SiteMinder

Federation Manager Java SDK ガイド

12.52 SP1

ca technologies

CA Technologies 製品リファレンス

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

- SiteMinder
- CA SiteMinder® Federation （旧 CA Federation Manager）

CA への連絡先

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

目次

第 1 章: CA SiteMinder® Federation Java SDK の概要 7

Java SDK の機能 7
Java SDK ファイル 8

第 2 章: Java SDK のインストール 9

Windows システムでの Java SDK のインストール 9
UNIX システムでの Java SDK のインストール 10

第 3 章: CA SiteMinder® Federation Java SDK プログラミング インターフェース 11

IFederationOpenIdentity インターフェース 11
オープン形式の Cookie 13
FedSdkLogger インターフェース 15

第 4 章: CA SiteMinder® Federation Java SDK の使用 17

オープン形式の Cookie を使用した、依存パーティでのプログラム フロー 18
オープン形式 Cookie を使用した分散代行認証 19
CA SiteMinder® Federation Java SDK ロギング 20
Java SDK サンプル アプリケーションの概要 20
Java SDK サンプル アプリケーションの展開 21
Java SDK サンプル アプリケーションの実行 24
Java SDK サンプル アプリケーションのカスタマイズ 24
SAML アサーションのカスタマイズ 25
Java アサーション ジェネレータ プラグイン インターフェースの実装 27
アサーション ジェネレータ プラグインの展開 29
UI 内のアサーション ジェネレータ プラグインの設定 30

マニュアルの変更点

SiteMinder r12.51.または 12.52 では、このガイドに変更はありませんでした。

第 1 章: CA SiteMinder® Federation Java SDK 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。

Java SDK の機能 (P. 7)
Java SDK ファイル (P. 8)

Java SDK の機能

CA SiteMinder® Federation Java SDK は、ユーザ ID 情報が含まれる HTTP Cookie と対話するためのライブラリです。Java SDK はオープン形式 Cookie をサポートします。 オープン 形式の Cookie は、UTF-8 バイトの文字列です。 この形式は、SiteMinder とエンド ユーザ アプリケーションの間で確実に情報を通信できるように設計された関連暗号化アルゴリズムを含んでいます。 アプリケーションは任意の共通 Web プログラミング言語で書き込むことができます。 Java SDK を使用するために、オープン形式の Cookie を作成する必要はありません。

通常、SiteMinder はエンド ユーザ アプリケーションによって消費用の HTTP Cookie へユーザ ID 情報を設定します。 エンド ユーザ アプリケーションは、Java SDK を使用して Cookie から ID 情報、認証コンテキスト、名前 ID および名前 ID 形式を抽出できます。 さらに、サードパーティ Web アクセス管理者は Cookie を作成し、SiteMinder にユーザ認証情報を提供できます。

Java SDK ファイル

CA SiteMinder® Federation Java SDK はいくつかの Java アーカイブ ファイルとして実装されます。 インストール中にそれらの場所を指定します。 最も重要なファイルである fedsdk.jar には IFederationOpenIdentity.java および他のサポートする Java クラスが含まれます。 Java アプリケーションは、このインターフェースの 1 つに対する実装オブジェクトをインスタンス化し、要件の指示通りにメソッドを呼び出す必要があります。

第 2 章: Java SDK のインストール

Windows システムでの Java SDK のインストール

以下の手順では、Windows プラットフォームでのインストールについて説明します。

重要: ターゲット システムに Java Runtime Environment (JRE) をインストールする必要があります。 サポートされているバージョンについては、テクニカル サポート サイトのプラットフォーム サポート マトリックスを参照してください。

インストール キットを見つける方法

1. テクニカル サポート サイトに移動します。
2. サイトにログオンします。
3. ［Download Center］をクリックします。

必要とするインストール キットの［Download Center］を検索し、ローカル システムにそれをダウンロードします。

Windows に CA SiteMinder® Federation Java SDK をインストールする方法

1. 実行中のすべてのアプリケーションを終了します。
2. インストール実行ファイルが置かれている場所に移動します。
3. ca-fedmgr-java-sdk-12.52-win32.exe をダブルクリックします。

   インストール ウィザードが起動されます。
4. インストール ウィザードのプロンプトに従います。
5. インストールが完了したら、システムを再起動します。

Windows での Java SDK のインストールが完了します。

UNIX システムでの Java SDK のインストール

Solaris および Linux のオペレーティング環境は、CA SiteMinder® Federation Java SDK をサポートします。

重要: ターゲット システムに Java Runtime Environment (JRE) をインストールする必要があります。 サポートされているバージョンについては、テクニカル サポート サイトのプラットフォーム サポート マトリックスを参照してください。

インストール キットを見つける方法

1. テクニカル サポート サイトに移動します。
2. サイトにログオンします。
3. ［Download Center］をクリックします。

必要とするインストール キットの［Download Center］を検索し、ローカル システムにそれをダウンロードします。

CA SiteMinder® Federation Java SDK を UNIX にインストールする方法

1. 実行中のすべてのアプリケーションを終了します。
2. インストール実行ファイルが置かれている場所に移動します。
3. プラットフォームのバイナリを実行します。

   Solaris： ca-fedmgr-java-sdk-12.52-sol.bin

   Linux： ca-fedmgr-java-sdk-12.52-linux.bin

   インストール ウィザードが起動されます。
4. インストール ウィザードの指示に従ってインストールを完了します。
5. インストールが完了したら、システムを再起動します。

Java SDK のインストールが完了します。

第 3 章: CA SiteMinder® Federation Java SDK プログラミング インターフェース

このセクションには、以下のトピックが含まれています。

IFederationOpenIdentity インターフェース (P. 11)
FedSdkLogger インターフェース (P. 15)

IFederationOpenIdentity インターフェース

IFederationOpenIdentity インターフェースにより、フェデレーション オープン形式 Cookie を操作するメソッドが定義されます。 インターフェースは以下のタスクをサポートします。

- アプリケーションに固有の SDK ロガーを初期化します。
- HTTP 要求、Java Cookie オブジェクト、または文字列形式内の Cookie からユーザ ID 情報を抽出します。
- Cookie 名、ドメインおよびセキュリティ ゾーンに対する値を初期化します。
- Cookie の暗号化および復号のためのキーを引き出すために使用される共有秘密キーを設定します。
- オープン形式の Cookie を作成します。
- ID 属性をアプリケーションへ渡します。
- AuthnContext および UserConsent 用の URI を取得し設定します。

IFederationOpenIdentity インターフェースの実装を取得するには、IdentityFactory で定義された実装メソッドの 1 つを呼び出します。 これらのメソッドでは、Cookie の暗号変換の文字列を指定する必要があります。

以下のパスワード ベースの暗号化の組み合わせが標準的なインストールに利用可能です。

- 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

パスワード ベースの暗号化（PBE）の組み合わせは FIPS に互換性がありません。 以下に記載された FIPS モードの暗号化の組み合わせについては、正しく作動するために Java SDK を使用する必要があります。

以下の暗号化の組み合わせは、FIPS に準拠しており、標準的なインストールにも利用可能です。

- AES128/CBC/PKCS5Padding
- AES192/CBC/PKCS5Padding
- AES256/CBC/PKCS5Padding
- 3DESEDE/CBC/PKCS5Padding

注: 暗号文字列およびそれらの対応する一定の名前はすべて IdentityCrypto.java にリスト表示されます。

オープン形式の Cookie

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 SiteMinder にアサートし、SiteMinder 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。

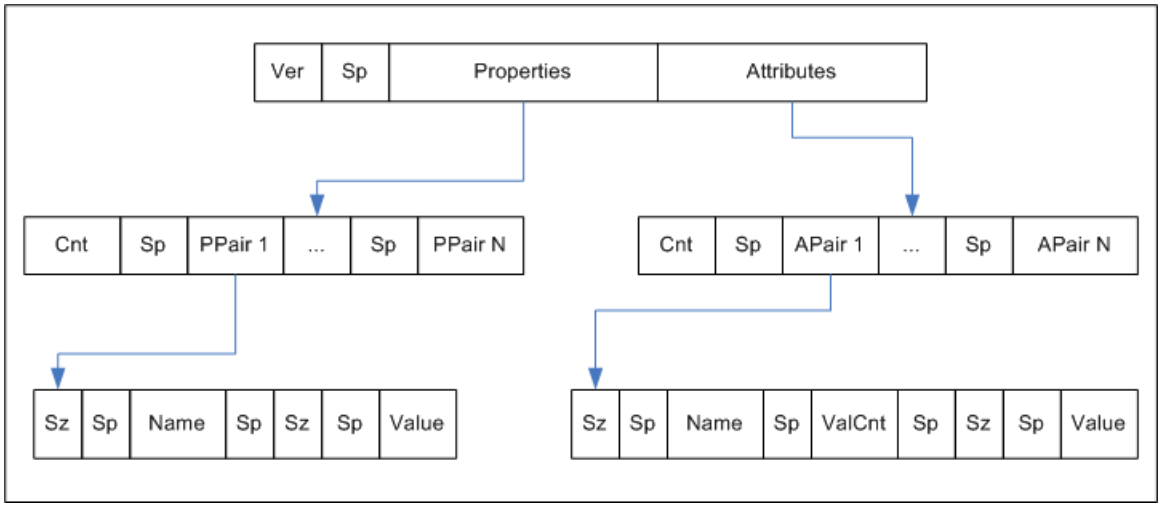
- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

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）

以下の図はオープン形式を表しています。



キー：

- Ver -- Cookie フォーマット バージョン。CA SiteMinder® Federation r12.1 の場合、この値は 1 です。
- Sp -- 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 -- 次に続く属性値の数。CA SiteMinder® Federation r12.1 については、属性に対する複数の値はサポートされていません。 この値は 1 に設定する必要があります。

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

FedSdkLogger インターフェース

FedSdkLogger インターフェースにより、カスタム ロギング メッセージを指定するための以下のメソッドが提供されます。

void logTrace (string fileName, string methodName, string msg)

トレース メッセージをログ記録します。

void logError (string fileName, string methodName, string msg)

エラー メッセージをログ記録します。

第 4 章: CA SiteMinder® Federation Java SDK の使用

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 を使用した、依存パーティでのプログラム フロー (P. 18)
オープン形式 Cookie を使用した分散代行認証 (P. 19)
CA SiteMinder® Federation Java SDK ロギング (P. 20)
Java SDK サンプル アプリケーションの概要 (P. 20)
Java SDK サンプル アプリケーションの展開 (P. 21)
Java SDK サンプル アプリケーションの実行 (P. 24)
Java SDK サンプル アプリケーションのカスタマイズ (P. 24)
SAML アサーションのカスタマイズ (P. 25)

オープン形式の Cookie を使用した、依存パーティでのプログラム フロー

依存パーティでの Java SDK プログラム フローの簡単な説明を以下に示します。

1. Java アプリケーションは、IdentityFactory インターフェースを使用して、IFederationOpenIdentity インターフェースの実装クラスを作成します。
2. Java アプリケーションは、extractCookie() メソッドを呼び出して HttpServletRequest オブジェクトから Cookie を抽出します。 このメソッドはまた、Cookie を復号し、ストレージ マップに ID 属性を入れます。
3. あるいは、Java アプリケーションは、processCookie() メソッドを呼び出して Cookie オブジェクトから属性をすべて抽出し、ストレージ マップにそれらを設定することもできます。
4. Java アプリケーションは、getAttributes()、getAttribute()、getAuthnContext()、getSessionID()、getNameID()、getNameIDFormat() および getUserConsent() メソッドを使用して、ストレージ マップに入れられたすべての属性に対する値を取得できます。
5. Java アプリケーションは、setAuthnContext() および setUserConsent() メソッドを使用して、Cookie に属性の値を設定できます。
6. Java アプリケーションは、スキュー時間を指定するしないにかかわらず、isExpired() メソッドを呼び出すことで Cookie が有効期限切れかどうかを判断できます。 このメソッドは、オプションのスキュー時間に追加して、Cookie 上の有効期限スタンプを現在の GMT 時間と比較します。 GMT 時間のほうが大きい場合、Cookie は期限切れです。 Cookie の有効期限スタンプは、Cookie が作成されるときに setTimeToLive () メソッドを使用して指定されます。

これらのメソッドの詳細情報については Javadoc の参考資料を参照してください。

オープン形式 Cookie を使用した分散代行認証

分散代行認証により、サードパーティ アクセス管理システムはユーザを認証し、アサーティング パーティで展開した SiteMinder とユーザ認証情報を共有することができます。 これらの認証情報は Cookie によって、またはクエリ文字列で共有されます。

注: このガイドでは、Cookie および Java SDK を使用した、分散代行認証について説明します。 クエリ文字列を使用した分散代行認証の詳細については、「*CA SiteMinder® Federation パートナーシップ フェデレーション ガイド*」を参照してください。

サードパーティ アクセス管理者およびアサーティング パーティが認証されたユーザ ID を通信するために Cookie を使用する予定である場合は、アクセス制御アプリケーションはこれらの手順に従うことができます。

1. CA SiteMinder® Federation Java SDK を実装します。
2. IFederationOpenIdentity インターフェースの実装クラスを構成します。
3. createCookie メソッドを呼び出します。

実装クラスを構成するには、アクセス制御管理者は CA SiteMinder® Federation で設定された Cookie ゾーンおよびパスワードを知っている必要があります。 これらの値は帯域外で通信されます。 サードパーティ アクセス管理システムは、アサーティング パーティと同じ Cookie ドメインにある必要があります。

分散代行認証用の Cookie を作成するときに使用する、IdentityFactory.java クラスからのコンストラクタが以下にリスト表示されています。

```
/**
 * Gets an implementation of the IFederationOpenIdentity interface.
 *
 * @param cryptoInstance A cryptographic string; supported values are
 *  listed in IdentityCrypto.java.
 * @param bUseHmac A Boolean value that indicates whether to use HMAC.
 */
  public static IFederationOpenIdentity getInstance(cryptoInstance, bUseHmac)
```

アクセス制御管理者はパスワード ベースの暗号化を使用して、Cookie 自体を暗号化できます。または、その Cookie は FIPS 準拠の暗号文字列の 1 つを使用できます。 FIPS 準拠の文字列を選んだ場合は、Java SDK によって提供される暗号化を使用します。

ここに、Cookie 作成のコード スニペットの例を示します。

```
IFederationOpernIdentity openID =
IdentityFactory.getInstance(IdentityCrypto.AES128, false);

String domain = ".moon.com";
String zone = "FED";
String name = "CryptoID"
String password = "";

openID.initCookieInfo(domain, zone, name, password);

openID.setLoginID = "TomJones";

openID.createCookie(HttpResponse);
```

createCookie メソッドは、ログイン ID を使用して、暗号化され HttpSevletResponse オブジェクトに追加される Cookie 値を作成します。 要求がリダイレクトされた後、サーブレット コンテナは自動的に Cookie を渡します。

CA SiteMinder® Federation Java SDK ロギング

デフォルト Java SDK ロガーは、標準出力ストリームへのメッセージを書き込みます。 ロギングは、デフォルトでは無効になっています。

CA SiteMinder® Federation Java SDK ロギングを有効にする方法

1. *sdkroot*¥ サンプル フォルダから sdkloggingconfig.properties ファイルをコピーし、任意のフォルダにそれを置きます。 フォルダが CLASSPATH にあることを確認します。
2. sdkloggingconfig.properties ファイルで sdk.logging.enable パラメータの値を Y に設定します。

ロギングが有効になります。

Java SDK サンプル アプリケーションの概要

Java SDK サンプル アプリケーションは依存パーティ Java アプリケーションをシミュレートします。 アプリケーションは、フェデレーション パートナーシップの依存パーティで実行する CA SiteMinder® Federation 展開によって送信された Cookie を消費します。

サンプル アプリケーションは、Java アプリケーションが受信要求から Cookie を取得し、依存パーティに送信されるユーザ ID 情報およびアサーション属性を抽出する方法について実証します。 このサンプル アプリケーションでは、CA SiteMinder® Federation が依存パーティでインストールされ、ユーザをサンプル アプリケーション サーブレットの URL にリダイレクトするよう設定される必要があります。

Java SDK サンプル アプリケーションの展開

Java SDK サンプル アプリケーションの展開には、依存パーティで Tomcat および CA SiteMinder® Federation をインストールする必要があります。

Java SDK サンプル アプリケーションを展開する方法

1. 推奨される任意の場所に Java SDK パッケージをインストールします。
2. 環境変数 FEDSDKROOT を Java SDK のインストール ディレクトリに設定します。

   注: FEDSDKROOT の値は SDK ディレクトリの場所を指します。 例: C:¥Program Files¥CA¥Federation Manager¥sdk

   この環境変数は、Windows 上で自動的に設定されますが、UNIX プラットフォーム上では手動でエクスポートする必要があります。
3. Tomcat 5.0 をインストールし、TOMCAT_HOME 環境変数を Tomcat ルート フォルダを指すよう設定します。

   注: Tomcat は、CA SiteMinder® Federation がインストールされているシステムから別のシステムにインストールされる必要があります。
4. FEDSDKROOT¥sample¥javasdk/war を TOMCAT_HOME¥webapps¥ フォルダにコピーすることにより、Web サーバにそれを展開します。
5. Tomcat サーバを起動します。
6. Tomcat がアップされ稼働中かどうか判断するためにリンク「http://<FQDN of Tomcat Host>:<port num>/」にアクセスしてみます。

7. オープン形式の Cookie を使用する場合は、fedsample.properties を以下のように TOMCAT_HOME¥webapps¥javasdk¥WEB-INF¥classes フォルダでアップデートします。

   - RedirectMode は値リダイレクト モードです。 オープン形式 Cookie に対しては OPEN を使用します。
   - CookieDomain は、CA SiteMinder® Federation の［パートナーシップの作成］ダイアログ ボックスで設定されるような Cookie ドメインの値です。
   - CookieName は、CA SiteMinder® Federation の［パートナーシップの作成］ダイアログ ボックスで設定されるような Cookie 名の値です。
   - CryptoInstance は暗号化変換の値です。それは CA SiteMinder® Federation の［パートナーシップの作成］ダイアログ ボックスで設定されます。
   - UseHmac は、CA SiteMinder® Federation の［パートナーシップの作成］ダイアログ ボックスで設定されるような［HMAC の有効化］チェック ボックスの値を指定します。チェック ボックスをオフにする場合は値 no を、チェック ボックスをオンにする場合は値 yes を使用します。
   - ShowAttributeMap は、アサーション マップ内のデータが表示されるかどうかを指定します。 アサーション マップ内のデータを表示しない場合は値を no にします。アサーション マップ内のデータすべてを表示する場合は値を yes にします。この値は、CA SiteMinder® Federation の［パートナーシップの作成］ダイアログ ボックスで設定されます。 値を no に設定した場合、SpSideAttributeKey パラメータに記載された属性のリストのみが表示されます。
   - SpSideAttributeKey は、その属性または表示される要求からの属性（カンマ区切り）を指定します。
   - CharsetEncoding は、画面上に表示されるレスポンスの charset エンコーディングを指定します。

8. ライト ウェイト プロビジョニングをテストしている場合は、以下のパラメータを更新する必要があります。
   - EnableProvisioningTest は、ライト ウェイト プロビジョニングが有効かどうかを指定します。 プロビジョニングを有効にしない場合は、値 no を使用します。 ライト ウェイト プロビジョニングのテストを有効にする場合は、値 yes を使用します。
   - AssertionConsumerUrl は、提供するアサーション コンシューマ URL を指定します。
   - UDType は、ユーザ ディレクトリ タイプに応じて、odbc または ldap を指定します。 タイプと関連付けられた接続パラメータを指定する必要があります。
9. ロギングを有効にするには、sdkloggingconfig.properties ファイルを更新します。 ロギングは、デフォルトでは無効になっています。
10. フェデレーション パートナーシップの依存パーティで CA SiteMinder® Federation をインストールし、アサーティング パーティ - 依存パーティのパートナーシップを定義します。
    a. 適切なリダイレクト モードを選択します。
    b. パートナーシップのターゲット URL を指定します。以下のいずれかを入力します。
       - SDK サンプル App の URL （たとえば http://<FQDN of target machine>:<Tomcat port>/javasdk/SpSideAttributeServlet）
       - 依存パーティの URL、http://<FQDN of SP>:CA Portal および proxyrules.xml 内（場所： %FEDROOT%¥ proxy-engine¥conf¥proxyrules.xml make the entry http://<FQDN of target machine>:<Tomcat port>/javasdk/SpSideAttributeServlet）

これでサンプル アプリケーションが展開され、実行する準備ができました。

Java SDK サンプル アプリケーションの実行

Tomcat および CA SiteMinder® Federation をインストールした後、Java SDK サンプル アプリケーションを実行できます。

Java SDK サンプル アプリケーションを実行する方法

1. サンプル アプリケーションが展開する Tomcat サーバを開始します。
   - Windows では、サービス コントロール パネルを使用します。
   - UNIX では、Tomcat の起動スクリプトを使用します。
2. 設定された連係トランザクションを実行して、SDK サンプル アプリケーションにリダイレクトします。

   サンプル アプリケーションは Cookie をデコードし、ユーザ アイデンティティ情報を表示します。

Java SDK サンプル アプリケーションのカスタマイズ

サンプル アプリケーションは、fedsdksample.jar を再生成するために build.bat または build.sh スクリプトを使用して変更できます。

サンプル Java アプリケーションをカスタマイズする方法

1. SpSideServlet または SpSideAttributeServlet.java を必要に応じて変更します。
2. JDK がインストールされ、JAVA_HOME が JDK インストールに対して適切に設定されていることを確認します。
3. fedsdksample.jar ファイルを構築するために build.bat （Windows）または build.sh （UNIX）を実行します。

サンプル アプリケーションのカスタマイズされたバージョンを実行する準備ができました。

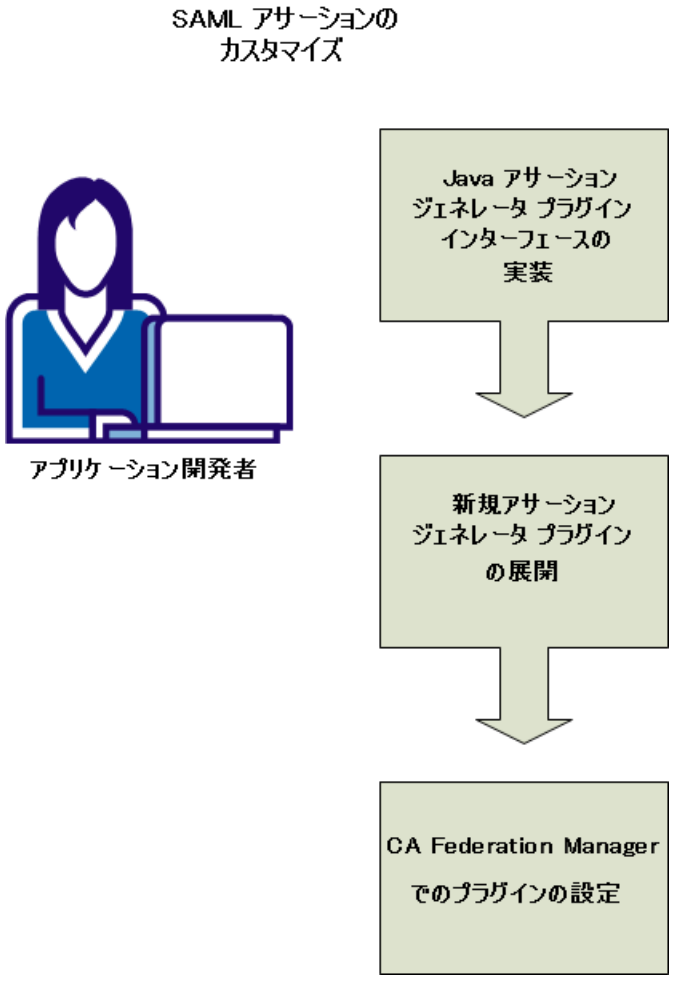
SAML アサーションのカスタマイズ

セキュリティ ドメインは、アサーションという名前のデータ パッケージを使用して認証と認可を交換します。 SAML（Security Assertion Markup Language）は、アサーションの形式を指定するオープン スタンダードです。フェデレーション パートナーシップは、アイデンティティ プロバイダ（アサーションのプロデューサ）およびサービス プロバイダ（アサーションのコンシューマ）から構成されます。

企業は、連係されたパートナー間のビジネス契約に基づいてアサーションのコンテンツを変更できます。 たとえば、あるパートナーは、アサーションで属性に対する使いやすい名前相当物を要求することができます。 または、あるパートナーは、アサーションに各属性の XML タイプ指定を含めるよう選ぶことができます。

CA SiteMinder® Federation は AssertionGeneratorPlugin.java インターフェースの実装で SAML アサーションを作成します。アプリケーション開発者は、既存の実装クラスを上書きすることで SAML アサーションのコンテンツを増強できます。

以下の図は、カスタム アサーション ジェネレータ プラグインを作成する過程を示します。



SAML アサーションをカスタマイズする過程にはこれらの手順が含まれます。

1. Java アサーション ジェネレータ プラグイン インターフェースを実装します (P. 27)。
2. 新しいアサーション ジェネレータ プラグインを展開します。
3. CA SiteMinder® Federation UI 内でアサーション ジェネレータ プラグインを設定します。

Java アサーション ジェネレータ プラグイン インターフェースの実装

AssertionGeneratorPlugin.java インターフェースを実装することでカスタム アサーション ジェネレータ プラグインを作成します。 実装クラスに関する最小要件を以下に記載します。

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であることを確実にするのに役立つコードを提供します。その結果、多くのスレッドが単一のプラグイン クラスを使用できます。
3. customizeAssertion メソッドへのコールを含めます。

   例

   この例では、アプリケーション開発者が使いやすい名前属性を作成するために handler.updateNameID を定義していると仮定します。

```
/**
     * <p>Performs Assertion Generator callout functionality to customize the
     * SAML assertion in the <code>AssertionGeneratorPlugin</code> object and
     * returns a result.</p>
     * @param apiContext     A context object that provides methods for sending
     *                     log, trace, and error messages to the Policy Server.
APIContext.getAttrMap() メソッドを使用して、アプリケーション URL で指定されたアプリケーションによってポストされる属性を取得します。
     * @param userContext    A context object that allows a custom object to set
     *                       and retrieve information about a user in a user
     *                       directory. The information includes user
     *                       attributes and directory attributes associated
     *                       with the user.
     * @param pluginParam    The string for Assertion plug-in parameters.
     * @param inputAssertion    The current XML token representing the SAML
Assertion.
     * @param outputAssertion   The final XML token representing the SAML
Assertion.
     * @return  0 if assertion is customized successfully, or -1 if no
customization or an error occurred.
     * If the method fails, the outputAssertion is ignored.
     * @throws java.lang.Exception For cases when the customization terminates
unexpectedly.
     *
     **/
```

```
    public int customizeAssertion(APIContext apiContext, UserContext
userContext,String pluginParam,
     String inputAssertion, final StringBuffer outputAssertion) throws Exception
{
        if (inputAssertion == null || inputAssertion.equals("")) {
            // Indicates non-zero for an error.
            apiContext.trace(PLUGIN_TAG, "Received null or empty response for
customization");
            return -1;
        }
        apiContext.trace(PLUGIN_TAG, "Entering customizeAssertion");
        StringBuffer newAssertion = new StringBuffer(inputAssertion);

        try
        {
            Saml1AssertionHandler handler =
                initHandler(apiContext, userContext);
            handler.updateNameID(newAssertion);
            handler.addAttributes(pluginParam, newAssertion);
        }
        catch(Throwable th)
        {
            apiContext.error("SAML1AssertionSample: " + th.getMessage());
            StringWriter writer = new StringWriter();
            th.printStackTrace(new PrintWriter(writer));
            writer.flush();
            apiContext.trace(PLUGIN_TAG,
                             "Error customizing Assertion:¥n" +
writer.toString());

            apiContext.trace(PLUGIN_TAG, "Done customizeAssertion");
            return -1;
        }

        outputAssertion.append(newAssertion);

        apiContext.trace(PLUGIN_TAG, "Done customizeAssertion");

        // return "success"
        return 0;
    }
```

注: 構文要件および customizeAssertion メソッドへ渡されるパラメータ文字列の使用は、カスタム オブジェクトの責任です。

アサーション ジェネレータ プラグインの展開

AssertionGeneratoPlugin.java インターフェース用の実装クラスをコード化した後、それをコンパイルし、SiteMinder 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 ジェネレータ 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、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_mgr_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_mgr_sdk_home*¥jar にあります。
   - 自分のプラグインを書き込む場合は、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_mgr_home*¥siteminder¥config に JVMOptions.txt ファイルを置きます。

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

UI 内のアサーション ジェネレータ プラグインの設定

アサーション ジェネレータ プラグインを設定するには、Administrative UI 内に設定に対する値を提供します。

注: プラグインを展開するまで、プラグイン設定を行わないでください。

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. 次のフィールドに値を入力します。

   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

   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、最終処理のため SiteMinder 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。コンパイルされたサンプル プラグインは、ディレクトリ *federation_mgr_sdk_home*/jar に含まれています。

   プラグイン パラメータ

   （オプション）。 SiteMinder がランタイムで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命令する整数を含めることができます。

アサーション ジェネレータ プラグインは、コード化およびコンパイルされ、配置済みです。 SiteMinder アサーション ジェネレータはフェデレーション パートナーによって定義されるとおり強化されたアサーションを作成します。