キー ストアをセキュリティ保護したためです。 r12.1 以降、フェデレーション システムでは認証機関による証明書の署名が必要です。

SSL キーと証明書ファイルの移行は次の状況で役立ちます。

- 既存のシステムをアップグレードする代わりに、新しいシステムで CA SiteMinder® Federation Standalone の別のバージョンに移動します。 既存のシステムから新しいシステムに SSL キーと証明書を移行します。
- クラスタのあるシステムから別のシステムに SSL キーと証明書を移行します。移行することで、キーと証明書を再利用できます。たとえば、ロード バランサが SSL リクエストをクラスタのフェデレーション システムに渡す場合、各システムが同じキーと証明書を使用する必要があります。 そのため、あるシステムから別のシステムにキーと証明書を移行します。

注: フェデレーション 12.0 システムを r12.52 SP1 にアップグレードする場合、インストーラは自動的に Apache と Tomcat SSL のキーと証明書ファイルを暗号化ファイルにアップグレードします。 これが移行に自動的に適用されることはありません。

証明書と秘密鍵ファイルは次のようになります。

Apache

- server.key ファイルには秘密鍵が含まれます。
- server.cert ファイルにはサーバ証明書が含まれます。

Tomcat

- r12.0 については、tomcat.keystore ファイルには自己署名証明書が含まれます。r12.1x については、tomcat.keystore ファイルには CA 署名証明書と秘密鍵のペアが含まれます。

これらのファイルを移行またはエクスポートするには、migratessl という名前の CA SiteMinder® Federation Standalone SSL ユーティリティを使用します。 移行ユーティリティは、Windows システムの場合はバッチ ファイルとして、UNIX システムの場合はシェル スクリプトとして製品に付属しています。 ユーティリティは、federation_install_dir/bin フォルダにインストールされます。

SSL ファイルを移行するプロセスは次のようになります。

1. r12.52 SP1 フェデレーション システムの任意の場所に既存の r12 フェデレーション システムからキーと証明書のファイルをコピーします。
2. キーと証明書のファイルをコピーした場所に migratessl ツールをコピーします。
3. 署名付き証明書を移行する場合、SSL 証明書に署名した認証機関証明書をエクスポートします。移行を続行する前に、CA 証明書をインポートします。

r12 System からキーと証明書をコピーします

SSL 移行ツールを使用するには、移行またはエクスポートを計画している CA SiteMinder® Federation Standalone システムのキーと証明書ファイルを最初に収集し、コピーします。

SSL キーと証明書ファイルをコピーする方法

1. 既存の CA SiteMinder® Federation Standalone システムでファイルの場所を確認します。

   Apache SSL キーと証明書ファイルは次の場所にあります。

   - *federation_install_dir*/secure-proxy/SSL/keys/server.key
   - *federation_install_dir*/secure-proxy/SSL/certs/server.crt

   Tomcat SSL キー ストア ファイルは次の場所にあります。

   - *federation_install_dir*/secure-proxy/SSL/keys/tomcat.keystore
2. 新しい CA SiteMinder® Federation Standalone マシンの任意の場所にキーと証明書のファイルをコピーします。

キー/証明書ファイルと同じフォルダに SSL 移行ツールをコピーします

SSL 移行ツールは、CA SiteMinder® Federation Standalone 12.1 SP3 で展開するソフトウェアを必要とします。CA SiteMinder® Federation Standalone 12.1 SP3 製品がインストールされているマシンでツールを実行します。 特に、ツールは移行するファイルをコピーした同じフォルダに置く必要があります。

SSL ユーティリティ ツールをコピーする方法

1. r12.52 SP1 システムの *federation_install_dir*/bin に移動します。
2. キーと証明書のファイルをコピーした、r12.52 SP1 システムの場所に migratessl ファイル（.bat または .sh）をコピーします。

SSL キーおよび証明書の移行またはエクスポート

migratessl ユーティリティを実行することにより、SSL キーまたは証明書ファイルの移行を実行します。

次の手順に従ってください:

1. 移行している SSL 証明書に最初に署名した認証機関証明書をインポートします。
   a. 移行元のシステムで、Administrative UI を使用して CA 証明書をエクスポートします。
   b. 移行先の新しいシステムで、Administrative UI を使用して CA 証明書をインポートします。
2. 既存のキーまたは証明書ファイルをコピーした新しいシステムで、コマンド ウィンドウを開きます。
3. コンポーネントをコピーしたフォルダに移動します。
4. 必要なコマンド引数と共に migratessl コマンドを指定します。 すべてのオプションについては、移行ツール コマンド引数のリストを参照してください。

   例

   - Apache SSL 接続用の SSL server.key を移行するには、以下を入力します。

   ```
   migratessl.bat -op migrate -keytype Apache
   -sourcefile server.key -certfile server.crt
   -sourcever 12.0 -sourceos Windows -oldpwd admin1
   -newpwd admin2 -issueralias trustedca
   ```

   - Tomcat SSL 接続用のキー/証明書ファイルを移行するには、以下を入力します。

   ```
   migratessl.sh -op migrate -keytype Tomcat
   -sourcefile tomcat.keystore -sourcever 12.1
   -sourceos UNIX -issueralias trustedca
   -oldpwd admin1 -newpwd admin2
   ```

   - Tomcat SSL 接続用のキー/証明書ファイルをエクスポートするには、以下を入力します。

   ```
   migratessl.sh -op export -keytype Tomcat
   -sourcefile tomcat.keystore -sourcever 12.1
   -sourceos UNIX  -dest ca/federationmgr/secure-proxy/
   SSL/keys/ -oldpwd admin1 -newpwd admin2
   ```

設定移行全体の一部として SSL キーおよび証明書を移行している場合は、パートナーシップを再アクティブ化することによって移行プロセスを完了します。

SSL 移行ツール コマンド引数

migratessl ツールはコマンド ラインで呼び出されます。 コマンドを入力する場合

- 各コマンド引数（Help フラグを除く）の後に値を 1 つだけ付けます。
- ディレクトリ パスなどのスペースがある値は、二重引用符で囲みます。

コマンド引数	意味
-op	Migrate または Export デフォルト： Migrate Apache のエクスポートの場合、-certfile 引数を指定した場合、ツールは server.key ファイルおよび server.crt ファイルをエクスポートします。 Tomcat の場合、ツールは PKCS#12 キー/証明書ファイルの tomcat.p12 ファイルをエクスポートします。
-keytype	Apache または Tomcat デフォルト： Apache
-sourcefile	SSL キー（Apache）または、キーと証明書を格納するキー ストア（Tomcat）を含むファイルの名前。
-certfile	Apache SSL サーバ証明書を含むファイルの名前（Apache のみ）。
-sourcever	12.0、12.1 など、キーまたは証明書が生成された CA SiteMinder® Federation Standalone のバージョン。 デフォルト： 12.0
-sourceos	キーが生成された環境のオペレーティング システム（Windows または UNIX）。 **注**: Linux のサポートは r12.1 SP3 で導入されたため、Linux オプションはありません。 デフォルト： ツールが実行されているマシンの OS。
-dest	出力ファイルのフォルダのパス。 移行の場合、このオプションは無視されます。 エクスポートの場合のデフォルト： 現在のフォルダ **重要**: 宛先フォルダを指定しない場合、移行しているファイルが上書きされます。

-issueralias	移行している証明書を署名した CA 証明書のエイリアス。 このエイリアスで CA 証明書を宛先の CA SiteMinder® Federation Standalone システムにインポートします （Migrate の場合にのみ使用され、Export の場合は無視されます）。
-oldpwd	キーのソースであるシステムの CA SiteMinder® Federation Standalone 管理パスワード。
-newpwd	キーの移動先のシステムの CA SiteMinder® Federation Standalone 管理パスワード。
-h	これらの使用手順を表示します。
-help	これらの使用手順を表示します。
-?	これらの使用手順を表示します。

第 28 章: フェデレーション アクティビティを監視するためのログ

このセクションには、以下のトピックが含まれています。

フェデレーション ログの概要

ログ記録を有効にすることにより、フェデレーション操作のトラブルシューティングを行います。 ログは、ユーザおよび CA サポートにとって重要な診断情報を提供します。

一部のログは、フェデレーション アクティビティに関する情報を提供します。 デフォルトでは、以下のログ記録が有効です。

- フェデレーション Web サービス（FWS）アプリケーション ログ（以下が含まれます）

  affwebservices.log -- このログ ファイルには、フェデレーション Web サービス (FWS) アプリケーションに関するメッセージが含まれます。このファイルのデフォルト パスは *federation_install_dir*¥logs¥fws です。

  FWSTrace.log -- このトレース ログには FWS ランタイム アクティビティに関する情報が含まれます。

- フェデレーション製品によって使用されるポリシー サーバに対するサーバ ログ（以下が含まれます）

  smtracedefault.log -- このトレース ログは、サーバ ランタイム アクティビティを追跡します。 このトレース ログのデフォルトの場所は、*federation_install_dir*¥logs¥server ディレクトリ内です。

  注: トレースを有効にすると、大きなログ ファイルが生成される可能性があります。

  smps.log -- このログ ファイルには、サーバに関する通知メッセージとトレース メッセージが含まれます。 このログ ファイルは、*federation_install_dir*¥logs¥server ディレクトリ内にあります。

- Administrative UI 運用ログ

  server.log -- このログ ファイルには、Administrative UI および組み込み SPS サーバに関するメッセージが含まれます。 このログ ファイルは、*federation_install_dir*¥logs¥ui ディレクトリ内にあります。

フェデレーション データ ストア オブジェクトのトレース ログ（XPSConfig_date_time_stamp.log）を有効にすることもできます。 このトレース ログは、データ ストア内のフェデレーション オブジェクトのトレース アクティビティを監視します。

チェックポイント ログ メッセージ

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、トランザクションを追跡するために使用できるトランザクション ID があります。 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 Web サービス（FWS）ログ

以下のログを有効にすることにより、FWS アプリケーション ランタイム アクティビティを監視できます。

- 参照用ログ記録（affwebserv.log）
- トレース ログ記録（FWSTrace.log）

ロールオーバ頻度やログ サイズなどのログ動作を管理するには、LoggerConfig.properties ファイル内の設定を変更します。

注: server.log ファイルを設定する logger.properties ファイルと LoggerConfig.properties ファイルを混同しないでください。 名前は似ていますが別のファイルです。

次の手順に従ってください:

1. 以下のディレクトリに移動します。
   federation_install_dir¥secure-proxy¥Tomcat¥webapps¥affwebservices¥WEB-INF¥classes¥

   注: UNIX オペレーティング環境では、パスにスラッシュ（/）を使用します。
2. テキスト エディタで LoggerConfig.properties ファイルを開きます。
3. （オプション）ログ設定を変更します。 LoggerConfig.properties ファイル内の各設定のオプションおよび説明を確認します。 設定項目は以下のとおりです。

   LoggingOn

   参照用ログ機能を有効または無効にします。

   LogFileName

   デフォルト： *federation_install_dir*¥¥logs¥¥fws¥¥affwebserv.log

   affwebserv.log はデフォルトのファイル名です。 名前は変更できます。

   LogLocalTime

   LogRollover

   LogSize

   LogCount

4. （オプション）FWS メッセージをログ記録するためのトレース設定を変更します。LoggerConfig.properties ファイル内の各設定のオプションおよび説明を確認します。

   TracingOn

   FWSTrace.log ファイルへの FWS トレース ログ記録を有効または無効にします。

   EnableDNSLookUp

   TraceFileName

   デフォルトの出力ファイル名は FWSTrace.log です。 この名前は変更できます。

   TraceConfigFile

   トレース設定ファイルを特定します。 この設定ファイルは、システムが監視してメッセージを記録する対象のコンポーネントおよびサブコンポーネントを特定します。

   TraceRollover

   TraceSize

   TraceCount

   TraceFormat

   TraceDelim

5. ファイルを保存して閉じます。
6. 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。

サーバトレース ログ

サーバ トレース ログ（smtracedefault.log ファイル）は、フェデレーション サーバのランタイム アクティビティを追跡します。このトレース ログのデフォルトの場所は、*federation_install_dir*¥logs¥server ディレクトリ内です。

注: トレースを有効にすると、大きなログ ファイルが生成される可能性があります。

サーバ側のトレース ログ記録をセットアップするには、以下の 2 つのタスクが必要です。

1. サーバ トレース ログの設定ファイルをセットアップします。（P. 464） 設定ファイルは、どのコンポーネントが監視され、smtracedefault.log ファイルに書き込まれるかを定義します。 デフォルト ファイル（smtracedefault.txt）を使用するか、または提供される他のテンプレートの 1 つを使用できます。
2. サーバ トレース ログ ファイル（smtracedefault.log）の動作を設定します（P. 465）。 ログ出力ファイルの場所、ログ設定ファイルの場所、ログ出力ファイルの形式、およびログ ロールオーバ頻度を指定します。

サーバトレース ログの設定ファイルのセットアップ

ログ設定ファイルをセットアップします。 ログ設定ファイルは、監視されるコンポーネント、それにより smtracedefault.log に書き込まれる内容を定義します。 フェデレーションに対して以下のいずれかのファイルを使用できます。

- smtracedefault.txt （デフォルト）
- samlidp_trace.template （アサーティング パーティのアクティビティ）
- samlsp_trace.template （依存パーティのアクティビティ）

効率を保つため、いずれか 1 つのテンプレートを使用してください。テンプレートの名前を LogTraceConfig パラメータに入力します。このパラメータにアクセスするには、XPSConfig コマンドで SM オプションを選択します。

注 ： 事前に設定済みのテンプレートは、*federation_install_dir*¥siteminder¥config¥profiler_templates にあります。

テンプレートの代わりに、デフォルトのファイルを使用し、すべてのフェデレーション コンポーネントをこのファイルに手動で追加できます。

次の手順に従ってください:

1. *federation_install_dir*¥siteminder¥config¥smtracedefault.txt に移動します。
2. テンプレート ファイルをバックアップします。
3. エディタで smtracedefault.txt ファイルを開きます。
4. 以下のテキストをコピーしてファイルに貼り付けることにより、ファイルを編集します。 既存のテキストを上書きします。

   ```
   components: Server/Policy_Server_General, IsProtected/Resource_Protection,
   Login_Logout/Authentication, Login_Logout/Policy_Evaluation,
   Login_Logout/Active_Expression, Login_Logout/Session_Management,
   IsAuthorized/Policy_Evaluation, JavaAPI,
   Fed_Server/Assertion_Generator, Fed_Server/Auth_Scheme,
   Fed_Server/Configuration
   data: Date, Time, Tid, TransactionID, SrcFile, Function, Domain,
   Resource, Action, User, SessionID, Data, AuthReason, Message
   ```

5. ファイルを保存します。

サーバトレース ログ ファイルの動作の設定

製品に付属している XPSConfig ツールを使用して、サーバ側のランタイム アクティビティを有効にできます。 XPSConfig は、対話型のコマンド ライン ユーティリティで、製品のパラメータを表示し、その設定を編集できるようにします。

次の手順に従ってください:

1. 製品がインストールされているシステム上でコマンド ウィンドウを開きます。
2. 「XPSConfig」と入力します。

   ここに示したとおりにコマンドを入力します。 コマンドでは大文字と小文字が区別されます。

   製品メニューが表示されます。
3. 「SM」と入力します。

   ［パラメータ］メニューが表示されます。 数字は、変更できる各パラメータと関連付けられています。
4. 変更するパラメータと関連付けられた数を入力します。
5. 値を変更するために「c」と入力します。
6. 「q」を入力して、パラメータリストに戻ります。
7. サーバのトレースの場合、以下の設定に対する値を指定します。

   LogTrace

   トレースを有効にします。 デフォルトではトレースがオフになります。これは、二重引用符の間のスペース（" "）によって示されます。 設定を空白のままにしないでください。

   LogTraceConfig

   この値はトレース設定ファイルをポイントします。 デフォルト値はありません。
   federation_install_dir¥siteminder¥config¥*template_name* を入力します。デフォルトのテンプレートは smtracedefault.txt です。 フェデレーション固有のその他トレース テンプレートがあります。

   例：

   federation_install_dir¥siteminder¥config¥samlidp_trace.template

   federation_install_dir¥siteminder¥config¥samlsp_trace.template

LogTraceConsole

メッセージがコンソール ウィンドウに表示されるかどうかを示します。 デフォルトでは、コンソールにログを出力しません。

LogTraceFormat

情報がログにどのように表示されるかを決定します。 デフォルトは sm です。 LogTraceDelimiter 設定と共に使用して、区切り文字として機能する文字を指定します。

LogTraceMode

トレース モードを指定します。 デフォルトは 0 です。

LogTraceDelimiter

ログ出力ファイルで区切り文字として機能する文字を特定します。 デフォルト値はありません。

LogTraceOutput

ログ出力ファイルの場所を指定します。 デフォルトは *federation_install_dir*¥logs¥server です。

8. ログ記録およびロールオーバー設定の変更による追跡ファイルのロールオーバーをどのぐらいの頻度で行うかを設定します。 変更するパラメータと関連付けられた数を入力します。

注: ログのロールオーバー設定の変更はすべて smtracedefault.log ファイルおよび smps.log ファイルに適用されます。

ロールオーバー パラメータは以下のとおりです。

LogFilesToKeep

維持するポリシー サーバ エラー ログの数を表します。 より古いファイルから削除されます。

LogRolloverDays

ロールオーバーが 1 日単位で発生するかどうかを示します。 ロールオーバーが行われるまでの日数に対応する数を入力します。

LogRolloverInterval

ロールオーバーが 1 時間単位で発生するかどうかを示します。 この値が設定された場合、LogRolloverDays が無視されます。

LogRolloverOnStart（**デフォルトで有効**）

サービスが開始された場合、ログ ファイルがロールオーバーされるかどうかを示します。

LogRolloverSize

ログ ファイルがロールオーバーされるサイズを示します。 次のロールオーバー間隔の前にサイズ制限に到達した場合も、ログ ファイルはロールオーバーされます。

LogRolloverTime

ロールオーバーを実行する時間を示します。 この設定は LogRolloverDays パラメータと共に使用されます。24 時間形式で、"時間：分" の形式で値を入力します。

例： "22:00"

9. パラメータの設定が完了したら、XPSConfig を終了するまで、「q」を入力し続けます。

XPSConfig ツールを終了するまで、XPSConfig で行なわれた変更は認識されません。 特記されている場合、変更によってはシステム サービスの再起動が必要です。

server.log ファイルのセットアップ

server.log ファイルは、製品の Administrative UI 操作を調査するのに役立ちます。 このログには、組み込まれた SPS サーバに関するメッセージも含まれます。 このログ ファイルはディレクトリ *federation_install_home*/logs/ui にあります。

logger.properties ファイルおよび log4j.properties ファイルには、server.log ファイルに記録される内容を決定するログ設定が含まれます。 これらの設定は、システムが実行時に読み取る名前/値のペアまたはディレクティブのグループです。

Logger.properties ファイル

logger.properties ファイルは *federation_install_dir*/secure-proxy/Tomcat/properties ディレクトリにあります。 ファイルの内容は、以下のセクションにグループ化されます。

- SvrConsoleAppender 設定
- SvrFileAppender 設定
- Server.conf 設定
- ログのロールオーバー設定

このファイルに記述されているディレクティブは、名=値という形式になります。 # 記号で始まるすべての行はコメントで、システムが設定を読み取る際に、読み取られることはありません。

注: Windows システム上のパス名は、2 つの円記号（¥¥）を使用します。

Log4j.properties

lo4j.properties ファイルは *federation_install_dir*/secure-proxy/Tomcat/webapps/fedui/WEB-INF/classes ディレクトリにあります。 このファイルは、Administrative UI 操作に対して記録されるログ レベルを決定します。

ログ ファイルを変更する手順は同じです。システムを再起動せずに、ファイルを変更できます。

次の手順に従ってください：

1. テキスト エディタでファイルを開きます。
2. 必要に応じて、ディレクティブを編集します。
3. ファイルを保存します。

ログ設定を変更します。

ログ設定

Server.conf 設定

logger.properties ファイル内の Server.conf 設定を使用して、ログ記録の有効化または無効化、ログ レベルの設定、ログ メッセージの出力形式の設定などが可能です。 このセクションで変更できるエントリは、以下の形式である必要があります。

```
log4j.rootCategory=<log_level>,<output_format>
```

log_level

メッセージのログ レベルを指定します。以下の値は、優先順位の低い順にリスト表示されています。

OFF、FATAL、ERROR、WARN、INFO、DEBUG、ALL

ログ記録を無効にするには、ログ レベルを OFF に設定します。値をそれ以外に設定すると、ログ記録は有効になります。

デフォルト： INFO

output_format

ログ メッセージが、コンソール、ファイル、またはその両方に表示されるかどうかを指定します。

デフォルト： SvrFileAppender

例： ログ レベルを INFO に設定し、メッセージをコンソールおよびファイルに表示されるようにするには、以下のエントリを使用します。

```
log4j.rootCategory=INFO,SvrConsoleAppender,SvrFileAppender
```

SvrConsoleAppender 設定

SvrConsoleAppender Settings セクションは、コンソールへのイベントのログ記録を制御します。 このセクションで変更できるエントリは以下のとおりです。

```
log4j.appender.SvrConsoleAppender.layout.ConversionPattern=<log_message_format>
```

log_message_format

コンソールに出力されるログ メッセージの形式を指定します。製品は log4j 日付パターン文字列をすべてサポートします。

デフォルト値： [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

SvrFileAppender 設定

SvrFileAppender Settings セクションは、ファイルへのイベントのログ記録を制御します。このセクションでは、ログ ロールオーバ頻度、およびファイルに書き込まれるログ メッセージの形式を定義します。 このセクションで変更できるエントリは、以下のとおりです。

```
log4j.appender.SvrFileAppender.File=<log_file_path>
log4j.appender.SvrFileAppender.Append=true
log4j.appender.SvrFileAppender.layout.ConversionPattern=<log_message_format>
```

log_file_path

ログ ファイルの名前とパスを指定します。

デフォルト名： server.log

デフォルト パス：
install_dir_home/secure-proxy/proxy-engine/logs/ui/server.log

true|false

既存のファイルにログ メッセージを追加するかどうかを指定します。この値を true に設定すると、既存のログ ファイルに新しいログ メッセージが追加されます。この値を false に設定すると、既存のログ ファイルがロールオーバされ、新しいログ ファイルが生成されます。

デフォルト値： true

log_message_format

server.log ファイルに書き込まれるログ メッセージの形式を指定します。 製品は log4j 日付パターン文字列をすべてサポートします。

デフォルト値： [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

使用されるログ ロールオーバのタイプ

log rolling セクションでは、いつ既存のログ ファイルがロールオーバされ、新しいログが生成されるかを決定します。 ファイルのサイズまたは日付に基づいてログがロールオーバされるようにします。

このセクションで変更できるエントリは、以下のとおりです。

```
log4j.appender.SvrFileAppender.MaxFileSize=1MB
log4j.appender.SvrFileAppender.MaxBackupIndex=10
#log4j.appender.SvrFileAppender.DatePattern='.'yyyy-MM-dd
```

MaxFileSize

ログ ファイルの最大サイズを指定します。このサイズに達すると、新しいログ ファイルが生成されます。

デフォルト値： 1 MB

MaxBackupIndex

システムが作成するログ ファイルの最大数を指定します。 ログ ファイルの数が MaxBackupIndex の数を超えると、最も古いログ ファイルが削除され、新しいファイルが生成されます。

デフォルト値： 10

DatePattern

ログ ファイルを作成する必要がある場合の日付を指定します。

デフォルト： yyyy-MM-dd

新しいログ ファイルは <*log_file_name*>.<*date_format*> という名前で作成されます

log_file_name

ログ ファイルの名前を指定します。

デフォルト： server.log

date_format

ログ ファイルが作成された日付を指定します。ファイルは log4j 日付パターン文字列をすべてサポートします。

デフォルト： yyyy-MM-dd

server.log 用の log4j.properties ファイル

log4j.properties ファイルは、server.log ファイルに書き込まれる追加の Administrative UI ログ記録を制御します。 このファイルは、*federation_install_dir*¥secure-proxy¥Tomcat¥webapps¥fedui¥WEB-INF¥classes ディレクトリにあります。

以下のエントリを変更できます。

```
log4j.appender.UIConsoleAppender.layout.ConversionPattern=<log_message_format>
```

log_message_format

コンソールに出力されるログ メッセージの形式を指定します。製品は log4j 日付パターン文字列をすべてサポートします。

デフォルト値： [%p] %c - %m%n

```
log4j.rootCategory=<log_level>,<output_format>
```

log_level

メッセージのログ レベルを指定します。以下の値は、優先順位の低い順にリスト表示されています。

OFF、FATAL、ERROR、WARN、INFO、DEBUG、ALL

ログ記録を無効にするには、ログ レベルを OFF に設定します。値をそれ以外に設定すると、ログ記録は有効になります。

デフォルト： INFO

output_format

ログ メッセージが、コンソール、ファイル、またはその両方に出力されるかどうかを指定します。

デフォルト： UIConsoleAppender

例： ログ レベルを INFO に設定し、メッセージをコンソールおよびファイルに表示されるようにするには、以下のエントリを使用します。

```
log4j.rootCategory=INFO,UIConsoleAppender,UIFileAppender
```

2 つの DEBUG エントリのコメントを解除することもできます。

フェデレーション データ オブジェクトのトレース ログ

XPS トレースを有効にしてフェデレーション データ ストア オブジェクトを監視します。 これらのアクティビティは smtracedefault.log に書き込まれます。 smtracedefault.log は、*federation_install_dir*¥logs¥server ディレクトリにあります。

次の手順に従ってください:

1. コマンド ウィンドウを開きます。
2. 「XPSConfig」と入力します。

   ここに示したとおりにコマンドを入力します。 コマンドでは大文字と小文字が区別されます。

   製品メニューが表示されます。
3. xTrace オプションに対して「X」を入力します。

   トレース メニューが表示されます。
4. fed オプションに関連付けられている数字を入力します。fed に関連するすべてのオプションが選択され、"x" でマークされます。
5. U を入力して選択を保存します。これによりトレース メニューが更新されます。
6. XPSConfig ツールが終了するまで q を入力します。
7. 変更を有効にするには、フェデレーション サービスを再起動します。
8. 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。

監査ログ

CA SiteMinder® Federation Standalone は、監査ログ（smaccess.log）をディレクトリ *federation_install_dir*/logs/server に自動的に作成します。XPSConfig コマンドを使用して、ユーザが認証イベントまたは認可イベント用のログ記録、あるいは両方のログ記録を有効にするまで、このログは空のままです。

注: UNIX プラットフォームでは、XPSConfig は大文字と小文字を区別します。

監査ログを有効にする方法

1. コマンド ウィンドウを開きます。
2. コマンド プロンプトで「XPSConfig」と入力します。

   製品メニューが表示されます。
3. 「SM」と入力します。

   現在の値が示されたパラメータのリストが表示されます。
4. （オプション）設定のリストをフィルタするには「**f**」を入力します。

   フィルタの入力プロンプトで、監査ログに関連するすべての設定を見つけるには**レポート**と入力します。
5. 有効にする監査ログのタイプに関連付けられている数字を入力します。

   ReportAuth

   認証イベント用のログ設定を指定します。

   ReportAz

   認可イベント用のログ設定を指定します。

6. 値を変更するために「c」と入力します。 デフォルトは 0 です。これは、イベントがログ記録されないことを意味します。
7. プロンプトで以下のいずれかの値を入力します。

   1 = イベントをすべてログ記録します

   2 = 拒否イベントのみをログ記録します
8. 製品メニューに戻るまで「q」を入力します。

   監査ログが有効になりました。

注: いつでもこの手順を繰り返し、監査ログ設定の設定を更新することができます。

監査ログ名および場所の設定（オプション）

監査ログのデフォルトの名前は smaccess.log で、デフォルトの場所は *federation_install_dir*/logs/server です。 これらの値は変更できます。

次の手順に従ってください:

1. コマンド ウィンドウを開きます。
2. コマンド ライン プロンプトで「XPSConfig」と入力します。

   注: UNIX プラットフォームでは、XPSConfig は大文字と小文字を区別します。

   製品メニューが表示されます。
3. 「SM」と入力します。

   パラメータのリストおよびそれらの値が表示されます。
4. （オプション）設定のリストをフィルタするには「**f**」を入力します。

   フィルタの入力プロンプトで、監査ログ テキスト ファイル名に関連する設定を見つけるには **text** と入力します。
5. ReportTextFile 設定と関連付けられた数を入力します。

   現在の値が表示されます。
6. ファイル名を変更するには、「c」と入力します。

7. 有効なパスおよび新しいファイル名を入力します。
8. システム コマンド プロンプトに戻るまで「q」を入力します。

新しいファイル名と場所が保存されます。

監査ログに対する ODBC データベースの使用（オプション）

デフォルトのテキスト ファイルを使用する代わりに、ODBC データベースを使用して監査データを記録できます。

次の手順に従ってください:

1. 監査ログのストレージ タイプを ODBC に変更します。
2. ODBC データ ソースを設定します。以下のいずれかの手順に従います。
   - Windows システムでの SQL Server データ ソースの作成 (P. 478)
   - UNIX システムでの SQL Server データ ソースの作成 (P. 479)
   - Windows での Oracle データ ソースの作成 (P. 480)
   - UNIX システムでの Oracle データ ソースの作成 (P. 482)

監査ログのストレージ タイプの変更

監査ログのデフォルトはテキスト形式です。 監査データを ODBC データベースに格納するには、ログのストレージ タイプを変更します。

重要: 監査ログ ストレージ タイプを「TEXT」から「ODBC」に変更した場合、それを元に戻すことはできません。

次の手順に従ってください:

1. コマンド ウィンドウを開きます。
2. コマンド ライン プロンプトで「XPSConfig」と入力します。

   注: UNIX プラットフォームでは、XPSConfig は大文字と小文字を区別します。

   ［製品］メニューが表示されます。

3. 「SM」と入力します。

   パラメータのリストおよびそれらの現在値が表示されます。

4. （オプション）設定のリストをフィルタするには「**f**」を入力します。

   監査ログのストレージ タイプに関連付けられているすべての設定を検索するには、［フィルタの入力］プロンプトで「store」と入力します。

5. LogStoreNamespace に数値を入力します。

   現在の値が表示されます。

6. ストレージ タイプを変更するには「c」と入力します。

7. プロンプトで「**ODBC**」と入力します。

   注: エントリにコロンを含めます。

8. パラメータのリストに戻るには、q を 2 回入力します。

9. これらの追加項目を設定します。 設定ごとに数値を入力し、変更を行います。

   注: 設定のリストをフィルタするには「**f**」と入力します。監査ログ データベースに関連付けられているすべての設定を検索するには、［フィルタの入力］プロンプトで「**Db**」と入力します。

   DbLogAdminName

   監査ログ用のデータ ソース ユーザ名を指定します。

   制限： 文字列。LogStoreNamespace が ODBC: に設定されている場合にのみ適用されます。

   DbLogAdminPassword

   監査ログ用のデータ ソース ユーザ パスワードを指定します。

   制限： 文字列。LogStoreNamespace が ODBC: に設定されている場合にのみ適用されます。

   DbLogDataSource

   監査ログ用のデータ ソース名を指定します。

   制限： 文字列。LogStoreNamespace が ODBC: に設定されている場合にのみ適用されます。

DbLogMaxConnections

監査ログ用のデータ ソースへの接続の最大数を指定します。

デフォルト： 15

制限： 整数である必要があります。LogStoreNamespace が ODBC: に設定されている場合にのみ適用されます。

DbLogUseDefault

監査ログがポリシー ストアと同じ ODBC データ ソースを使用するかどうかを指定します。

デフォルト： FALSE

制限： TRUE または FALSE。LogStoreNamespace が ODBC: に設定されている場合にのみ適用されます。

10. システム コマンド プロンプトに戻るには、何度も「q」を入力します。
11. 監査データの記録に ODBC データベースを使用するには、データ ソースをセットアップ (P. 476)します。

Windows システムでの SQL Server データ ソースの作成

ODBC で、SQL Server ワイヤ プロトコル用のデータ ソースを設定する必要があります。

Windows 上でデータ ソースを作成する方法

1. 以下のいずれかを実行します。
   - サポートされた 32 ビットの Windows オペレーティング システムを使用している場合は、［スタート］をクリックし、［プログラム］-［管理ツール］-［ODBC データ ソース］を選択します。
   - サポートされる 64 ビットの Windows オペレーティング システムを使用している場合：
     a. *install_home*¥Windows¥SysWOW64 に移動します。
     b. odbcad32.exe をダブルクリックします

   ［ODBC データ ソース アドミニストレータ］が表示されます。
2. ［システム DSN］タブをクリックします。

   システム データ ソース設定が表示されます。

3. ［追加］をクリックします。

   ［データ ソースの新規作成］ダイアログ ボックスが表示されます。

4. SiteMinder SQL Server Wire Protocol を選択し、［完了］をクリックします。

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

5. ［データ ソース名］フィールドにデータ ソース名を入力します。

   例： CA SiteMinder® Federation Standalone Data Source

   注: 指定したデータ ソース名を書き留めます。 この情報は、ポリシー ストアとしてデータベースを設定するときに必要です。

6. ［サーバ］フィールドに MS SQL Server ホスト システムの名前を入力します。

7. ［データベース名］フィールドにデータベース名を入力します。

8. ［テスト］をクリックします。

   接続設定がテストされ、接続に成功したことを示すメッセージが表示されます。

9. ［OK］をクリックします。

   SQL Server データ ソースが設定され、［システム データ ソース］リストに表示されます。

UNIX システムでの SQL Server データ ソースの作成

CA SiteMinder® Federation Standalone ODBC データ ソースは、system_odbc.ini ファイルを使用して設定します。このファイルは、*federation_install_dir*/siteminder/db にある sqlserverwire.ini の名前を system_odbc.ini に変更することによって作成します。 この system_odbc.ini ファイルには、使用可能な ODBC データ ソースの名前すべてと、それらのデータ ソースと関連付けられた属性が含まれています。このファイルは、サイトごとに機能するようカスタマイズする必要があります。 また、このファイルにはデータ ソースを追加できます。たとえば、SiteMinder 用の追加の ODBC ユーザ ディレクトリを定義できます。

system_odbc.ini ファイルの最初のセクション [ODBC Data Sources] には、現在使用可能なデータ ソースすべてのリストが含まれています。「=」の前の名前は、個別のデータ ソースそれぞれを説明する、ファイルの後続のセクションを示しています。「=」の後には、コメント フィールドがあります。

注: データ ソース エントリの最初の行である [SiteMinder Data Source] を変更する場合、変更内容を書き留めておきます。この値は、ODBC データベースをポリシー ストアとして設定するときに必要になります。

system_odbc.ini ファイル内には、各データ ソースの属性を記述するセクションがあります。最初の属性は、このデータ ソースが SiteMinder によって使用されるときにロードされる ODBC ドライバです。残りの属性は、そのドライバに固有です。

MS SQL Server データ ソースを追加する場合は、ファイルの [ODBC Data Sources] セクションに新しいデータ ソース名を追加し、データ ソースと同じ名前を使用してデータ ソースを記述するセクションを追加する必要があります。 新しいサービス名を作成したり、別のドライバを使用する場合は、system_odbc.ini ファイルを変更する必要があります。 [SiteMinder Data Source] の下に Oracle または SQL ドライバのエントリがあります。

MS SQL Server データ ソースを設定するには、最初に *federation_install_dir*/siteminder/db ディレクトリに system_odbc.ini ファイルを作成する必要があります。 このためには、*federation_install_dir*/siteminder/db にある sqlserverwire.ini の名前を system_odbc.ini に変更する必要があります。

Windows での Oracle データ ソースの作成

Oracle データベース用の ODBC データ ソースを作成します。

次の手順に従ってください:

1. 以下のいずれかを実行します。
   - サポートされた 32 ビットの Windows オペレーティング システムを使用している場合は、［スタート］をクリックし、［プログラム］-［管理ツール］-［ODBC データ ソース］を選択します。

- サポートされた 64 ビットの Windows オペレーティング システムを使用している場合：

  a. *install_home*¥Windows¥SysWOW64 に移動します。

  b. odbcad32.exe をダブルクリックします

［ODBC データ ソース アドミニストレータ］が表示されます。

2. システム［システム DSN］タブをクリックし、次に［追加］をクリックします。

   ［データ ソースの新規作成］ダイアログ ボックスが表示されます。

3. SiteMinder Oracle Wire Protocol を選択し、［完了］をクリックします。

   ODBC Oracle Wire Protocol ドライバのセットアップ ダイアログ ボックスが表示されます。 ［一般］タブが前面に移動します。

4. ［データ ソース名］フィールドにデータ ソースを識別する名前を入力します。

   注: この名前を記録しておいてください。 このデータ ソース名は、ポリシー サーバに参照データベースを指定するときに必要です。

5. ［ホスト名］フィールドに、Oracle データベースがインストールされているマシンの名前を入力します。

6. ［ポート番号］フィールドに、マシン上で Oracle データベースがリスニングするポートの番号を入力します。

7. ［SID］フィールドに、接続する Oracle インスタンスの名前を入力します。

   注: サービス名は tnsnames.ora ファイル内で指定されます。SID はデータベース インスタンス用のシステム識別子です。 tnsnames.ora ファイルは、Oracle インスタンスを識別し、接続するために Oracle が使用するサービス名と詳細が含まれています。

   例： tnsnames.ora ファイルが Oracle インスタンスの以下のエントリを含んでいる場合、［SID］フィールドに instance1 を入力します。

```
instance1=
    (Description=
    (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
    (Connect_DATA_ = (SID = SIDofinstance1))
    )
```

8. ［テスト接続］をクリックします。

   接続設定がテストされ、接続に成功したことを示すメッセージが表示されます。

9. ［OK］をクリックします。

   Oracle のデータ ソースがワイヤ プロトコル ドライバに対して設定されます。

UNIX システムでの Oracle データ ソースの作成

SiteMinder ODBC データ ソースは、system_odbc.ini ファイルを使用して設定します。このファイルは、*federation_install_dir*/*siteminder*/db にある oraclewire.ini の名前を system_odbc.ini に変更することによって作成します。 この system_odbc.ini ファイルには、使用可能な ODBC データ ソースの名前すべてと、それらのデータ ソースと関連付けられた属性が含まれています。 このファイルは、サイトごとに機能するようカスタマイズする必要があります。 また、このファイルにはデータ ソースを追加できます。たとえば、SiteMinder 用の追加の ODBC ユーザ ディレクトリを定義できます。

system_odbc.ini ファイルの最初のセクション [ODBC Data Sources] には、現在使用可能なデータ ソースすべてのリストが含まれています。「=」の前の名前は、個別のデータ ソースそれぞれを説明する、ファイルの後続のセクションを示しています。「=」の後には、コメント フィールドがあります。

注: データ ソース エントリの最初の行である [SiteMinder Data Source] を変更する場合、変更内容を書き留めておきます。この値は、ODBC データベースをポリシー ストアとして設定するときに必要になります。

system_odbc.ini ファイル内には、各データ ソースの属性を記述するセクションがあります。最初の属性は、このデータ ソースが SiteMinder によって使用されるときにロードされる ODBC ドライバです。残りの属性は、そのドライバに固有です。

Oracle データ ソースの追加には、ファイルの [ODBC Data Sources] セクションに新しいデータ ソース名を追加することと、データ ソースと同じ名前を使用して、そのデータ ソースを記述するセクションを追加することが含まれます。 新しいサービス名を作成したり、別のドライバを使用する場合は、system_odbc.ini ファイルを変更する必要があります。 [SiteMinder Data Source] の下に SQL Server または Oracle ドライバのエントリがあります。

Oracle データ ソースを設定するには、最初に *federation_install_dir*/siteminder/db ディレクトリに system_odbc.ini ファイルを作成する必要があります。 このためには、*federation_install_dir*/siteminder/db にある oraclewire.ini の名前を system_odbc.ini に変更する必要があります。

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 Standalone 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 Standalone 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]]
```

第 29 章: フェデレーション システム設定のリストア

このセクションには、以下のトピックが含まれています。

前の設定へシステムをリストアする方法

以前にバックアップされた設定に戻ることによりシステム設定をリストアできます。 現在の設定に問題がある場合、前の設定に戻ることは役立つ可能性があります。

前の設定に戻るプロセスを以下に示します。

1. リストアしたいシステム用の既存の CA SiteMinder® Federation Standalone 設定をバックアップします。
2. ユーザがバックアップ設定を作成したときにあったものと同じ設定を使用して、このシステムで設定ウィザードを実行します。

   ユーザが設定ウィザードを実行する場合、設定は同じままである必要があります。

   以下の設定が元の設定に一致する必要があります。

   - **展開設定**

     新しいシステム用に同じ展開モード（プロキシまたはスタンドアロン）を選択します。

   - **ポート番号**

     バックアップされたシステムが使用するのと同じポートを指定します。

- **仮想ホスト名**

  最初に設定されたとき、システムが仮想ホストを使用した場合は、同じ仮想ホスト名を使用します。 さらに、システム用にホストファイルで適切なエントリをします。

- **SiteMinder コネクタ**

  システムが SiteMinder コネクタを使用した場合は、再度 SiteMinder コネクタを選択します。

3. システムへバックアップされた設定をインポートします。

以下のセクションは、プロセスについて詳述します。

既存の設定のバックアップ

ユーザ設定のバックアップは、フェデレーション システムのリカバリや移行に役立ちます。

注: この手順はバージョン r12.52 SP1 以上に適用されます。

設定をバックアップするには、設定データをエクスポートします。 製品に付属する XPSExport ツールを使用して、設定データを XML ファイルにエクスポートできます。

重要: エクスポート プロセス中は、フェデレーション トランザクションが正常に処理できません。

次の手順に従ってください:

1. コマンド ウィンドウを開きます。
2. 設定をエクスポートするには、以下のコマンドを入力します。

   XPSExport *export_file_name* -xe -xp -passphrase *passphrase*

   export_file_name

   エクスポートの結果の出力ファイルに名前を付けます。 XPSExport からの出力は、XML 形式であるため、ファイル名は拡張子 **.xml** で終わる必要があります。

   passphrase

   機密データを暗号化するために必要なパスフレーズを指定します。パスフレーズは、8 文字以上で、1 つ以上の数字、1 つ以上の大文字、および 1 つ以上の小文字を含んでいる必要があります。 パスフレーズにスペースが含まれる場合、引用符で囲む必要があります。

   注: パスフレーズを直接入力しない場合は、コマンドからそれを外します。 そうすると、XPSExport はパスフレーズの入力およびパスフレーズの確認を求め、画面に表示されません。

エクスポートにより、暗号化された設定データが含まれる XML ファイルが生成されます。 設定をリストアするにはこのファイルを使用します。

バックアップ設定に戻す

既存の CA SiteMinder® Federation Standalone 設定に問題が発生した場合、同じシステムの以前にバックアップした設定に戻します。

設定をリストアするには、製品に付属する XPSImport ツールを使用して、XML ファイルをインポートします。

重要: 説明の通りに正確にインポート手順に従います。手順が完了するまで、Administrative UI の［証明書 & キー］タブにアクセスしないでください。

次の手順に従ってください:

1. フェデレーション データ用の新しいデータベース インスタンスを設定します。

   重要: この手順では既存のデータベースを使用しないでください。 使用すると、インポートは失敗します。

2. 入力を求められたら、新しいデータベース インスタンスを指定して、設定ウィザードを実行します。

   この新しい設定には、元の設定に使用されたのと同じ設定を使用します。 これらの設定には以下のものが含まれます。

   - 展開モード
   - ポート番号
   - 仮想ホスト名
   - SiteMinder コネクタ

3. プラットフォームに応じてフェデレーション サービスを停止します。

   Windows

   停止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。

   ［スタート］、［すべてのプログラム］、［CA］、［Federation Standalone］、［サービスの停止］を選択します。

   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

      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。

4. XPSImport コマンドを使用して、設定データをすべてリストアします。

   XPSImport *export_file_name* -passphrase *passphrase*

   export_file_name

   元の設定のエクスポートの結果の XML ファイルに名前を付けます。ファイル名は拡張子 **.xml** で終わる必要があります。

   passphrase

   機密データを復号化するために必要なパスフレーズを指定します。パスフレーズは、8 文字以上で、1 つ以上の数字、1 つ以上の大文字、および 1 つ以上の小文字を含んでいる必要があります。 パスフレーズにスペースが含まれる場合、引用符で囲む必要があります。

5. 設定ウィザードを再実行します。

   この新しい設定には、元の設定に使用されたのと同じ設定を使用します。 これらの設定には以下のものが含まれます。

   - 展開モード
   - ポート番号
   - 仮想ホスト名
   - SiteMinder コネクタ

6. （オプション）元の設定で、SiteMinder コネクタが有効にされていた場合は、以下の手順に従って、コネクタを再設定します。

   a. Administrative UI にログインします。

   b. ［インフラストラクチャ］タブをクリックし、［展開設定］を選択します。

   c. 元の設定で使用されていた同じ値を使用して、コネクタ設定を再設定します。

   d. ［ホストの登録］をクリックして、ポリシー サーバにフェデレーション システムを再登録します。

設定が元の状態にリストアされます。

第 30 章: トラブルシューティング

このセクションには、以下のトピックが含まれています。

システム パフォーマンス トラブルシューティング

以下の問題は、システム パフォーマンス トラブルシューティングを説明しています。

高負荷環境でのセッション ストア タイムアウトの設定

高負荷環境では、アイドル タイムアウトしたセッションや期限切れになったセッションの削除など、セッション ストアの保守タスクに必要な、実行時間の長いクエリがタイムアウトになる可能性があります。MaintenanceQueryTimeout レジストリ設定の値を増加させることにより、セッション ストアの保守タスクのタイムアウト（デフォルトでは 60 秒）を調節します。 メンテナンス スレッドがタスクを正常に完了できるように、値を増やします、

MaintenanceQueryTimeout レジストリ設定は次の場所にあります。

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Netegrity¥SiteMinder¥CurrentVersion¥SessionServer
```

プロキシ エンジンのハングアップおよびリクエスト処理の停止

症状:

数日間のリクエスト処理後、CA SiteMinder® Federation Standalone のビルトイン プロキシ エンジンがハングアップします。

解決方法:

Apache Web サーバ（HTTP リスナ）とプロキシ エンジン（Tomcat サーブレット エンジン）との間の接続用プロキシ エンジンの server.conf 内にあるチューニング パラメータを変更します。

コンポーネントとなる mod_jk が、変更されたパラメータを使用します。mod_jk は Apache JServ プロトコル（AJP）を使用して、Apache Web サーバと Tomcat との間の通信を有効にする Tomcat コネクタとして働きます。

server.conf ファイルを変更する方法

1. 以下のディレクトリに移動します。

   federation_install_dir/secure-proxy/proxy-engine/conf

2. エディタで server.conf ファイルを開きます。

3. 以下のパラメータを変更します

   worker.ajp13.reply_timeout

   プロキシ エンジンから受信した任意の 2 つのパケット間で経過できる最大時間をミリ秒単位で指定します。 このタイムアウトが期限切れになった後、Apache サーバ（HTTP リスナ）とプロキシ エンジンとの間の接続がドロップされます。 0 の値は、レスポンスが受信されるまでプロキシ エンジンが無期限に待機することを示します。

   接続がプロキシ エンジンからのレスポンスを無期限に待たないことを確認するには、この値を増加させます。

   デフォルト： 0

   worker.ajp13.retries

   mod_jk コンポーネントが通信エラーの場合にプロキシ エンジンに接続リクエストを送信する最大回数を示します。 再試行の回数が満たされ、プロキシ エンジンからのレスポンスがない場合、接続はドロップされます。

   接続リクエスト用の再試行の回数を増やすには、この値を増加させます。

   デフォルト： 2

4. server.conf ファイルを保存します。

署名検証の失敗の解決

悪意のあるユーザは、署名を無効にせずにドキュメントのコンテンツを変更することにより、XML シグネチャ ラッピング攻撃を実行できます。 デフォルトでは、ポリシー サーバおよび Web エージェント オプション パックのソフトウェア制御には、シグネチャ ラッピング攻撃に対する防御が設定されています。 ただし、サードパーティ製品は、XML 仕様に準拠しない方法で XML ドキュメントを発行できます。 その結果、デフォルトの署名確認によって署名検証が失敗する場合があります。

署名検証の失敗は、以下の理由で発生します。

- 重複した ID 要素が XML ドキュメント内にあり、署名がこの重複した ID を参照している場合。 重複した ID 属性は許可されていません。
- XML 署名が、想定された親要素を参照していない場合。シグネチャ ラッピングの脆弱性がログに記録されます。

フェデレーション トランザクションが失敗する場合は、署名検証の失敗に関する smtracedefault.log ファイルおよび fwstrace.log ファイルを確認します。 これらのエラーは、受信した XML ドキュメントが XML 標準に準拠していないことを示す場合があります。 回避策として、シグネチャ ラッピング攻撃に対するデフォルトのポリシー サーバおよび Web エージェント保護を無効にできます。

重要: 署名の脆弱性に対する保護を無効にした場合は、これらの攻撃に対する別の保護対策を決定します。

XML シグネチャ ラッピングの確認を無効にする方法

1. xsw.properties ファイルに移動します。このファイルは、ポリシー サーバと Web エージェントで別の場所に存在します。
   - ポリシー サーバの smtracedefault.log ファイルでエラー メッセージが発生した場合は、*siteminder_home*/config/properties に移動します
   - Web エージェントの fwstrace.log でエラー メッセージが発生した場合は、

     *web_agent_option_pack_ home/*affwebservices/web-INF/classes に移動します。

     注: Web エージェント オプション パックが Web エージェントと同じシステムにインストールされている場合、このファイルは *web_agent_home* ディレクトリに存在します。
2. 以下の xsw.properties 設定を true に変更します。
   - DisableXSWCheck=true （ポリシー サーバ設定のみ）
   - DisableUniqueIDCheck=true （ポリシー サーバおよび Web エージェント オプション パック設定）

     注: DisableUniqueIDCheck 設定の値は、ポリシー サーバと Web エージェント オプション パックで同じである必要があります。
3. ファイルを保存します。

2 つの SSO トランザクションで同じブラウザ セッションを使用すると失敗する

症状:

ユーザが、同じブラウザ セッションで 2 つのシングル サインオン トランザクションを試行します。 そのトランザクションは、同じアサーティング パーティから別の依存パーティへのものです。 最初のトランザクションは成功しますが、2 番目のトランザクションはアサーティング パーティで許可失敗になります。2 つのパートナーシップで異なるアサーティング パーティ ユーザ ディレクトリを使用するように設定されるため、問題が発生します。

CA SiteMinder® Federation Standalone がアサーティング パーティでシングル サインオン トランザクションを開始する場合、ブラウザにセッション Cookie を配置します。 このセッション Cookie は、ユーザ ID およびアサーティング パーティ ユーザ ディレクトリに関する情報を含みます。一度にブラウザに存在できるのは、1 つの CA SiteMinder® Federation Standalone セッション Cookie のみです。

最初のトランザクションと同じブラウザ セッションで別のトランザクションを試行する場合、最初のトランザクションのセッション Cookie はブラウザに残ります。ただし、このセッション Cookie は、2 番目のパートナーシップについての正しい情報を持っていないため、許可操作は失敗します。

解決方法:

各パートナーシップのアサーティング パーティ ユーザ ディレクトリが同じである場合にのみ、異なるシングル サインオン トランザクションに同じブラウザ セッションを使用します。

パートナーシップごとに異なるアサーティング パーティ ユーザ ディレクトリが設定されている場合は、最初のブラウザ セッションを閉じ、新しいブラウザ セッションを開始して 2 番目のトランザクションを試行します。

システムのトラブルシューティングのためのセキュア プロキシ エンジン ログの確認

パートナーシップ用の CA SiteMinder® Federation Standalone には、トラフィックをバックエンド サーバに転送するセキュア プロキシ エンジンが含まれます。 セキュア プロキシ エンジンには、以下のコンポーネントが含まれます。

- Apache Web サーバ

  HTTP リスナとして働き、適切に設定すれば、受信要求用 HTTP トラフィックを処理します。HTTPS トラフィックを処理することもできます。

- Tomcat サーバ

  Administrative UI の操作のためのサーブレット コンテナを提供します。Apache Web サーバは mod_jk という名前の Tomcat コネクタを介して Tomcat サーバに通信します。

CA SiteMinder® Federation Standalone 環境で問題をトラブルシュートするために、これらのコンポーネントと関連するログ ファイルを伴う CA サポートを提供できます。

CA SiteMinder® Federation Standalone トラブルシューティングを支援する 2 つの Apache のログは、以下のとおりです。

mod_jk.log

製品では、mod_jk.log がデフォルトで有効になっています。 フェデレーション サーバとの最初の接続の後、このファイルへの情報のログが開始されます。

このログ ファイルを変更する方法

1. *federation_install_dir*¥secure-proxy¥httpd¥conf に移動します。
2. httpd.conf ファイルを開きます。
3. これらの設定を反映させるように、ファイルの以下の行を設定します。

   ```
   JkLogFile "path_to_mod_jk_log"
   JkLogLevel debug
   JkRequestLogFormat "%w %V %T %m %h %p %U %s"
   ```

   注: パス「logs/mod_jk.log」は、JkLogFile エントリのデフォルトの場所です。 デフォルトを使用するか、または任意の場所にこのパスを設定します。

mod_jk.log を無効にするには、コメント アウトするか、またはファイルからこれらの行を消去します。

httpclient.log

デバッグする場合のみ、httpclient.log を有効にできます。httpclient.log ファイルは、*federation_install_dir*¥secure-proxy¥proxy-engine¥logs にあります。

このログ ファイルを変更する方法

1. *federation_install_dir*¥secure-proxy¥proxy-engine¥conf に移動します。
2. server.conf ファイルを開きます
3. 以下の行を変更します。

   ```
   httpclientlog="yes"
   ```

httpclient.log ファイルの場所とログ レベルを変更するには、httpclientlogging.properties ファイルを編集します。 このファイルをディレクトリ *federation_install_dir*¥secure-proxy¥Tomcat¥properties に配置します。

第 31 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。

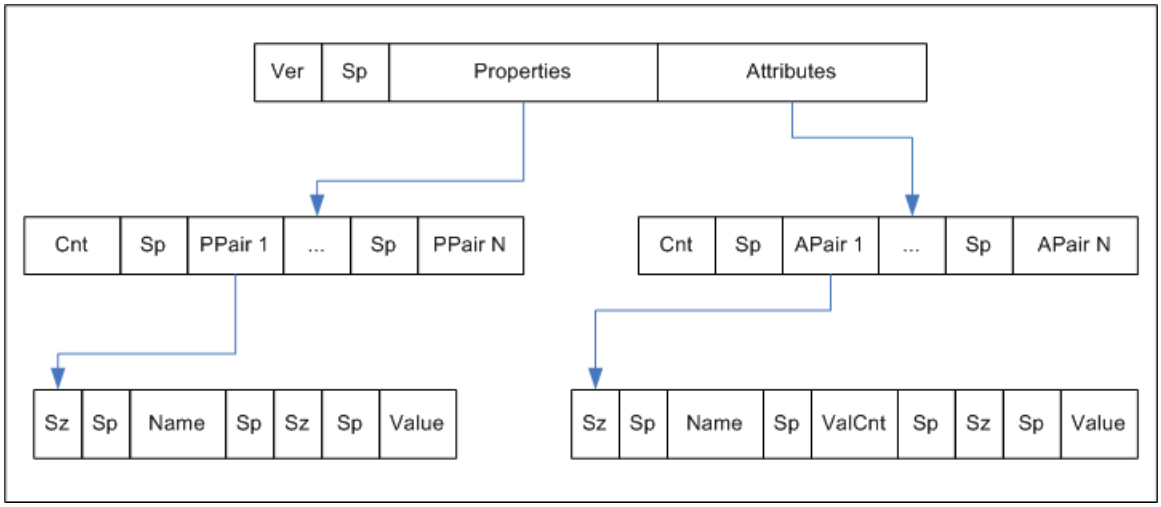
- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

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 Standalone 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 Standalone 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 Cookie により、アプリケーションはユーザ属性を CA SiteMinder® Federation Standalone にアサートし、CA SiteMinder® Federation Standalone によりカプセル化されたユーザ属性を消費することができます。 オープン形式 Cookie には以下の一般的特性があります。

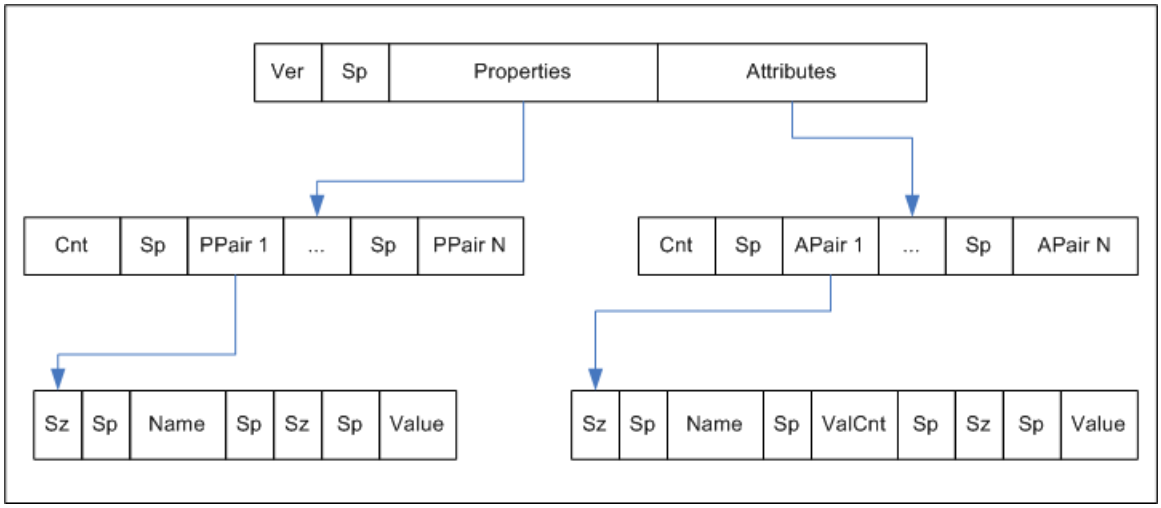
- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

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）
- UserConsent
- ログイン ID
- ExpiresON （有効期限）

以下の図はオープン形式を表しています。



キー：

- Ver -- Cookie 形式バージョン。 値は 1 です。
- Sp -- ASCII スペース文字。読みやすくするためにのみ使用されます。
- 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 -- 次に続く名前値ペアの数。ASCII で表されます。
- Sz -- 次に続く名前または値の長さ
- ValCnt -- 属性値の数

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

CA SiteMinder® Federation Standalone 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

CA SiteMinder® Federation Standalone 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 ログアウトで使用されるバック チャネル通信の場合、CA SiteMinder® Federation Standalone は FipsMode 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

バックエンド通信アルゴリズム（SPS サーバ）

バックエンド通信（SPS バックエンド サーバ）の場合、以下の暗号はセットアップの FipsMode に応じてサポートされます。 これらは、<fedroot>¥secure-proxy¥proxy-engine¥conf¥server.conf で定義されています。

FIPS_Compat モードおよび FIPS_Migration モード

ciphers="-RSA_With_Null_SHA,+RSA_With_Null_MD5,-RSA_With_RC4_SHA,+RSA_With_RC4_MD5,+RSA_With_RC2_CBC_MD5,+RSA_With_DES_CBC_SHA,+RSA_With_DES_CBC_MD5,+RSA_With_3DES_EDE_CBC_MD5,+RSA_Export_With_RC4_40_MD5,-RSA_Export_With_DES_40_CBC_SHA,+RSA_Export_With_RC2_40_CBC_MD5,-DH_RSA_With_DES_CBC_SHA,-DH_RSA_With_3DES_EDE_CBC_SHA,-DH_RSA_Export_With_DES_40_CBC_SHA,-DH_DSS_With_DES_CBC_SHA,-DH_DSS_Export_With_DES_40_CBC_SHA,-DH_Anon_With_RC4_MD5,-DH_Anon_With_DES_CBC_SHA,-DH_Anon_With_3DES_EDE_CBC_SHA,-DH_Anon_Export_With_DES_40_CBC_SHA,-DH_Anon_Export_With_RC4_40_MD5,-DHE_RSA_With_DES_CBC_SHA,-DHE_RSA_Export_With_DES_40_CBC_SHA,-DHE_DSS_With_DES_CBC_SHA,-DHE_DSS_Export_With_DES_40_CBC_SHA,-Null_With_Null_Null"

FIPS_ONLY モード

fipsciphers="+DHE_DSS_With_AES_256_CBC_SHA, +DHE_RSA_With_AES_256_CBC_SHA, +RSA_With_AES_256_CBC_SHA, +DH_DSS_With_AES_256_CBC_SHA, +DH_RSA_With_AES_256_CBC_SHA, +DHE_DSS_With_AES_128_CBC_SHA, +DHE_RSA_With_AES_128_CBC_SHA, +RSA_With_AES_128_CBC_SHA, +DH_DSS_With_AES_128_CBC_SHA, +DH_RSA_With_AES_128_CBC_SHA, +DHE_DSS_With_3DES_EDE_CBC_SHA, +DHE_RSA_With_3DES_EDE

Java SDK 暗号化アルゴリズム

CA SiteMinder® Federation Standalone 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」

フェデレーション システムの暗号化アルゴリズム

FMCrypto 暗号化/復号アルゴリズム

AES_128

内部キー暗号化アルゴリズム

CA SiteMinder® Federation Standalone は操作の FIPS モードに応じて、以下の内部キー暗号化/復号化アルゴリズムを使用します。

FIPS_MIGRATE および FIPS_ONLY モード

AES_128

FIPS_COMPAT モード

RC2

Apache Web サーバおよび Administrative UI の SSL キー アルゴリズム

CA SiteMinder® Federation Standalone は、埋め込み Apache Web サーバ SSL 通信に以下のアルゴリズムを使用します。

Apache SSL キーの生成

SHA1withRSA

キー暗号化

DES-EDE3-CBC

CA SiteMinder® Federation Standalone は、Administrative UI への SSL 通信に以下のアルゴリズムを使用します。

SSL キー パスワード暗号化

aes-128-cbc