CA SiteMinder Federation Standalone

Java SDK ガイド

r12.52 SP1

ca technologies

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

CA SiteMinder® Federation Standalone の以前のリリースでの問題の結果として 12.52 のドキュメントの更新は行われていません。

目次

第 1 章: CA SiteMinder® Federation Standalone 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

CA SiteMinder® Federation Standalone 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 2 つの Cookie 形式をサポートします。

- オープン形式の Cookie
- レガシー（以前は FEDProfile）Cookie

オープン 形式の Cookie は、UTF-8 バイトの文字列です。 このフォーマットは、CA SiteMinder® Federation Standalone 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。

Java SDK の現在のバージョンでは、SDK の以前のバージョンを使用するアプリケーションのレガシー Cookie をサポートします。

通常、CA SiteMinder® Federation Standalone 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、ユーザ許可、名前 ID および名前 ID 形式を抽出できます。 アプリケーションはユーザ許可および認証コンテキストに対して URI を設定できます。 さらに、サードパーティ Web アクセス管理者は Cookie を作成し、CA SiteMinder® Federation Standalone にユーザ認証情報を提供できます。

Java SDK ファイル

CA SiteMinder® Federation Standalone 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、2 つの Java インターフェース（IFederationOpenIdentity.java およびレガシー FederationIdentity.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、これら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。

smapi.jar アーカイブには、アサーション ジェネレータ プラグインをカスタマイズすることをサポートするクラスが含まれます。 Javadoc は、これらのクラス内のメソッド、および SDK 内の他のすべてのクラスに関する情報を含みます。

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 Standalone 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-r12.52 SP1-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 Standalone 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 Standalone 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-r12.52 SP1-sol.bin

   Linux： ca-fedmgr-java-sdk-r12.52 SP1-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 Standalone 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。

FederationIdentity インターフェース

FederationIdentity インターフェースにより、フェデレーション レガシー 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。
- レガシー（FEDPROFILE）Cookie を作成します。
- ID 属性をアプリケーションへ渡します。
- Cookie の暗号化および復号化用のパスワードを設定します。

重要: FederationIdentity インターフェースはパスワード ベースの暗号化のみをサポートします。これは FIPS 準拠ではありません。 FIPS のみのインストールを使用している場合は、IFederationOpenIdentity インターフェースを実装します。

Cookie-Related Parameters

CA SiteMinder® Federation Standalone はこれらの Cookie 関連のパラメータを以下のデフォルト値に設定します。

- Cookie ドメインは "" に設定されます。
- Cookie 最長有効期間には最大限度がありません。
- Cookie パスは「/」に設定されます。
- Cookie パスワードは "" に設定されます。
- Cookie SSO セキュリティ ゾーン名は "FED" に設定されます。

CA SiteMinder® Federation Standalone 管理 UI を使用する SSO セキュリティ ゾーン名およびパスワードを変更できます。 これらのパラメータを再設定する場合、帯域外の通信で CA SiteMinder® Federation Standalone Java SDK を使用して、値を任意のパートナーに知らせる必要があります。 そうしない場合、Cookie は復号化できません。

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 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 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。

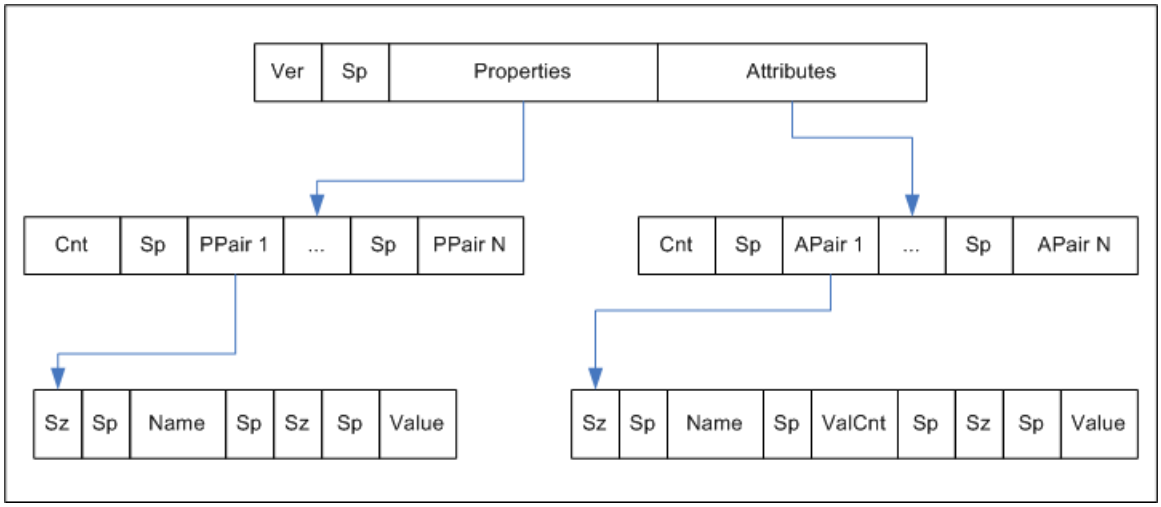
- Cookie は、任意のプログラミング言語で書かれたアプリケーションによってアクセス可能です。
- Cookie コンテンツは、UTF-8 バイトの文字列から構成され、それは国際文字セットをサポートします。
- UTF-8 バイトの各名前/値ペアの合わせたサイズは、名前/値ペアに先行します。
- スペース文字は読みやすいように追加されます。
- Cookie は簡単に解析でき、容易に拡張可能です。

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

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 Standalone 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

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。

レガシー 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、FederationIdentity インターフェース用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()、および getNameIDFormat() 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。

オープン形式 Cookie を使用した委任認証

委任認証により、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 CA SiteMinder® Federation Standalone 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、委任認証について説明します。クエリ文字列を使用した委任認証の詳細については、「CA SiteMinder® Federation Standalone 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 Standalone 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 Standalone 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。

委任認証用の 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。

レガシー Cookie を使用した委任認証

委任認証により、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 CA SiteMinder® Federation Standalone と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。 Cookie は、CA SiteMinder® Federation Standalone が復号化できるように CA SiteMinder® Federation Standalone Java SDK を使用して生成されます。

注: このドキュメントでは、Cookie および Java SDK を使用した委任認証について説明します。 クエリ文字列を使用した委任認証の詳細については、「Federation Standalone 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必要があります。

1. Java SDK を実装します。
2. FederationIdentity インターフェースの実装クラスを構築します。
3. createProfileCookie メソッドを呼び出します。

実装クラスを構築するには、アクセス制御管理者は帯域外通信を通して 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。 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。

委任された認証用の Cookie を作成する場合に使用するコンストラクタを以下に示します。

```
/**
 * This constructor loads customized parameters for the cookie.
 *
 * @param zoneName Cookie zone name (the default is FED)
 * @param password String used for cookie encryption
 * @param domain string used to indicate the cookie domain
 * @param obj the object of FedSdkLogger class
 */
public FederationIdentityImpl(String zoneName, String password, String domain,
                              FedSdkLogger obj) throws JavaSDKException
```

注: 最後のパラメータは FedSdkLogger オブジェクトです。サードパーティ アクセス管理システムが独自のロガーを実装する場合、参照はここで渡されます。そうしない場合、NULL が渡されます。また、SDK はデフォルト ログ記録実装を使用します。

createProfileCookie メソッドを呼び出すには、サードパーティ アクセス制御アプリケーションは、［アサーティング パーティ］->［依存パーティ］パートナーシップで設定されたリモート エンティティ サービス プロバイダの ID を知っている必要があります。

createProfileCookie メソッド シグネチャを以下に示します。

```
/**
 * Creates a <ZONE>PROFILE cookie and populates it with the passed in values.
 * The zone to use was configured when this object was constructed.
 * @param providerID - the provider for whom to create the cookie
 * @param loginID - the user ID
 * @param cookieVersion - the value to set the cookie version to.
 * @param response - the response object
 * @throws JavaSDKException
 */
public void createProfileCookie(String providerID,
        String loginID,
        HttpServletResponse response) throws JavaSDKException;
```

ここに、Cookie 作成のコード スニペットの例を示します。

```
String zone = request.getParameter("FED");
String domain = request.getParameter(".ca.com");
String password = request.getParameter("password");
FederationIdentity fedIdentity =
                new FederationIdentityImpl(zone, password, domain, null);
fedIdentity.createProfileCookie("ServiceProviderID", "JaneDoe",
httpServletResponse);
```

createProfileCookie メソッドは、プロバイダ ID とユーザ 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 Standalone 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 Standalone 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 Standalone 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 Standalone 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 Standalone をインストールする必要があります。

次の手順に従ってください:

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 Standalone¥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 Standalone 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. レガシー（FEDPROFILE）Cookie を使用する場合は、TOMCAT_HOME¥webapps¥javasdk¥WEB-INF¥classes フォルダ内の fedsample.properties を次のように更新します。

   - RedirectMode はリダイレクト モードの値です。レガシー Cookie には LEGACY を使用します。

     デフォルト値： OPEN

   - CookieZone は、CA SiteMinder® Federation Standalone Administrative UI の［展開設定］ダイアログ ボックスで設定される CookieZone の値です。

     デフォルト値： FED

   - EncryptionPassword は、CA SiteMinder® Federation Standalone Administrative UI の［展開設定］ダイアログ ボックスで設定されるパスワードの値です。

     デフォルト値： 空白

   - ProviderId は、［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プロバイダ識別子の値です。

     デフォルト値： 空白

   - 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。

     デフォルト値： UTF-8

   - 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、［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。

     デフォルト値： no

   - 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。

     デフォルト値： 空白

8. オープン形式の Cookie を使用する場合は、TOMCAT_HOME¥webapps¥javasdk¥WEB-INF¥classes フォルダの fedsample.properties を以下のように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、［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 Cookie ドメインの値です。
   - CookieName は、［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 Cookie 名の値です。
   - CryptoInstance は、［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暗号化変換の値です。
   - UseHmac は、［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［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、［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。

9. 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。
10. 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。
11. 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 Standalone 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 Standalone 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。

第 5 章: アサーション ジェネレータ プラグインを使用したアサーションのカスタマイズ

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 ジェネレータ プラグインの概要

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。

アサーション ジェネレータ プラグインを展開する方法

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

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

     Windows: build_plugin.bat

     UNIX: build_plugin.sh

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

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

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

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

   注: プラグインで Apache Xerces または Xalan を使用するには、CA SiteMinder® Federation Standalone 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 Standalone 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、CA SiteMinder® Federation Standalone UI 内で設定することにより、このプラグインを有効にします。 UI パラメータにより、CA SiteMinder® Federation 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 (P. 32)するまで、プラグイン設定を実行しないでください。

アサーション ジェネレータ プラグインを有効にする方法

1. 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 6 章: メッセージ コンシューマ プラグイン

SiteMinder SAML（1.x および 2.0）認証方式は、レスポンス メッセージを処理します。 ビジネス上の理由で、たとえばレスポンスをさらに処理するために追加の手順が必要となる場合があります。 メッセージ コンシューマ拡張 API は、認証処理中に SAML レスポンスを 2 つの方法を使用して詳細に作成することを可能にするインターフェースを定義します。

- ユーザの特定中に失敗理由の詳細をレポートする方法
- ユーザ認証情報の検証をカスタマイズする方法

Java MessageConsumerPlugin API は、メッセージ コンシューマ拡張（MCE）インターフェースを実装します。 自分の要件に基づいてコードを書き、カスタム プラグインを CA SiteMinder® Federation Standalone に統合することができます

MessageConsumerPlugin クラスには、以下の 4 つのメソッドが含まれています。

メソッド	説明
init()	プラグインが必要とするあらゆ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、CA SiteMinder® Federation Standalone は各プラグイン インスタンスに対してこのメソッドを 1 回呼び出します。
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。 CA SiteMinder® Federation Standalone がシャットダウンしているとき、CA SiteMinder® Federation Standalone は各プラグイン インスタンスに対してこのメソッドを 1 回呼び出します。
postDisambiguateUser()	認証方式によって実行できない場合に、ユーザの特定を実行する処理を提供します。または、新規フェデレーション ユーザに対するデータをユーザ ストアに追加するための処理を提供します。 このメソッドが復号されたアサーションを受け取ることに注目してください。 復号されたアサーションは、キー「_DecryptedAssertion」の下の MCP に渡されるプロパティ マップに追加されます。

メソッド	説明
postAuthenticateUser()	ポリシー サーバ処理結果が成功または失敗に関係なく、アサーション処理の最終結果を決定する任意の追加のコードを提供します。

CA SiteMinder® Federation Standalone から、メッセージ コンシューマ プラグイン クラスの以下のサンプルが提供されます。

- *fed_sdk_home*¥sample¥com¥ca¥federation¥sdk¥plugin¥sample¥MessageConsumerPluginSample
- *fed_sdk_home*¥sample¥com¥ca¥federation¥sdk¥plugin¥sample¥MessageConsumerSAML2