

SiteMinder フェデレーション

レガシー フェデレーション ガイド

12.52 SP1



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複写、譲渡、変更、開示、修正、複製することはできません。

本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし、CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本書の制作者は CA および CA Inc. です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

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

CA Technologies 製品リファレンス

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

- SiteMinder
- CA SiteMinder® Web エージェント オプション パック
- CA SiteMinder for Secure Proxy Server

CA への連絡先

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

マニュアルの変更点

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

- ガイド全体で **SAML** アフィリエイトエージェントへの参照をすべて削除しました。この製品はサポートされなくなりました。
- [smfedexport のコマンド オプション](#) (P. 427) - smfedexport ユーティリティに **-decryptionkeyalias** コマンド オプションを追加しました。

目次

第 1 章: レガシー フェデレーションの概要 17

フェデレーションにおけるパートナーに関する用語	17
レガシー フェデレーションのコンポーネント	18
レガシー フェデレーション認証方式	20
セキュリティゾーンを使用したフェデレーション シングル サインオン	20
Secure プロキシ サーバ フェデレーション ゲートウェイ	23
レガシー フェデレーションにおける国際化	23
デバッグ機能	24
レガシー フェデレーション用 API	25
ポリシー管理 API	25
Java メッセージ コンシューマ プラグイン API	26
Java アサーション ジェネレータ プラグイン API	26
レガシー フェデレーション設定フローチャート	27

第 2 章: サンプル設定の使用によるレガシー フェデレーションについての理解 29

手動の SiteMinder 対 SiteMinder 展開の概要	29
必須コンポーネントのインストール確認	31
サンプル フェデレーション ネットワーク	31
基本的な設定におけるアイデンティティ プロバイダ データ	32
詳細設定におけるアイデンティティ プロバイダ データ	34
基本的な設定におけるサービス プロバイダ データ	35
高度な設定におけるサービス プロバイダ データ	37
サンプル ネットワーク用のアイデンティティ プロバイダのセットアップ	37
IdP ユーザ ストアのセットアップ	38
ポリシー サーバが IdP LDAP ポリシー ストアを指すための設定	38
IdP でのポリシー サーバ トレース ロギングの有効化	40
Web エージェント オプション パックによる Web サーバの設定	40
IdP での Web エージェント オプション パック ロギングの有効化	46
IdP ポリシー サーバのユーザ ストアの指定	46
IdP でのアフィリエイト ドメインのセットアップ	47
IdP のアフィリエイト ドメインへのユーザ ディレクトリの追加	48
IdP のアフィリエイト ドメインへのサービス プロバイダの追加	48
IdP がアサーションを生成する対象のユーザの選択	51

アサーションの名前 ID の設定.....	52
IdP での POST シングル サインオンの設定.....	53
基本サンプル展開での署名処理の無効化.....	54
サービス プロバイダ オブジェクト設定の完了.....	54
サービス プロバイダの設定.....	54
サンプル ネットワーク用のサービス プロバイダのセットアップ.....	54
SP ユーザ ストアのセットアップ.....	55
ポリシー サーバが SP LDAP ポリシー ストアを指すための設定.....	56
SP でのフェデレーション コンポーネントのトレース ロギングの有効化.....	57
Web エージェント オプション パックによる Web サーバの設定.....	57
SP での Web エージェント オプション パック ロギングの有効化.....	61
SP ポリシー サーバのユーザ ストアの指定.....	61
SP での SAML 2.0 認証方式の設定.....	62
SP でのターゲット リソースの保護.....	65
SAML 2.0 シングル サインオンのテスト.....	67
フェデレーション展開への機能の追加.....	68
シングル ログアウトの設定.....	69
SAML 2.0 Artifact シングル サインオンの設定.....	71
アサーションへの属性の追加.....	79
デジタル署名と検証の設定.....	80
アサーションの暗号化と復号.....	83

第 3 章: SiteMinder Federation のセットアップの概要 87

フェデレーションセットアップの概要.....	87
インストール概要手順の表記規則.....	88
アサーティング パーティ コンポーネントのセットアップ.....	90
アサーティング パーティ ポリシー サーバのインストール.....	91
アフィリエイト ドメインのセットアップおよびこれらのドメインへのサイトの追加.....	91
アサーティング パーティでの Web エージェントまたは SPS フェデレーション ゲートウェイ のインストール.....	93
Web エージェント オプション パック (アサーティング パーティ) 用のアプリケーション サー バのインストール.....	93
アサーティング パーティ Web エージェント オプション パックのインストール.....	94
フェデレーション Web サービス (アサーティング パーティ) の設定.....	95
フェデレーション Web サービスへのアクセスの許可 (アサーティング パーティ).....	96
SAML POST レスポンスの署名の有効化.....	97
ターゲット リソースへのリンクの作成 (オプション).....	97
依存パーティコンポーネントのセットアップ.....	101
依存パーティ ポリシー サーバのインストール.....	102

SAML 認証方式または WS フェデレーション認証方式の設定	103
依存パーティでのターゲット リソースの保護	104
Web エージェントまたは SPS フェデレーション ゲートウェイのインストール (依存パーティ)	104
Web エージェント オプション パック (依存パーティ) 用の Web サーバまたはアプリケーション サーバのインストール	105
依存パーティでの Web エージェント オプション パックのインストール	106
依存パーティでのフェデレーション Web サービスの設定	107
フェデレーション Web サービスへのアクセスの許可 (アサーティング パーティ)	108
Artifact シングル サインオンの場合の証明書データ ストアの変更 (オプション)	109
シングル サインオンを開始するためのリンクの作成 (オプション)	109
第 4 章: SAML 1.x アサーション ジェネレータ ファイルの設定	113
第 5 章: JVM 用の JVMOptions ファイルの確認	115
第 6 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	117
セッションストアに格納されるフェデレーション データ	117
セッションストアの有効化	119
共有セッションストアを必要とする環境	120
第 7 章: フェデレーション環境の保護	123
フェデレーション通信の保護	123
アサーションの単一使用条件の設定	123
フェデレーション環境間の接続のセキュリティ保護	124
クロス サイト スクリプティングに対する保護	126
第 8 章: フェデレーションのためのキーと証明書の管理	127
第 9 章: フェデレーション用ユーザ ディレクトリの設定	129
第 10 章: アフィリエイトドメインの作成	131
アフィリエイト ドメインの概要	131
アフィリエイト ドメインの設定	132
アフィリエイト ドメインへのエンティティの追加	133

第 11 章: フェデレーション Web サービスへのアクセス権の付与 135

フェデレーション Web サービスを保護するポリシー	135
FWS ポリシーに関連付けられた機能	137
フェデレーション Web サービスを保護するポリシーの適用	138

第 12 章: SAML 1.x プロデューサの設定 139

アサーティング パートナーの前提条件 (レガシー)	139
プロデューサを設定する方法	140
アフィリエイトの識別のオプション設定タスク	141
レガシー フェデレーション ダイアログ ボックスへの移動	141
SAML 1.x アフィリエイトとアフィリエイト ドメインの関連付け	142
アフィリエイトの一般設定の入力	143
SiteMinder セッションのないユーザの認証 (SAML 1.x)	144
SAML 1.x コンシューマの時間制限の設定 (オプション)	146
SAML 1.x コンシューマの IP アドレス制限の設定 (オプション)	147
アサーションが生成される対象のユーザの選択	148
リソースへのアクセスからのユーザまたはグループの除外	149
ネストされたグループによるリソースへのアクセスの許可	149
手動設定によるユーザの追加	150
SAML 1.x アサーションの設定	151
SAML 1.x アサーション関連のセキュリティ問題	153
シングル サインオン のアサーション有効期間	153
アサーションの単一使用の設定	156
アサーションを取得するサービスへのアクセスの許可 (Artifact SSO)	156
フェデレーション エージェント グループへの Web エージェントの追加	157
アサーションを取得するための FWS ポリシーへの依存パートナーの追加	158
アサーション検索サービスのベーシック保護の検証	160
Artifact サービスを保護する認証方式の設定	160
アサーションを取得するサービスを保護するベーシック認証	161
Basic over SSL によるアサーション検索サービスの保護	162
アサーションを検索するサービスを保護ためのクライアント証明書認証	162
属性を SAML 1.x アサーションに含める設定 (オプション)	166
SAML 1.x アサーションの属性の設定	167
アサーション属性の最大文字数を指定します	169
レスポンス属性の新規作成スクリプトの使用	170
SAML アサーション レスポンスのカスタマイズ (オプション)	171
AssertionGeneratorPlugin の実装	171
アサーション ジェネレータ プラグインの展開	172

アサーション ジェネレータ プラグインの有効化	173
コンシューマ リソースへのリンクの作成 (SAML 1.x)	174
サイト間転送 URL を保護するかどうかの選択.....	176

第 13 章: SAML 1.x コンシューマとしての設定 177

依存パートナーの前提条件	177
SAML 1.x コンシューマを設定する方法	177
コンシューマに関するオプション設定タスク	178
レガシー フェデレーション ダイアログ ボックスへの移動.....	178
SAML 1.x 認証方式.....	179
SAML 1.x Artifact 認証方式の概要	181
SAML 1.x POST プロファイル認証方式の概要	183
SAML 1.x 認証方式の前提条件	184
SiteMinder ポリシー サーバをインストールします	184
プロデューサおよびコンシューマでのフェデレーション Web サービスのインストール	185
POST レスポンスの署名および検証を行うための証明書データ ストアのセットアップ	186
SAML 1.x Artifact 認証の設定	186
HTTP Artifact SSO のバックチャネル設定	188
SAML 1.x POST プロファイル認証の設定	188
メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ.....	190
MessageConsumerPlugin の実装.....	191
メッセージ コンシューマ プラグインの展開	192
SAML 1.x のメッセージ コンシューマ プラグインの有効化.....	193
SAML 1.x 認証試行が失敗した後のユーザのリダイレクト.....	194
HTTP ヘッダとしての SAML 属性の供給.....	195
HTTP ヘッダとしての SAML 属性のユース ケース.....	196
属性を HTTP ヘッダとして提供するため設定の概要.....	198
SAML 属性を保存するためのリダイレクト モードの設定	198
ユーザを検証するための認可ルールの作成	199
属性を HTTP ヘッダとして送信するレスポンスの設定	200
属性を HTTP ヘッダとして実装するためのポリシーの作成.....	201
バックチャネル対応のクライアント証明書認証の有効化 (オプション)	202
証明書データ ストアへのクライアント証明書の追加	203
バックチャネル認証のクライアント証明書オプションの選択	204
SAML 1.x 認証方式によってリソースを保護する方法	204
各認証方式に対する固有なレルムの設定	205
すべての認証方式に対する単一のターゲット レルムの設定	206

第 14 章: SAML 2.0 アイデンティティプロバイダの設定 213

アサーティング パートナーの前提条件 (レガシー)	213
アイデンティティ プロバイダを設定する方法	213
サービス プロバイダの識別に関するオプション設定タスク	214
レガシー フェデレーション ダイアログ ボックスへの移動.....	215
アフィリエイト ドメインへの SAML 2.0 サービス プロバイダの追加.....	216
サービス プロバイダ オブジェクトの一般情報の設定.....	216
SiteMinder セッションのないユーザの認証	218
サービス プロバイダ利用可能時間の制限の設定 (オプション)	220
サービス プロバイダの IP アドレス制限の設定 (オプション)	221
プロキシ サーバの識別 (オプション)	222
アサーションが生成される対象のユーザの選択	223
リソースへのアクセスからのユーザまたはグループの除外	224
ネストされたグループによるリソースへのアクセスの許可	224
手動設定によるユーザの追加.....	225
SAML 2.0 アサーションの名前 ID の指定.....	226
SAML アサーション レスポンスのカスタマイズ (オプション)	228
AssertionGeneratorPlugin の実装.....	228
アサーション ジェネレータ プラグインの展開	229
アサーション ジェネレータ プラグインの有効化	230
Web アプリケーションの属性によるアサーションのカスタマイズ.....	231
SAML 2.0 の場合のシングル サインオンの設定	233
シングル サインオンのアサーション有効期間	233
さまざまなシングル サインオン バインドに対するインデックス付きエンドポイントの定義.....	236
SSO に対する認証方式保護レベルの適用.....	241
デジタル署名オプションの決定.....	242
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	243
Allow/Create の有効化によるユーザ識別子の作成.....	246
SP からの認証コンテキストの無視	247
アサーションの単一使用の設定.....	247
IdP での HTTP エラー処理	248
アサーションでのセッション期間のカスタマイズ	249
アサーションを取得するサービスへのアクセスの許可 (Artifact SSO)	250
フェデレーション エージェント グループへの Web エージェントの追加	251
アサーションを取得するための FWS ポリシーへの依存パートナーの追加	252
Artifact サービスを保護する認証方式の設定	253
アサーションを取得するサービスを保護するベーシック認証	254
Basic over SSL によるアサーション検索サービスの保護.....	255
アサーションを検索するサービスを保護ためのクライアント証明書認証	255

バックチャネル認証に必要な WebLogic 設定	259
IdP または SP からシングルサインオンを開始します。	259
アイデンティティプロバイダで開始された SSO (POST または Artifact バインド)	260
サービスプロバイダで開始された SSO (POST バインドまたは Artifact バインド)	263
アサーションの属性の設定 (オプション)	270
SSO アサーションの属性の指定	271
アサーション属性の最大文字数を指定します	273
SSO の属性および属性クエリ リクエスト	274
シングルログアウトの設定 (オプション)	275
シングルログアウト リクエスト有効期間	277
シングルログアウト確認ページのガイドライン	278
IdP でのアイデンティティプロバイダディスカバリの設定	279
アイデンティティプロバイダディスカバリ プロファイルの有効化 (オプション)	279
IdP ディスカバリ ターゲットの攻撃からの保護	280
署名済みリクエストおよびレスポンスの検証	281
NameID とアサーションの暗号化	283
暗号化の有効化	283
IdP においてプロキシサーバで処理されるリクエスト	284
プロキシサーバを使用して処理するリクエストの設定	285

第 15 章: SAML 2.0 サービスプロバイダの設定 287

サービスプロバイダのセットアップ	287
SAML 認証リクエストのプロセス	289
依存パートナーの前提条件	291
SAML 2.0 認証方式を設定する方法	291
サービスプロバイダに関するオプション設定タスク	292
レガシー フェデレーションダイアログ ボックスへの移動	292
認証方式タイプの選択	293
SAML 2.0 認証方式に関する一般情報の指定	294
SAML 2.0 認証用のユーザ レコードの検索	294
認証方式の一部としての不明瞭解消のローカル設定	295
SAML アフィリエーションを使用したユーザ レコードの検索 (オプション)	297
SP でのシングルサインオンの設定	298
使い捨てポリシーの適用によるセキュリティ強化	299
SSO の名前識別子の作成の許可	301
HTTP-Artifact SSO に関するバックチャネルの設定	302
サービスプロバイダでの ECP の設定	303
シングルログアウトの有効化	304
シングルログアウトのバインディング	305

シングル ログアウトの設定.....	305
サービス プロバイダでのデジタル署名オプション	306
シングル サインオンのアサーション暗号化要件の適用	307
SSO の暗号化のセットアップ	308
カスタム SAML 2.0 認証方式の作成（オプション）	308
サービス プロバイダでの IDP ディスカバリ設定	309
SP でのアイデンティティ プロバイダ ディスカバリの設定.....	310
IdP ディスカバリ ターゲットの攻撃からの保護	311
メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ.....	312
MessageConsumerPlugin の実装	314
メッセージ コンシューマ プラグインの展開	315
SAML 2.0 におけるメッセージ コンシューマ プラグインの有効化	316
HTTP ヘッダとしての SAML 属性の供給.....	316
HTTP ヘッダとしての SAML 属性のユース ケース.....	317
属性を HTTP ヘッダとして提供するため設定の概要.....	320
SAML 属性を保存するためのリダイレクト モードの設定	320
ユーザを検証するための認可ルールの作成	321
属性を HTTP ヘッダとして送信するレスポンスの設定	322
属性を HTTP ヘッダとして実装するためのポリシーの作成	323
SAML 2.0 認証が失敗した場合のリダイレクト URL の指定	325
SP においてプロキシ サーバで処理されるリクエスト.....	326
SP においてプロキシ サーバを使用して処理するリクエストの設定.....	327
バックチャネル対応のクライアント証明書認証の有効化（オプション）	328
証明書データ ストアへのクライアント証明書の追加	329
バックチャネルに関するクライアント証明書オプションの設定	330
SAML 2.0 認証方式でターゲット フェデレーション リソースを保護する方法.....	331
各認証方式に対する固有なレルムの設定	332
すべての認証方式に対する単一のターゲット レルムの設定	333

第 16 章: WS-フェデレーション アカウント パートナーの設定 339

アサーティング パートナーの前提条件（レガシー）	339
アカウント パートナーを設定する方法	340
アカウント パートナーに関するオプション設定タスク	341
レガシー フェデレーション ダイアログ ボックスへの移動.....	341
アフィリエイト ドメインへのリソース パートナーの追加.....	342
リソース パートナーオブジェクトに関する一般情報の設定	342
SiteMinder セッションのないユーザの認証	343
シングル サインオンのアサーション有効期間	346
リソース パートナー利用可能時間の制限の設定（オプション）	348

リソース パートナーの IP アドレス制限の設定 (オプション)	349
アサーションが生成される対象のユーザの選択	350
リソースへのアクセスからのユーザまたはグループの除外	351
ネストされたグループによるリソースへのアクセスの許可	352
手動設定によるユーザの追加	353
WS フェデレーションアサーションの名前 ID の設定	354
WS フェデレーションの場合のシングル サインオンの設定	355
アカウント パートナーでのシングル サインオンの開始	356
リソース パートナーでのシングル サインオンの開始	356
SAML アサーション レスポンスのカスタマイズ (オプション)	357
AssertionGeneratorPlugin の実装	357
アサーション ジェネレータ プラグインの展開	358
アサーション ジェネレータ プラグインの有効化	359
Web アプリケーションの属性によるアサーションのカスタマイズ	360
WS-Federation のサインアウトの設定	362
WS フェデレーション アサーションの属性の設定 (オプション)	363
WS-Federation のアサーション属性の設定	364
アサーション属性の最大文字数を指定します	366
属性の新規作成スクリプトの使用	367

第 17 章: SiteMinder の WS-フェデレーション リソース パートナーとしての設定

369

依存パートナーの前提条件	369
リソース パートナーを設定する方法	370
リソース パートナーに関するオプション設定タスク	371
レガシー フェデレーション ダイアログ ボックスへの移動	371
WS フェデレーション認証方式の概要	372
WS フェデレーション認証方式タイプの選択	373
WS フェデレーション認証方式の一般情報の指定	374
認証用のユーザ レコードの検索	374
WS フェデレーション ユーザの LoginID の取得	375
検索仕様を使用した WS フェデレーション ユーザの検索	376
リソース パートナーでの WS フェデレーション シングル サインオンの設定	377
WS フェデレーション サインアウトの実装	378
サインアウトの有効化	379
カスタム WS フェデレーション認証方式の作成	379
メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ	380
MessageConsumerPlugin の実装	381
メッセージ コンシューマ プラグインの展開	382

WS フェデレーションでのメッセージ コンシューマ プラグインの有効化	383
WS フェデレーションの認証試行が失敗した後のユーザのリダイレクト	385
HTTP ヘッダとしての SAML 属性の供給	386
HTTP ヘッダとしての SAML 属性のユース ケース	387
属性を HTTP ヘッダとして提供するため設定の概要	390
SAML 属性を保存するためのリダイレクト モードの設定	390
ユーザを検証するための認可ルールの作成	391
属性を HTTP ヘッダとして送信するレスポンスの設定	392
属性を HTTP ヘッダとして実装するためのポリシーの作成	393
WS-Federation 認証方式でターゲット リソースを保護する方法	395
各認証方式に対する固有なレルムの設定	395
すべての認証方式に対する単一のターゲット レルムの設定	396

第 18 章: SAML 2.0 アフィリエーションの設定 403

アフィリエーションの概要	403
アフィリエーションによるシングル サインオン	403
アフィリエーションによるシングル ログアウト	404
SAML 2.0 アフィリエーションの設定	404
アイデンティティ プロバイダでのアフィリエーションの割り当て	405
サービス プロバイダでのアフィリエーションの割り当て	406

第 19 章: アサーション クエリの属性を使用したユーザの認可 407

属性機関による認可の実行	407
ユーザ属性を持つユーザを認可するフロー図	410
属性機関および SAML リクエストの設定方法	411
属性機関のセットアップ	412
属性機関での属性の設定	413
依存パートナーの属性機関サービスへのアクセスの許可	414
属性クエリを生成する SAML リクエストをセットアップする方法	417
属性クエリの有効化と属性の指定	417
属性クエリに関する NameID の設定	418
属性クエリに関するバックチャネルの設定	419
フェデレーション属性変数の作成	419
フェデレーション属性変数を使用したポリシー式の作成	420

第 20 章: 設定を簡略化する SAML 2.0 プロバイダ メタデータの使用 421

SAML 2.0 のメタデータ ツール	421
メタデータのエクスポート ツール	422

smfedexport ツールの実行	426
smfedexport のコマンド オプション	427
smfedexport ツールの例	430
メタデータのインポート ツール	432
smfedimport ツールの実行	433
smfedimport ツールの例	434
smfedimport のコマンド オプション	435
複数の SAML 2.0 プロバイダを使用したインポート ファイルの処理	437
複数の証明書エイリアスを使用したインポート ファイルの処理	438

第 21 章: レガシー フェデレーショントレース ロギング 439

追跡ロギング	439
トレース ログの FWS キャッシュのクリア	440
Fed_Client コンポーネントのログ メッセージ	441
トレース ロギングの設定	442
Fed_Server コンポーネントのログ メッセージ	443
フェデレーション サービス トレース ログ (smtracedefault.log)	443
ログ内の FWS データの更新	445
トレース設定テンプレートによるロギングの簡略化	446
FWS のトレース ロギング テンプレート	446
IdP および SP に関するトレース ロギング テンプレート	448

第 22 章: 同じ値を使用する必要がある設定 451

設定テーブルを使用する方法	451
SAML 1.x の一致設定	452
SAML 2.0 設定項目の一致設定	454
WS フェデレーションの設定	455

第 23 章: SiteMinder によって使用されるフェデレーション Web サービス URL 457

フェデレーション サービス URL	457
アサーティング パーティのサービスの URL	458
サイト間の転送サービス URL (SAML 1.x)	458
アサーション検索サービス URL (SAML 1.x)	459
Artifact 解決サービス URL (SAML 2.0)	460
シングルサインオン サービス URL (SAML 2.0)	462
シングルサインオン サービス URL (WS-Federation)	463
IdP のシングル ログアウト サービス URL (SAML 2.0)	464

AP のサインアウト サービス URL (WS フェデレーション)	465
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 サービス URL (SAML 2.0)	466
属性 サービス URL (SAML 2.0)	467
AP の WSFedDispatcher サービス URL.....	468
依存パーティでのサービスの URL.....	468
SAML 認証情報コレクタ サービス URL (SAML 1.x)	469
認証リクエスト サービス (SAML 2.0)	470
アサーション コンシューマ サービス URL (SAML 2.0)	471
セキュリティ トークン コンシューマ サービス URL (WS フェデレーション)	472
SP のシングル ログアウト サービス URL (SAML 2.0)	473
RP のサインアウト サービス URL (WS フェデレーション)	474
RP の WSFedDispatcher サービス URL.....	475
Web.xml ファイル	476

第 24 章: レガシー フェデレーションのトラブルシューティング 477

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	477
一般的な考慮事項.....	479
無効な smjavaagent.dll のために Web エージェント オプション パックを初期化できない.....	479
cookie ドメイン不一致エラー	479
コンシューマ/SP での認証完了後のエラー.....	480
コンシューマでアサーションを取得する際の HTTP 404 エラー	480
フェデレーション Web サービスがプロデューサ/IdP に SAML リクエストを送信しない.....	481
一致パラメータの大文字と小文字の区別による設定問題	481
ログオフ後のポリシー サーバシステムの障害	482
アサーション内のマルチバイト文字が正しく処理されない	482
ServletExec を使用する IIS Web サーバでトレース ログが表示されない.....	483
JVM の初期化中のエラー	483
SAML 1.x のみの問題	484
SAML 1.x Artifact プロファイル シングル サインオン の失敗.....	484
アサーション 検索サービスへのアクセスの認証に失敗する	484
認証方式を変更した後に認証が失敗する	485
SAML Artifact シングル サインオン に対するクライアント 認証が失敗する	485
SAML 2.0x のみの問題	486
アサーション 検索サービスへのアクセスの認証に失敗する	486
セッション ストアから有効期限データを削除するときの ODBC エラー	487

付録 A: パートナシップ モデルでのレガシー設定の再作成 489

第 1 章: レガシー フェデレーションの概要

このセクションには、以下のトピックが含まれています。

[フェデレーションにおけるパートナーに関する用語](#) (P. 17)

[レガシー フェデレーションのコンポーネント](#) (P. 18)

[レガシー フェデレーション認証方式](#) (P. 20)

[セキュリティゾーンを使用したフェデレーション シングル サインオン](#)
(P. 20)

[Secure プロキシ サーバ フェデレーション ゲートウェイ](#) (P. 23)

[レガシー フェデレーションにおける国際化](#) (P. 23)

[デバッグ機能](#) (P. 24)

[レガシー フェデレーション用 API](#) (P. 25)

[レガシー フェデレーション設定フローチャート](#) (P. 27)

フェデレーションにおけるパートナーに関する用語

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

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

- SAML 1.x プロデューサ
- SAML 2.0 ID プロバイダ
- WS フェデレーション アカウント パートナー

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

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

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

レガシー フェデレーションのコンポーネント

レガシー フェデレーションは、**SiteMinder** オブジェクトの設定（アフィリエイト ドメイン、アフィリエイト パートナー、認証方式、フェデレーション リソースを保護するポリシーなど）に基づきます。

SiteMinder フェデレーション製品、レガシー フェデレーションのユースケース、プロセスフロー図の詳細については、「[ユーザのエンタープライズでのフェデレーション](#)」を参照してください。

レガシー フェデレーションでは、以下のコンポーネントが使用されます。

SAML アサーション ジェネレータ

アサーティングパーティで **SAML** アサーションを作成するポリシー サーバ コンポーネントです。

SAML アサーション ジェネレータは、プロデューサ/IdP サイトにセッションを持つユーザに関するアサーションを作成します。パートナーが **SAML** アサーションをリクエストすると、**Web** エージェントは **SAML** アサーション ジェネレータを呼び出します。アサーション ジェネレータは、ユーザ セッションおよびポリシー ストア内の情報に基づくアサーションを作成します。

アサーション ジェネレータは、認証プロファイルまたは設定されたバインドに従って、以下のようにアサーションを処理します。

- **SAML Artifact** プロファイル/バインド

アサーション ジェネレータは、アサーションを **SiteMinder** セッションストアに保存します。アサーションへの参照が **SAML Artifact** の形式で **Web** エージェントに返されます。

- **SAML POST** プロファイル/バインド

SiteMinder は **HTTP** 形式に埋め込み **SAML** レスポンスとしてブラウザ経由でアサーションを返します。

Web エージェントには、**SAML** プロファイルに従って、**SAML Artifact**、**SAML** レスポンス、または **WS-フェデレーションセキュリティ** トークン レスポンスを依存パーティに送信する役割があります。依存パーティでは、クライアントが **SAML Artifact** またはレスポンス メッセージの処理に使用可能であることが必要です。**SiteMinder** が依存パーティである場合、**SAML 1.x** 認証情報コレクタ、または **SAML 2.0** アサーション コンシューマをクライアントとして使用できます。

アサーション ジェネレータ プラグインを設定することにより、SAML アサーションの内容をカスタマイズできます。このプラグインを使用して、フェデレーション環境の内容をカスタマイズできます。

WS-フェデレーション アサーション ジェネレータ

SAML アサーションが含まれる WS-フェデレーション

RequestSecurityTokenResponse メッセージを作成するポリシー サーバ コンポーネントです。

WS-フェデレーション アサーション ジェネレータは、アカウント パートナーにセッションを持つユーザに関する SAML 1.1 アサーションを作成します。ユーザがリソースをリクエストすると、Web エージェントは WS-フェデレーション アサーション ジェネレータを呼び出します。ポリシー サーバは、ユーザセッションおよびポリシー ストアに設定されている情報に基づくアサーションを作成します。その後、アサーション ジェネレータは、WS-フェデレーション RequestSecurityTokenResponse メッセージにアサーションを配置します。

Web エージェントは、WS-フェデレーション パッシブ リクエスト プロファイルに従って、ブラウザを使用してセキュリティ トークン レスポンス メッセージを依存パーティに送信します。リソース パートナーでは、WS-フェデレーション アサーション コンシューマなどのクライアントがアサーションの処理に使用できることが必要です。

アサーション ジェネレータ プラグインを設定することにより、SAML アサーションの内容をカスタマイズできます。このプラグインを使用して、フェデレーション環境の内容をカスタマイズできます。

フェデレーション認証方式

SAML または WS-フェデレーション アサーションを検証し、アサーション データを依存パーティのローカル ユーザにマップするポリシー サーバ コンポーネントです。

サポートされる認証方式は次のとおりです。

- SAML 1.x Artifact
- SAML 1.x POST
- SAML 2.0 (Artifact および POST バインド)
- WS-フェデレーション

フェデレーション Web サービス

アサーティング パーティでのアサーション取得、セッション同期、および通知アラートをサポートする **Web** エージェント コンポーネントです。依存パーティにおいて、これらのサービスはアサーションを収集します。

レガシー フェデレーション認証方式

レガシー フェデレーションでは、以下の認証方式がサポートされます。

- SAML 1.x Artifact
- SAML 1.x POST
- SAML 2.0
- WS-フェデレーション

それぞれの認証方式を使用して、依存パーティの **SiteMinder** が **SAML** アサーションを処理できます。アサーションを受信すると、認証方式は以下のように動作します。

- SAML アサーションを検証する
- データをローカル ユーザにアサーションをマッピングする
- アサーションを消費するサイトで **SiteMinder** セッションを確立する

SAML 認証方式では、アサーティング パーティのリモート ユーザを依存パーティのローカル ユーザにマッピングするように動作します。ユーザをマッピングすることにより、認証方式が認証用の正しいユーザ レコードを検索できます。

セキュリティゾーンを使用したフェデレーション シングル サインオン

SiteMinder 環境では、**Web** サービス保護用の **Web** アプリケーション環境およびフェデレーション リソース保護用のフェデレーション環境を含めたセットアップができます。この方法により、**SiteMinder** 展開をより効率化できます。

特定のフェデレーション機能では、永続的なユーザセッションが必要になります。すなわち、**SAML** アサーションがポリシー サーバのセッションストアに保存されます。

該当する機能は、以下のとおりです。

Artifact シングル サインオン

SAML 1.x および **SAML 2.0** の場合、**SAML** アサーションは、依存パーティが後で取得する永続セッションに保存されます。

シングル ログアウト

プロデューサ サイトおよびコンシューマ サイトでのシングル ログアウト (**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)。連携したログアウト中のパートナーの通知を容易にするために、パートナー データは永続的なユーザセッションに保存されます。

永続的なユーザセッションを使用すると、アサーションの取得やログアウト リクエストの処理に必要なコールがセッションストアに対して行われるため、パフォーマンスが低下します。パフォーマンスに対する影響を限定するには、セキュリティ ゾーンを使用します。

セキュリティゾーンは、単一の **Cookie** ドメインのセグメントです。セキュリティゾーンを使用することにより、アプリケーションを分割して、リソース アクセスに応じた異なるセキュリティ要件を許可することができます。1つのゾーン内のすべてのアプリケーションが、相互にシングルサインオンを許可できます。別のゾーンにあるアプリケーションについては、信頼関係の設定によりシングルサインオンが決定されます。

アサーティング パーティにおけるフェデレーション アプリケーションでは、以下のセットアップを実行します。

- 専用のセキュリティ ゾーンを作成する。
- 連携されないすべてのアプリケーションに対して、別のゾーンを 1 つ作成する。
- 非連携ゾーンを信頼するように連携ゾーンを設定する。

別のゾーンを使用することにより、コールがフェデレーション アプリケーション専用のセッション サーバに限定されます。

注: フェデレーション環境では、**Web** エージェントのみ、セキュリティゾーンを使用するように設定できます。セキュア プロキシエージェントおよびアプリケーション サーバはこの機能をサポートしていません。

セキュリティゾーンを設定するには、以下の **Web** エージェント パラメータの値を入力します。

SSOZoneName

シングルサインオンセキュリティ ゾーンを識別します。ゾーンをドメインに関連付けるために、ゾーン名が **Cookie** ドメイン名に追加されます。

注: この項目では、英語の文字のみがサポートされています。その他の言語の文字はサポートされていません。

SSOTrustedZone

トラステッドセキュリティ ザーンの番号付きリストを表示します。ゾーンおよびトラステッドゾーン リストを定義することにより、**Web** エージェントが読み書きできる **Cookie** が決定されます。

これらのパラメータは、エージェント設定オブジェクトまたはローカルのエージェント設定ファイルの一部です。

セキュリティゾーンの詳細については、「**Web** エージェント設定ガイド」を参照してください。

Secure プロキシ サーバ フェデレーション ゲートウェイ

CA SiteMinder for Secure Proxy Server フェデレーション ゲートウェイは、フェデレーション ネットワーク内のアクセス制御にプロキシベースのソリューションを提供します。従来のプロキシは、通常、インターネット リソースをリクエストするユーザのグループとして機能します。SPS フェデレーション ゲートウェイは、リバース プロキシであり、企業のリソースをリクエストするユーザの代表として動作します。

SPS フェデレーション ゲートウェイは、自己完結型システムです。このゲートウェイには、独自のサーブレット エンジンおよび Web サーバ ビルトインが組み込まれています。SPS ゲートウェイは、フェデレーション パートナーから保護されたリソースへのリクエストの処理において、そのプロキシエンジンに依存します。フェデレーション ゲートウェイとして動作するように SPS を機能強化することで、迅速な展開が可能になります。

レガシー フェデレーション のコンポーネントとして、SPS フェデレーション ゲートウェイは、Web エージェントおよび Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーションのサービスを提供できます。単一の SPS フェデレーション ゲートウェイは、必要な Web エージェントの台数を抑制することにより、リソースをアクセスするために必要とされる設定の量を制限できます。

注: CA SiteMinder for Secure Proxy Server には、SiteMinder とは別のライセンスが必要です。

レガシー フェデレーション における国際化

レガシー フェデレーション は、I18N 国際化に対応するために以下の機能をサポートします。

- レガシー フェデレーション 設定オブジェクト、Java、および C++ コードは、国際化の目的のための UTF-8 形式でエンコードされます。
- マルチバイトのユーザ ID と属性値を持つデフォルト アサーションおよびカスタム アサーションの作成および消費。
- エンコードされたターゲット URL およびリダイレクト URL。これらの URL は HTTP 1.1 RFC 2616 でエンコードされるため、マルチバイトパスおよびファイル名は正しく処理されます。

アサーションにマルチバイト文字が含まれる場合は、使用しているオペレーティングシステムの **LANG** 設定を以下の **UTF-8** 形式に設定してください。

LANG=xx_xx.UTF-8

たとえば、日本語の場合のエントリは次のとおりです。

LANG=ja_JP.UTF-8

デバッグ機能

レガシー フェデレーション コンポーネントは、フェデレーション ネットワークでのアクティビティを監視およびデバッグするために特定のイベントをロギングします。

Web エージェント ログ

プロデューサ サイトで **SAML** アサーションを生成するリクエストに関する情報を表示します。

フェデレーション Web サービス ログ

SAML アサーションの取得および **SAML** アサーションの消費のリクエストに関する情報を表示します。

ポリシー サーバ ログ

SAML アサーション ジェネレータおよび **SAML Artifact** 認証方式からのコールの結果を表示します。また、設定したポリシー サーバ トレース メッセージも表示します。これらのメッセージは、ポリシー サーバ 管理プロファイラを使用するか、提供されるプロファイラ テンプレート ファイルの 1 つを使用して設定できます。

Web エージェント オプション パック ログ

FWS トレース メッセージを表示します。これらのメッセージは、**FWSTrace.conf** ファイルを使用するか、提供されるトレース テンプレート ファイルの 1 つを使用して設定できます。

レガシー フェデレーション用 API

以下の API は、レガシー フェデレーション のサポートを提供します。

- ポリシー管理 API
- Java メッセージ コンシューマ プラグイン API
- Java アサーション ジェネレータ プラグイン API

ポリシー管理 API

C および Perl のポリシー管理 API は、SiteMinder フェデレーションのサポートとして新規の言語要素を提供します。これらの新規言語要素には、以下のものがあります。

- フェデレーション オブジェクト用の C 構造体および Perl パッケージ。これらのオブジェクトには、アフィリエイト ドメイン、アフィリエイト、ID、サービス プロバイダ、リソース、アカウント パートナーが含まれます。
- SAML 1.x、SAML 2.0、および WS フェデレーション設定用の C 関数および Perl メソッド。
- SAML 2.0 メタデータ定数。
- WS フェデレーション メタデータ定数。

ポリシー管理 API の詳細については、「Perl 用 SiteMinder プログラミングガイド」または「C 言語用 SiteMinder プログラミングガイド」を参照してください。

Java メッセージ コンシューマ プラグイン API

SiteMinder Java MessageConsumerPlugin API は、SAML 1.x、SAML 2.0、および WS-Federation Message Consumer Extension インターフェースを実装します。この API を使用することで、ユーザの不明瞭解消および認証の独自処理が可能になります。独自の要件に合わせてコードをカスタマイズした後、カスタム プラグインを SiteMinder に統合することで、SAML アサーション レスポンスや WS フェデレーション セキュリティ トークン レスポンスの処理および操作を拡張できます。

詳細については、「SiteMinder Java 用プログラミング ガイド」を参照してください。

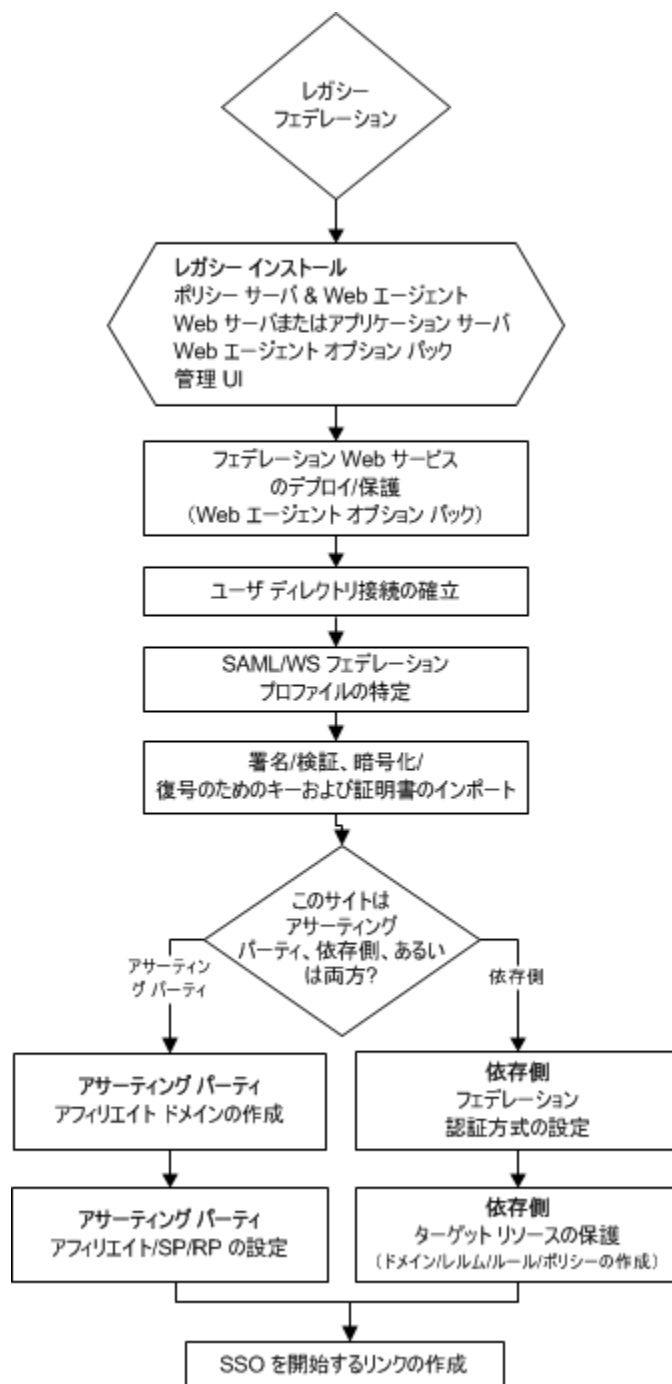
Java アサーション ジェネレータ プラグイン API

SiteMinder Java アサーション ジェネレータ プラグイン API は Assertion Generator Framework を実装します。プラグインを使用すると、パートナーとベンダーの間のビジネス契約のアサーション コンテンツを変更できます。

詳細については、「SiteMinder Java 用プログラミング ガイド」を参照してください。

レガシー フェデレーション設定フローチャート

次のフローチャートは、レガシー フェデレーション を設定する一般的なプロセスの要点を示しています。



第 2 章：サンプル設定の使用によるレガシー フェデレーションについての理解

このセクションには、以下のトピックが含まれています。

[手動の SiteMinder 対 SiteMinder 展開の概要](#) (P. 29)

[必須コンポーネントのインストール確認](#) (P. 31)

[サンプル フェデレーション ネットワーク](#) (P. 31)

[サンプル ネットワーク用のアイデンティティ プロバイダのセットアップ](#)
(P. 37)

[サンプル ネットワーク用のサービス プロバイダのセットアップ](#) (P. 54)

[SAML 2.0 シングル サインオンのテスト](#) (P. 67)

[フェデレーション展開への機能の追加](#) (P. 68)

手動の SiteMinder 対 SiteMinder 展開の概要

展開を手動で実行できます。手動の展開タスクは、単純な設定（POST バインドを使用したシングル サインオン）から始めます。基本的な設定から始めることによって、最少の手順でフェデレーションがどのように機能するかを確認することができます。

POST シングル サインオンが動作するようになった後、Artifact バインド、デジタル署名、暗号化の設定などの追加タスクについて説明されます。実稼働環境に応じてこれらの機能を追加できます。

重要：この展開実習は、SAML 2.0 のみに対応しています。これらの手順は、SAML 1.x や WS フェデレーション設定には該当しません。

手動の展開例は、以下の点でサンプルアプリケーションの展開と異なります。

- 説明されている展開は、各システム上にポリシー サーバと **Web** エージェントを持つ 2 つのシステムの間でセットアップされます。これら 2 つのシステムは、**IdP** および **SP** を表します。
- サンプルアプリケーションでセットアップしない手動の設定で実現される追加の機能が記載されています。以下のような機能があります。
 - **Artifact** バックチャネルに対する **SSL** の設定
 - アサーションへの属性の追加
 - デジタル署名およびアサーションの検証
 - アサーションの暗号化および復号

手動の展開の手順では、全体を通してサンプルデータが使用されています。実際の環境でのデータを使用するには、ユーザのアイデンティティ プロバイダおよびサービス プロバイダの設定に対するエントリを指定してください。

必須コンポーネントのインストール確認

フェデレーションには、以下のコンポーネントのインストールが要求されます。

- SiteMinder ポリシー サーバ
- 管理 UI
- Web エージェント
- Web エージェント オプション パック

このサンプル フェデレーション 展開例では、これらのコンポーネントがインストールされており作動しているものと仮定します。

オプションで、以下の機能をセットアップします。

- SSL 通信用の Web またはアプリケーション サーバを有効にします。

SSL は、HTTP-Artifact シングル サインオンでのバックチャネル通信のセキュリティを保護するためのオプションです。

- セッション ストアとして有効化されるデータベースをセットアップします。

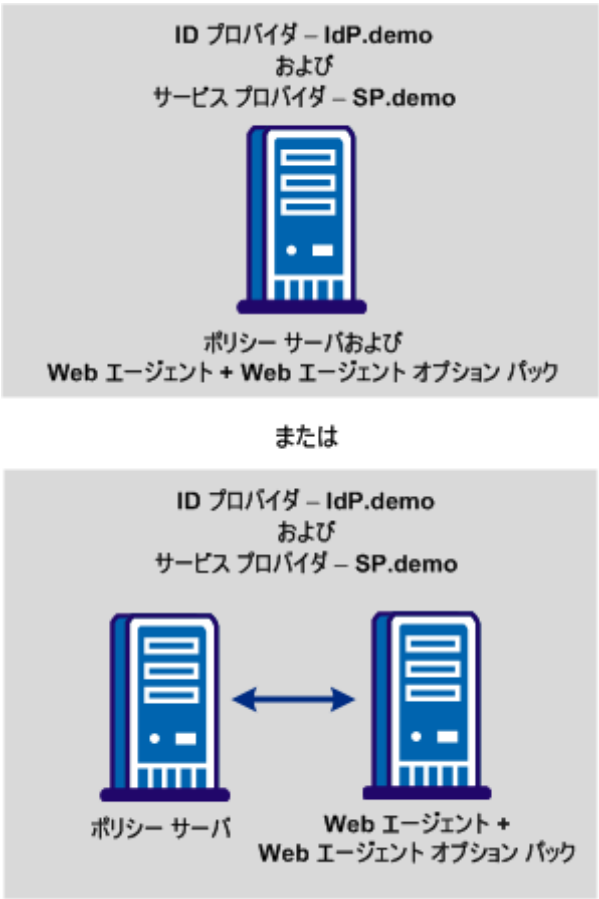
セッション ストアは、HTTP-Artifact バインドを使用する場合に、アサーティング パーティでのアサーションの保存に必要とされます。また、セッション ストアは、アサーション パーティや依存パーティでのシングル ログアウトにも必要とされます。

セッション ストア データベースのセットアップ手順については、「ポリシー サーバインストール ガイド」を参照してください。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。手順については、「ポリシー サーバ管理ガイド」を参照してください。

サンプル フェデレーション ネットワーク

SiteMinder フェデレーション ネットワークのサンプル Web サイトは、idp.demo という名前のアイデンティティ プロバイダと sp.demo という名前のサービス プロバイダで構成されます。ビジネス パートナリシップは、idp.demo と sp.demo の間で確立されています。

次の図は、サンプル フェデレーション ネットワークを示しています。



基本的な設定におけるアイデンティティプロバイダ データ

IdP.demo はアイデンティティ プロバイダです。以下の表には、最も基本的な SAML 2.0 POST シングル サインオン設定のサンプルデータが含まれます。

アイデンティティプロバイダ コンポーネン サンプル ネットワーク	
ト	
IdP ポリシー サーバ	サーバ : www.idp.demo:80
	サーバ タイプ : IIS Web Server

アイデンティティプロバイダ コンポーネン ト

IdP ポリシー ストア	IP アドレス : www.idp.demo:389 ストレージ : LDAP (Sun One Directory Server) ルート DN : o=idp.demo 管理者ユーザ名 : cn=Directory Manager パスワード : federation
ユーザ ストア	ディレクトリ名 : IdP LDAP サーバ : www.idp.demo:42088 サーバタイプ : Sun One Directory Server (LDAP) ユーザ ストア : LDAP ディレクトリには以下のユーザが含まれます。 ■ user1 ■ user2 ユーザパスワード : test メール : <user_name>@idp.demo ルート : dc=idp,dc=demo 開始 : uid= 終了 : ,ou=People,dc=idp,dc=demo
Web エージェント オプション パック のある IdP Web エージェント	サーバ : www.idp.demo:80 サーバタイプ : IIS Web Server エージェント名 : idp-webagent
アサーション コンシューマ サービス URL	URL : http://www.sp.demo:81/affwebservices/ public/saml2assertionconsumer
アサーション取得サービス URL	URL : http://www.idp.demo:80/affwebservices/assertionretriever

アイデンティティプロバイダ コンポーネン ト サンプル ネットワーク

認証 URL	URL : http://www.idp.demo/siteminderagent/ redirectjsp/redirect.jsp

詳細設定におけるアイデンティティプロバイダ データ

以下の表には、Artifact プロファイル、アサーションの署名および暗号化など、より高度な SAML 2.0 機能のサンプルデータが含まれています。

アイデンティティプロバイダ コンポーネン ト サンプル ネットワーク

セッション ストア	サーバ : www.idp.demo データベース タイプ : ODBC データベース ソース情報 : SiteMinder セッション データ ソース ユーザ名 : admin パスワード : dbpassword
SSL-enabled サーバ	サーバ : www.idp.demo:443 サーバ タイプ : IIS 6.0 Web Web エージェント オプション パックのある Web サーバは Artifact バインドに対して SSL 対応です
証明機関の証明書 (CA)	CA の証明書 : docCA.crt DER エンコードされた証明書 : docCA.der この CA は、SSL を有効にするためにサーバサイド証 明書に署名します
SAML レスポンスに署名する秘密キー/証 明書ペア	証明書 : post-cert.crt 秘密キー : post-pkey.der パスワード : fedsvcs
暗号化用の証明書 (公開キー)	公開キー : sp-encrypt.crt

アイデンティティプロバイダ コンポーネント サンプル ネットワーク

アサーションに含める属性	属性： 未指定（デフォルト） 属性の種類： ユーザ DN 変数名： firstname 変数値： givenname

基本的な設定におけるサービス プロバイダ データ

サービス プロバイダは SP.demo です。以下の表には、最も基本的な SAML 2.0 POST シングル サインオン設定のサンプルデータが含まれます。

サービス プロバイダ コンポーネン ト サンプル ネットワーク

SP ポリシー サーバ	サーバ： www.sp.demo:80 サーバ タイプ： IIS Web Server
SP ポリシー ストア	P アドレス： www.sp.demo:389 ストレージ： LDAP （Sun One ディレクトリ サーバ） ルート DN： o=ca.com 管理者ユーザ名： cn=Directory Manager パスワード： federation

サービス プロバイダ コンポーネント	サンプル ネットワーク
ユーザ ストア	ディレクトリ名 : SP LDAP サーバ : www.sp.demo:32941 サーバタイプ : LDAP (Sun 1 ディレクトリ サーバ) ユーザ ストア : LDAP ディレクトリには以下のユーザが含まれます。 ■ user1 ■ user2 ユーザ パスワード : customer メール : <user_name>@sp.demo ルート : dc=sp,dc=demo 開始 : uid= 終了 : ,ou=People,dc=sp,dc=demo
SP Web エージェントおよび Web エージェント オプション パック	サーバ : www.sp.demo:81 サーバタイプ : Sun ONE 6.1 Web Server エージェント名 : sp-webagent
シングル サインオン サービス	SSO サービス : http://www.idp.demo:80/affwebservices/public/saml2sso
ターゲット リソース	ターゲット リソース : http://www.sp.demo:81/spsample/protected/target.jsp

高度な設定におけるサービス プロバイダ データ

以下の表は、Artifact プロファイルのセットアップ、アサーションの署名および暗号化など、より高度な SAML 2.0 機能のサンプルデータを示しています。

サービス プロバイダ コンポーネント	サンプル ネットワーク
Artifact 解決 サービス	解決サービス : https://www.idp.demo:443/affwebservices/saml2artifactresolution
証明機関の証明書	CA の証明書 : docCA.crt DER エンコードされた証明書 : docCA.der この CA は、SSL を有効にするためにサーバサイド証明書に署名します
証明書（公開キー） SAML レスポンスの署名を検証するために使用されます	証明書 : post-cert.crt
秘密キー/証明書ペア 復号とデジタル署名に使用されます	秘密キー : sp-encrypt.der 公開キー : sp-encrypt.crt パスワード : fedsvcs 発行者 DN : CN=Certificate Manager,OU=IAM,O=CA.COM シリアル番号 : 008D 8B6A D18C 46D8 5B

サンプル ネットワーク用のアイデンティティ プロバイダのセットアップ

アイデンティティ プロバイダでレガシー フェデレーションを展開するためのタスクの詳細について、以下のセクションで説明します。各セクションのエントリは、ベーシック設定用に提供されているサンプルデータを反映したものです。

注: これらの手順では、必要なコンポーネントがすでにインストールされているものと仮定しています。

IdP ユーザ ストアのセットアップ

アイデンティティ プロバイダでは、ユーザが定義されたユーザ ストアが必要です。アイデンティティ プロバイダは、これらのユーザに関するアサーションを作成できます。この展開では、ユーザ ストアは **Sun ONE LDAP** ユーザ ディレクトリです。このユーザ ストアにユーザを追加するために、**Sun ONE** サーバ コンソールが使用されています。

ユーザ ストアを設定する方法

1. 以下のユーザを追加します。

- user1
- user2

2. user1 および user2 の属性に以下のように入力します。

user1

ユーザ パスワード : test

メール : user1@idp.demo

user2

ユーザ パスワード : test

メール : user2@idp.demo

重要: 電子メールアドレスは、同じユーザに対してサービス プロバイダ ユーザ ストア内で同じである必要があります。

3. [トレース ロギングを有効化](#) (P. 40) します。

ポリシー サーバが IdP LDAP ポリシー ストアを指すための設定

この展開では、LDAP ポリシー ストアが使用されます。ポリシー サーバが LDAP ポリシー ストアを指していることを確認してください。

注: この手順では、読者がその展開内のユーザ ストアへのユーザの追加方法について既知であるものとしています。

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を開きます。
2. [データ] タブを選択します。

3. 以下のフィールドに値を入力します。

データベース

ポリシー ストア

ストレージ

LDAP

IP アドレス (LDAP ディレクトリ)

www.idp.demo : 389

ルート DN

o=idp.demo

管理者ユーザ名

cn=Directory Manager

パスワード

password

パスワードの確認入力

password

4. [OK] をクリックして変更内容を保存し、ダイアログを閉じます。
5. 「[IdP ユーザ ストアのセットアップ](#) (P. 38)」に進みます。

IdP でのポリシー サーバトレース ログिंगの有効化

アイデンティティ プロバイダで、ポリシー サーバに関するログングを有効にします。 `smtracedefault.log` ログ ファイルを表示して、シングルサインオンおよびシングルログアウトに関するトレース メッセージを調べることができます。このログ ファイルは、`policy_server_home/siteminder/log` ディレクトリにあります。

次の手順に従ってください:

1. ポリシー サーバ管理コンソールを開きます。
2. [プロファイラ] タブをクリックし、トレース ログの内容をカスタマイズします。

注: フェデレーション トレース メッセージを参照するには、ログに `Fed_Server` コンポーネントを含めます。

ポリシー サーバ管理コンソールを使用して、ポリシー サーバでトレース ログングを設定してください。

3. IdP Web エージェントをインストールします。

Web エージェント オプション パックによる Web サーバの設定

サンプル展開用にフェデレーション Web サービス (FWS) アプリケーションを設定します。

FWS をセットアップするには、以下の作業を行います。

- [フェデレーション Web サービス用の JDK のインストール](#) (P. 41)
- [IdP において FWS と連携するための ServletExec のインストールと設定](#) (P. 41)
- [IdP での AffWebServices.properties ファイルの設定](#) (P. 44)
- [IdP でのフェデレーション Web サービスのテスト](#) (P. 45)

フェデレーション Web サービス用の JDK のインストール

Web エージェント オプション パックは、フェデレーション Web サービス アプリケーションを起動する JDK を必要とします。

適切な JDK のバージョンを調べるには、[テクニカル サポート サイト](#)に移動し、SiteMinder プラットフォーム サポート マトリックスで該当リリースを検索してください。

IdP において FWS と連携するための ServletExec のインストールと設定

FWS が動作するには、ServletExec をインストールするか、サポートされる任意のアプリケーション サーバをインストールする必要があります。このサンプル ネットワークでは、IIS 6.0 Web サーバ上で ServletExec を使用します。

注: SiteMinder 12.52 SP1 には ServletExec_AS_6_license_key.txt という名前の ServletExec ライセンス キー ファイルが付属しています。このライセンス キーがない場合は、[CA テクニカル サポート](#)にご連絡ください。このライセンス ファイルからライセンス キーをコピーし、ServletExec 管理コンソールの [ServletExec ライセンス] ダイアログ ボックスに入力します。ServletExec のライセンスの詳細については、New Atlanta Communication <http://www.newatlanta.com> Web サイトで ServletExec のドキュメントを参照してください。

使用する ServletExec のサポートされているバージョン用の最新のホットフィックスを必ず適用してください。ホットフィックスは、フェデレーション Web サービスが ServletExec と連携するために必要となります。ホットフィックスを取得するには、[エラー! ハイパーリンクの参照に誤りがあります。](#) の Web サイトに移動してください。

ServletExec をセットアップする方法

1. ServletExec をインストールします。詳細については、New Atlanta のドキュメントを参照してください。
2. ServletExec 管理コンソールを開きます。
3. [Web アプリケーション] の下で、[管理] を選択します。
[Web アプリケーションの管理] ダイアログ ボックスが表示されます。
4. [Web アプリケーションの追加] をクリックします。

5. 以下の情報を入力します。

アプリケーション名

affwebservices

URL コンテキスト パス

/affwebservices/

場所

C:\¥program files¥ca¥webagent¥affwebservices

注: セットアップの方法によって affwebservices の場所が異なることがあります。適切な場所を入力してください。

6. [サブミット] をクリックします。
7. ServletExec コンソールを終了します。
8. IIS デフォルト ユーザ アカウントのディレクトリ セキュリティ設定を変更します。

重要: IIS ユーザ アカウントには、IIS がすべてのプラグインにファイル システムへの書き込みを許可するための適切な権限が必要です。そのため、フェデレーション Web サービスが ServletExec と連携するためには、IIS デフォルト ユーザ アカウントのディレクトリ セキュリティ設定の変更が必要となります。

詳細情報:

[IIS ファイル システムに書き込む ServletExec の有効化](#) (P. 42)

[IdP での FWS プロパティ ファイルの設定](#) (P. 44)

IIS ファイル システムに書き込む ServletExec の有効化

IIS サーバ ユーザ アカウントには、IIS がプラグインによるファイル システムへの書き込みを許可するための適切な権限が必要です。ServletExec がフェデレーション ログ ファイルに書き込むためには、ServletExec と関連付けられる匿名 ユーザ アカウントがファイル システムに対する書き込み権限を持っている必要があります。

次の手順に従ってください:

1. ServletExec がインストールされているシステム上の IIS Internet Information Services Manager を開きます。
2. [Web サイト]、[デフォルト Web サイト] に移動します。
一連のアプリケーションが右ペインに表示されます。
3. [ServletExec] を選択し、[プロパティ] を右クリックします。
4. [プロパティ] ダイアログ ボックスで [ディレクトリ セキュリティ] タブを選択します。
5. [認証およびアクセス制御] セクションで [編集] をクリックします。
[認証方法] ダイアログ ボックスが開きます。
6. 制御を以下のように設定します。
 - a. [匿名ユーザ アクセスの有効化] を選択します。
匿名アクセスについては、Windows ファイル システムの権限の許可を持つユーザ アカウントの名前およびパスワードを入力します。ユーザ アカウントにこの権限を付与するには、Windows のドキュメントを参照してください。たとえば、匿名アクセスに対して IUSR インターネット ゲスト アカウントを使用することができます。
 - b. [基本認証] をクリアします。
 - c. [統合 Windows 認証] をクリアします。
7. 入力を指示された場合、セキュリティ変更を Web サーバのすべての子コンポーネントに適用します。
8. Web サーバを再起動します。

ServletExec に関連付けられるユーザ アカウントは現在、IIS ファイル システ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. [コントロール パネル] - [管理ツール] - [ローカル セキュリティ ポリシー] - [ローカル ポリシー] - [ユーザ権限割り当て] を開きます。
[ローカル セキュリティ 設定] ダイアログ ボックスが表示されます。
2. [オペレーティング システムの一部として機能] をダブルクリックします。
[オペレーティング システムの一部として機能] プロパティ ダイアログが開きます。

3. [ローカルセキュリティ設定] ダイアログ ボックスに匿名のユーザー アカウントを追加します。
4. [OK] をクリックします。
5. コントロール パネルを終了します。
6. オプションで、IIS Web サーバを保護する Web エージェントのエージェント設定オブジェクトを確認することを強くお勧めします。このオブジェクトは、匿名のユーザによるファイル システムへの書き込み防止に関して `SetRemoteUser` パラメータが [yes] に設定されていることを確認します。

IdP での FWS プロパティ ファイルの設定

`affwebservices.properties` ファイルには、フェデレーション Web サービスの初期化パラメータがすべて含まれます。このファイル内の設定の少なくとも 1 つを変更します。

`affwebservices.properties` ファイルを変更する方法

1. Web エージェント オプション パックのある IdP システムで、`C:\Program Files\ca\webagent\affwebservices\WEB-INF\classes` ディレクトリに移動します。

2. `AgentConfigLocation` パラメータに `WebAgent.conf` ファイルの場所を設定します。このパラメータには、値の設定が必要です。

この展開の場合、IIS Web サーバが FWS アプリケーションをホストします。したがって、`WebAgent.conf` ファイルのパスは次のようになります。

`C:\Program Files\ca\webagent\bin\IIS\WebAgent.conf`

注: フェデレーション Web サービスは Java コンポーネントです。したがって、Windows パスにはダブルバックスラッシュが含まれる必要があります。この形式に該当するのは、Windows のみです。

このパスが 1 行に入力されていることを確認してください。

3. ファイルを保存して閉じます。
4. [IdP でフェデレーション Web サービスをテスト](#) (P. 45) します。

IdP でのフェデレーション Web サービスのテスト

フェデレーション Web サービスをセットアップした後、アプリケーションが正しく作動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. Web ブラウザを開き、次の URL を入力します。

`http://<fqhn>:<port_number>/affwebservices/assertionretriever`

`fqhn`

完全修飾ホスト名を定義します。

`port_number`

Web エージェントおよび Web エージェント オプション パックがインストールされているサーバのポート番号を定義します。

この展開の場合、次のように入力します。

`http://www.idp.demo:80/affwebservices/assertionretriever`

フェデレーション Web サービスが正しく作動している場合、次のメッセージが表示されます。

アサーション取得サービスは正常に初期化されました。(Assertion Retrieval Service has been successfully initialized.)

リクエストされたサーブレットは、HTTP POST リクエストのみを受け付けます。(The requested servlet accepts only HTTP POST requests.)

このメッセージは、フェデレーション Web サービスがデータ アクティビティをリスニングしていることを示します。フェデレーション Web サービスが正しく作動していない場合、アサーション検索サービスが失敗したというメッセージを受け取ります。アサーション検索サービスが失敗する場合は、フェデレーション Web サービス ログを確認してください。

2. [IdP で Web エージェント オプション パック ロギングを有効化](#) (P. 46) します。

IdP での Web エージェント オプション パック ロギングの有効化

IdP において、Web エージェント オプション パックのあるシステムのロギングを有効にします。以下のログを表示できます。

- affwebservices.log
- FWSTrace.log

次の手順に従ってください:

1. `LoggerConfig.properties` ファイルをセットアップすることにより、`affwebservices.log` を設定します。
2. FWS トレース ロギングを設定します。
3. IdP ポリシー サーバのユーザ ストアを指定します。

IdP ポリシー サーバのユーザ ストアの指定

IdP ユーザ ディレクトリは、アイデンティティ プロバイダがアサーションを生成する対象のユーザ レコードから構成されます。

以下の手順では、管理 UI でユーザ ディレクトリを設定する方法を説明します。IdP LDAP ディレクトリ は、`user1` および `user2` が含まれる Sun ONE LDAP ディレクトリです。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [ディレクトリ] - [ユーザ ディレクトリ] をクリックします。
3. [ユーザ ディレクトリの作成] をクリックします。
4. 以下のフィールドに値を入力します。

名前

IdP LDAP

ネームスペース

LDAP

サーバ

`www.idp.demo:42088`

5. [LDAP 設定] セクションの以下のフィールドに入力します。

ルート

dc=idp,dc=demo

その他の値はデフォルトのままにします。

[LDAP ユーザ DN の検索] の以下のフィールドに入力します。

開始

uid=

終了

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

6. [内容の表示] をクリックして、ディレクトリの内容を表示できることを確認します。
7. [サブミット] をクリックします。
8. IdP でアフィリエイト ドメインをセットアップします。

IdP でのアフィリエイト ドメインのセットアップ

アイデンティティ プロバイダに対応するサービス プロバイダを識別するには、アフィリエイト ドメインを作成し **sp.demo** 用のサービス プロバイダ オブジェクトを追加します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [フェデレーション] - [レガシー フェデレーション] - [アフィリエイト ドメイン] をクリックします。
3. [アフィリエイト ドメインの作成] をクリックします。
4. 以下のフィールドに値を入力します。

名前

フェデレーション サンプル パートナー

説明

sp.demo のドメイン

5. このダイアログ ボックスを開いたまま、[IdP でユーザ ディレクトリをアフィリエイト ドメインに追加](#) (P. 48) します。

IdP のアフィリエイトドメインへのユーザ ディレクトリの追加

ユーザ ディレクトリをアフィリエイト ドメインに関連付けます。

次の手順に従ってください:

1. [アフィリエイト ドメイン] ダイアログ ボックスで [ユーザ ディレクトリ] セクションに入力します。
2. IdP LDAP ディレクトリを追加します。
実際のネットワークの場合は、IdP でセットアップしたユーザ ストアを選択します。
3. [OK] をクリックします。
4. 「[IdP のアフィリエイト ドメインへのサービス プロバイダの追加](#) (P. 48)」に進みます。

IdP のアフィリエイトドメインへのサービス プロバイダの追加

sp.demo という名前の サービス プロバイダをアフィリエイト ドメインに追加します。

次の手順に従ってください:

1. 管理 UI で、[フェデレーション]-[レガシー フェデレーション]-[SAML サービス プロバイダ] の順に移動します。
2. [SAML サービス プロバイダの作成] を選択します。
3. 設定ウィザードに従います。
4. Federation Sample Partners をドメインとして選択した後、[次へ] をクリックします。
5. [一般] 手順で以下のフィールドに入力します。

名前

sp.demo

説明

サービス プロバイダ

SP ID

sp.demo

IdP ID

idp.demo

スキュー時間(秒)

デフォルトを受け入れる

認証 URL

<http://www.idp.demo/siteminderagent/redirectjsp/redirect.jsp>

この **redirect.jsp** は、アイデンティティ プロバイダサイトでインストールされる Web エージェント オプション パックに付属しています。この展開の場合、そのサーバは **www.idp.demo** です。ユーザーに **SiteMinder** セッションがない場合、IdP の SSO サービスは、ログインする認証 URL にユーザーをリダイレクトします。

認証が成功した後、**redirect.jsp** アプリケーションは、アサーション生成のために SSO サービスにユーザーをリダイレクトして戻します。**SiteMinder** ポリシーがこの URL を保護する必要があります。

有効

このオプションが選択されていることを確認します。デフォルトでは、このオプションが有効になっています。

6. UI を開いたまま、IdP がアサーションを生成する対象となる [ユーザーの選択] に移動します。

認証 URL の保護 (SAML 2.0)

認証 URL は、SiteMinder ポリシーを使用して保護する必要があります。認証 URL を保護することで、保護されたフェデレーション リソースをリクエストするユーザが IdP に SiteMinder セッションを持たない場合でも、それらのユーザに認証のチャレンジが確実に与えられることが保証されます。

次の手順に従ってください:

1. [ドメイン] から、認証 URL 保護ドメインと呼ばれるポリシー ドメインを作成します。
2. [ユーザ ディレクトリ] ページで、IdP LDAP ユーザ ディレクトリを追加します。
3. 認証 URL 保護ドメインから、以下のフィールド エントリを持つ永続的なレルムを作成します。

名前

認証 URL 保護レルム

エージェント

ルックアップ ボタンを使用して、FSS Web エージェントを選択します

これは、Web エージェント オプション パックのあるサーバを保護する Web エージェントです。

リソース フィルタ

/siteminderagent/redirectjsp/redirect.jsp

その他の設定はデフォルトのままにします。

[セッション] タブ

[永続セッション] を選択します

4. IDP 認証 URL 保護レルムから、レルムの下に以下のフィールド エントリを持つルールを作成します。

名前

認証 URL 保護ルール

レルム

認証 URL 保護レルム

リソース

*

Web エージェント アクション

Get

その他の設定はデフォルトのままにします。

5. 認証 URL 保護ドメインから、以下のエントリを持つポリシーを作成します。

名前

認証 URL 保護ポリシー

[ユーザ]タブ

IdP LDAP ユーザ ディレクトリから **user1** を追加します

[ルール]タブ

認証 URL 保護ルールを追加します

アイデンティティ プロバイダにおいて認証 URL を保護するポリシーが作成できました。

IdP がアサーションを生成する対象のユーザの選択

アフィリエイト ドメインにサービス プロバイダを指定する際、アサーション ジェネレータが SAML アサーションを生成する対象のユーザおよびグループのリストを含めます。アフィリエイト ドメインにあるディレクトリからのユーザおよびグループのみを追加します。

アサーション生成対象のユーザを選択する方法

1. [ユーザ] 手順に移動します。
2. [ユーザ ディレクトリ] セクションで、以前に設定した LDAP ユーザ ディレクトリの [メンバーの追加] を選択します。
[ユーザ/グループ] ダイアログ ボックスが表示されます。
3. 以下のフィールドに入力して、**user1** および **user2** を検索します。

検索タイプ

属性/値

属性

uid

値

*

これらの従業員が IdP LDAP にリスト表示されます。

4. [OK] をクリックします。
5. ウィザードの次の手順に進み、アサーションの名前 ID を設定します。

アサーションの名前 ID の設定

名前 ID は、アサーション内のユーザを識別するための独自の方法です。管理 UI に入力する NameID は、アサーションに含まれます。

名前 ID を設定する方法

1. [名前 ID] 手順に移動します。
[名前 ID] ダイアログ ボックスが表示されます。
2. 以下のフィールドに値を入力します。

名前 ID 形式

電子メール アドレス

形式の値として電子メール アドレスを指定すると、名前 ID がユーザを識別するためにユーザ ディレクトリの 電子メール アドレスを使用する必要があることを意味します。

[名前 ID タイプ] セクション

ユーザ属性

名前 ID フィールド -- 属性名

mail

3. ui を開いたまま、ウィザードの次の手順に進みます。

IdP での POST シングル サインオンの設定

HTTP-POST をシングル サインオンの SAML 2.0 バインドとして指定します。

次の手順に従ってください:

1. [SAML プロファイル] 手順に移動します。
2. 以下のフィールドに値を入力します。

オーディエンス

sp.demo

認証コンテキスト クラス参照

urn:oasis:names:tc:SAML:2.0:ac:classes:Password (default)

アサーション コンシューマ サービス

http://www.sp.demo:81/affwebservices/public/
saml2assertionconsumer

アサーション コンシューマ サービスの URL を指定します。実際のネットワークの場合、指定するサーバは、Web エージェント オブジェクト パックがインストールされた SP Web サーバです。

認証レベル

5 (デフォルト)

有効期間秒数

60 (デフォルト)

テスト環境で、以下のメッセージがポリシー サーバトレース ログに表示される場合、有効期間の値を 60 より大きくしてください。

```
Assertion rejected(_b6717b8c00a5c32838208078738c05ce6237) -current time  
(Fri Sep 09 17:28:33 EDT 2005) is after SessionNotOnOrAfter time (Fri Sep 09  
17:28:20 EDT 2005)
```

HTTP-POST

このチェック ボックスを選択

3. 残りのフィールドは無視します。
4. ウィザードの次の手順に進みます。

基本サンプル展開での署名処理の無効化

実稼働環境では、アサーションを署名するための署名処理は必須です。ただし、基本サンプル展開の場合、署名処理を無効にします。

重要: SAML 2.0 実稼働環境では絶対に署名処理を無効にしないでください。

次の手順に従ってください:

1. [暗号化 & 署名] 手順に移動します。
2. ページの [署名] セクションで、[署名の処理を無効にする] を選択します。
3. [次へ] をクリックして、ウィザードの [属性] 手順に移動します。

サービス プロバイダ オブジェクト設定の完了

属性は、サービス プロバイダ設定の最終手順です。基本設定の場合には、属性を設定しないでください。その代わりに、[終了] をクリックして、サービス プロバイダ設定を完了します。設定がサブミットされます。アイデンティティ プロバイダ用のサービス プロバイダ オブジェクトが識別されました。

サービス プロバイダの設定

アイデンティティ プロバイダでの設定が完了したら、サービス プロバイダの設定が必要です。

サンプル ネットワーク用のサービス プロバイダのセットアップ

サービス プロバイダでレガシー フェデレーションを展開するためのタスクの詳細について、以下のセクションで説明します。各セクションのエントリは、ベーシック設定用に提供されているサンプルデータを反映したものです。

注: これらの手順では、必要なコンポーネントがすでにインストールされているものと仮定しています。

SP ユーザ ストアのセットアップ

SP において、ユーザ ストアを設定し、アサーションを必要とするユーザに関するユーザ レコードを追加します。認証中にアサーションが提示されると、サービス プロバイダは、ユーザ ストアでユーザ レコードを検索します。

この展開の場合、Sun ONE LDAP ユーザ ディレクトリがユーザ ストアです。Sun ONE サーバ コンソールを使用して、ユーザをディレクトリに追加します。

ユーザ ストアを設定する方法

1. 以下のユーザを追加します。

- user1
- user2

2. user1 および user2 の属性に以下のように入力します。

user1

ユーザ パスワード : customer

メール : user1@sp.demo

user2

ユーザ パスワード : customer

メール : user2@sp.demo

重要: 電子メールアドレスは、同じユーザに対してアイデンティティ プロバイダのユーザ ストア内で同じである必要があります。

3. [トレース ロギングを有効化](#) (P. 57) します。

ポリシー サーバが SP LDAP ポリシー ストアを指すための設定

ポリシー サーバと LDAP ポリシー ストアの間の接続を確立します。

次の手順に従ってください:

1. ポリシー サーバ管理コンソールを開きます。
2. [データ] タブを選択します。

以下のフィールドに値を入力します。

データベース

ポリシー ストア

ストレージ

LDAP

LDAP IP アドレス

sp.demo:389

ルート DN

o=sp.demo

管理者ユーザ名

cn=Directory Manager

パスワード

federation

パスワードの確認入力

federation

3. [OK] をクリックします。
4. [SP ユーザ ストアをセットアップ](#) (P. 55) します。

SP でのフェデレーション コンポーネントのトレース ロギングの有効化

SP ポリシー サーバで、トレース ログ `smtracedefault.log` にフェデレーション コンポーネントをログ記録するように SiteMinder Profiler を設定し、トレース メッセージを確認します。

ログ記録を有効にする方法

1. ポリシー サーバ管理コンソールを開きます。
2. [プロファイラ] タブをクリックし、トレース ログの内容をカスタマイズします。フェデレーション トレース メッセージを参照するには、ログに `Fed_Server` コンポーネントを含めます。

ポリシー サーバでトレース ロギングを設定するには、ポリシー サーバ管理コンソールを使用します。

3. SP Web エージェントをインストールします。

Web エージェント オプション パックによる Web サーバの設定

Web エージェント オプション パックにより、フェデレーション Web サービス (FWS) アプリケーションがインストールされました。サンプル展開用に FWS アプリケーションを設定します。

FWS が動作するには、以下の手順を行います。

1. [フェデレーション Web サービス用の JDK のインストール](#) (P. 57)
2. [SP において FWS と連携するための ServletExec のインストールと設定](#) (P. 58)
3. [AffWebServices.properties ファイルの設定](#) (P. 59)
4. [Web エージェント オプション パック ロギングの有効化](#) (P. 61)
5. [フェデレーション Web サービスのテスト](#) (P. 60)

フェデレーション Web サービス用の JDK のインストール

Web エージェント オプション パックは、フェデレーション Web サービス アプリケーションを起動する JDK を必要とします。必要な特定のバージョンについては、[テクニカル サポート サイト](#)に移動し、SiteMinder プラットフォーム サポート マトリックスで該当リリースを検索してください。

SP において FWS と連携するための ServletExec のインストールと設定

この展開で FWS が作動するには、ServletExec Sun を ONE 6.1 Web サーバにインストールする必要があります。

注: SiteMinder 12.52 SP1 には ServletExec_AS_6_license_key.txt という名前の ServletExec ライセンス キー ファイルが付属しています。このライセンス キーがない場合は、[CA テクニカル サポート](#)にご連絡ください。このライセンス ファイルからライセンス キーをコピーし、ServletExec 管理コンソールの [ServletExec ライセンス] ダイアログ ボックスに入力します。ServletExec のライセンスの詳細については、New Atlanta Communication <http://www.newatlanta.com> Web サイトで ServletExec のドキュメントを参照してください。

ServletExec のサポートされているバージョン用の最新のホット フィックスを適用してください。ホット フィックスは、フェデレーション Web サービスが ServletExec と連携するために必要となります。ホット フィックスを取得するには、New Atlanta Communication <http://www.newatlanta.com> の Web サイトに移動してください。

ServletExec をセットアップする方法

1. ServletExec をインストールします。
手順については、「New Atlanta Communications」のドキュメントを参照してください。
2. ServletExec 管理コンソールを開きます。
3. [Web アプリケーション] の下で、[管理] を選択します。
[Web アプリケーションの管理] ダイアログ ボックスが表示されます。
4. [Web アプリケーションの追加] をクリックします。
5. 以下の情報を入力します。

アプリケーション名

affwebservices

URL コンテキスト パス

/affwebservices/

場所

C:\program files\ca\webagent\affwebservices

ネットワーク内の **affwebservices** の場所は異なることがあります。
適切な場所を入力してください。

6. [サブミット] をクリックします。
7. ServletExec コンソールを終了します。
8. [AffWebServices.properties ファイルを設定](#) (P. 59) します。

FWS.properties ファイルの設定

AffWebServices.properties ファイルには、フェデレーション Web サービスの初期化パラメータがすべて含まれます。このファイルに **WebAgent.conf** ファイルの場所を指定します。

次の手順に従ってください:

1. Web エージェント オプション パックのある SP システムで、
C:\Program Files\ca\webagent\affwebservices\WEB-INF\classes ディレクトリに移動します。

2. AgentConfigLocation パラメータに **WebAgent.conf** ファイルの場所を設定します。このパラメータの値の設定は必須です。

この展開の場合、サービス プロバイダで FWS アプリケーションをホストする Web サーバは、Sun ONE Web サーバです。したがって、**WebAgent.conf** ファイルのパスは次のようになります。

C:\Sun\WebServer6.1\https-sp.demo\config\WebAgent.conf

注: フェデレーション Web サービスは Java コンポーネントです。したがって、Windows パスにはダブルバックスラッシュが含まれる必要があります。このエントリは 1 行に指定してください。

3. ファイルを保存して閉じます。
4. [フェデレーション Web サービスをテスト](#) (P. 60) します。

フェデレーション Web サービスのテスト

フェデレーション Web サービス アプリケーションをセットアップした後、アプリケーションが正しく作動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. Web ブラウザを開き、次の URL を入力します。

`http://fqhn:port_number/affwebservices/assertionretriever`

`fqhn`

完全修飾ホスト名を定義します。

`port_number`

Web エージェントおよび Web エージェント オプション パックがインストールされているサーバのポート番号を定義します。

この展開の場合、次のように入力します。

`http://www.sp.demo:81/affwebservices/assertionretriever`

フェデレーション Web サービスが正しく作動している場合、次のメッセージが表示されます。

アサーション取得サービスは正常に初期化されました。(Assertion Retrieval Service has been successfully initialized.)

リクエストされたサーブレットは、HTTP POST リクエストのみを受け付けます。(The requested servlet accepts only HTTP POST requests.)

このメッセージは、フェデレーション Web サービスがデータ アクティビティをリスニングしていることを示します。フェデレーション Web サービスが正しく作動していない場合、アサーション検索サービスが失敗したというメッセージを受け取ります。アサーション検索サービスが失敗する場合は、フェデレーション Web サービス ログを確認してください。

2. [Web エージェント オプション パックのロギングを有効にします。](#) (P. 61)

SP での Web エージェント オプション パック ログイングの有効化

SP において、Web エージェント オプション パックのあるシステムのログイングを有効にして、以下のログを表示できるようにします。

- affwebserv.log

エラー ログイング メッセージが含まれています。

- FWSTrace.log

エラーおよびトレースのログイングを有効にする方法

1. `LoggerConfig.properties` ファイルを開きます。このファイルは、`web_agent_home/affwebservices/WEB-INF/classes` ディレクトリにあります。
2. `LoggingOn` パラメータを Y に設定します。
3. `LogFileName` 設定のデフォルトの名前と場所 (`affwebserv.log` ファイルを指しています) はそのままにしておきます。
4. `TracingOn` を Y に設定します。
5. `TraceFileName` 設定の名前と場所 (`FWSTrace.log` ファイルを指しています) はそのままにしておきます。

以上で、ログイングが有効になりました。

SP ポリシー サーバのユーザ ストアの指定

SP ユーザ ディレクトリは、サービス プロバイダが認証に使用するユーザ レコードで構成されます。

管理 UI のユーザ ディレクトリを設定します。SP LDAP という名前のディレクトリは、`user1` および `user2` というユーザが含まれる Sun ONE LDAP ディレクトリです。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [ディレクトリ] - [ユーザ ディレクトリ] をクリックします。
3. [ユーザ ディレクトリの作成] をクリックします。

4. 以下のフィールドに値を入力します。

名前

SP LDAP

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

ネームスペース

LDAP

サーバ

www.sp.demo:32941

6. [LDAP 検索] セクションで、以下のフィールドに入力します。

ルート

dc=sp,dc=demo

その他の値はデフォルトのままにします。

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

開始

uid=

終了

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

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

SP での SAML 2.0 認証方式の設定

サービス プロバイダでユーザを認証するには、**SAML 2.0** 認証方式を設定します。IdP のアサーションにより、認証用の認証情報が提供されます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [認証] - [認証方式] をクリックします。

3. 以下のフィールドに値を入力します。

[各方式共通セットアップ] セクション

名前

Partner IDP.demo 認証方式

認証方式タイプ

SAML 2.0 テンプレート

保護レベル

5 (デフォルト)

4. [SAML 2.0 設定] をクリックします。

一般およびユーザの特定を指定するためのダイアログ ボックスが表示されます。

5. [一般] セクションで以下の設定を指定します。

SP ID

sp.demo

IdP ID

idp.demo

SAML バージョン

2.0 (デフォルト)

スキュー時間

30 (デフォルト)

注: SP ID および IdP ID の値は、IdP での値に一致する必要があります。

6. [ユーザの特定] セクションで、以下の項目を設定します。

LDAP

Username=%s

7. [次へ] をクリックして、シングル サインオン設定に進みます。

詳細情報:

[サービス プロバイダでの署名の検証の有効化 \(P. 82\)](#)

SP でのシングル サインオンに関する HTTP-POST の設定

認証方式として、シングルサインオン バインドが使用されることを示し、アイデンティティ プロバイダとの通信方法をサービス プロバイダが認識できるようにします。

次の手順に従ってください:

1. [SSO] 設定で、以下のフィールドに入力します。

リダイレクト モード

302 Cookie データ (デフォルト)

セッション Cookie で HTTP 302 リダイレクトを介してユーザをリダイレクトしますが、他のデータは使用しません。

SSO サービス

`http://www.idp.demo:80/affwebservices/public/saml2sso`

オーディエンス

`sp.demo`

この値は、アイデンティティ プロバイダでの値に一致する必要があります。

ターゲット

`http://www.sp.demo:81/spsample/protected/target.jsp`

ターゲットが `http` で始まる場合は、リソースまでの完全パスを入力します。SAML 2.0 認証方式を使用する SiteMinder ポリシーは、ターゲットを保護します。

2. [バインディング] セクションで [HTTP-POST] を選択します。
3. [単一使用ポリシーを強制する] チェック ボックスをオフにします。
このオプションを無効にすることにより、サンプル ネットワークは SAML 2.0 準拠でなくなります。使い捨てポリシー機能の使用を有効にするには、サービス プロバイダでセッション ストアをセットアップします。

4. [署名と暗号化] の手順に達するまで [次へ] をクリックします。
5. [署名の処理を無効にする] を選択します。

重要: 署名の無効化は、初期的なシングル サインオン設定のデバッグのみを意図したものです。実稼働環境においては、署名処理は必須セキュリティ要件です。SP において、署名の検証を有効にし、署名を検証するように証明書データ ストアをセットアップします。

6. 設定手順が終了するまで [次へ] をクリックします。
7. [完了] をクリックします。
ベーシック認証方式設定が完了しました。
8. 管理 UI を開いたまま、「SAML 2.0 認証を使用したターゲット リソースの保護」 (Protect the Target Resource Using SAML 2.0 Authentication) に進みます。

SP でのターゲット リソースの保護

SAML 2.0 認証方式を設定した後、サービス プロバイダにおいてターゲット リソースを保護するポリシーでこの方式を使用します。

次の手順に従ってください:

1. [インフラストラクチャ] - [エージェント] - [エージェント] と順に移動し、sp-webagent という名前の Web エージェントを作成します。このエージェントは、Web エージェント オプション パックがインストールされているサーバを保護します。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
3. 以下の値を持つポリシー ドメインを作成します。

名前

IdP.demo ビジター用のドメイン

[ユーザ ディレクトリ]セクション

user1 と user2 を保持するユーザ ディレクトリを追加します。

4. [レルム] ページに移動し、以下の値を持つ永続的なレルムを設定します。

名前

SP ターゲット ページ保護レルム

エージェント

sp-webagent

リソース フィルタ

/spsample/protected.jsp

サービス プロバイダ Web サーバでターゲット リソースへのパスを定義します。

デフォルトリソース保護

保護されている

認証方式

Partner IdP.demo 認証方式

5. 以下の値を持つルールをレルムに追加します。

名前

SP ターゲット ページ保護ルール

レルム

SP ターゲット ページ保護レルム

リソース

*

アクション

Web エージェント アクション

Get

他のすべてのフィールドはデフォルトのままにします。

6. [ポリシー] ページに移動し、以下の値を持つポリシーを作成します。

[一般] ページ

名前

SP ターゲット ページ保護ポリシー

ユーザ pagexs

SP LDAP ディレクトリの場合は、[メンバーの追加] をクリックします。user1 を追加し、このユーザにターゲットへのアクセス権を付与します。

[ルール] ページ

SP ターゲット ページ保護ルールを追加します。

7. [サブミット] をクリックします。
ターゲット リソースの保護ポリシーの作成が完了しました。
8. 管理 UI を終了します。
9. HTML ページを使用して、[フェデレーションのセットアップをテスト](#) (P. 67) します。

SAML 2.0 シングル サインオンのテスト

SiteMinder 対 SiteMinder ネットワークにおけるシングル サインオンをテストするには、独自の HTML ページを使用します。この HTML ページには、認証リクエスト サービスへのハードコードされたリンクが含まれる必要があります。この展開の場合、POST バインド用のサンプル リンクは次のとおりです。

<http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo>

認証リクエスト サービスは、ユーザの認証コンテキストを取得するためのリンクで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 リソースにユーザを戻します。

注: 認証リクエスト リンク内の ProviderID は、SP において SAML 認証方式で指定された [IdP ID] フィールドの値に一致する必要があります。[IdP ID] フィールドは、[認証方式プロパティ] ダイアログ ボックスの [方式のセットアップ] タブにあります。

フェデレーション シングル サインオンをテストする方法

1. ブラウザを開きます。
2. シングルサインオンをトリガするリンクのある Web ページの URL を入力します。
ログインのチャレンジを表示する必要があります。
3. ユーザストアにある既存ユーザのログインを使用して、ユーザ認証情報を入力します。たとえば、**user1** がユーザストアに含まれるユーザである場合、このユーザの認証情報を入力します。

シングルサインオンが成功すると、ユーザのターゲット ページが表示されます。
4. シングルサインオンのテスト後、[機能をフェデレーション展開に追加 \(P. 68\)](#) できます。

フェデレーション展開への機能の追加

POST シングル サインオンの設定が終了した後、追加の機能をフェデレーション ネットワークに付加することができます。

この展開例で取り上げる追加タスクは以下のとおりです。

- シングル ログアウトの設定
- Artifact シングル サインオンの設定
- アサーションへの属性の追加
- アサーションのデジタル署名の有効化
- アサーションの暗号化および復号

これらの追加機能の中には、実稼働環境におけるシングル サインオンで必須のものがあります。たとえば、POST バインドにおけるデジタル署名などです。 必須タスクには注記が付いています。

シングル ログアウトの設定

シングル ログアウト プロトコル (SLO) は、特定のユーザに関するすべてのセッションの同時終了を実行し、セキュリティの保証に役立ちます。ログアウトを開始したブラウザとこれらのセッションを関連付けます。シングル ログアウトにより、特定ユーザに関するすべてのセッションが必ずしも終了するとは限りません。

シングル ログアウトの設定により、アイデンティティ プロバイダおよびサービス プロバイダがシングル ログアウト プロトコルをサポート可能になります。また、設定により、シングル ログアウトの処理方法が決定されます。

IdP でのシングル ログアウトの有効化

IdP でシングル ログアウトを開始できます。IdP (idp.demo) で、SP 単位でシングル ログアウトを有効化します。

次の手順に従ってください:

1. 管理 UI にログインし、sp.demo の SAML サービス プロバイダ オブジェクトにアクセスします。
2. [SAML プロファイル] ページに移動します。
3. [HTTP-Redirect] を選択します。
残りのフィールドがアクティブになります。
4. 次のフィールドに値を入力します。

SLO ロケーション URL

`http://www.sp.demo:81/affwebservices/public/saml2slo`

SP の SLO サブレットを定義します。

SLO 確認 URL

`http://www.idp.demo:80/idpsample/SLOConfirm.jsp.`

5. 他のフィールドはデフォルトのままにします。
6. [サブミット] をクリックします。
7. ポリシー サーバ管理コンソールにログインし、セッション ストアを有効にします。

手順については、「ポリシー サーバ管理ガイド」を参照してください。

SP でのシングル ログアウトの有効化

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。

次の手順に従ってください：

1. 保護されたリソースを持つレلمが永続セッションに対して設定されていることを確認します。
2. Partner IDP.demo Auth Scheme という名前の認証方式に移動します。
3. この認証方式の SAML 2.0 設定を変更して [SLO] タブにアクセスします。
4. [SLO] タブで、[HTTP-リダイレクト] を選択します。
残りのフィールドがアクティブになります。
5. フィールドには以下のように入力します。

SLO ロケーション URL

`http://www.idp.demo:80/affwebservices/public/saml2slo`

SLO 確認 URL

`http://www.sp.demo:81/spsample/SLOConfirm.jsp`

6. 他のすべてのフィールドは、デフォルト値のままにしておきます。
7. ポリシー サーバ管理コンソールにログインし、セッション ストアを有効にします。

手順については、「ポリシー サーバ管理ガイド」を参照してください。

シングル ログアウトのテスト

シングル ログアウトをテストするには、独自の Web ページを使用します。SP で開始されるシングル サインオンをテストするための HTML ページには、シングル ログアウト サービスへのハードコードされたリンクが含まれることを確認してください。

シングル サインオンのテストに成功した後に、シングル ログアウトをテストできます。作成したランディング・ページから、HTTP リダイレクト バインディングを使用してシングル ログアウト URL にブラウザを誘導するリンクをクリックします。

SAML 2.0 Artifact シングル サインオンの設定

Artifact シングル サインオンを設定するタスクをアイデンティティ プロバイダおよびサービス プロバイダで実行します。

アイデンティティ プロバイダでの必須タスクは以下のとおりです。

- [IdP セッション ストアのセットアップ](#) (P. 71)
- [IdP Web サーバに対する SSL の有効化](#) (P. 72)
- Artifact 解決サービス ポリシーへのアクセスの許可
- [IdP でアサーションを保存するための永続セッションの有効化](#) (P. 74)
- IdP での Artifact バインドの選択

サービス プロバイダでの必須タスクは以下のとおりです。

- SP の証明書データ ストアへの CA 証明書の追加
- SP での Artifact バインドの有効化
- [Artifact シングル サインオンのテスト](#) (P. 78)

Artifact シングル サインオンを使用するための IdP セッション ストアのセットアップ

Artifact バインドを使用するには、IdP でセッション ストアのセットアップおよび有効化を行います。Artifact バインドを使用する場合、アサーションが Artifact で取得される前に、アサーションを保存するためのセッション ストアが必要です。

セッション ストアを有効にする方法

1. ODBC データベースをインストールし、セッション ストアとして機能するように設定します。この展開の場合、Microsoft SQL Server を使用しています。
手順については、「ポリシー サーバインストールガイド」を参照してください。
2. ポリシー サーバ管理コンソールを開きます。
3. [データ] タブを選択します。
4. [データベース] ドロップダウンリストから、[セッション サーバ]を選択します。

5. 以下のフィールドに値を入力します。

データソース情報

SiteMinder セッション データ ソース

ユーザ名

admin

パスワード

dbpassword

パスワードの確認入力

dbpassword

最大接続数

16 (デフォルト)

6. [セッション サーバの有効化] チェック ボックスを選択します。
7. [OK] をクリックして設定を保存します。
8. [Artifact シングル サインオンを使用するために IdP Web サーバに対して SSL を有効化](#) (P. 72) します。

IdP Web サーバで Artifact シングル サインオンを使用するための SSL の有効化

Web エージェント オプション パックがインストールされている Web サーバに対して SSL を有効にします。SSL の有効化により、アサーションが渡されるバック チャネルが安全であることが確認されます。

次の手順に従ってください:

1. サーバサイド証明書リクエストを作成します。
2. 認証機関にサーバサイド証明書の署名を依頼します。
3. Web サーバ設定にサーバサイド証明書を指定します。
サンプル ネットワークで使用する IIS Web サーバの場合、IIS 認証ウィザードが使用されます。
4. IdP でアサーションを保存するための永続セッションを有効化します。

Artifact 解決サービスの FWS ポリシーへのアクセス許可

Web エージェント オプションパックにより、フェデレーション Web サービス (FWS) アプリケーションがインストールされます。Web エージェントと同じ IdP に対応したポリシー サーバをインストールすると、FWS アプリケーション内のサービスに関するいくつかのポリシーが自動的に作成されます。これらのポリシーの 1 つにより、HTTP-Artifact シングル サインオンの Artifact 解決サービスが保護されます。

この Artifact 解決ポリシーの保護を適用することにより、Artifact 解決サービスにアクセスできる依存パートナーを指定します。

IdP で次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [エージェント] - [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. [名前] フィールドに **idp-webagent** を入力します (これは、このサンプル展開におけるエージェントの名前です)。[サブミット] をクリックします。
5. [インフラストラクチャ] - [エージェント グループ] を選択します。
6. FederationWebServicesAgentGroup エントリを選択します。
[エージェント グループ] ダイアログ ボックスが表示されます。
7. [追加/削除] をクリックします。[エージェント グループのメンバ] ダイアログ ボックスが表示されます。
8. **idp-webagent** を [使用可能なメンバー] リストから [メンバーの選択] リストに移動させます。
9. [OK] ボタンをクリックして、[エージェント グループ] ダイアログに戻ります。

10. [サブミット] クリックした後、[閉じる] クリックしてメインページに戻ります。
11. フェデレーション サンプル パートナー アフィリエイト ドメイン内のすべてのサービス プロバイダが **Artifact** 解決サービスにアクセスできるように指定します。以下の手順を実行します。
 - a. [インフラストラクチャ] - [ポリシー] - [ドメイン] - [ドメイン] を選択します。
 - b. [FederationWebServicesDomain] を選択します。
 - c. [ポリシー] タブを選択した後、[変更] をクリックします。
 - d. [ポリシー リスト] から、
[SAML2FWSArtifactResolutionServicePolicy] エントリの右側にある編集の矢印をクリックします。
[ポリシー] ダイアログ ボックスが表示されます。
 - e. [ユーザ] タブから、**SAML2FederationCustomUserStore** ディレクトリの [メンバーの追加] を選択します。
[ユーザ/グループ] ダイアログ ボックスが表示されます。
 - f. リストから [affiliate:FederationSamplePartners] を選択し、[OK] をクリックします。
 - g. [サブミット] をクリックして変更を終了します。

Artifact 解決サービスを保護するポリシーが適用されました。

IdP でアサーションを保存するための永続セッションの有効化

保護された認証 URL が含まれるレルムの永続セッションを有効にします（「認証 URL の保護」）。永続セッションは、**SAML Artifact** バインドのアサーションを保存するために必要とされます。

認証 URL を保護したときに永続セッションを有効化していない場合は、ここで有効化します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [ドメイン] - [ドメイン] の順に移動します。
3. 認証 URL のドメインにアクセスします。レルム。

4. [レルム]ページから、認証 URL を持つレルムを選択して変更します。
5. [セッション] セクションで、[永続] を選択します。
6. [OK] をクリックします。
7. IdP で Artifact バインドを選択します。

IdP での Artifact バインドの選択

Artifact シングルサインオンの場合、Artifact バインドを有効にします。

次の手順に従ってください:

1. [フェデレーション] - [レガシー フェデレーション] - [SAML サービスプロバイダ] をクリックします。
2. このパートナーの設定にアクセスするには、sp.demo を選択します。
3. [変更] をクリックし、[SAML プロファイル] ページを選択します。
4. 以下のフィールドに値を入力します。

オーディエンス

sp.demo

この値は、サービス プロバイダの値と一致する必要があります。

アサーション コンシューマ サービス

`http://www.sp.demo:81/affwebservices/public/
saml2assertionconsumer`

認証レベル、有効期間、認証コンテキスト クラス参照

デフォルトを使用

テスト環境では、ポリシー サーバのトレース ログに以下のメッセージが存在する場合、有効期間を 60（デフォルト）より大きな値に増加させることができます。

Assertion rejected (_b6717b8c00a5c32838208078738c05ce6237) – current time (Fri Sep 09 17:28:33 EDT 2005) is after SessionNotOnOrAfter time (Fri Sep 09 17:28:20 EDT 2005)

5. [Artifact バインディング] セクションで [HTTP-Artifact] を選択します。

6. [Artifact エンコーディング] フィールドでは URL を選択します。
Artifact は、URL にエンコードされたクエリ文字列に追加されます。
7. [属性] ページに移動します。
8. [バックチャネル] セクションで、以下のフィールドに入力します。
パスワード
smfederation
パスワードの確認入力
smfederation
アイデンティティ プロバイダは、バックチャネルでの安全な通信にこのパスワードを使用します。
9. [サブミット] をクリックします。
10. SP で CA 証明書を証明書データ ストアに追加します。

SP での SSL バックチャネルの CA 証明書の追加

Artifact シングル サインオンにおいて、Basic over SSL が Artifact 解決サービスを保護する認証方式である場合は、サービス プロバイダの証明書データ ストアに証明書を追加します。

証明書データ ストアには、サービス プロバイダとアイデンティティ プロバイダの間の SSL 接続を確立する認証機関証明書が保持されます。証明書は、アサーションが送信されるバックチャネルのセキュリティを保護します。Artifact 解決サービスを保護し、バックチャネルのセキュリティを保護することで、信頼できる機関により SSL 接続のセキュリティが保護されることをサービス プロバイダが認識できるようにします。

一般的なルート CA および中間 CA のセットが SiteMinder に付属しています。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。

この展開の場合、エイリアスは sampleAppCertCA、CA の証明書は docCA.crt です。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [認証機関] をクリックします。

3. [新規インポート] 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. [ファイルの選択] 手順では、**docCA.crt** を選択します。
ウィザードの [パスワード] 手順がスキップされます。
5. [エントリの選択] 手順で、[エイリアス] 列に **sampleAppCertCA** と入力します。
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。
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。
7. SP において **SAML 認証** に対する **Artifact バインド** を有効化します。

重要: システムが使用する他の証明書の信頼チェーンに含まれている CA 証明書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。

サービス プロバイダでの Artifact バインドの有効化

サービス プロバイダで **SAML 認証** 方式に対応するシングル サインオン バインドを設定します。設定により、サービス プロバイダ通信方法がアイデンティティ プロバイダに指定されます。

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] - [認証方式] をクリックします。
2. [Partner IDP.demo Auth Scheme] を選択します。この認証方式は、ベシック設定用に作成した認証方式です。
3. [変更] - [SAML 2.0 設定] - [SSO] タブを選択します。
4. [解決サービス] フィールドに次に値を入力します。

<https://www.idp.demo:443/affwebservices/saml2artifactresolution>

5. [暗号化 & 署名] ページに移動します。
6. [バックチャネル] セクションで、以下のフィールドに入力します。

認証

基本

SP 名

sp.demo

パスワード

smfederation

パスワードの確認入力

smfederation

パスワードは、アイデンティティ プロバイダにおけるパスワードと一致する必要があります。このパスワードにより、アイデンティティ プロバイダにおいて **Artifact** 解決サービスまでのバックチャネル全域の安全なアクセスが可能になります。

7. [OK] をクリックします。
8. [SP 開始の Artifact シングル サインオンをテスト](#) (P. 78) します。

Artifact シングル サインオンのテスト

自分の Web ページを使用して、SiteMinder 対 SiteMinder のネットワークでシングル サインオンをテストします。

独自の HTML ページには、認証リクエスト サービスへのハードコードされたリンクが含まれる必要があります。この展開では、Artifact バインドのリンクは以下のとおりです。

```
http://<server:port>/affwebservices/public/saml2authnrequest?ProviderID=
IdP_ID&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

server:port

Web エージェント オプション パックがインストールされてた SP でのサーバの名前およびポートを定義します。

IdP_ID

プロバイダ ID を定義します。

この展開では、リンクは以下のとおりです。

```
http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=
idp.demo&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
```

このリンクを含めた HTML ソース ファイルは以下の例のようになります。

```
<a
href="http://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderI
D=
idp.demo&ProtocolBinding=urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
Link for ARTIFACT Single Sign-on</a>
```

認証リクエスト サービスは、ユーザの認証コンテキストを取得するためのリンクで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にユーザを戻します。

注: 認証リクエストリンク内の **ProviderID** は、SP において SAML 認証方式の [IdP ID] フィールドの値に一致する必要があります。[IdP ID] フィールドは、[認証方式プロパティ] ダイアログボックスの [方式のセットアップ] タブにあります。

次は、[SP で開始されたシングルサインオンのテスト](#) (P. 67) の手順に従います。

アサーションへの属性の追加

ユーザストアレコードから SAML アサーションに属性を追加して、ユーザを識別することができます。属性は、ターゲットリソースへのアクセスをリクエストしているユーザに対応するアイデンティティプロバイダのユーザストアに存在する必要があります。

この展開の場合、ユーザレコードで **user1** の **givenname** を表す属性を追加します。

アイデンティティプロバイダで次の手順に従ってください:

1. 管理 UI にログインします。
2. [フェデレーション] - [レガシーフェデレーション] - [SAML サービスプロバイダ] をクリックします。
3. **sm.demo** を選択します。

4. [変更] をクリックし、[属性] ページに移動します。
5. [属性] セクションの [追加] をクリックします。
6. [属性の追加] ダイアログ ボックスで、以下のフィールドに入力します。

属性タイプ

未指定 (デフォルト)

属性の種類

ユーザ属性

変数名

firstname

属性名

givenname

givenname は user1 のプロファイル内の属性です。

7. [OK] をクリックして変更を保存し、[属性] ページに戻ります。
8. [サブミット] をクリックします。

デジタル署名と検証の設定

SAML 2.0 POST シングルサインオンの場合、アイデンティティプロバイダは SAML レスポンスに署名する必要があります。アイデンティティプロバイダでの設定タスクにより、デジタル署名が有効化されます。サービスプロバイダでの設定タスクにより、署名の検証が有効化されます。

重要: 実稼働環境においては、署名処理は必須セキュリティ要件です。

- [アイデンティティプロバイダでの署名の有効化](#) (P. 81)
- [サービスプロバイダでの署名の検証の有効化](#) (P. 82)

アイデンティティプロバイダでの署名の有効化

POST バインドの SAML アサーションを署名するキーおよび証明書は、証明書データストアに保存されます。SAML レスポンスの署名が必要であるため、アイデンティティプロバイダの証明書データストアには該当するキー/証明書ペアが含まれている必要があります。

サンプルアプリケーションが自動的にインストールするキー/証明書ペアを使用して、サンプルアプリケーションを展開します。新しいキー/証明書ペアをインポートする場合は、以下の手順を実行します。

秘密キー/証明書ペアをインポートする方法

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [信頼された証明書] - [秘密キー] の順に移動します。
3. 秘密キー/証明書 `idp.demo` を証明書データストアにインポートします。
`idp.demo` は SAML レスポンスに署名します。
4. [フェデレーション] - [レガシー フェデレーション] - [SAML サービスプロバイダ] を選択します。
5. サービスプロバイダ `sp.demo` を選択した後、[暗号化 & 署名] タブに移動します。
6. [署名の処理を無効にする] チェックボックスをオフにします。このチェックボックスの選択解除は、署名処理が有効になることを意味します。
7. 以下のフィールドに値を入力します。

署名エイリアス

秘密キー/証明書ペアのインポート時に指定したエイリアスを入力します。

署名アルゴリズム

RSASh1withRSA (デフォルト)

POST 署名オプション

アサーションの署名 (デフォルト)

8. [サブミット] をクリックします。

サービス プロバイダでの署名の検証の有効化

POST シングル サインオンの場合、アイデンティティ プロバイダは SAML アサーションにデジタル署名する必要があります。そのため、サービス プロバイダが署名を検証する必要があります。

デジタル署名を検証する方法

- 証明書（公開キー）を証明書データ ストアにインポートします。
- 発行者の DN および証明書のシリアル番号を指定します。

公開キーをインポートする方法

1. 管理 UI で、[インフラストラクチャ] - [X509 証明書管理] - [信頼された証明書] - [秘密キー] の順に移動します。
2. 公開キー/証明書ペアを証明書データ ストアに追加します。この展開の場合、証明書は `post-cert.crt` です。
キー/証明書ペアがデータ ストアに追加されました。
3. [サブミット] をクリックします。
4. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
5. SAML 2.0 認証方式 (Partner IdP.demo Auth Scheme) を選択します。
6. [暗号化 & 署名] タブを選択します。
7. [D-Sign 情報] セクションで、[署名の処理を無効にする] チェックボックスをオフにして、署名の処理を有効にします。
8. 以下のフィールド値を指定します。

発行者 DN

CN=Certificate Manager,OU=IAM,O=CA.COM

シリアル番号

008D 8B6A D18C 46D8 5B

D-Sig 情報を使用して、サービス プロバイダが SAML レスポンスの署名を確認できます。[発行者 DN] と [シリアル番号] の値は、サービス プロバイダの証明書データ ストアに保存された証明書から得られます。

9. [OK] をクリックします。
以上で、検証設定が完了しました。
10. POST シングル サインオンをテストします。

アサーションの暗号化と復号

セキュリティ強化のために、アサーションを暗号化できます。暗号化は、基本的なシングル サインオン ネットワークを設定した後に実行可能なオプションのタスクです。

アイデンティティ プロバイダは、サービス プロバイダがアサーションの復号に使用する秘密キー/証明書ペアに対応した証明書を持つアサーションを暗号化します。

設定タスクは、アイデンティティ プロバイダとサービス プロバイダで実行可能です。

- [アイデンティティ プロバイダでの暗号化の有効化](#) (P. 83)
- [サービス プロバイダでの復号の有効化](#) (P. 84)

IdP でのアサーションの暗号化の有効化

この展開では、sp_encrypt.crt が暗号化される証明書です。

IdP での暗号化を有効にする方法

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [X509 証明書管理] - [信頼された証明書] - [秘密キー] の順に移動します。
3. sp-encrypt.crt 証明書を証明書データ ストアにインポートします。
4. [フェデレーション] - [レガシー フェデレーション] - [SAML サービス プロバイダ] の順に移動します。
5. sp.demo を選択します。
6. [変更] - [SAML 2.0 設定] を選択します。
7. [暗号化 & 署名] タブに移動します。
8. [アサーションを暗号化する] を選択します。

9. [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] のデフォルトをそのまま使用します。

10. [発行者 DN] で、証明書の発行者を入力します。この展開の場合、DN は次のとおりです。

CN=Doc Certificate Authority、OU=Doc、O=CA.COM

注: [発行者 DN] フィールドに入力する値は、証明書データ ストア内の証明書の発行者 DN に一致する必要があります。一致する値を入力したときに検証する DN を確認してください。

11. [シリアル番号] フィールドに、証明書データ ストア内の証明書のシリアル番号を入力します。この展開の場合、値は 00EFF6AFB49925C3F4 です

数値は 16 進数であることが必要です。

12. [OK] をクリックして、変更を保存します。

13. SP で暗号化アサーションを復号します。

SP でのアサーションの復号の有効化

アサーションがアイデンティティ プロバイダで暗号化されている場合、サービス プロバイダは秘密キーおよび対応する証明書を証明書データ ストアに保存している必要があります。

サービス プロバイダは、アサーションを復号するための秘密キー/証明書ペアを保有する限り、アイデンティティ プロバイダからの暗号化されたアサーションを受理します。

注: 暗号化されたアサーションをサービス プロバイダで受理するために [暗号化されたアサーションを要求する] 機能を有効にする必要はありません。

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. [インフラストラクチャ] - [X509 証明書管理] - [信頼された証明書] - [秘密キー] の順に移動します。

3. 秘密キー/証明書ペア `sp-encrypt.crt` を証明書データ ストアに追加します。このペアのエイリアスは `sp1privkey` です。秘密キーのパスワードは `fedsvcs` です。
4. シングル サインオンをテストします。以下のいずれかに進んでください。
 - SAML 2.0 POST シングル サインオンのテスト
 - Artifact シングル サインオンのテスト

第 3 章: SiteMinder Federation のセットアップの概要

このセクションには、以下のトピックが含まれています。

[フェデレーション セットアップの概要 \(P. 87\)](#)

[インストール概要手順の表記規則 \(P. 88\)](#)

[アサーティング パーティ コンポーネントのセットアップ \(P. 90\)](#)

[依存パーティコンポーネントのセットアップ \(P. 101\)](#)

フェデレーション セットアップの概要

この概要では、フェデレーション ネットワークのセットアップについて概説します。

各手順のステップは、アサーションを生成するパーティによるタスクとアサーションを消費するパーティによるタスクによって分割されています。この構成内では、各サイトのポリシー サーバのタスクと **Web** エージェントのタスクによって手順がさらに分割されています。

このガイドでは、汎用性を考慮して、フェデレーション関係の両側を区別するためにアサーティング パーティと依存パーティという用語を使用しています。

アサーションを生成するパーティをアサーティング パーティと呼びます。以下のパートナーのいずれもがアサーティング パーティとして機能できます。

- SAML 1.x プロデューサ
- SAML 2.0 アイデンティティ プロバイダ
- WS-フェデレーション アカウント パートナー

認証の目的でアサーションを消費するパーティを依存パーティと呼びます。以下のパートナーのいずれもが依存パーティとして機能できます。

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

注: すべてのインストール タスクを先に実行してから、管理 UI を使用してソフトウェア設定を実行することができます。

これらの手順は、最新の SiteMinder リリースに対応しています。特定リリースのバージョン情報については、プラットフォーム マトリックスを参照してください。

プラットフォーム サポート マトリックスにアクセスする方法

1. [テクニカル サポート サイト](#)にログオンします。
2. プラットフォーム サポート マトリックスを検索します。

以下の情報に注意してください。

- この製品は、同じ Cookie ドメインを使用する 2 つのシステム間のフェデレーションをサポートしていません。
- レガシー フェデレーションは、コアの SiteMinder とは別にライセンスされるアイテムです。

インストール概要手順の表記規則

インストールおよび設定の手順では、以下の変数が使用されています。

web_agent_home

Web エージェントのインストール場所を指定します。

policy_server_home

ポリシー サーバのインストール場所を指定します。

web_server_home

Web サーバのインストール場所を示します。

fqhn

完全修飾ホスト名を指定します。

port_number

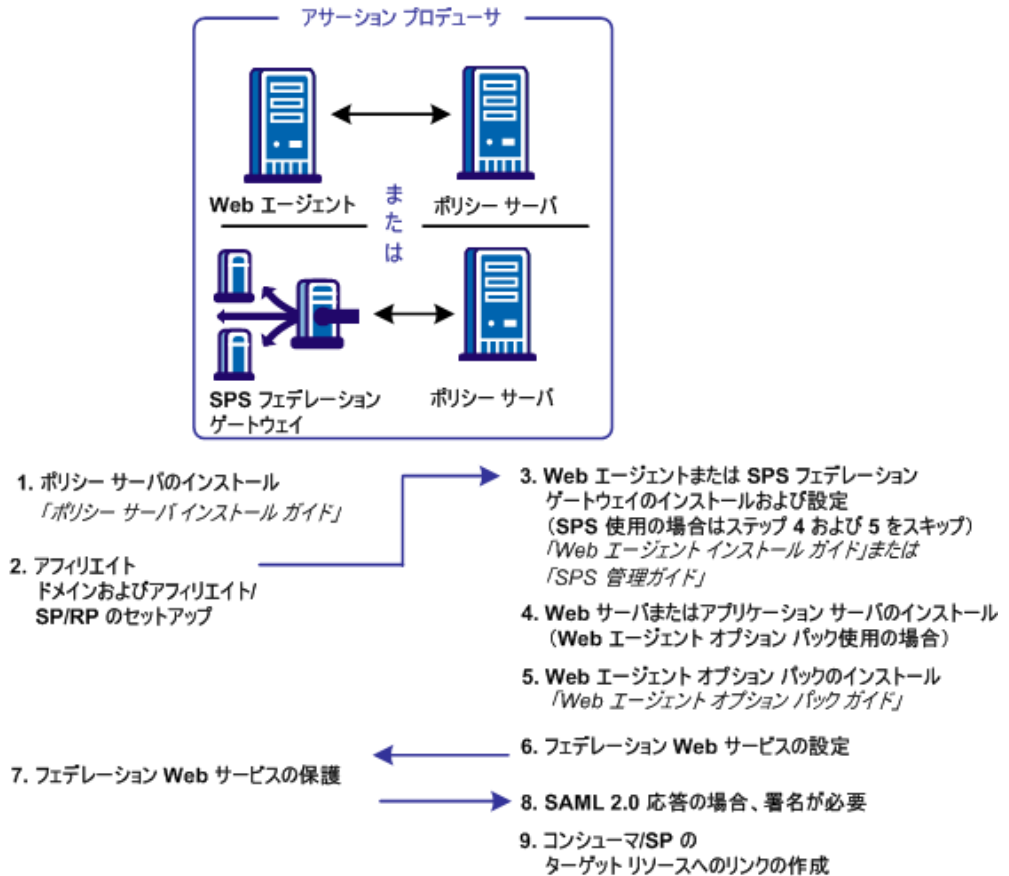
サーバのポート番号を指定します。

sps_home

CA SiteMinder for Secure Proxy Server のインストール場所を指定します。

アサーティング パーティ コンポーネントのセットアップ

次の図は、SAML 1.x プロデューサ、SAML 2.0 アイデンティティ プロバイダまたは WS フェデレーション アカウント パートナーのセットアップを示しています。



特に記載がある場合を除き、設定手順については、このガイドを参照してください。

注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure* プロキシ サーバ管理ガイド」(Secure Proxy Server Administration Guide)を参照してください。

アサーティング パーティ ポリシー サーバのインストール

アサーティング パーティでのセットアップは以下のとおりです。

1. ポリシー サーバをインストールします。
「ポリシー サーバインストール ガイド」を参照してください。
2. **Artifact** シングル サインオン 専用セッション ストアおよびそのデータベースをセットアップします。

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。

セッション ストアは、**Artifact** シングル サインオンのみに必要となります。セッション サーバでは、アサーションが取得される前にこれを保存するためです。
3. ポリシー サーバによって使用されるポリシー ストアをセットアップします。手順については、「**ポリシー サーバインストールガイド**」を参照してください。
4. 「**ポリシー サーバ設定ガイド**」の説明に従って、ユーザ ディレクトリをセットアップします。

このユーザ ディレクトリには、アサーションが生成される対象のユーザが含まれる必要があります。
5. (オプション) ポリシー サーバがアサーティング パーティと依存パーティの間の通信を確認するには、エラーおよびトレース ロギングを有効にします。

アフィリエイトドメインのセットアップおよびこれらのドメインへのサイトの追加

フェデレーション **Web** サービスをセットアップする前に、アフィリエイトドメインを確立し、アサーションを消費するサイトをアフィリエイトドメインに追加します。アフィリエイトドメインは、アサーションを生成するサイトのパートナーを識別します。

アサーティング パーティで以下のように操作します。

1. 管理 UI にアクセスします。
2. アフィリエイトドメインを作成します。
3. アサーティング パーティ (プロデューサ、IdP、AP) がアサーションを生成するユーザ用のユーザ ストアを追加します。

4. 依存パーティ（コンシューマ、SP、RP）ごとにオブジェクトをアフィリエイト ドメインに追加します。

依存パーティとドメインに追加される各オブジェクトとの間に、一対一の対応関係が存在する必要があります。

5. サイトをアフィリエイト ドメインに追加した後、認証 URL が保護されていることを検証します。この検証により、フェデレーション リソースのリクエストを処理する前に、ユーザがアサーティング パーティにセッションを持つことが確認されます。

このタスクを実行するには、以下の手順に従います。

- a. ポリシー ドメインを作成します。
- b. Web エージェントでポリシー ドメインを保護します。Web エージェント オプション パックのあるサーバを保護している Web エージェントを使用します。
- c. レルム、ルール、および認証 URL を保護するポリシーをこのポリシー ドメインに追加します。

詳細情報:

[SiteMinder セッションのないユーザの認証 \(SAML 1.x\)](#) (P. 144)

アサーティング パーティでの Web エージェントまたは SPS フェデレーション ゲートウェイのインストール

Web エージェントは、SiteMinder フェデレーション ネットワークにおける必須コンポーネントです。 **Web** エージェントは **Web** サーバにインストールします。または **SPS** フェデレーション ゲートウェイをインストールする方法もあります (**SPS** フェデレーション ゲートウェイには **Web** エージェントが組み込まれています)。

アサーティング パーティで、以下のコンポーネントをセットアップします。

1. 以下のコンポーネントのいずれかをインストールします。
 - **Web** エージェント
手順については、「**Web** エージェント インストール ガイド」を参照してください。
 - **SPS** フェデレーション ゲートウェイ
手順については、「*Secure Proxy Server Administration Guide*」を参照してください。
2. **Artifact** シングル サインオンを使用するには、**Web** エージェントがインストールされた **Web** サーバまたは **SPS** フェデレーション ゲートウェイがインストールされたシステムで **SSL** を有効にします。

Web エージェント オプション パック(アサーティング パーティ)用のアプリケーション サーバのインストール

Web エージェントおよび **Web** エージェント オプション パックと共に レガシー フェデレーション を実装する場合は、**Web** エージェント オプション パックをインストールします。 このコンポーネントは、**Web** サーバまたはアプリケーション サーバにインストールしてください。

アサーティング パーティで以下の作業を実行します。

1. フェデレーション Web サービス（Web エージェント オプション パックと共にインストールされるアプリケーション）を実行するサーバとして以下のいずれかをインストールします。
 - ServletExec を実行する Web サーバ
 - WebLogic アプリケーション サーバ
 - WebSphere アプリケーション サーバ
 - JBOSS アプリケーション サーバ
 - Tomcat アプリケーション サーバ
2. フェデレーション Web サービスをこれらのシステムに展開します。
3. Artifact シングルサインオンを使用するには、Web エージェント オプション パックがインストールされた Web サーバで SSL を有効にします。

アサーティング パーティ Web エージェント オプション パックのインストール

Web エージェント オプション パックは、SiteMinder レガシー フェデレーションの必須コンポーネントであるフェデレーション Web サービス アプリケーションを提供します。

アサーティング パーティで以下の作業を実行します。

1. Web エージェント オプション パックをインストールします。

手順については、「[Web エージェント オプション パック ガイド](#)」（Web Agent Option Pack Guide）を参照してください。
2. JDK がインストールされていることを確認します。Web エージェント オプション パックでは JDK が必要となります。

サポートされる JDK のバージョンについては、[テクニカルサポート サイト](#)にログオンし、SiteMinder プラットフォーム サポート マトリックスで該当リリースを検索してください。

注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「[Secure プロキシ サーバ管理ガイド](#)」（Secure Proxy Server Administration Guide）を参照してください。

フェデレーション Web サービス(アサーティング パーティ)の設定

フェデレーション Web サービス アプリケーションは、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 が設定されたサーバにインストールされます。

アサーティング パーティでフェデレーション Web サービスを設定する方法

1. サポートされるアプリケーション サーバのいずれかで Web エージェント オプション パック を使用するよう設定します。Web エージェント オプション パック の展開手順を参照してください。

SPS フェデレーション ゲートウェイ には、フェデレーション Web サービスがすでに展開されています。

2. `AffWebServices.properties` ファイル内の `AgentConfigLocation` パラメータが `WebAgent.conf` ファイルへの完全パスに設定されることを確認します。構文が正しく、パスがファイル内で 1 行に表示されていることを確認します。

`AffWebServices.properties` ファイルには、フェデレーション Web サービスの初期化パラメータが含まれます。このファイルは、以下のディレクトリのいずれかに配置されています。

- `web_agent_home/affwebservices/WEB-INF/classes`
- `sps_home/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes`

web_agent_home

Web エージェントのインストール場所を表します。

sps_home

SPS フェデレーション ゲートウェイのインストール場所を表します。

3. フェデレーション Web サービス アプリケーションに関するエラーおよびトレース ロギングを有効にします。`LoggerConfig.properties` ファイルでロギングを有効にします。ログを使用して、アサーティング パーティと依存パーティの間の通信を確認できます。
 - エラー ロギングは、`affwebserv.log` ファイル (デフォルトのエラー ログ ファイル) に記録されます。
 - トレース ロギングは、`FWSTrace.log` ファイル (デフォルト トレース ログ) に記録されます。

4. Web ブラウザを開き以下のリンクを入力することにより、フェデレーション Web サービスをテストします。

`http://fqhn:port_number/affwebservices/assertionretriever`

fqhn

完全修飾ホスト名を定義します。

port_number

フェデレーション Web サービス アプリケーションがインストールされているサーバのポート番号を定義します。

例：

`http://myhost.ca.com:81/affwebservices/assertionretriever`

フェデレーション Web サービスが正しく作動している場合、次のメッセージが表示されます。

アサーション取得サービスは正常に初期化されました。(Assertion Retrieval Service has been successfully initialized.)

リクエストされたサーブレットは、HTTP POST リクエストのみを受け付けます。(The requested servlet accepts only HTTP POST requests.)

このメッセージは、フェデレーション Web サービスがデータ アクティビティをリスニングしていることを示します。フェデレーション Web サービスが正しく作動していない場合、アサーション検索サービスが失敗したというメッセージを受信します。テストが失敗する場合は、フェデレーション Web サービス ログを確認してください。

フェデレーション Web サービスへのアクセスの許可 (アサーティング パーティ)

ポリシー サーバのインストール時、SiteMinder はフェデレーション Web サービス (FWS) アプリケーション用のポリシーを作成します。FWS アプリケーションは Web エージェント オプション パックを使用してインストールされます。一部のフェデレーション機能では、依存パーティが保護された FWS サービスへのアクセス許可を必要とします。依存パートナーをポリシーに追加するタスクは、アサーティングパーティのみで実行します。

たとえば、シングルサインオンの HTTP-Artifact バインドの場合、ポリシーは、SiteMinder がアサーションを取得するサービスを保護します。SiteMinder が特定の依存パートナーのアサーションを取得するには、そのパートナーがサービスを保護するポリシーにユーザとして追加されている必要があります。

フェデレーション パートナリシップに対して設定された機能に適用される [特定の FWS ポリシーへのアクセス権を付与](#) (P. 135) します。

SAML POST レスポンスの署名の有効化

SAML POST レスポンスの署名は、SAML 仕様書要件の 1 つです。SAML POST レスポンスを署名するには、アサーティング パーティの証明書データ ストアに秘密キーと証明書を追加します。

キーと証明書をデータ ストアにインポート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。

ターゲット リソースへのリンクの作成(オプション)

以下のいずれかに移動します。

- [SAML 1.x シングルサインオン用のリンク](#) (P. 97)
- [アイデンティティ プロバイダでの SAML 2.0 シングルサインオン用のリンク](#) (P. 99)
- [WS フェデレーション シングルサインオンを開始するリンク](#) (P. 100)

プロデューサでの SAML 1.x シングル サインオンの開始

SAML 1.x 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、サイト間転送 URL にアクセスする必要があります。これにより、プロデューサ側 Web エージェントにリクエストが送信されます。その後、ユーザはコンシューマ サイトにリダイレクトされます。

プロデューサでユーザが選択するリンクには、所定のクエリ パラメータが含まれる必要があります。これらのパラメータは、プロデューサ Web エージェントに対する HTTP GET リクエストの一部です。

SAML Artifact プロファイルの場合、サイト間転送 URL の構文は以下のとおりです。

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSERTIONREF=
QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url?query_paramete
r_name%3Dquery_parameter_value%26query_parameter_name%3Dquery_parameter_val
ue&SMCONSUMERURL=http://consumer_site/affwebservices/public/samlcc&AUTHREQUIR
EMENT=2
```

producer_site

フェデレーション ネットワークにインストールされているコンポーネントに応じて、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するシステムのサーバおよびポート番号を指定します。

consumer_site

フェデレーション ネットワークにインストールされているコンポーネントに応じて、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するシステムのサーバおよびポート番号を指定します。

SAML POST プロファイルの場合、サイト間転送 URL の構文は以下のとおりです。

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSERTIONREF=
QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url
```

producer_site

フェデレーション ネットワークにインストールされているコンポーネントに応じて、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するシステムのサーバおよびポート番号を指定します。

consumer_site

フェデレーション ネットワークにインストールされているコンポーネントに応じて、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するシステムのサーバおよびポート番号を指定します。

注: SAML POST プロファイルでは SMCONSUMERURL パラメータおよび AUTHREQUIREMENT パラメータは使用されません。ただし、これらのパラメータのいずれかをサイト間転送 URL に含める場合には、もう一方のパラメータも含めてください。

詳細情報:

[コンシューマ リソースへのリンクの作成 \(SAML 1.x\)](#) (P. 174)

アイデンティティプロバイダでの SAML 2.0 シングル サインオンの開始

ユーザがサービス プロバイダ (POST または Artifact バインド) に移動する前にアイデンティティプロバイダにアクセスする場合は、アイデンティティプロバイダで未承認応答を開始します。未承認応答を開始するには、フェデレーション Web サービス アプリケーションおよびアサーションジェネレータが HTTP Get リクエストおよびクエリ パラメータを受け付ける必要があります。このクエリ パラメータは、IdP がレスポンスを生成する対象のサービス プロバイダ ID を示します。

SAML 2.0 Artifact プロファイルまたは POST プロファイルの場合、リンクの構文は以下のとおりです。

`http://IdP_server:port/affwebservices/public/saml2sso?SPID=SP_ID`

`idp_server:port`

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する Web サーバおよびポートを識別します。

`SP_ID`

サービス プロバイダ ID の値。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。

どのバインドが有効化されるかに応じて、[ProtocolBinding クエリ パラメータ](#) (P. 261)をこのリンクに追加します。

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。

サービス プロバイダでシングル サインオンを開始することもできます。

詳細情報:

[IdP によって使用される未承認応答のクエリ パラメータ](#) (P. 261)

アカウント パートナーでの WS フェデレーション シングル サインオンの開始

WS フェデレーション シングル サインオンを開始するには、ユーザがハードコードされた **HTML** リンクを持つページをクリックします。この **HTML** リンクにより、そのユーザのブラウザがアカウント パートナーでのシングル サインオン サービスに導かれます。その後、アカウント パートナーによりユーザがリソース パートナーにリダイレクトされます。

シングル サインオンを開始するリンクは、任意のサイトに含めることができますが、常に最初にアカウント パートナーにユーザを導く必要があります。

このリンクの構文は、次のとおりです。

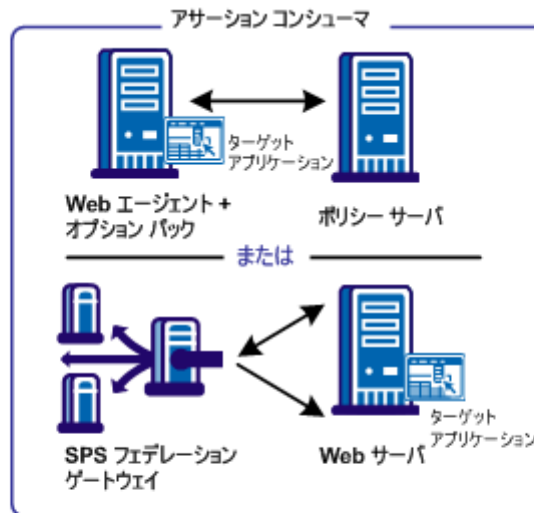
```
https://AP:port/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=RP_ID
```

ap_server:port

アカウント 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 オブション 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。

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。

依存パーティコンポーネントのセットアップ



1. ポリシー サーバのインストール
「ポリシー サーバインストール ガイド」
2. プロデューサ/IdP/AP ごとの SAML 認証方式の設定
3. ターゲットリソースを保護するレルム、ルール、ポリシーの定義。
4. Web エージェントまたは SPS フェデレーション ゲートウェイのインストールおよび設定
「Web エージェント インストール ガイド」または「SPS 管理ガイド」
(SPS 使用の場合はステップ 6 および 7 をスキップ)
5. フェデレーション Web サービス用の Web サーバまたはアプリケーション サーバのインストール
6. Web エージェント オプション パックのインストール
「Web エージェント オプション パック ガイド」
7. フェデレーション Web サービスの設定
8. フェデレーション Web サービスの保護
9. アーティファクト SSO の場合、証明書 データ ストアのセットアップ

特に記載がある場合を除き、設定手順については、このガイドを参照してください。

依存パーティでポリシー サーバと Web エージェントをセットアップするための手順の多くは、アサーティングパーティでのセットアップ手順に類似していますが、以下の点に違いがあります。

- コンシューマ、サービス プロバイダ、リソース パートナーの設定は行いません
- ポリシー サーバで SAML または WS フェデレーション認証方式を設定します

次の図は、**SAML 1.x** コンシューマ、**SAML 2.0** サービス プロバイダ、**WS** フェデレーション リソース パートナーでの必要なタスクを示しています。

注: この手順では、ターゲット リソースが依存パーティ Web サイトに存在すると仮定しています。

依存パーティ ポリシー サーバのインストール

依存パーティ サイトでポリシー サーバをインストールします。ポリシー サーバは、フェデレーション認証方法やアサーション ジェネレータなどの機能を提供します。

詳細については、「[ポリシー サーバインストール ガイド](#)」および「[ポリシー サーバ設定ガイド](#)」を参照してください。

依存パーティで以下の手順を実行します。

1. ポリシー サーバをインストールします。
2. ポリシー ストアをセットアップします。

重要: 新しいポリシー ストアを初期化すると、ポリシー サーバインストーラが自動的にアフィリエイト オブジェクトを **ampolicy.smdif** ファイルにインポートします。これらのオブジェクトは、連携に必要となります。既存のポリシー ストアを使用する場合は、アフィリエイト オブジェクトを手動でインポートします。インポートが成功したことを確認するには、管理 UI にログインし、[ポリシー] - [ドメイン] - [ドメイン] の順に移動します。インポートが成功している場合は、**FederationWebServices** ドメイン オブジェクトがリストに表示されています。

3. ユーザ ストアをセットアップし、ターゲット リソースへのアクセスが許可されたユーザを追加します。

SAML 認証方式または WS フェデレーション認証方式の設定

依存パーティのポリシー サーバで、各アサーティングパーティの認証方式（Artifact、POST プロファイル、SAML 2.0、WS フェデレーション）を設定します。

重要: 認証方式に対して指定するパートナーの名前は、アサーティングパーティで指定する依存パーティの名前に一致する必要があります。

具体的な内容は次のとおりです。

- SAML 1.x 認証方式の場合、方式設定の [アフィリエイト名] フィールドは、プロデューサ サイトにおけるアフィリエイト オブジェクトの [アフィリエイト名] に一致する必要があります。
- SAML 2.0 の場合、相当するフィールドは [SP ID] です。このフィールドは、アイデンティティ プロバイダの [SP ID] に一致する必要があります。
- WS フェデレーションの場合、方式設定の [リソース パートナー ID] はアカウント パートナーの [リソース パートナー ID] に一致する必要があります。

詳細情報:

[SAML 1.x コンシューマとしての設定](#) (P. 177)

[SAML 2.0 サービス プロバイダの設定](#) (P. 287)

[SiteMinder の WS-フェデレーション リソース パートナーとしての設定](#) (P. 369)

依存パーティでのターゲットリソースの保護

SAML 認証方式または WS フェデレーション認証方式を作成した後、それらの方式を一意的なレールまたは単一のカスタム レールに割り当てます。レールは、ユーザ アクセスに対してアサーションを要求とする依存パーティ ターゲット リソースのコレクションです。依存パーティは、以下のいずれかの方法でターゲット リソースを識別します。

- サイト間転送 URL 内の TARGET 変数 (SAML 1.x)。
- 認証リクエスト URL (SAML 2.0 および WS フェデレーション)。
- 認証方式設定 (SAML 2.0 および WS フェデレーション)。

作成したレールに SAML 認証方式または WS フェデレーション認証方式を割り当てた後、レールのルールを作成し、リソースを保護するポリシーにそのルールを追加します。

Web エージェントまたは SPS フェデレーション ゲートウェイのインストール(依存パーティ)

Web エージェントは、SiteMinder レガシー フェデレーション ネットワークにおける必須コンポーネントです。Web エージェントは Web サーバにインストールしますが、SPS フェデレーション ゲートウェイをインストールする方法もあります (SPS フェデレーション ゲートウェイには Web エージェントが組み込まれています)。

依存パーティで、以下のコンポーネントをセットアップします。

1. 以下のコンポーネントのいずれかをインストールします。
 - Web エージェント
手順については、「Web エージェント インストール ガイド」を参照してください。
 - SPS フェデレーション ゲートウェイ
手順については、「Secure プロキシサーバ管理ガイド」 (Secure Proxy Server Administration Guide) を参照してください。
2. Web エージェントまたは SPS フェデレーション ゲートウェイを設定します。

Web エージェント オプション パック (依存パーティ) 用の Web サーバまたはアプリケーション サーバのインストール

Web エージェントおよび Web エージェント オプション パック (SPS フェデレーション ゲートウェイではない) と共に レガシー フェデレーション を実装する場合は、Web エージェント オプション パックをインストールします。このコンポーネントは、Web サーバまたはアプリケーション サーバにインストールしてください。

依存パーティで以下の作業を実行します。

1. フェデレーション Web サービス (Web エージェント オプション パックと共にインストールされるアプリケーション) を実行するサーバとして以下のいずれかをインストールします。
 - ServletExec を実行する Web サーバ
 - WebLogic アプリケーション サーバ
 - WebSphere アプリケーション サーバ
 - JBOSS アプリケーション サーバ
 - Tomcat アプリケーション サーバ
2. フェデレーション Web サービスをこれらのシステムに展開します。

依存パーティでの Web エージェント オプション パックのインストール

Web エージェント オプション パックは、レガシー フェデレーション の必須コンポーネントであるフェデレーション Web サービス アプリケーションを提供します。

依存パーティで以下の作業を実行します。

1. Web エージェント オプション パックをインストールします。

手順については、「[Web エージェント オプション パック ガイド](#)」
(Web Agent Option Pack Guide) を参照してください。

2. JDK がインストールされことを確認します。Web エージェント オプション パックではこの JDK が必要となります。

必要な JDK のバージョンを調べるには、[テクニカル サポート サイト](#)に移動し、SiteMinder プラットフォーム マトリックスを検索してください。

注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「[Secure プロキシ サーバ管理ガイド](#)」 (Secure Proxy Server Administration Guide) を参照してください。

依存パーティでのフェデレーション Web サービスの設定

以下の手順を使用して、フェデレーション Web サービス アプリケーションをセットアップできます。フェデレーション Web サービス アプリケーションは、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 ゲートウェイが設定されたサーバにインストールされます。

依存パーティでフェデレーション Web サービスを設定する方法

1. サポートされるアプリケーション サーバのいずれかで Web エージェント オプション パックを使用するように設定します。Web エージェント オプション パックの展開手順を参照してください。

SPS フェデレーション ゲートウェイを使用している場合、フェデレーション Web サービス アプリケーションはすでに展開されています。

2. `AffWebServices.properties` ファイル内の `AgentConfigLocation` パラメータに `WebAgent.conf` ファイルへの完全パスに設定します。構文が正しく、パスがファイル内で 1 行に表示されていることを確認します。

`AffWebServices.properties` ファイルには、フェデレーション Web サービスの初期化パラメータが含まれます。このファイルは、以下のディレクトリのいずれかに配置されています。

- `web_agent_home/affwebservices/WEB-INF/classes`
- `sps_home/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes`

`web_agent_home`

Web エージェントのインストール場所を表します。

`sps_home`

SPS フェデレーション ゲートウェイのインストール場所を表します。

3. フェデレーション Web サービス アプリケーションに関するエラーおよびトレースのロギングを有効にします。ロギングは、`LoggerConfig.properties` ファイルで有効化されます。ログを使用して、アサーティングパーティと依存パーティの間の通信を確認できます。
 - エラー ロギングは、`affwebserv.log` ファイル（デフォルトのエラー ログ ファイル）に記録されます。
 - トレース ロギングは、`FWSTrace.log` ファイル（デフォルト トレース ログ）に記録されます。

4. Web ブラウザを開き以下のリンクを入力することにより、フェデレーション Web サービスをテストします。

`http://fqhn:port_number/affwebservices/assertionretriever`

fqhn

完全修飾ホスト名を定義します。

port_number

フェデレーション Web サービス アプリケーションがインストールされているサーバのポート番号を定義します。

例：

`http://myhost.ca.com:81/affwebservices/assertionretriever`

フェデレーション Web サービスが正しく作動している場合、次のメッセージが表示されます。

アサーション取得サービスは正常に初期化されました。(Assertion Retrieval Service has been successfully initialized.)

リクエストされたサーブレットは、HTTP POST リクエストのみを受け付けます。(The requested servlet accepts only HTTP POST requests.)

このメッセージは、フェデレーション Web サービスがデータ アクティビティをリスニングしていることを示します。フェデレーション Web サービスが正しく作動していない場合、アサーション検索サービスが失敗したというメッセージが表示されます。テストが失敗する場合は、フェデレーション Web サービス ログを確認してください。

詳細情報：

[フェデレーション Web サービス \(アサーティング パーティ\) の設定](#) (P. 95)

フェデレーション Web サービスへのアクセスの許可 (アサーティング パーティ)

ポリシー サーバのインストール時、SiteMinder はフェデレーション Web サービス (FWS) アプリケーション用のポリシーを作成します。FWS アプリケーションは Web エージェント オプションパックを使用してインストールされます。一部のフェデレーション機能では、依存パーティが保護された FWS サービスへのアクセス許可を必要とします。依存パートナーをポリシーに追加するタスクは、アサーティングパーティのみで実行します。

たとえば、シングルサインオンの HTTP-Artifact バインドの場合、ポリシーは、SiteMinder がアサーションを取得するサービスを保護します。SiteMinder が特定の依存パートナーのアサーションを取得するには、そのパートナーがサービスを保護するポリシーにユーザとして追加されている必要があります。

フェデレーション パートナリシップに対して設定された機能に適用される [特定の FWS ポリシーへのアクセス権を付与](#) (P. 135) します。

Artifact シングル サインオンの場合の証明書データストアの変更(オプション)

証明書データストアには、PKI 操作（暗号化、復号、署名、検証、クライアント認証など）のキーおよび証明書が保持されています。

Artifact シングル サインオンを実装する場合、アサーティングパーティの証明書データストアには、SSL 接続を確立するための認証機関の証明書が保持されます。この SSL 接続は、依存パーティとアサーティングパーティの間で確立されます。この SSL 接続により、Artifact シングル サインオン用にアサーションが送信されるバックチャネルのセキュリティが保護されます。

一般的なルート CA のセットが証明書データストア内に付属しています。データストアに存在しない Web サーバに対するルート CA を使用する場合は、これらのルート CA をインポートしてください。

証明書データストアに関する詳細については、「ポリシー サーバ設定ガイド」を参照してください。

シングル サインオンを開始するためのリンクの作成(オプション)

SAML 2.0 および WS フェデレーションにおいて、ユーザがアサーティングパーティにアクセスする前に依存パーティにアクセスする場合は、ハードコードされたリンクを作成します。ハードコードされたリンクによって、ユーザがアサーティングパーティにリダイレクトされ、認証コンテキストが取得されます。依存パーティでは、この認証コンテキストを構成する特性を利用して、ユーザがどのように認証されたかを認識できます。

詳細情報:

[SP での SAML 2.0 シングル サインオンの開始 \(オプション\) \(P. 110\)](#)
[リソース パートナーでの WS フェデレーション シングル サインオンの開始 \(P. 111\)](#)

SP での SAML 2.0 シングル サインオンの開始(オプション)

ユーザがアイデンティティ プロバイダにアクセスする前にサービス プロバイダにアクセスする場合は、サービス プロバイダがユーザをアイデンティティ プロバイダにリダイレクトする必要があります。サービス プロバイダにおいて、認証リクエスト サービスへのハードコードされたリンクが含まれる **HTML** ページを作成してください。認証リクエスト サービスでは、認証コンテキストを取得するために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ます。

注: HTML ページは保護されていないレلمに配置される必要があります。

サービス プロバイダでユーザがクリックするハードコードリンクには、所定のクエリ パラメータが含まれる必要があります。これらのパラメータは、認証リクエスト サービスに対する **HTTP GET** リクエストの一部になります。認証リクエスト サービスは、サービス プロバイダのポリシー サーバ上にあります。

SAML 2.0 (Artifact またはプロファイル) の場合、リンクの構文は次のとおりです。

`http://sp_server:port/affwebservices/public/saml2authnrequest?ProviderID=IdP_ID`

`sp_server:port`

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するサービス 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 ID を指定します。

どのバインドが有効化されるかに応じて、このリンクに **ProtocolBinding** クエリ パラメータを追加できます。サービス プロバイダでのリンク設定の詳細については、「**IdP** または **SP** でシングル サインオンを開始するためのリンクのセットアップ」を参照してください。

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。

アイデンティティ プロバイダでリンクを作成することもできます。

リソース パートナーでの WS フェデレーション シングル サインオンの開始

ユーザがアカウント パートナーをアクセスする前にリソース パートナーをアクセスする場合は、リソース パートナーがユーザをアカウント パートナーにリダイレクトする必要があります。認証に使用するアカウント パートナーへのリンクが含まれるサイト選択ページなどの **HTML** ページを作成してください。ユーザがリンクを選択すると、アカウント パートナーのシングル サインオン サービスに導かれます。

注: サイト選択ページは、保護されていないレルムに配置される必要があります。

リソース パートナーでユーザがクリックするハードコードリンクには、所定のクエリ パラメータが含まれる必要があります。これらのパラメータは、アカウント パートナーのポリシー サーバのシングル サインオン サービスに対する **HTTP GET** リクエストの一部です。

このリンクの構文は、次のとおりです。

https://host:port/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=RP_ID

host:port

シングル サインオン サービスが配置されたサーバおよびポート番号を示します。

RP_ID

リソース パートナー ID を指定します

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。

第 4 章: SAML 1.x アサーション ジェネレータ ファイルの設定

プロデューサのポリシー サーバには、アサーション ジェネレータという名前のコンポーネントが含まれます。SAML 1.x の場合のみ、アサーション ジェネレータがアサーションを生成するために

AMAssertionGenerator.properties ファイルが必要とされます。このプロパティ ファイルには、コメント付きの手順も含まれます。ファイル内での設定に関する詳細について参照できます。

このファイルは、次の場所にインストールされます。

policy_server_home/config/properties

アサーション ジェネレータは、このファイル内の設定を変更しなくても作動します。ただし、ファイルには、アサーションで使用されるデフォルト値が含まれているため、これらの値を実際のネットワークに合わせて変更する必要があります。

AmAssertionGenerator.properties ファイルへの更新は、ポリシー サーバが再起動された後に取り出されます。

AMAssertionGenerator.properties ファイルを設定する方法

1. *policy_server_home/config/properties* ディレクトリに移動します。
2. テキスト エディタで **AMAssertionGenerator.properties** ファイルを開きます。
3. 以下のパラメータを変更します。

AssertionIssuerID

アサーションを発行するサイトを識別する URL を指定します。

この URL は、SAML 認証方式の [発行者] フィールドに対する値と同じであることが必要です。

SecurityDomain

プロデューサのドメインを識別します（example.com など）。

SourceID

SAML 1.x Artifact プロファイルの場合のみ、Artifact 内でプロデューサを識別するの固有の ID を指定します。詳細については、**エラー! ハイパーリンクの参照に誤りがあります。**で SAML 仕様を参照してください。

重要: このファイルの値は、コンシューマ サイトでの同等の設定に対する値と一致する必要があります。

第 5 章: JVM 用の JVMOptions ファイルの確認

JVMOptions.txt ファイルには、ポリシー サーバが Java 仮想マシンを作成するときに使用する設定が含まれています。Java 仮想マシンは、フェデレーション Web サービスのサポートに使用されます。SAML 1.x、SAML 2.0、WS-Federation でこのファイルが使用されます。

ポリシー サーバアップグレード中に、既存の JVMOptions.txt ファイルの名前が JVMOptions.txt.backup に変更されます。新しい JVMOptions.txt ファイルが作成されます。元のファイルにカスタマイズされたパラメータが含まれていた場合、これらのカスタム パラメータが含まれるよう、作成されたファイルを変更してください。

このファイルは、次の場所にインストールされます。

`policy_server_home/config/`

重要: JVMOptions.txt ファイルを更新する場合、変更を有効にするには、ポリシー サーバを再起動する必要があります。

注:

- 環境によっては、ポリシー サーバの実行中にシステムからログオフすると、ポリシー サーバのサービスが失敗します。この失敗は JVM の問題に起因するものです。失敗を防止するには、JVMOptions.txt ファイル内の固有のコマンド行に `-Xrs` コマンドを追加してください。この Java コマンドにより、Java 仮想マシンによるオペレーティング システム シグナルの使用が削減されます。

このコマンドは大文字と小文字が区別されるため、必ず `X` を大文字にしてください。

- 必要なクラスが欠落しているというエラーが表示された場合には、このファイルの `classpath` 命令を変更してください。JVMOptions.txt ファイルに含まれる設定の詳細については、Java のマニュアルを参照してください。Java コンパイラ命令 `java.endorsed.dirs` は、クラス ロードを制御するために JVMOptions.txt ファイルで使用されます。

第 6 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。

[セッションストアに格納されるフェデレーション データ \(P. 117\)](#)

[セッションストアの有効化 \(P. 119\)](#)

[共有セッションストアを必要とする環境 \(P. 120\)](#)

セッションストアに格納されるフェデレーション データ

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。

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 (POST バインド) 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保証できます。

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。

- 認証セッション変数永続性 (SAML 1.x および SAML 2.0)

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。永続属性モードは、アサーションから抽出した属性をセッションストアに保存するようにポリシー サーバに指定します。その後、属性は HTTP ヘッダ変数として提供できます。

- 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。ソフトウェアでは、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。

アカウント パートナーおよびリソース パートナーでは、永続セッションが必須です。

セッションストアの有効化

セッションストアを有効にすると、SAML Artifact シングル サインオン、シングル ログアウト、WS フェデレーション サインアウト、使い捨てポリシーの使用時にデータが保存されます。

セッションサーバデータベースは、ポリシー サーバのセッションサーバが永続セッションデータを格納する場所です。

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لم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: 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実行することはできません。そのため、ポリシーストアデータベースを使用するオプションは無効化されています。

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

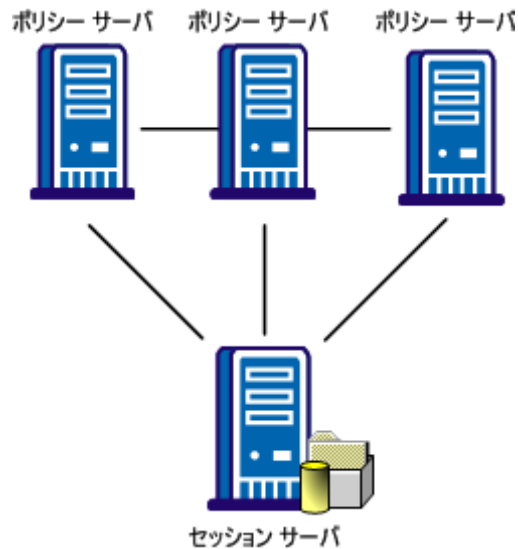
以下の SiteMinder 機能では、SAML アサーションおよびユーザ セッション情報を保存するために共有セッション ストアを必要とします。

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 7 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](#) (P. 123)

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。

SiteMinder 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。

- アサーションの使い捨ての適用。
- アサーティング パーティおよび依存パーティでの接続の保護。
- 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。

アサーションの単一使用条件の設定

SAML 1.x および 2.0 の仕様に準拠して、SiteMinder はアサーションの単一使用を適用できます。単一使用に意図されたアサーションを生成することにより、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定されます。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。

SiteMinder がアサーティング パーティ（プロデューサ/IdP）として機能している場合、アサーションの使い捨てを設定できます。SAML 1.x アフィリエイトの場合、**「DoNotCache 条件の設定」**を選択します。SAML 2.0 IdP の場合、**「OneTimeUse 条件の設定」**設定を選択できます。これらの環境設定によって、SiteMinder は使い捨て条件を示すアサーションに適切な要素を挿入できます。

注: アサーションの使い捨てと SAML 1.x および 2.0 の HTTP-POST シングルサインオン用使い捨てポリシーを混同しないように注意してください。使い捨てポリシーは POST トランザクション専用です。一方、単一使用機能は HTTP-Artifact および HTTP-POST に対応します。

フェデレーション環境間の接続のセキュリティ保護

セキュリティ保護された接続で通信する場合、フェデレーションパートナー間またはパートナーとアプリケーション間で送信される ID 情報が最も厳重に保護されます。

依存パーティとターゲットアプリケーション間の接続のセキュリティ保護

依存パーティからクライアント サイト ターゲット アプリケーションへのデータ伝送のセキュリティを保護します。セキュリティ保護された接続を通信チャネルとして使用することで、セキュリティ攻撃に対する環境の脆弱性が改善されます。

たとえば、アサーションには、依存パーティが抽出してクライアント アプリケーションに送信する属性が含まれることがあります。依存パーティでは、HTTP ヘッダ変数や Cookie を使用して、これらの属性をアプリケーションに渡すことができます。ヘッダや Cookie に保存された属性はクライアント側で上書きできるため、悪意のあるユーザが他のユーザになりますことが可能になります。SSL 接続を使用することで、この種のセキュリティ侵害から環境が保護されます。

ベストプラクティスとして、該当するエージェント設定オブジェクト (ACO) に **UseSecureCookies** パラメータを設定することにより、この脆弱性から保護します。**UseSecureCookies** パラメータを設定すると、「**secure**」フラグが付けられた **Cookie** を生成するようにフェデレーション **Web** サービスに指定されます。このフラグは、**Cookie** が **SSL** 通信チャネルのみに送信されることを示します。

注: 変更対象となる **ACO** は、フェデレーション環境のセットアップに応じて異なります。**Web** エージェントがインストールされているシステムと同じシステムにフェデレーション **Web** サービスを展開する場合は、**Web** エージェントに対応した **ACO** を編集してください。**Web** エージェントとは異なるシステムにフェデレーション **Web** サービスを展開する場合は、フェデレーション **Web** サービスに対して作成した固有の **ACO** を編集してください。

SiteMinder アサーティング パーティでの初期認証のセキュリティ保護

SiteMinder アサーティング パーティでのユーザの初期認証は、潜在的な脆弱性を与えます。ユーザがアサーティング パーティでユーザセッションを確立するために最初に認証する際に、セッション ID **Cookie** がブラウザに書き込まれます。**cookie** が非 **SSL** 接続の上で送信される場合、攻撃者は **Cookie** を取得してユーザの機密情報を盗むことができます。そして、攻撃者は、情報を使用してインパーソネーションや個人情報の盗難を実行できます。

ベストプラクティスとして、**Web** エージェントパラメータ **UseSecureCookies** を設定することにより、この脆弱性に対して保護します (このパラメータはエージェント設定オブジェクトで変更できます)。**UseSecureCookies** パラメータを設定すると、「**secure**」フラグが付けられた **Cookie** を生成するように **Web** エージェントに指定されます。このフラグは、ブラウザが **SSL** 接続上のみで **Cookie** を渡すことを示し、その結果、セキュリティが向上します。概して、すべての **URL** に対して **SSL** 接続を確立することが推奨されます。

クロス サイト スクリプティングに対する保護

実行可能スクリプトの形成可能な文字をフィルタリングせずに、アプリケーションがブラウザからの入力テキストを表示する場合に、クロスサイトスクリプティング (XSS) 攻撃が発生する可能性があります。通常、入力テキストは、ポストからのデータまたは URL に関するクエリ パラメータのデータです。これらの文字がブラウザに表示された場合、望ましくないスクリプトがブラウザ上で実行される可能性があります。

SiteMinder では、SiteMinder フェデレーション機能と連動して使用するための JSP をいくつか提供しています。これらの JSP は、出力ストリーム内の安全でない情報がブラウザに表示されないように、リクエスト内のキャラクタをチェックします。

SiteMinder がフェデレーション リクエストを受信すると、以下の JSP がデコードされた値をスキャンして、クロス サイト スクリプティング文字を調べます。

- **idpdiscovery.jsp**
アイデンティティ プロバイダ ディスカバリ用に依存パーティで使用されます。
- **linkaccount.jsp**
動的なアカウント リンク用に依存パーティで使用されます。
- **sample_application.jsp**
シングル サインオンを開始する IDP で使用されます。このサンプルアプリケーションを使用して、ユーザを SSO サービスに導いた後、カスタム Web アプリケーションに導くことができます。通常は独自のアプリケーションを使用します。
- **signoutconfirmurl.jsp**
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
+	プラス記号

SiteMinder 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、文字セットを拡張することができます。

第 8 章：フェデレーションのためのキーと証明書の管理

アサーションの保護およびアサーション内のデータの暗号化は、パートナーシップ設定の重要な部分です。フェデレーション環境で、キー/証明書ペアおよびスタンドアロン証明書は多くの機能に役立ちます。

- アサーションの署名/検証（3 つのすべてのプロファイル）
- 認証リクエストの署名/検証（SAML 2.0 のみ）

- シングル ログアウト リクエストおよびレスポンスの署名/検証 (SAML 2.0)
- HTTP-Artifact SSO のバックチャネル リクエストおよびレスポンスの署名 (SAML 1.1 および 2.0)
- アサーション全体またはアサーションの一部の暗号化/復号化 (SAML 2.0)
- Artifact シングル サインオン用のバックチャネル全体のクライアント 認証情報 (SAML 1.1 および 2.0)

「ポリシー サーバ設定ガイド」には、キーおよび証明書の管理に関する情報と手順が記載されています。

SSL サーバ証明書を使用して、以下のタスクを実行できます。

- SSL 接続でのフェデレーション トラフィックを管理する。
- Artifact シングル サインオンでのバックチャネルの通信のセキュリティを保護する。

SiteMinder Web エージェントがインストールされている Web サーバに対して SSL を有効にする手順を参照してください。

注: SSL を有効にすると、Base URL パラメータも含めて、すべてのサービスの URL に影響があります。具体的には、すべてのサービス URL が https:// で始まる必要があります。

SAML 2.0 署名アルゴリズム

SAML 2.0 の場合、タスクに署名するための署名アルゴリズムを選択するオプションがあります。アルゴリズムを選択する機能は以下のユース ケースをサポートします。

- IdP が RSAwithSHA1 または RSAwithSHA256 アルゴリズムで、アサーション、レスポンスおよび SLO-SOAP メッセージに署名する IdP から SP へのパートナーシップ。
- SP が RSAwithSHA1 または RSAwithSHA256 アルゴリズムで、認証リクエストおよび SLO-SOAP メッセージに署名する SP から IdP へのパートナーシップ。

署名検証によって、署名済みドキュメントで使用中のアルゴリズムを自動検出して、それを確認します。署名検証の設定は必要ありません。

第 9 章: フェデレーション用ユーザ ディレクトリの設定

ディレクトリ接続により、SiteMinder 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。SiteMinder はこれらの接続を使用して、ユーザ ID を確認し、ユーザ レコード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。

フェデレーション トランザクションのユーザを選択する前に、ユーザ ディレクトリを設定してください。ユーザ ディレクトリを設定するには、「ポリシー サーバ設定ガイド」を参照してください。

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

第 10 章: アフィリエイトドメインの作成

このセクションには、以下のトピックが含まれています。

[アフィリエイトドメインの概要 \(P. 131\)](#)

[アフィリエイトドメインの設定 \(P. 132\)](#)

[アフィリエイトドメインへのエンティティの追加 \(P. 133\)](#)

アフィリエイトドメインの概要

アフィリエイトドメインは、1つ以上のユーザディレクトリに関連付けられたフェデレーションエンティティの論理グループです。

アフィリエイトドメインは、フェデレーションエンティティが含まれているだけでなく、どのユーザディレクトリがドメインと関連付けられるかの定義も行います。アサーションを生成するには、SiteMinder がアイデンティティプロバイダとして、ユーザレコードが定義されるユーザディレクトリにアクセスする必要があります。ポリシーサーバは、アフィリエイトドメインの検索順で指定されるユーザディレクトリを問い合わせることにより、ユーザレコードの場所を特定します。

検索順は、ユーザディレクトリ接続をアフィリエイトドメインに追加するときに定義します。ディレクトリの順序を移動することもできます。

アフィリエイトドメインでは、ドメインのオブジェクトを変更できる管理者アカウントが 1つ以上必要です。システムレベルの管理者は、すべてのドメイン内の全オブジェクトを管理でき、アフィリエイトの管理権限を持ちます。ポリシードメインに対する制御権限を他の管理者に付与できるシステム管理者は、システムおよびドメインオブジェクトの管理権限を持ちます。

アフィリエイトドメインの設定

ドメインオブジェクトを追加して、コンシューマ、サービスプロバイダ、またはリソースパートナーのリソースにアクセス可能なユーザを選択し、関連するエンティティを追加することができます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [フェデレーション] - [レガシー フェデレーション] - [アフィリエイト ドメイン] をクリックします。
3. [アフィリエイト ドメインの作成] をクリックします。
4. [一般] 設定に、アフィリエイト ドメインの名前および簡単な説明を入力します。
5. [ユーザディレクトリ] セクションで、[追加/削除] をクリックします。
6. ドメインに関連付けるユーザディレクトリを [使用可能なメンバー] から [メンバーの選択] に移動させます。

アフィリエイト リソースへのアクセスを許可するユーザのレコードが保存されたディレクトリを指定してください。

7. [OK] をクリックします。

選択したディレクトリが、[ユーザディレクトリ] テーブルに表示されます。

既存のディレクトリが存在しない場合は、[作成] をクリックしてユーザディレクトリを作成してください。必要な情報の入力が完了すると、作成したディレクトリが [ユーザディレクトリ] テーブルに表示されます。

8. 必要に応じて、[ユーザディレクトリ] テーブルの右側の矢印を使用してテーブル内のディレクトリの順序を調節します。ディレクトリの詳細を編集する場合は、左側の矢印を使用します。

ディレクトリが表示される順序は、**SiteMinder** がユーザ レコードを検索する順序です。リストの最上位から検索が開始されます。

9. [サブミット] をクリックします。

アフィリエイト ドメインが作成されます。

次の手順では、パートナーをアフィリエイト ドメインに追加し、フェデレーション パートナリシップ内のアサーティング パーティとして **SiteMinder** を設定します。

詳細情報:

[SAML 1.x プロデューサの設定 \(P. 139\)](#)

[SAML 2.0 サービス プロバイダの設定 \(P. 287\)](#)

[SiteMinder の WS-フェデレーション リソース パートナーとしての設定 \(P. 369\)](#)

アフィリエイトドメインへのエンティティの追加

フェデレーション パートナリシップにおいてアサーティング パーティのロールを実行するように **SiteMinder** を設定します。 **SiteMinder** がアサーティング パーティとして機能するには、パートナーをアフィリエイト ドメインに追加します。 パートナーが認証リクエストを送信すると、**SiteMinder** がそれに応じてアサーションを生成することができます。

以下のエンティティをアフィリエイト ドメインに追加できます。

- SAML 1.x アフィリエイト
- SAML 2.0 サービス プロバイダ
- WS フェデレーション リソース パートナー

注: これらのエンティティには、アサーティング パーティで [フェデレーション Web サービスにアクセス \(P. 158\)](#) するための許可が必要です。

パートナーのアフィリエイトドメインの追加する手順については、以下のセクションのいずれかを参照してください。

- 「[SAML 1.x プロデューサとしての SiteMinder の設定](#) (P. 139)」
(Configure SiteMinder as a SAML 1.x Producer)
- [SAML 2.0 アイデンティティプロバイダとしての SiteMinder の設定](#) (P. 213)
- 「[アカウントパートナーとしての SiteMinder の設定](#)」 ([Configure SiteMinder as an Account Partner](#)) (P. 339)

詳細情報:

[SAML 1.x プロデューサの設定](#) (P. 139)

[SAML 2.0 アイデンティティプロバイダの設定](#) (P. 213)

[WS-フェデレーション アカウントパートナーの設定](#) (P. 339)

第 11 章: フェデレーション Web サービスへのアクセス権の付与

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 Web サービスを保護するポリシー](#) (P. 135)

[FWS ポリシーに関連付けられた機能](#) (P. 137)

[フェデレーション Web サービスを保護するポリシーの適用](#) (P. 138)

フェデレーション Web サービスを保護するポリシー

ポリシー サーバをインストールすると、いくつかのサービスに対するポリシーが **SiteMinder** で作成されます。これらのサービスに、フェデレーション Web サービス (FWS) アプリケーションが含まれます。一部のフェデレーション機能では、依存パーティが関連の保護されたサービスへのアクセス許可を必要とします。

依存パートナーをポリシーに追加するタスクは、アサーティングパーティのみで実行します。

たとえば、HTTP-Artifact バインドの場合、ポリシーは、**SiteMinder** がアサーションを取得するサービスを保護します。**SiteMinder** が特定の依存パートナーのアサーションを取得するには、そのパートナーがサービスを保護するポリシーにユーザとして追加されている必要があります。

次の表は、FWS サービスに関連する FWS ポリシー オブジェクトの一覧を示します。

オブジェクト タイプ	オブジェクト名
ドメイン	FederationWebServicesDomain
レルム	FederationWebServicesRealm public
エージェント グループ	FederationWebServicesAgentGroup

オブジェクト タイプ	オブジェクト名
ルール	SAML2FWSAttributeServiceRule
	FederationWSSessionServiceRule
	SAML2FWSArtifactResolutionRule
	FederationWSAssertionRetrievalServiceRule
	FederationWSNotificationServiceRule
ポリシー	SAML2FWSArtifactResolutionServicePolicy
	SAML2FWSAttributeServicePolicy
	FederationWSAssertionRetrievalServicePolicy
	FederationWSNotificationServicePolicy
	FederationWSSessionServicePolicy
変数	AllowNotification
	AllowSessionSync
ユーザ ディレクトリ	FederationWSCustomUserStore
	SAML2FederationCustomUserStore

FWS ポリシーに関連付けられた機能

SiteMinder によって作成されるポリシーは、以下の レガシー フェデレーション 機能をサポートします。

FWS ポリシー	フェデレーション機能
SAML2FWSArtifactResolutionServicePolicy	SAML 2.0 Artifact シングル サインオン用に Artifact 解決サービスを保護します
FederationWSAssertionRetrievalServicePolicy	SAML 1.x Artifact シングル サインオン用にアサーション検索サービスを保護します
SAML2FWSAttributeServicePolicy	SAML 2.0 の属性機関サービスを保護します
FederationWSNotificationServicePolicy	通知サービスを保護します。通知は、SAML アフィリエイトエージェントがコンシューマである場合のみ利用可能です。
FederationWSSessionServicePolicy	セッション管理のためにセッションサービスを保護します。セッション管理は、SAML アフィリエイトエージェントがコンシューマである場合のみ利用可能です。

フェデレーション Web サービスを保護するポリシーの適用

FWS ポリシーを備えたフェデレーション機能を実装する場合、依存パーティは保護されたサービスにアクセスするための許可を必要とします。

アクセス許可を付与するには、以下のタスク実行する必要があります。

- FWS アプリケーションを保護する Web エージェントを FederationWebServicesAgentGroup エージェント グループに追加する。
- 特定のサービスへのアクセスを許可されたユーザとして依存パートナーを追加する。

ユーザを所定のポリシーに追加する以外、その他すべてのポリシー オブジェクトは自動的にセットアップされます。

HTTP-Artifact アサーション取得および属性機関ポリシーを適用するための詳細な手順については、それらの機能の関連セクションを参照してください。

詳細情報:

[アサーションを取得するサービスへのアクセスの許可 \(Artifact SSO\)](#) (P. 156)

[依存パートナーの属性機関サービスへのアクセスの許可](#) (P. 414)

第 12 章: SAML 1.x プロデューサの設定

このセクションには、以下のトピックが含まれています。

- [アサーティング パートナーの前提条件 \(レガシー\)](#) (P. 139)
- [プロデューサを設定する方法](#) (P. 140)
- [SAML 1.x アフィリエイトとアフィリエイト ドメインの関連付け](#) (P. 142)
- [アフィリエイトの一般設定の入力](#) (P. 143)
- [アサーションが生成される対象のユーザの選択](#) (P. 148)
- [SAML 1.x アサーションの設定](#) (P. 151)
- [アサーションを取得するサービスへのアクセスの許可 \(Artifact SSO\)](#) (P. 156)
- [Artifact サービスを保護する認証方式の設定](#) (P. 160)
- [属性を SAML 1.x アサーションに含める設定 \(オプション\)](#) (P. 166)
- [SAML アサーション レスポンスのカスタマイズ \(オプション\)](#) (P. 171)
- [コンシューマ リソースへのリンクの作成 \(SAML 1.x\)](#) (P. 174)

アサーティング パートナーの前提条件(レガシー)

アサーティング パートナーを設定するには、以下の条件を確認します。

- ポリシー サーバがインストールされています。
- 以下のオプションのいずれかがインストールされています。
 - **Web エージェントおよび Web エージェント オプション パック。**
Web エージェントは、ユーザを認証し、SiteMinder セッションを確立します。 オプション パックはフェデレーション Web サービス アプリケーションを提供します。 ネットワーク内の適切なシステムに必ず FWS アプリケーションを展開してください。
 - **SPS フェデレーション ゲートウェイでは、Web エージェントが組み込まれており、また組み込まれた Tomcat Web サーバ上にフェデレーション Web サービス アプリケーションが配置されています。**

詳細については、「Web エージェント オプション パック ガイド」を参照してください。

- 秘密キーと証明書は、メッセージの署名および復号を必要とする機能に対してインポートされます。

- 依存パートナーは、フェデレーション ネットワーク内でセットアップされます。

プロデューサを設定する方法

SiteMinder は、SAML プロデューサとして、そのビジネス パートナー、すなわちコンシューマ用のアサーションを生成します。フェデレーション パートナーシップを確立するために、プロデューサは、各パートナーに関する情報を必要とします。各パートナーは、管理 UI ではアフィリエイトと呼ばれます。パートナーごとにアフィリエイト オブジェクトを作成し、2 つのエンティティがどのように通信してアサーションを渡し、シングルサインオンなどのプロファイルを満たすかを定義してください。

プロデューサにおいて以下の設定タスクが必要になります。

1. アフィリエイトをアフィリエイト ドメインに関連付けます。
2. [アフィリエイトの一般設定項目を設定します](#) (P. 143)。
3. [プロデューサがアサーションを生成する対象のユーザを選択します](#) (P. 148)。
4. [アサーションを設定します](#) (P. 151)。
5. (HTTP-Artifact SSO のみ)
 - a. セッション ストアによるアサーションの保存を有効にします。ポリシー サーバ管理コンソールを使用してセッション ストアを管理します。
 - b. 適用可能な依存パーティごとに [アサーション検索サービスへのアクセスを許可します](#) (P. 158)。

6. [シングルサインオンを開始するリンクを作成します](#) (P. 174)。
7. [オプションの設定タスクを実行します](#) (P. 141)。

ヒント：

- 設定が機能するためには、プロデューサとコンシューマの特定のパラメータ値が一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定](#) (P. 451)」 (Configuration Settings that Must Use the Same Values) に記載されています。
- フェデレーション Web サービス サブレットの正しい URL を使用していることを確認してください。これらの URL は、「[SiteMinder 設定で使用されるフェデレーション Web サービス URL](#) (P. 457)」 (Federation Web Services URLs Used in SiteMinder Configuration) に一覧表示されています。

アフィリエイトの識別のオプション設定タスク

アフィリエイトを識別する際に以下のオプションのタスクがあります。

- [アサーションに含める属性の設定](#) (P. 166)。
- [アフィリエイト動作の時間制限の設定](#) (P. 146)。
- アフィリエイトのアクセスに使用されるアドレスを制限するための [IP アドレス制限の設定](#)。 (P. 147)
- アサーション ジェネレータ プラグインを使用した [アサーションのコンテンツのカスタマイズ](#)。 (P. 171)

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。

- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

SAML 1.x アフィリエイトとアフィリエイトドメインの関連付け

アフィリエイト ドメインは、フェデレーション パートナーの論理グループです。SiteMinder が認識できるように、アフィリエイトをアフィリエイト ドメインに関連付けます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [フェデレーション] - [レガシー フェデレーション] - [アフィリエイト] をクリックします。
3. [アフィリエイトの作成] をクリックします。
4. このアフィリエイトが属するアフィリエイト ドメインを選択します。
5. [次へ] をクリックします。

このアフィリエイトがアフィリエイト ドメインに関連付けられます。次の手順では、アフィリエイトに関する[一般情報 \(P. 143\)](#)を設定します。

詳細情報:

[SiteMinder セッションのないユーザの認証 \(SAML 1.x\)](#) (P. 144)

アフィリエイトの一般設定の入力

アフィリエイトの一般設定項目を設定します。

アフィリエイトに関する一般情報を提供する方法

1. まず、設定ウィザードの [一般] 手順に移動します。
2. [一般] セクションで以下のフィールドに入力します。

- 名前
- パスワード/パスワードの確認
- 認証 URL

この URL は `redirect.jsp` ファイルを指す必要があります -- 例を以下に示します。

`http://myserver.mysite.com/siteminderagent/redirectjsp/redirect.jsp`

myserver は、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 ゲートウェイのある Web サーバを特定します。

認証 URL を保護するポリシーを必ず作成してください。

3. [有効] を選択して、アフィリエイト オブジェクトをアクティブ化します。
4. (オプション) [安全な URL を使用] を選択します。

[安全な URL を使用] 機能は、SiteMinder セッションを確立するためにユーザをリダイレクトする前に、認証 URL に追加する `SMPORTALURL` クエリ パラメータを暗号化するように SSO サービスに指定します。`SMPORTALURL` を暗号化することにより、悪意のあるユーザによるこのパラメータの改ざんが防止されます。

注: このチェック ボックスをオンにする場合は、[認証 URL] フィールドを以下の URL に設定してください。

`http(s)://idp_server:port/affwebservices/secure/securedirect.`

このフィールドの詳細については、[ヘルプ] をクリックしてください。

5. (オプション) [制限] セクションおよび [詳細] セクションのフィールドに入力します。
6. [次へ] をクリックします。

SiteMinder セッションのないユーザの認証(SAML 1.x)

コンシューマをアフィリエイト ドメインに追加する際、[認証 URL] フィールドの設定が必要となります。認証 URL が **redirect.jsp** ファイルを指す必要があります。この URL の目的は、プロデューサでのセッションを確立することです。

redirect.jsp ファイルは、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 ゲートウェイがインストールされているプロデューサでインストールされます。**redirect.jsp** ファイルは SiteMinder ポリシーで保護し、保護されたリソースをリクエストするユーザの認証が依頼されるようにします。ユーザが SiteMinder セッションを持たないため、Web エージェントからチャレンジが出力されます。

ユーザが認証され、正常に **redirect.jsp** ファイルにアクセスした後、セッションが確立されます。このユーザは、**redirect.jsp** ファイルによりリダイレクトされ、プロデューサ Web エージェントに戻ります。このエージェントでリクエストの処理が可能です。また、SAML アサーションの生成ができます。

認証 URL を保護するための手順は、以下のすべてのセットアップにおいて同じです。

- Web エージェントと同じシステムにインストールされる Web エージェント オプションパック。
- Web サーバプロキシにインストールされた Web エージェントのあるアプリケーションサーバ。
- アプリケーションサーバ エージェントと共にインストールされるアプリケーションサーバ。
- アサーティングパーティでインストールされる SPS フェデレーションゲートウェイ。

認証 URL を保護するポリシーの設定

認証 URL を保護する方法

1. 管理 UI にログインします。
2. アサーティング パーティ **Web** サーバに対して定義したレルムにバインドする **Web** エージェントを作成します。 **Web** サーバと **FWS** アプリケーションに個別のエージェント名を割り当てるか、両方に同じエージェント名を使用します。
3. コンシューマ リソースへのアクセス試行時に認証情報が要求されるユーザに対してポリシー ドメインを作成します。
4. ポリシー ドメインに含まれるリソースへのアクセス権が必要なユーザを選択します。
5. ポリシー ドメインに対して以下の値を持つレルムを定義します。

エージェント

アサーティング パーティ **Web** サーバのエージェント

リソース フィルタ

Web エージェント r6.x QMR 6、r12.0 SP2、r12.0 SP3、および SPS フェデレーション ゲートウェイによって次の値が入力されます。

`/siteminderagent/redirectjsp/`

リソース フィルタ `/siteminderagent/redirectjsp/` は、**FWS** アプリケーションが自動的にセットアップするエイリアスです。 エイリアス参照には次の内容が含まれます。

■ Web エージェント

`web_agent_home/affwebservices/redirectjsp`

■ SPS フェデレーション ゲートウェイ

`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`

永続セッション

SAML Artifact プロファイルの場合のみ、[レルム] ダイアログ ボックスの [セッション] セクションで [永続] チェック ボックスをオンにします。 永続セッションを設定しない場合、ユーザはコンシューマ リソースにアクセスできません。

残りの設定項目は、デフォルトのままにするか、必要に応じ変更します。

6. [OK] をクリックしてレلمを保存します。
7. レلمのルールを作成します。レلمに対応するすべてのリソースを保護するには、[レلم] フィールドでデフォルト値のアスタリスク (*) をそのまま使用します。
8. 前の手順で作成したルールが含まれるアサーティング パーティ Web サーバに対するポリシーを作成します。
9. 「[アサーションが生成される対象のユーザの選択 \(P. 148\)](#)」のタスクを実行します。

SAML 1.x コンシューマの時間制限の設定(オプション)

コンシューマ リソースの利用可能な時間に制限を指定できます。時間制限を指定すると、コンシューマ リソースへのアクセスは指定された期間中のみ可能となります。ユーザが指定の期間外にリソースへのアクセスを試みても、プロデューサは **SAML** アサーションを生成しません。

注: 時間制限は、ポリシー サーバがインストールされたサーバのシステムクロックに基づきます。

時間制限を指定する方法

1. まず、[一般] 設定に移動します。
ページの [制限] セクションで、[時間を設定] を選択します。
[時間制限] ページが表示されます。
2. スケジュールを設定します。このスケジュール グリッドは、ルール オブジェクトの [時間制限] グリッドと同一です。詳細については、「ポリシー サーバ設定ガイド」を参照してください。
3. [OK] をクリックします。

時間制限のスケジュールが設定されました。

SAML 1.x コンシューマの IP アドレス制限の設定(オプション)

コンシューマにアクセスするブラウザが実行されている Web サーバの IP アドレス、範囲アドレス、またはサブネット マスクを指定できます。IP アドレスを指定する場合、コンシューマでは、適切な IP アドレスからのユーザのみを受け付けます。

IP アドレスを指定する方法

1. まず、管理 UI の [一般] 設定に移動します。

ページの [制限] セクションで、[IP アドレス] 領域の [追加] をクリックします。

[IP 制限] ページが表示されます。

2. 追加する IP アドレス タイプのオプションを選択し、そのアドレス タイプに関連するフィールドに入力します。

IP アドレスが不明でも、アドレスのドメイン名が既知の場合は、[DNS の検索] ボタンをクリックしてください。このボタンにより、[DNS の検索] ページが表示されます。[ホスト名] フィールドに完全修飾ホスト名を入力し、[OK] をクリックします。

オプションを以下に示します。

- [単一ホスト] -- ブラウザをホストする単一の IP アドレスを指定します。単一の IP アドレスを指定する場合、ユーザは指定された IP アドレスからのみコンシューマにアクセスできます。
- [ホスト名] -- Web サーバをそのホスト名で指定します。ホスト名を指定する場合、コンシューマは指定されたホストからのユーザのみにアクセス可能です。
- [サブネットマスク] -- Web サーバのサブネット マスクを指定します。サブネット マスクを指定する場合、サービス プロバイダは指定されたサブネット マスクからのユーザのみにアクセス可能です。このボタンを選択する場合、[アドレスとサブネットマスクの追加] ダイアログ ボックスが表示されます。左矢印ボタンと右矢印ボタンを使用するか、スライダー をクリックしてからドラッグして、サブネット マスクを選択してください。
- [範囲] -- IP アドレス範囲を指定します。IP アドレスの範囲を指定する場合、コンシューマは、アドレスの範囲内の IP アドレスからのユーザのみを許可します。範囲の開始アドレス ([FROM]) および終了アドレス ([TO]) を入力します。

3. [OK] をクリックして設定を保存します。

アサーションが生成される対象のユーザの選択

アサーティング パーティでの設定の一部として、アサーション ジェネレータが **SAML アサーション** を生成する対象のユーザおよびグループのリストを含めます。アサーティング パーティは、**SAML 1.x** プロデューサ、**SAML 2.0** アイデンティティ プロバイダ、または **WS フェデレーション** アカウント パートナーのいずれかです。

アフィリエイト ドメインにあるディレクトリからのユーザおよびグループのみを追加できます。

フェデレーション トランザクションのユーザおよびグループを指定する方法

1. 設定するパートナーの [ユーザ] 設定に移動します。

[ユーザ ディレクトリ] ページには、ポリシー ドメインのユーザ ディレクトリごとにエントリが表示されます。

2. ユーザ ディレクトリからポリシーにユーザまたはグループを追加します。

各ユーザ ディレクトリ テーブルで、[メンバーの追加] - [エントリの追加] - [すべて追加] を選択できます。選択したメソッドに応じて、ダイアログ ボックスが表示され、ユーザの追加が可能になります。

- [メンバーの追加] を選択すると、[ユーザ/グループ] ウィンドウが表示されます。個々のユーザは、自動的に表示されません。検索ユーティリティを使用して、ディレクトリの 1 つに含まれる特定のユーザを見つけることができます。
- [エントリの追加] を選択する場合は、[ユーザ ディレクトリ 検索式編集] ダイアログ ボックスで [手動設定](#) (P. 150) によってユーザを選択してください。

ユーザやグループの編集または削除を行うには、それぞれ右向き矢印 (>) またはマイナス記号 (-) をクリックします。

3. 好きな方法で個別のユーザ、ユーザ グループ、または両方を選択して [OK] をクリックします。

[ユーザ ディレクトリ] ページが再度開き、新しいユーザがユーザ ディレクトリ テーブルに表示されます。

詳細情報:

[リソースへのアクセスからのユーザまたはグループの除外](#) (P. 149)

[ネストされたグループによるリソースへのアクセスの許可](#) (P. 149)

[手動設定によるユーザの追加](#) (P. 150)

リソースへのアクセスからのユーザまたはグループの除外

ユーザやユーザのグループをアサーションの取得から除外できます。

次の手順に従ってください:

1. [ユーザ] 設定に移動します。
2. 特定のユーザ ディレクトリのリストからユーザまたはグループを選択します。
3. 選択したユーザまたはグループを除外するには、[除外] をクリックします。

選択内容が 管理 UI に反映されます。

4. [OK] をクリックして、変更を保存します。

ネストされたグループによるリソースへのアクセスの許可

LDAP ユーザ ディレクトリには、サブグループを持つグループが含まれることがあります。複雑なディレクトリの場合、大量のユーザ情報を組織化する方法の 1 つとして、他のグループの階層内にグループをネストすることがあります。

ネストされたグループ内のユーザの検索を有効にする場合、リクエストされたユーザ レコードがすべてのネストされたグループで検索されます。ネストされたグループを有効にしない場合、指定したグループのみをポリシー サーバが検索します。

ネストされたグループでの検索を有効にする方法

1. [ユーザ] 設定に移動します。

関連するアフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、それぞれのユーザ ディレクトリが独自のセクションに表示されます。

2. ネストされたグループ内の検索を有効にするには、[ネストされたグループの許可] チェック ボックスをオンにします。

手動設定によるユーザの追加

アサーション生成のユーザを指定する場合の選択肢の 1 つとして、手動設定によりユーザを識別します。

次の手順に従ってください:

1. 設定するパートナーの [ユーザ] 設定に移動します。

アフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、すべてのディレクトリが [ユーザ ディレクトリ] ページに表示されます。

2. [エントリの追加] をクリックします。

[ユーザ ディレクトリ 検索式編集] ページが表示されます。

3. 検索オプションを選択した後、その検索オプションのフィールドに入力します。

検索する場所

LDAP ディレクトリの場合、ドロップダウン リストからオプションを選択します。

DN の確認

LDAP 検索はディレクトリ内でこの DN を検索します。

ユーザの検索

LDAP 検索の実行範囲は、ユーザ エントリ内の一致に限定されます。

グループの検索

LDAP 検索の実行範囲は、グループ エントリ内の一致に限定されます。

組織の検索

LDAP 検索の実行範囲は、組織エントリ内の一致に限定されます。

エントリの検索

LDAP 検索の実行範囲は、ユーザ エントリ、グループ エントリ、および組織エントリ内の一致に制限されています。

- Microsoft SQL Server、Oracle、および Windows NT ディレクトリの場合は、[手動設定] フィールドにユーザ名を入力できます。
- Microsoft SQL Server または Oracle では、SQL クエリを代わりに入力することもできます。例：

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

ポリシー サーバは、ユーザ ディレクトリの [認証情報と接続] タブの [ユーザ名] フィールドに指定されたデータベース ユーザとしてクエリを実行します。[手動設定] フィールドに入力する SQL 文を作成する場合には、ユーザ ディレクトリのデータベーススキーマに関する知識が必要です。たとえば、SmSampleUsers スキーマを使用している場合に特定のユーザを追加するには、SmUser テーブルからユーザ エントリを選択します。

- LDAP ディレクトリの場合は、[手動設定] フィールドに「all」と入力して、すべてのディレクトリ エントリを追加します。

4. [OK] をクリックして、変更を保存します。

SAML 1.x アサーションの設定

プロデューサ サイトにおいて、SAML アサーションをコンシューマに配信する方法を決定します。アサーションは、ユーザを識別してコンシューマに通知します。

アサーションは、以下の情報が含まれる XML ドキュメントです。

- コンシューマに関する情報
- セッション情報
- ユーザ属性

SAML アサーションに関する詳細については、[エラー! ハイパーリンクの参照に誤りがあります。](#)の SAML 仕様を参照してください。

SAML 1.x アサーションを設定する方法

1. [アサーション] 設定に移動します。
2. [アサーション] ページのフィールドに入力します。フィールドの説明については、[ヘルプ] をクリックしてください。

- アサーション コンシューマ URL (SAML POST の場合は必須、SAML Artifact の場合はオプション)

SAML 1.x Artifact バインドの場合、アサーション コンシューマ URL が SMCONSUMERURL クエリ パラメータ (必須のサイト間転送 URL パラメータ) より優先されます。ユーザがシングルサインオンを開始する場合にこの URL を選択します。悪意のあるユーザは、クエリ パラメータを変更し、不正なサイトにユーザを送って Artifact を取得する可能性があります。ユーザが不正に導かれるのを防止するには、アサーション コンシューマ URL の値を指定してください。

- スキュー時間 (秒)

プロデューサのシステム クロックとコンシューマのシステム クロックの差を秒単位で指定します。スキュー時間は、シングルサインオンおよびシングルログアウトに使用されます。

シングルサインオンの場合、スキュー時間の値とシングルサインオン有効期間によって、アサーションが有効な時間が決定されます。スキュー時間についての理解を深めるには、[アサーション有効期間の計算](#) (P. 153)方法について再確認してください。

3. [終了] をクリックして選択内容を保存します。

[アサーション] ページには、オプションの [\[属性 \(P. 166\)\]](#) セクションも含まれます。このセクションでは、属性をアサーションに含めることができます。

詳細情報:

[SAML 1.x アサーション関連のセキュリティ問題 \(P. 153\)](#)

SAML 1.x アサーション関連のセキュリティ問題

SAML アサーション ジェネレータは、任意の認証方式保護レベルで認証されたユーザのセッションに基づいたアサーションを作成します。プロデューサがアサーションを生成する対象ユーザを制御できます。それらのユーザが認証される保護レベルは制御できません。

複数のリソースで 1 つの特定の保護レベルをリクエストすることができます。ソースを別の保護レベルでセキュリティ保護することができます。ユーザが認証する際に、それらのユーザが必要な保護レベルで認証していることを確認してください。

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、スキュー時間と有効期間の値によって、アサーションの合計有効時間を SiteMinder が計算する方法が決定されます。SiteMinder 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。

注: この説明におけるアサーティング パーティは、SAML 1.x プロデューサ、SAML 2.0 アイデンティティ プロバイダ、または WS フェデレーション アカウント パートナーです。依存パーティは、SAML 1.x コンシューマ、SAML 2.0 サービス プロバイダ、または WS フェデレーション リソース パートナーです。

アサーション ドキュメントで、NotBefore と NotOnOrAfter の値は、有効期間の開始と終了を表します。

アサーティング パーティで、SiteMinder はアサーション有効期間を設定します。有効期間は、アサーションが生成されたときのシステム時間です。SiteMinder は、この時間を使用してアサーションに IssueInstant 値を設定した後、IssueInstant 値からスキュー時間の値を引き算します。その結果の時間が NotBefore 値になります。

NotBefore=IssueInstant - スキュー時間

有効期間の終了を決定するために、SiteMinder 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、SiteMinder は、アサーティングパーティで受信したアサーションが有効かどうか判別するために実行する計算と同じ計算を実行します。

パートナーシップの両側での SiteMinder によるアサーション有効期間の計算

SiteMinder がパートナーシップの両側にある場合、アサーション有効期間は、スキュー時間の 2 倍と有効期間の総和になります。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) + 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 + 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: レガシー フェデレーション の場合、有効期間はアサーティングパーティのみで設定されます。

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
- 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、アサーションから **NotBefore** と **NotOnOrAfter** の値を使用し、それらの値にスキュー時間を適用します。この数式は、依存パーティが新しい **NotBefore** と **NotOnOrAfter** の値を計算する方法です。

- スキュー時間 = 180 秒 (3 分)
- NotBefore = 4:56PM
- NotOnOrAfter=5:05PM

アサーション有効期間

この例の値を使用したアサーションの合計有効期間の計算は、次のとおりです。

$$120 \text{ 秒 (2x60)} + 60 \text{ 秒} + 360 \text{ 秒 (2x180)} = 540 \text{ 秒 (9 分)}$$

アサーションの単一使用の設定

SAML 1.x の仕様に準拠して、**SiteMinder** はアサーションの単一使用を適用できます。単一使用のアサーションを生成することにより、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ことが依存パーティで認識されます。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 情報を使用した認証判断という結果になります。

アサーションの単一使用を設定する方法

1. アフィリエイト オブジェクトの [一般] 設定に移動します。
2. [詳細] セクションで、[DoNotCache 条件の設定] を選択します。
3. [サブミット] をクリックします。

アサーションを取得するサービスへのアクセスの許可 (Artifact SSO)

HTTP Artifact シングル サインオンの場合、依存パーティは、アサーションを取得するための FWS サービスを保護するポリシーへのアクセスを許可する必要があります。

アクセスを許可するには、以下の手順に従います。

- **FederationWebServicesAgentGroup** エージェント グループに、FWS アプリケーションを保護する **Web** エージェントを追加します。
- 特定のサービスへのアクセスを許可された [ユーザとして依存パートナーを追加](#) (P. 158) します。

ユーザを所定のポリシーに追加する以外、その他すべてのポリシー オブジェクトは自動的にセットアップされます。

フェデレーション エージェント グループへの Web エージェントの追加

FederationWebServicesAgentGroup エージェント グループに、FWS アプリケーションを保護する Web エージェントを追加します。

- ServletExec の場合、このエージェントは、Web エージェント オプション パックがインストールされている Web サーバにあります。
- WebLogic や JBOSS などのアプリケーション サーバの場合、この Web エージェントは、アプリケーション サーバ プロキシがインストールされている場所にインストールされています。Web エージェント オプション パックは別のシステムに存在することもあります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [エージェント] - [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. 展開内の Web エージェントの名前を指定します。[サブミット] をクリックします。
5. [インフラストラクチャ] - [エージェント] - [エージェント グループ] をクリックします。
6. [FederationWebServicesAgentGroup] エントリを選択します。
7. [追加/削除] をクリックします。[エージェント グループのメンバ] ダイアログ ボックスが表示されます。
8. Web エージェントを [使用可能なメンバ] リストから [選択されたメンバ] リストに移動します。
9. [OK] ボタンをクリックして、[エージェント グループ] ダイアログに戻ります。
10. [サブミット] クリックした後、[閉じる] クリックしてメイン ページに戻ります。

アサーションを取得するための FWS ポリシーへの依存パートナーの追加

シングルサインオンに HTTP-Artifact バインドを使用している場合、パートナーシップ内の依存パーティはアサーション検索サービスへのアクセス許可を必要とします。SiteMinder は、ポリシーを使用して SAML 1.x および 2.0 検索サービスを保護します。

ポリシー サーバをインストールする際に、デフォルトで FederationWebServicesDomain がインストールされます。このドメインには、SiteMinder がアサーションを取得するサービスに対応して以下のポリシーが含まれます。

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

注: WS フェデレーションは、HTTP-Artifact プロファイルを使用しません。そのため、この手順はリソース プロバイダには適用されません。

これらのポリシーが任意の関連依存パートナーにアクセスするための権限を付与します。

次の手順に従ってください:

1. 管理 UI で、[ポリシー] - [ドメイン] - [ドメイン ポリシー] の順に移動します。

ドメイン ポリシーのリストが表示されます。

2. SAML プロファイルのポリシーを選択します。

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

[ドメイン ポリシー] ページが表示されます。

3. [変更] をクリックしてポリシーを変更します。
4. [ユーザ] タブを選択します。

5. 該当するユーザディレクトリのダイアログボックスで、[メンバーの追加] をクリックします。

SAML 1.x

FederationWSCustomUserStore

SAML 2.0

SAML2FederationCustomUserStore

[ユーザ/グループ] ページが表示されます。

先に設定したアフィリエイト ドメインが [ユーザ/グループ] ダイアログボックスにリスト表示されます。たとえば、アフィリエイト ドメインが **fedpartners** という名前の場合、エントリは **affiliate:fedpartners** となります。

6. サービスへのアクセスを必要とするパートナーが存在するアフィリエイト ドメインの横のチェックボックスをオンにします。[OK] をクリックします。

ユーザディレクトリのリストに戻ります。

7. [サブミット] をクリックします。

ポリシーのリストに戻ります。

アサーション検索サービスのベーシック保護の検証

アサーション検索サービスを保護するためにベーシック認証を設定する場合、保護を検証します。

次の手順に従ってください:

1. Web ブラウザを開きます。

フェデレーション Web サービス アプリケーションがインストールされているサーバ用の完全修飾ホスト名およびポート番号を入力することにより、フェデレーション Web サービスにアクセスします。例:

SAML 1.x: `http://idp-fws.ca.com:81/affwebservices/assertionretriever`

SAML 2.0: `http://idp-fws.ca.com:81/affwebservices/saml2artifactresolution`

サービスが保護されている場合、SiteMinder から認証情報の入力が必要です。認可されたアフィリエイトのみにフェデレーション Web サービスへのアクセスが許可されます。

2. 依存パートナー用にポリシー サーバで設定された有効な名前とパスワードを入力します。名前とパスワードが認証のチャレンジの認証情報です。

認証のチャレンジは、サービスが保護されていることを示します。

SiteMinder がチャレンジを示さない場合、ポリシーの設定が不適切です。

Artifact サービスを保護する認証方式の設定

HTTP-Artifact プロファイルの場合、アサーション検索サービス (SAML 1.x) および Artifact 解決サービス (SAML 2.0) は、アサーティングパーティでアサーションを取得します。これらのサービスでは、アサーション レスポンスを依存パーティに送信する際、セキュリティで保護されたバックチャネルを使用します。これらのサービスおよびバックチャネル上の通信を不正なアクセスから保護することを強くお勧めします。

注: WS フェデレーション は HTTP-Artifact プロファイルをサポートしません。

これらのサービスを保護するには、アサーティングパーティでサービスが含まれるレلمに対する認証方式を指定します。認証方式により、依存パーティで消費サービスがバックチャネル上の関連サービスにアクセスするために提供する必要がある認証情報のタイプが指定されます。

以下のいずれかの認証方式を選択できます。

- [基本](#) (P. 161)
- [SSL を介した基本](#) (P. 162)
- [X.509 クライアント証明書](#) (P. 162)

アサーションを取得するサービスを保護するベーシック認証

HTTP Artifact シングル サインオンの場合、アサーティング パーティは、安全なバックチャネルでアサーションを依存パーティに送信します。ベーシック認証の場合、Artifact を解決し、アサーションを取得するサービスにアクセスするためのパスワードを設定します。すると、サービスは、バックチャネルでアサーションを依存パーティに送信します。

SSL を有効にしてベーシック認証を使用できます。ただし、SSL は必須ではありません。

注: バックチャネル全域でベーシックまたは Basic over SSL を認証方式として使用する場合にのみパスワードが関連します。

SAML 1.x アサーション検索サービスの場合、次の手順に従ってください:

1. 管理 UI にログインします。
2. プロデューサに関する [一般] 設定に移動します。
3. 以下のフィールドに値を入力します。
 - パスワード
 - パスワードの確認入力
4. [サブミット] をクリックして変更内容を保存します。

SAML 2.0 Artifact 解決サービスの場合、次の手順に従ってください:

1. 管理 UI にログインします。
2. アイデンティティ プロバイダに関する [属性] 設定に移動します。
3. [バックチャネル] セクションで、以下のフィールドに値を入力します。
 - パスワード
 - パスワードの確認入力
4. [サブミット] をクリックして変更内容を保存します。

Basic over SSL によるアサーション検索サービスの保護

アサーション検索サービス (SAML 1.x) や Artifact 解決サービス (SAML 2.0) の保護として、Basic over SSL 認証方式を使用できます。アサーティングパーティにおいて、ポリシー サーバのインストール時に、サービスを保護するための一連のデフォルト ポリシーがすでに設定されています。

必要な設定は、各パートナーで SSL を有効にすることだけです。アサーティングパーティや依存パーティでの他の設定は不要です。依存パーティでは、証明書データ ストアのデフォルト ルート認証機関 (CA) の 1 つを使用して、SSL 接続を確立できます。デフォルト CA の代わりに独自のルート CA を使用する場合は、CA 証明書をデータ ストアにインポートします。

Basic over SSL 認証方式を使用する場合には、すべてエンドポイント URL で SSL 通信が使用される必要があります。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。

アサーションを検索するサービスを保護ためのクライアント証明書認証

クライアント証明書認証方式を使用して、アサーション検索サービス (SAML 1.x) および Artifact 解決サービス (SAML 2.0) を保護できます。アサーティングパーティがクライアント証明書認証を必要とするように設定されている場合、依存パーティは、接続をアサーティングパーティに戻し、クライアント証明書の表示を試みます。

クライアント証明書認証方式の使用方法

1. 関連するサービスを保護するためのポリシーをアサーティングパーティで作成します。このポリシーは、クライアント証明書認証方式を使用します。
2. 依存パーティでのバックチャネル設定用にクライアント証明書認証を有効化します。
3. パートナーシップのそれぞれの側で SSL を有効化します。

クライアント証明書認証を使用する場合、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。すなわち、URL が **https://** で始まる必要があります。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。

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)。

検索サービスを保護するポリシーの作成

アサーティングパーティがアサーションを取得するサービスを保護するためのポリシーをアサーティングパーティで作成します。

次の手順に従ってください:

1. アサーションをリクエストするアフィリエイトごとに、個別のエントリをユーザディレクトリに追加します。ユーザディレクトリを作成するか既存のディレクトリを使用します。

管理 UI のアフィリエイト一般設定の [名前] フィールドで指定されている値と同じ値をユーザレコードに入力します。たとえば、アフィリエイトの [名前] フィールドの値が **Company A** の場合、ユーザディレクトリのエントリは次のようになります。

`uid=CompanyA, ou=Development, o=CA`

ポリシーサーバは、アフィリエイトクライアント証明書の Subject DN 値をこのディレクトリエントリにマップします。

2. 設定したユーザディレクトリを **FederationWebServicesDomain** に追加します。

3. 証明書マッピング エントリを作成します。

属性名を、アフィリエイトのユーザディレクトリ エントリにマップします。属性は、アフィリエイトの証明書における **Subject DN** エントリを表します。たとえば、**CN** を属性名として選択すると、この値は **cn=CompanyA,ou=Development,o=partner** というアフィリエイトを表します。

マップを設定するには、[インフラストラクチャ] - [ディレクトリ] - [証明書マッピング] の順に移動します。

4. X509 クライアント証明書認証方式を設定します。
5. 以下のエントリが含まれるレルムを **FederationWebServicesDomain** の下に作成します。

名前

any_name

例： cert assertion retrieval

エージェント

FederationWebServicesAgentGroup

リソース フィルタ

/affwebservices/certassertionretriever (SAML 1.x)

/affwebservices/saml2certartifactresolution (SAML 2.0)

認証方式

前の手順で作成されたクライアント証明書認証方式。

6. 以下の情報が含まれるルールを cert assertion retrieval レルムの下に作成します。

名前

any_name

例： cert assertion retrieval rule

リソース

*

Web エージェント アクション

GET、POST、PUT

7. FederationWebServicesDomain の下に Web エージェント レスポンス ヘッダを作成します。

アサーション検索サービスは、この HTTP ヘッダを使用して、アフィリエイトがアサーションを取得するサイトであることを確認します。

以下の値が含まれるレスポンスを作成します。

名前

any_name

属性

WebAgent-HTTP-Header-Variable

属性の種類

ユーザ属性

変数名

consumer_name

属性名

アフィリエイト名の値が含まれる使用ディレクトリ属性を入力します。

例：uid=CompanyA

以下のエントリに基づいて、Web エージェントは、HTTP_CONSUMER_NAME という名前のレスポンスを返します。

8. 以下の値が含まれるポリシーを FederationWebServicesDomain の下に作成します。

名前

any_name

ユーザ

この手順で先に作成したユーザ ディレクトリからユーザを追加します。

ルール

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

Artifact 解決サービスを保護するポリシーが完成しました。

依存パーティにおいて、管理者は、以下のように関連するアサーションサービスに接続するバックチャネルでクライアント証明書認証を有効にする必要があります

SAML 1.x: アサーション検索サービスに対して [クライアント証明書認証を有効化](#) (P. 202) します

SAML 2.0: Artifact 解決サービスに対して [クライアント証明書認証を有効化](#) (P. 328) します

属性を SAML 1.x アサーションに含める設定(オプション)

属性をアサーションに含めることができます。サーブレットやアプリケーションでは、属性を使用してユーザに関するカスタマイズされたコンテンツを表示できます。ユーザ属性、DN 属性、または静的データのすべてをアサーション内でプロデューサからコンシューマに渡すことができます。属性を Web アプリケーションで使用する場合、属性を使用して、コンシューマでのユーザのアクティビティを制限できます。たとえば、プロデューサは **Authorized Amount** という名前の属性を送信します。コンシューマでは、この属性をユーザの支出可能最高金額に設定します。

属性は名前/値ペアの形式をとり、属性には、メールアドレス、肩書き、許可されたトランザクション支出限度などの情報が含まれます。コンシューマでは、アサーションを受信すると、属性を抽出します。コンシューマは、属性を HTTP ヘッダ変数または HTTP Cookie 変数としてアプリケーションで使えるようにします。

属性を渡すには、レスポンスを設定する必要があります。この目的に利用可能なレスポンスは次のとおりです。

- Affiliate-HTTP-Header-Variable
- Affiliate-HTTP-Cookie-Variable

HTTP ヘッダおよび HTTP Cookie にはサイズ制限があり、アサーション属性でそれらを超えるはできません。サイズ制限は以下のとおりです。

- HTTP ヘッダ：SiteMinder は、Web サーバのヘッダに関するサイズ制限までの属性をヘッダで送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダ サイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。
- HTTP Cookie：SiteMinder は、Cookie のサイズ制限までの Cookie を送信できます。それぞれのアサーション属性は、それ自身の Cookie として送信されます。Cookie のサイズ制限はブラウザ固有です。また、その制限は、属性ごとではなく、アプリケーションに渡されるすべての属性に適用されます。Cookie のサイズ制限を調べるには、使用する Web ブラウザのドキュメントを参照してください。

詳細情報:

[SAML 1.x アサーションの属性の設定 \(P. 167\)](#)

[レスポンス属性の新規作成スクリプトの使用 \(P. 170\)](#)

SAML 1.x アサーションの属性の設定

属性を SAML アサーションからコンシューマサイトのターゲットアプリケーションへ渡すようにレスポンスを設定できます。

アサーションの属性を設定する方法

1. [アサーション] 設定に移動します。
2. [属性] セクションの [追加] をクリックします。
[属性の追加] ダイアログ ボックスが表示されます。
3. [属性タイプ] ドロップダウンから、ヘッダを設定するか Cookie 変数を設定するかを選択します。

4. 「属性のセットアップ」セクションから、以下のいずれかのオプションを「属性の種類」セクションで選択します。

- スタティック
- ユーザ属性
- DN 属性

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。

選択の結果により、「属性フィールド」セクションで利用可能なフィールドが決定されます。

5. 選択した「属性の種類」のフィールドに入力します。「属性の種類」の選択結果により、設定が必要な追加フィールドが決定されます。

スタティック

以下のフィールドに入力します。

- 変数名

SiteMinder がアフィリエイトに返す属性の名前を入力します。

- 変数値

名前/値ペアの値として静的なテキストを入力します。

たとえば、`show_content=yes` という名前/値ペアを返すには、変数名として `show_content` と入力し、変数値として `yes` と入力します。

ユーザ属性

以下のフィールドに入力します。

- 変数名

SiteMinder がコンシューマに返す属性の名前を入力します。

- 属性名

名前/値ペアに対応するユーザ ディレクトリの属性を入力します。

たとえば、ユーザの電子メールアドレスをコンシューマに返すには、変数名として `email_address` と入力し、属性名として `email` と入力します。

DN 属性

以下のフィールドに入力します。

- 変数名

SiteMinder がコンシューマに返す属性の名前を入力します。

- DN 仕様

SiteMinder がユーザ属性を取得するユーザ グループの識別名を入力します。DN の値はコンシューマに返されます。DN がわからない場合は、[検索] をクリックします。SiteMinder [ユーザの検索] ダイアログ ボックスを使用して、ユーザ グループを検索し、DN を選択します。

- 属性名

名前/値ペアのこの属性に対応するユーザ ディレクトリの属性を入力します。

[属性] メニューから [アフィリエイト] - [HTTP] - [Cookie] - [変数] を選択すると、[変数名] フィールドのラベルが [Cookie 名] に変化します。

6. (オプション) ネストされたグループから DN 属性を取得するには、[属性の種類] セクション内の [ネストされたグループの許可] チェック ボックスをオンにします。
7. [OK] をクリックして、変更を保存します。

アサーション属性の最大文字数を指定します

ユーザ アサーション属性の最大長は設定可能です。アサーション属性の最大長を変更するには、EntitlementGenerator.properties ファイルの設定を変更します。

ファイル内のプロパティ名は、設定のプロトコルに固有です。

次の手順に従ってください:

1. ポリシー サーバがインストールされているシステムで、`policy_server_home¥config¥properties¥EntitlementGenerator.properties` に移動します。
2. テキスト エディタでファイルを開きます。

- 現在の環境で使用中のプロトコルでのユーザ属性の最大長を調節します。各プロトコルの設定を以下に示します。

WS-フェデレーション

プロパティ名：

`com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength`

プロパティ タイプ：正の整数値

デフォルト値：1024

説明：WS-FED アサーション属性の最大属性長を示します。

SAML 1.x

プロパティ名：

`com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength`

プロパティ タイプ：正の整数値

デフォルト値：1024

説明：SAML1.1 アサーション属性の最大属性長を示します。

SAML 2.0

プロパティ名：

`com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength`

プロパティ タイプ：正の整数値

デフォルト値：1024

説明：SAML2.0 アサーション属性用の最大属性長を示します。

- これらのパラメータのいずれかを変更した後は、ポリシー サーバを再起動します。

レスポンス属性の新規作成スクリプトの使用

〔属性の追加〕 ページの〔詳細〕 セクションには、〔スクリプト〕 フィールドが含まれます。このフィールドには、〔属性のセットアップ〕 セクションの入力に基づいて SiteMinder が生成したスクリプトが表示されます。このフィールドの内容をコピーし、別のレスポンス属性の〔スクリプト〕 フィールドに貼り付けることができます。

注：〔スクリプト〕 フィールドの内容をコピーして別の属性に貼り付ける場合は、〔属性の種類〕 セクションで適切なオプション ボタンを選択してください。

SAML アサーション レスポンスのカスタマイズ(オプション)

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を使用して、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次のとおりです。

1. SiteMinder 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（SDK に含まれています）を実装します。
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. 管理 UI でアサーション ジェネレータ プラグイン パラメータを有効にします。

アサーション ジェネレータ プラグインに関する追加の情報は、以下を参照してください。

- 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 クラスの新規コンストラクタが「*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。
- 認証および認可の API に関する概要および概念情報については、「Java 用 SiteMinder プログラミング ガイド」を参照してください。

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。

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。

- 要件を満たすようにインターフェースにメソッドを実装します。

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。既存の実装を上書きできます。例については、以下のサンプルクラスを参照してください。

SAML 1.x / WS フェデレーション

AssertionSample.java

SAML 2.0

SAML2AssertionSample.java

サンプルクラスは、`/sdk/samples/assertiongeneratorplugin` ディレクトリにあります。

実装によって `customizeAssertion` メソッド内に渡されるパラメータ文字列のコンテンツは、カスタム オブジェクトから提供されます。

アサーション ジェネレータ プラグインの展開

AssertionGeneratorPlugin 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、SiteMinder 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

- アサーション プラグイン Java ファイルをコンパイルします。

コンパイルには以下の `.jar` ファイルが必要となります。これらのファイルは、ポリシー サーバと共にインストールされています。

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

- 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。この変更により、変更されたクラスパスによるプラグインのロードが可能になります。

`installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

注: `xercesImpl.jar`、`xalan.jar`、`SMJavaApi.jar` のクラスパスを変更しないでください。

- プラグインを有効化します。

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、管理 UI 内で設定することにより、このプラグインを有効にします。UI パラメータにより、SiteMinder 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. 172)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. 管理 UI にログオンします。
2. [フェデレーション] - [レガシー フェデレーション] - [アフィリエイト] をクリックします。
3. 既存のアフィリエイト エントリを選択するか、新規に作成します。
4. [一般] 設定に移動します。
5. [アサーション ジェネレータ 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。

Java クラス名

既存のプラグインの Java クラス名を指定します。

プラグイン クラスは、アサーションの解析および変更を行った後、最終的な処理のために結果をアサーション ジェネレータに返すことができます。

各アフィリエイトには 1 つのプラグインのみが対応します。たとえば、次のようになります。

`com.mycompany.assertiongenerator.AssertionSample`

パラメータ

(オプション) [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。

注: 管理 UI でアサーション プラグインを有効にする代わりに、ポリシー管理 API (C または Perl) を使用して、プラグインを統合することができます。詳細については、「C 言語用 SiteMinder プログラミング ガイド」または「Java 言語用 SiteMinder プログラミング ガイド」を参照してください。

6. ポリシー サーバを再起動します。

ポリシー サーバを再起動することにより、再コンパイル後に最新のアプリケーション プラグインのバージョンが選択されます。

コンシューマ リソースへのリンクの作成 (SAML 1.x)

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、サイト間転送 URL にアクセスする必要があります。これにより、プロデューサ側 Web エージェントにリクエストが送信されます。その後、ユーザはコンシューマ サイトにリダイレクトされます。

SAML Artifact プロファイルの場合、サイト間転送 URL の構文は以下のとおりです。

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSTIONREF=QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url?query_parameter_name%3Dquery_parameter_value%26query_parameter_name%3Dquery_parameter_value&SMCONSUMERURL=http://consumer_site/affwebservices/public/samlcc&AUTHREQUIREMENT=2
```

SAML POST プロファイルの場合、サイト間転送 URL の構文は以下のとおりです。

```
http://producer_site/affwebservices/public/intersitetransfer?SMASSTIONREF=QUERY&NAME=affiliate_name&TARGET=http://consumer_site/target_url
```

サイト間転送 URL 内の変数を以下に示します。

producer_site

ユーザが認証される Web サイトを指定します。

affiliate_name

アフィリエイト ドメインで設定されたアフィリエイトの名前を示します。

consumer_site

ユーザがプロデューサ サイトからアクセスするサイトを示します。

target_url

コンシューマ サイトのターゲット ページ。

ユーザが選択するサイト間転送 URL には、後に続くテーブルにリストされたクエリ パラメータが含まれる必要があります。

注: SAML Artifact プロファイル用のクエリ 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。

クエリ パラメータ	意味
SMASSERTIONREF (必須)	内部使用のみ。値は常に QUERY です。この値は変更しないでください。
NAME (必須)	アフィリエイト ドメインで設定されたアフィリエイトの名前。
TARGET (必須)	コンシューマ サイトのターゲット URL。
SMCONSUMERURL (Artifact プロファイルのみに必要)	コンシューマ サイトの URL がアサーションを処理し、ユーザを認証します。 SAML 1.x Artifact バインドにおいて、値がアサーション コンシューマ URL に対して指定されている場合、その値がこのクエリ パラメータの値よりも優先されます。
AUTHREQUIREMENT = 2 (Artifact プロファイルのみに必要)	内部使用のみ。値は常に 2 です。この値は変更しないでください。

注: SAML POST プロファイルでは **SMCONSUMERURL** パラメータおよび **AUTHREQUIREMENT** パラメータは使用されません。ただし、これらのパラメータのいずれかをサイト間転送 URL に含める場合には、もう一方のパラメータも含める必要があります。

例: **Artifact** プロファイルの場合のサイト間転送 URL の例を以下に示します。

```
http://www.smartway.com/affwebservices/public/intersitetransfer?SMAS  
SERTIONREF=QUERY&NAME  
=ahealthco&TARGET=http://www.ahealthco.com:85/smartway/index.jsp&S  
MCONSUMERURL=  
http://www.ahealthco.com:85/affwebservices/public/samlcc&AUTHREQUIR  
EMENT=2
```

例: **POST** プロファイルの場合のサイト間転送 URL の例を以下に示します。

```
http://www.smartway.com/affwebservices/public/intersitetransfer?SMAS  
SERTIONREF  
=QUERY&NAME=ahealthco&TARGET=http://www.ahealthco.com/index.ht  
ml
```

サイト間転送 URL を保護するかどうかの選択

サイト間転送 URL リンクを持つ Web ページが、永続セッション用に設定された **SiteMinder** の保護レلمに含まれることがあります。ユーザが保護されたページ上でリンクの 1 つを選択すると、**SiteMinder** は認証のチャレンジをユーザに表示します。ユーザがログインした後、永続セッションが確立されることがあり、その場合、**SAML** アサーションの保存が必要になります。

これらのページが非保護の場合、プロデューサは、**SiteMinder** セッションを持たないアフィリエイトユーザを認証 URL に導きます。この URL は、ログインして **SiteMinder** セッションを受信するようにユーザに促します。管理 UI にアフィリエイトを設定する場合、認証 URL を定義します。

注: 永続セッションを設立するには、セッションストアを設定してください。セッションストアのセットアップは、ポリシー サーバ管理コンソールを使用して行います。

第 13 章: SAML 1.x コンシューマとしての設定

依存パートナーの前提条件

SiteMinder が依存パートナーとして機能するには、以下のタスクを完了する必要があります。

- ポリシー サーバをインストールします。
- 以下のコンポーネントのいずれかをインストールします。
 - Web エージェントおよび Web エージェント オプション パック。
Web エージェントは、ユーザを認証し、セッションを確立します。オプション パックはフェデレーション Web サービス アプリケーションを提供します。ネットワーク内の適切なシステムに必ず FWS アプリケーションを展開してください。
 - SPS フェデレーション ゲートウェイ (Web エージェントが組み込まれており、組み込まれた Tomcat Web サーバ上にフェデレーション Web サービス アプリケーションが配置されています)。

詳細については、「Web エージェント オプション パック ガイド」を参照してください。

- 検証およびメッセージの暗号化を必要とする機能のために秘密キーと証明書をインポートします。
- フェデレーション ネットワーク内でアサーティング パートナーをセットアップします。

SAML 1.x コンシューマを設定する方法

SAML 1.x コンシューマとして SiteMinder を設定するには、以下のタスクが必要となります。

1. SAML 1.x 認証方式の前提条件を実行します。
2. 認証方式タイプを選択し、これに名前を割り当てます。
3. SAML 1.x 認証方式で認証されるユーザのネームスペースを指定します。

4. このコンシューマがサポートするシングル サインオン プロファイル (Artifact または POST) を選択します。
5. フェデレーション パートナーとしてアサーションを生成するプロデューサごとに SAML 認証方式を設定します。各方式をレルムにバインドします。レルムには、フェデレーション リソースのターゲット URL が含まれていることが必要です。これらのリソースを SiteMinder ポリシーで保護します。

ヒント :

- 設定が機能するためには、プロデューサとコンシューマの特定のパラメータ値が一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」 (Configuration Settings that Must Use the Same Values) に記載されています。
- フェデレーション Web サービス サブレットの正しい URL を使用していることを確認してください。これらの URL は、「[SiteMinder 設定で使用されるフェデレーション Web サービス URL \(P. 457\)](#)」 (Federation Web Services URLs Used in SiteMinder Configuration) に一覧表示されています。

コンシューマに関するオプション設定タスク

コンシューマを設定する際には、以下のオプション タスクがあります。

- メッセージ コンシューマ プラグインを使用したアサーションのカスタマイズ
- 失敗した認証の試行のリダイレクト

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。

- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

SAML 1.x 認証方式

コンシューマは、SAML 1.x アサーションを使用してユーザを認証するためのサイトです。

注: 1つのサイトが SAML プロデューサと SAML コンシューマとして機能できます。

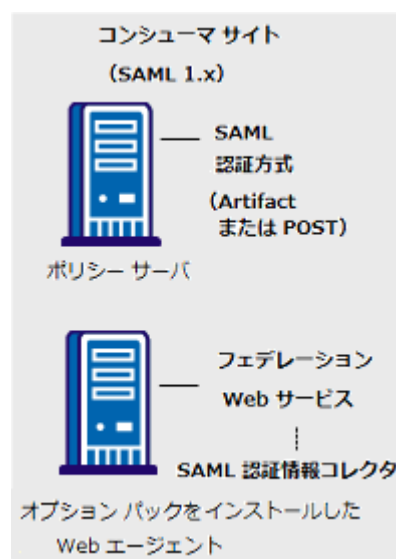
レガシー フェデレーション 機能を持つ任意の SiteMinder サイトが SAML 1.x アサーションを消費することができ、これらのアサーションを使用してユーザを認証できます。アサーションが消費される場合、サイトでは、認証プロセスを完了するために、アサーションからの情報をユーザ ディレクトリと対照して比較することが必要になります。

SiteMinder では、以下の SAML 1.x 認証方式が提供されます。

- SAML Artifact プロファイル
- SAML POST プロファイル

SAML ベースの認証方式では、コンシューマ サイトがユーザを認証できます。SAML アサーションを消費し、SiteMinder セッションを確立することにより、クロスドメイン シングル サインオンが可能になります。ユーザが識別された後、コンシューマ サイトでは、特定のリソースに対してユーザに認可を与えることができます。

次の図は、コンシューマ サイトでの認証に関する主要コンポーネントを示します。



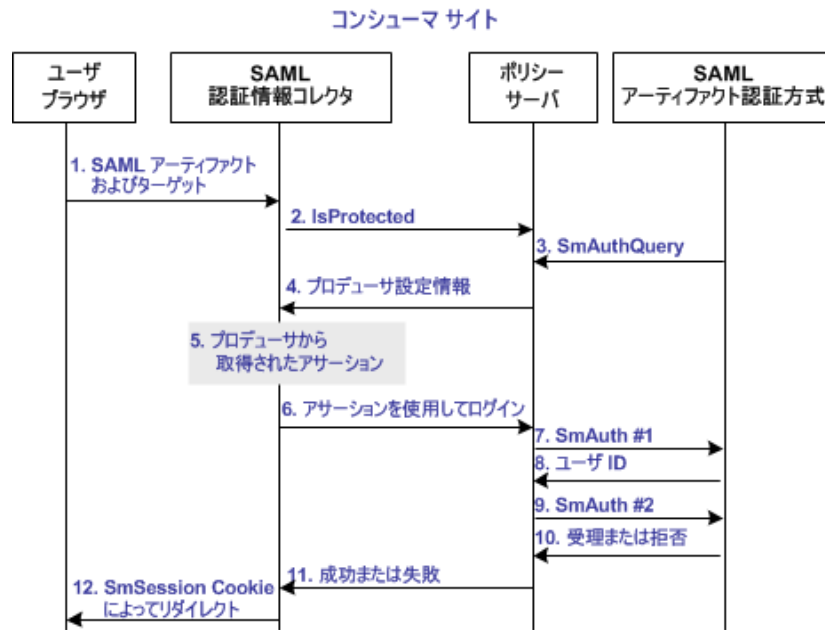
注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシ サーバ管理ガイド*」(Secure Proxy Server Administration Guide)を参照してください。

コンシューマ側のポリシー サーバで SAML 1.x 認証方式が設定されています。SAML 認証情報コレクタは、フェデレーション Web サービス アプリケーションのコンポーネントです。認証情報コレクタは、コンシューマの側 Web エージェントまたは SPS フェデレーション ゲートウェイにインストールされます。認証情報コレクタは、ポリシー サーバにおいて SAML 認証方式から情報を取得し、その情報を使用して SAML アサーションにアクセスします。

SAML アサーションは、コンシューマ サイトでポリシー サーバへのアクセス権を付与する認証情報になります。ユーザの認証および認可が行われ、認可が成功すると、このユーザはターゲット リソースにリダイレクトされます。

SAML 1.x Artifact 認証方式の概要

次の図は、SAML 1.x Artifact 認証方式がどのようにリクエストを処理するかを示しています。



注: SPS フェデレーション ゲートウェイ（すなわち Web エージェントと Web エージェント オプションパック）がエージェントおよび SAML の認証情報コレクタ機能を提供します。

特に明記されている場合を除き、このプロセスにおけるすべてのアクティビティはコンシューマサイトで実行されます。

1. ユーザが SAML Artifact およびターゲット URL と共に SAML 認証情報コレクタにリダイレクトされます。

Artifact およびターゲット URL は、プロデューササイトで Web エージェントから生成されます。

2. SAML 認証情報コレクタは、リクエストされたリソースを SAML Artifact 認証方式が保護しているかどうかを判別するために、ポリシーサーバをコールします。
3. ポリシーサーバは、必要なデータを SAML Artifact 認証方式に渡し、ここでプロデューサ設定情報が抽出されます。

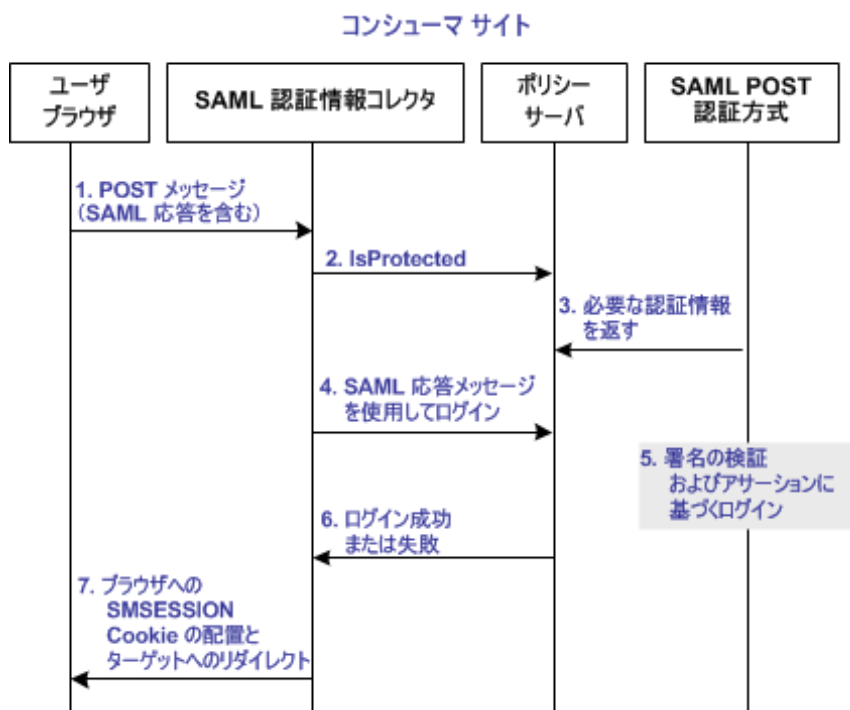
4. ポリシー サーバは、プロデューサ設定情報を **SAML 認証情報コレクタ** に返します。この情報を使用して、認証情報コレクタ サブレットがプロデューササイトをコールし、**SAML アサーション**を取得できるようになります。
5. **SAML 認証情報コレクタ**は、ポリシー サーバからデータを取得し、これを使用して **SAML アサーション**を取得します。
6. アサーションが返された後、認証情報コレクタは、アサーションを認証情報として使用し、ポリシー サーバにログインします。
7. ポリシー サーバは、**SAML 認証方式**に対して最初のユーザ不明瞭解消コールを実行します。
8. **SAML 認証方式**は、認証方式データおよびアサーションを使用してユーザを特定し、ユーザの固有の識別番号を認証情報コレクタに返します。
9. ポリシー サーバは、**SAML 認証方式**に対して 2 番目のユーザ認証コールを実行します。

注: SiteMinder 認証 AP により、2 段階の認証プロセスが指定されます。詳細については、「**C 言語用 SiteMinder プログラミング ガイド**」または「**Java 言語用 SiteMinder プログラミング ガイド**」を参照してください。

10. **SAML 認証方式**は、**SAML アサーション**を検証し、受理または拒否のメッセージをポリシー サーバに返します。
11. ポリシー サーバは、受理または拒否のメッセージを認証情報コレクタに送信します。
12. **SAML 認証情報コレクタ**は、セッション **Cookie**を作成し、これをブラウザに配置した後、ユーザをターゲットリソースにリダイレクトします。ログインに失敗した場合、認証情報コレクタは、ユーザを **No Access URL**にリダイレクトします。

SAML 1.x POST プロファイル認証方式の概要

次の図は、SAML 1.x POST プロファイル認証方式がどのようにリクエストを処理するかを示しています。



注: SPS フェデレーション ゲートウェイまたは Web エージェント オプションパックが SAML 認証情報コレクタ機能を提供します。

特に明記されている場合を除き、以下のプロセスはコンシューマ サイトで実行されます。

1. ブラウザが HTML フォームを SAML 認証情報コレクタ URL にポストします。このフォームには、プロデューサで生成された SAML レスポンス メッセージおよびターゲット URL のアドレスが含まれます。
2. SAML 認証情報コレクタは、ターゲット リソースが保護されているかどうかを判別するために、ポリシー サーバにアクセスします。
3. ポリシー サーバは、SAML POST プロファイル認証方式がターゲット URL を保護しているという応答を返します。ポストされたフォームから署名付きのレスポンスが返されます。これが、ログイン コールに必要な認証情報です。
4. SAML 認証情報コレクタは、ポリシー サーバにログイン コールを送信し、デジタル署名された SAML レスポンスを認証情報として渡します。

5. SAML POST プロファイル認証方式は、署名、およびレスポンスとアサーションの他のフィールドを確認します。
6. 確認が終了し、ユーザがディレクトリ内で検出されると、認証は成功します。いずれかの確認に失敗した場合、認証が失敗となります。
7. SAML 認証情報コレクタは、**SMSESSION Cookie** を作成します。この **Cookie** はブラウザに配置され、ユーザはターゲットリソースにリダイレクトされます。ログインに失敗した場合、認証情報コレクタは、設定済みの **No Access URL** にユーザをリダイレクトします。

SAML 1.x 認証方式の前提条件

SAML 認証方式を設定するための前提条件は以下のとおりです。

- プロデューサおよびコンシューマで **SiteMinder** ポリシー サーバをインストールします。
- プロデューサおよびコンシューマで フェデレーション **Web** サービスをインストールします。
- SAML POST レスポンスに署名するための証明書データ ストアを準備します。

SiteMinder ポリシー サーバをインストールします

SiteMinder ポリシー サーバには レガシー フェデレーション 機能が含まれます。

ポリシー サーバをインストールするには、「ポリシー サーバインストール ガイド」を参照してください。

プロデューサおよびコンシューマでのフェデレーション Web サービスのインストール

フェデレーション Web サービス (FWS) は、Web アプリケーションの 1 つです。FWS は、SAML 認証情報コレクタ サブレットを提供します。このサブレットは、アサーションおよびフェデレーション ネットワーク設定のための他のサービスを消費します。

FWS アプリケーション機能を使用するには、Web エージェントおよび Web エージェント オプションパックインストールするか、SPS フェデレーション ゲートウェイをインストールします (SPS フェデレーション ゲートウェイには、プロデューサおよびコンシューマのサイトで FWS が組み込まれています)。

インストールおよび設定の手順については、以下のガイドを参照してください。

- *Web エージェント オプションパック ガイド*
- *Secure プロキシ サーバ管理ガイド (Secure Proxy Server Administration Guide)*

DefaultAgentName 設定に対する値の指定

Web エージェントをインストールする際に、すべてのコンシューマ Web エージェントに対する Web エージェント パラメータ DefaultAgentName の値を定義します。この値は、Web エージェント識別情報を指定します。

DefaultAgentName を識別する指定エージェントを、ターゲット リソースを保護するレルムのリソース フィルタに含めます。エージェント設定オブジェクトまたはローカル エージェント設定ファイルに

DefaultAgentName パラメータを設定します。DefaultAgentName パラメータを省略する場合や、レルム リソース フィルタ内の AgentName パラメータで指定された値を使用する場合には、シングル サインオン プロファイルと無関係に SAML 1.x 認証が失敗します。

POST レスポンスの署名および検証を行うための証明書データストアのセットアップ

SAML POST プロファイルを使用してアサーションを渡すには、そのアサーションが含まれる SAML レスポンスにプロデューサが署名する必要があります。コンシューマ サイトのアサーション コンシューマは、その署名を検証する必要があります。

これらのタスクを実行するには、署名、検証またはその両方を行うための秘密キー/証明書ペアを証明書データストアに追加します。証明書データストアを使用することで、SAML レスポンスの署名や検証に必要なキーおよび証明書の管理や取得を行えます。

証明書データストアの詳細については、「ポリシー サーバ設定ガイド」を参照してください。

SAML 1.x Artifact 認証の設定

SAML Artifact 認証方式をレルムに割り当てるには、方式を設定する必要があります。

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] - [認証方式] の順に移動します。
2. [認証方式の作成] をクリックします。
3. [認証方式タイプの新規オブジェクトの作成] を選択します。
[認証方式] ページが表示されます。
4. 認証方式の名前を入力します。
5. [認証方式タイプ] ドロップダウン リストから、[SAML Artifact テンプレート] を選択します。

SAML Artifact に対応して [認証方式] ダイアログ ボックスの内容が変化します。

6. 方式セットアップを設定します。

設定の説明については、[ヘルプ] をクリックしてください。

重要: [アフィリエイト名]、[パスワード]、および [パスワードの確認] フィールドは、フェデレーション ネットワーク内の他の値に一致する必要があります。詳細については、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」を参照してください。

7. (オプション) [デフォルト ターゲット URL] フィールドにターゲット リソースを指定します。このフィールドは、ページの [追加設定] セクションにあります。ターゲットは、コンシューマの保護された フェデレーション リソースです。

コンシューマは、デフォルト ターゲットを使用する必要はありません。シングルサインオンを開始するリンクには、ターゲットを指定するクエリ パラメータが含まれます。

または、認証レスポンス URL の TARGET クエリ パラメータの値を使用して、ターゲット リソースを指定します。このオプションを有効にするには、[クエリ パラメータ TARGET でデフォルト ターゲット URL 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。

8. (オプション) メッセージ コンシューマ API などの機能を設定し、[追加設定] セクションで認証エラーの URL をリダイレクトします。
9. [OK] をクリックして方式を保存します。

以上で、SAML 1.x Artifact 認証方式が設定されました。

詳細情報:

[HTTP Artifact SSO のバックチャネル設定 \(P. 188\)](#)

HTTP Artifact SSO のバックチャネル設定

SAML Artifact プロファイルの場合、アサーティング パーティは、バックチャネル上でアサーションをコンシューマに送信します。認証方式を使用してバックチャネルを保護します。ベーシックまたはクライアント証明書認証方式を使用して、バックチャネルのセキュリティ保護を行えます。

■ ベーシック認証

ベーシック認証を使用し、SiteMinder が両方のパートナーに存在する場合、各サイトのアフィリエイト名はコンシューマの名前です。アサーティング パーティが SiteMinder ではない場合、アサーティング パーティ管理者がこのサイトを識別するために使用している名前をアサーティング パーティ管理者が提供する必要があります。アフィリエイト名として提供された名前を認証方式設定に指定してください。

■ クライアント証明書認証

バックチャネルにクライアント証明書認証を使用する場合、管理 UI 内のアフィリエイト名はクライアント証明書のエイリアスであることが必要です。さらに、証明書件名の CN がアフィリエイト名に一致する必要があります。アフィリエイト名、エイリアス、および CN が一致する必要があります。

ポリシー サーバは、FIPS 140 以外で暗号化された証明書を使用して、バックチャネル上のクライアント証明書認証をサポートします。ポリシーサーバが FIPS のみのモードで作動している場合であっても同様です。ただし、厳密な FIPS のみのインストールの場合、FIPS 140 互換のアルゴリズムで暗号化された証明書のみを使用してください。

クライアント証明書は、証明書データ ストアに保存されます。

SAML 1.x POST プロファイル認証の設定

SAML POST プロファイル認証方式を設定する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
2. [認証方式の作成] をクリックします。
3. [認証方式タイプの新規オブジェクトの作成] を選択します。
[認証方式] ページが表示されます。
4. 認証方式の名前を入力します。

5. [認証方式タイプ] ドロップダウン リストから、[SAML POST テンプレート] を選択します。

SAML POST 方式に対応して [認証方式] ダイアログ ボックスの内容が変化します。

6. 方式セットアップを設定します。

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。

重要: [アフィリエイト名]、[パスワード]、および [パスワードの確認] フィールドは、フェデレーション ネットワーク内の他の値に一致する必要があります。詳細については、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」を参照してください。

7. (オプション) [デフォルト ターゲット URL] フィールドにターゲット リソースを指定します。このフィールドは、ページの [追加設定] セクションにあります。ターゲットは、コンシューマの保護された フェデレーション リソースです。

コンシューマは、デフォルト ターゲットを使用する必要はありません。シングルサインオンを開始するリンクには、ターゲットを指定するクエリ パラメータが含まれます。

または、認証レスポンス URL の TARGET クエリ パラメータの値を使用して、ターゲット リソースを指定します。このオプションを有効にするには、[クエリ パラメータ TARGET でデフォルト ターゲット URL 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。

8. (オプション) メッセージ コンシューマ API などの機能を設定し、[追加設定] セクションで認証エラーの URL をリダイレクトします。
9. [OK] をクリックして方式を保存します。

以上で、SAML 1.x POST 認証方式が設定されました。

メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ

メッセージ コンシューマ プラグインは、メッセージ コンシューマ プラグインを実装する 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。

注: 認証と不明瞭解消に関するステータス コードの詳細については、「Java 用 SiteMinder 用プログラミング ガイド」を参照してください。

認証時、SiteMinder 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、SiteMinder 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、SiteMinder 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、SiteMinder 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、SiteMinder 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。

メッセージ コンシューマ プラグインに関する追加の情報については、以下を参照してください。

- 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「*Java Developer Reference*」にあります。 `MessageConsumerPlugin` インターフェースを参照してください。
- 認証および認可の API に関する概要および概念情報については、「Java 用 SiteMinder プログラミング ガイド」を参照してください。

プラグインを設定する方法

1. SiteMinder SDK をインストールします（未インストールの場合）。
2. MessageconsumerPlugin.java インターフェースを実装します（SiteMinder SDK に含まれています）。
3. メッセージ コンシューマ プラグイン実装クラスを展開します。
4. 管理 UI でメッセージ コンシューマ プラグインを有効にします。

MessageConsumerPlugin の実装

MessageConsumerPlugin.java インターフェースを実装するにより、カスタム メッセージ コンシューマ プラグインを作成します。実装クラスの最小要件は、以下の手順に示されています。

次の手順に従ってください:

1. パラメータが含まれない公のデフォルト コンストラクタ メソッドを提供します。
2. 実装がステートレスになるように、コードを提供します。多数のスレッドが 1 つのプラグインクラスを使用できる必要があります。
3. 現実の要件に応じて、インターフェース内のメソッドを実装します。

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

`init()`

プラグインが必要とする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、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」の下に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最終結果を決定する追加のコードを提供します。

SiteMinder から、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
(*installation_home*¥sdk¥samples¥messageconsumerplugin)

MessageConsumerSAML20.java
(*installation_home*¥sdk¥samples¥authextensionsaml20)

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、SiteMinder が実行可能ファイルを検索できることを確認します。

メッセージ コンシューマ プラグインを展開するには、以下の手順を実行します。

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、ポリシー サーバと共にインストールされています。

installation_home¥siteminder¥bin¥jars¥SmJavaApi.jar

SmJavaApi.jar の同一のコピーが SiteMinder SDK と共にインストールされています。このファイルは、*installation_home*¥sdk¥Java¥SmJavaApi.jar ディレクトリにあります。

開発時にそれらのいずれも使用できます。

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。 `installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

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ポリシー サーバの再起動 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ことに必要です。
4. プラグインを有効化します。

SAML 1.x のメッセージ コンシューマ プラグインの有効化

メッセージ コンシューマ プラグインを作成してコンパイルした後に、管理 UI 内で設定することにより、このプラグインを有効にします。UI 設定により、`SiteMinder` にプラグインの検索場所が指定されます。

[プラグインを展開](#) (P. 192)するまで、プラグイン設定を実行しないでください。

メッセージ コンシューマ プラグインを有効にする方法

1. 管理 UI にログオンします。
2. 該当する **SAML 1.x** 方式の [認証方式] ダイアログ ボックスに移動します。 [追加設定] セクションで、以下のフィールドに入力します。

完全 Java クラス名

プラグインの Java クラス名を指定します。たとえば、`SiteMinder 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`

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。

管理 UI でプラグインを設定する代わりに、ポリシー管理 API (C または Perl) を使用して、`IdpPluginClass` および `IdpPluginParameters` を設定します。

3. ポリシー サーバを再起動します。

SAML 1.x 認証試行が失敗した後のユーザのリダイレクト

コンシューマがシングルサインオン トランザクション中にユーザを認証できない場合、コンシューマは、後続の処理用にカスタマイズされた URL にそのユーザをリダイレクトできます。

認証が失敗した場合の処理として複数のリダイレクト URL をオプションで設定できます。これらのリダイレクト URL を使用することで、ユーザのリダイレクトをより詳細に制御できます。たとえば、ユーザストアでユーザを検索できない場合、ユーザが見つからない場合のリダイレクト URL に入力することができます。

「ステータス リダイレクト URL とモード」は、「認証ダイアログ ボックス」の「追加構成」セクションにあります。リダイレクト URL は、特定のステータス条件に対応します。

- ユーザが見つかりません
- シングルサインオン メッセージが無効です
- ユーザ認証情報は受理されません

いずれかの状態が発生すると、リダイレクト URL は、後続アクション用のアプリケーションまたはカスタマイズされたエラー ページにユーザをリダイレクトできます。

注: リダイレクト URL の設定は必須ではありません。

リダイレクト URL を設定しない場合、SiteMinder の標準の処理が行われます。認証が失敗した場合の処理方法は、認証方式の設定によって異なります。

ステータス リダイレクト URL を設定する方法

1. SAML Artifact または SAML POST 認証方式のページに移動します。
2. 「ステータス リダイレクト URL とモード」セクションで、1つまたは複数のフィールドに URL に入力します。

設定の説明については、「ヘルプ」をクリックしてください。

フェデレーション Web サービスでは、設定されたリダイレクト URL の 1 つに認証理由をマッピングすることにより、エラーを処理します。エラーを報告するために、ユーザがその URL にリダイレクトされることがあります。

3. 以下のモードのいずれかを選択します。

- 302 データなし
- HTTP POST

4. [OK] をクリックして、変更を保存します。

注: これらのリダイレクト URL を SiteMinder メッセージ コンシューマ プラグインと共に使用して、より詳細なアサーション処理を実行できます。認証が失敗した場合、プラグインは、指定されたリダイレクト URL の 1 つにユーザをリダイレクトできます。

詳細情報:

[SAML 1.x Artifact 認証の設定 \(P. 186\)](#)

[SAML 1.x POST プロファイル認証の設定 \(P. 188\)](#)

HTTP ヘッダとしての SAML 属性の供給

アサーション レスポンスには、アサーションの属性を含めることができます。これらの属性は HTTP ヘッダ変数として提供できるため、クライアント アプリケーションでは、これらを使用して、より細かい粒度のアクセス制御を実行できます。

属性を HTTP ヘッダに含める場合には、以下のような利点があります。

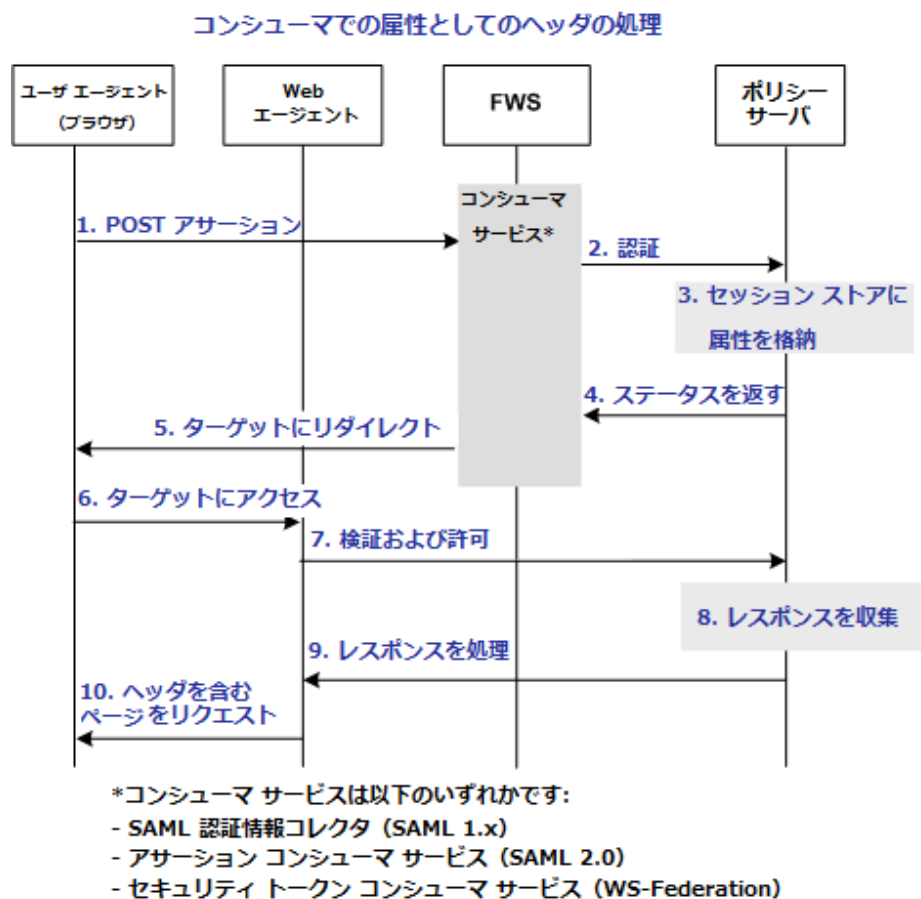
- HTTP ヘッダは永続的ではありません。これらのヘッダは、それらが含まれるリクエストまたはレスポンス内のみに存在します。
- HTTP ヘッダは、SiteMinder Web エージェントによって提供される場合と同様に、ブラウザ内で非表示であるため、セキュリティ上の心配が軽減されます。

注: HTTP ヘッダにはサイズの制限があり、属性はその制限を超えることができません。SiteMinder では、ヘッダに関する Web サーバのサイズ制限に達するまで、属性をヘッダに送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダサイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。

HTTP ヘッダとしての SAML 属性のユース ケース

認証時、一連の SAML 属性がアサーションから抽出され、HTTP ヘッダとして提供されます。認可プロセス時、これらのヘッダはカスタムアプリケーションに返されます。

次のフロー図は、ランタイムのイベントのシーケンスを表します。



注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション 機能を提供できます。フロー図では、Web エージェント ブロックは SPS フェデレーション ゲートウェイに組み込まれた Web エージェント になります。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシ サーバ 管理 ガイド*」 (Secure Proxy Server Administration Guide) を参照してください。

属性を HTTP ヘッダとして処理する場合のイベントのシーケンスは以下のとおりです。

1. アサーションがアサーティング パーティで生成された後、アサーションが依存パーティの適切なコンシューマサービスに送信されます。配信方法 (POST または **Artifact** または **WS フェデレーション**) には無関係です。

注: コンシューマ サービスは、**SAML 認証情報コレクタ (SAML 1.x の場合)**、**アサーション コンシューマ サービス (SAML 2.0 の場合)**、または**セキュリティ トークン コンシューマ サービス (WS フェデレーションの場合)** です。

2. コンシューマサービスは、そのローカル ポリシー サーバをコールし、設定済みの認証方式を使用してアサーションによるユーザ認証を実行します。
3. 認証方式のリダイレクト モードパラメータが **PersistAttributes** に設定されている場合、ポリシー サーバはセッション変数として属性をセッションストアにキャッシュします。
4. 認証の結果はコンシューマ サービスに返されます。
5. コンシューマ サービスは、保護されたターゲット リソースにブラウザをリダイレクトします。
6. ブラウザは、ターゲット リソースへのアクセスを試みます。
7. **Web** エージェントは、ポリシー サーバをコールして、ユーザセッションを検証すると共に、ユーザのターゲット リソースへのアクセスが許可されていることを確認します。
8. ポリシー サーバは、設定済みのレスポンスによって属性を取得します。
9. ポリシー サーバは、レスポンスを処理し、属性を **Web** エージェントに送信します。
10. **Web** エージェントは、必要に応じて HTTP ヘッダを設定します。

属性を HTTP ヘッダとして提供するため設定の概要

セッションストアにキャッシュされた SAML 属性を取得し、それらを HTTP ヘッダとして提供するには、複数の設定手順が必要となります。

次の手順に従ってください:

1. SAML 認証方式のリダイレクトモードとして **PersistAttributes** を選択します。これにより、SAML 属性が HTTP ヘッダとして返されるようになります。
2. ターゲット リソースが含まれるレルムに対する認可ルールを設定します。
3. ターゲット リソースを保護するレルムで **PersistentRealm** を設定します。
4. ヘッダとして提供される SAML 属性ごとに、アクティブなレスポンスタイプを使用するレスポンスを設定します。
5. 認可ルールとアクティブなレスポンスをバインドして属性のユーザを HTTP ヘッダとして実装するポリシーを作成します。

SAML 属性を保存するためのリダイレクトモードの設定

依存パーティが SAML アサーションでユーザを認証した後、SAML 属性はセッションストアに書き込まれます。その後、ブラウザはターゲット リソースにリダイレクトされます。

属性データを持つブラウザをリダイレクトする方法

1. 管理 UI にログインします。
2. SAML 認証方式の設定ページに移動します。

3. [リダイレクトモード] パラメータを[属性を保持する]に設定します。[リダイレクトモード] フィールドの場所は以下のように特定されます。

SAML 1.x

[リダイレクトモード] は、設定のメインページの[方式のセットアップ] セクションにあります。

SAML 2.0

[SAML 2.0 設定] - [SSO] を選択します。[リダイレクトモード] は、ページの [SSO] セクションにあります。

WS-フェデレーション

[WS-Federation の設定] - [SAML プロファイル] をクリックします。[リダイレクトモード] は、ページの [SSO] セクションにあります。

4. [サブミット] をクリックして、変更を保存します。

以上で、属性データを渡すようにリダイレクトモードが設定されました。

ユーザを検証するための認可ルールの作成

保護されたターゲット リソースが含まれるレルムの場合、セッションストアから SAML 属性を取得するルールを作成します。

ルールは、認可イベント (onAccessAccept) に基づきます。ユーザは、FWS アプリケーションによってすでに認証されています。Web エージェントは、ユーザを再認証して HTTP ヘッダを渡すことはできません。属性の取得は、認可の段階で行われます。

レルムの OnAccessAccept ルールを作成する方法

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [レルム] と順に移動します。
3. ターゲット リソースが含まれるレルムを選択します。
4. [ルール] セクションの[作成] をクリックします。
[ルールの作成] ページが表示されます。
5. 名前、および必要に応じて説明を入力します。
6. [リソース] フィールドにアスタリスク (*) を入力します。

7. [アクション] セッションで [許可イベント] および [OnAccessAccept] を選択します。
8. [許可/拒否と有効/無効] セクションで [有効] を選択します。
9. [OK] ボタンをクリックしてルールを保存します。

以上で、保護されたリソースのあるレルムに対して認可ルールが定義されました。

属性を HTTP ヘッダとして送信するレスポンスの設定

SAML 属性を HTTP ヘッダとして Web エージェントに送信するレスポンスを設定します。Web エージェントは、レスポンスを処理して、ヘッダ変数がクライアント アプリケーションに利用可能になるようにします。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
3. ターゲット リソースのドメインを選択し、[変更] をクリックします。
4. [レスポンス] タブを選択します。
5. [作成] をクリックします。
[レスポンス] ダイアログが表示されます。
6. 名前を入力します。
7. エージェントのタイプが **SiteMinder Web** エージェントであることを確認します。
8. [レスポンス属性の作成] をクリックします。
[レスポンス属性] ダイアログが表示されます。
9. [属性] フィールドで [WebAgent-HTTP-Header-Variable] を選択します。
10. [属性の種類] の [アクティブ レスポンス] を選択します。

11. フィールドには以下のように入力します。

変数名

ヘッダ変数の名前を指定します。この名前が割り当てられます。

ライブラリ名

`smfedattrresponse`

この値がこのフィールドに入力される必要があります。

関数名

`getAttributeValue`

この値がこのフィールドに入力される必要があります。

パラメータ

アサーションに表示される属性名を指定します。

アサーション内の属性は、フェデレーション パートナーとの合意に基づいて決定してください。

12. [OK] をクリックして属性を保存します。
13. HTTP ヘッダ変数の各属性について手順を繰り返します。1 つのレスポンスに対して複数の属性を設定できます。
- [レスポンス] タブに戻ります。作成して属性は、[属性リスト] セクションに一覧表示されます。
14. [OK] をクリックしてレスポンスを保存します。
- [レスポンス] タブに戻ります。
15. [サブミット] をクリックしてドメインを保存します。

レスポンスは、HTTP ヘッダになるように属性を Web エージェント上に送信します。

属性を HTTP ヘッダとして実装するためのポリシーの作成

SAML 属性の使用を HTTP ヘッダとして実装するには、認可イベント ルールおよびアクティブ レスポンスを 1 つのポリシーにまとめます。

次の手順に従ってください:

1. 管理 UI にログオンします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。

3. ターゲット リソースが含まれるドメインを選択し、[変更] をクリックします。
4. [ポリシー] タブを選択し、[ポリシー] セクションの [作成] をクリックします。
[ポリシーの作成] ダイアログが表示されます。
5. [名前] フィールドにわかりやすい名前を入力します。
6. 保護されたリソースへのアクセスを許可するユーザを [ユーザ] タブで選択します。
7. 先に [ルール] タブで作成した認可ルールを追加します。
8. 認可ルールを選択し、[レスポンスの追加] をクリックします。
[使用可能なレスポンス] ダイアログ ボックスが表示されます。
9. 先に作成したアクティブ レスポンスを選択し、[OK] をクリックします。
[ルール] タブに戻ります。レスポンスが認証ルールと共に表示されます。
10. [サブミット] をクリックしてポリシーを保存します。

SAML 属性を HTTP ヘッダとして使用できるようにするためのポリシーが作成されました。

バックチャネル対応のクライアント証明書認証の有効化(オプション)

HTTP-Artifact シングル サインオンを使用する場合、プロデューサでアサーション検索サービスを保護するためにクライアント証明書認証を選択できます。このサービスは、アサーションを取得してコンシューマに送信します。

注: クライアント証明書認証はオプションです。ベーシック認証を使用することもできます。

SAML 認証情報コレクタは、SAML Artifact 認証方式を呼び出します。SAML 認証情報コレクタは、認証方式から情報を収集して、SAML アサーションをアサーションから取得します。アサーション検索サービスが含まれるレルムに使用する認証方式の指定が要求されます。SAML 認証情報コレクタは、アサーションを取得するために提供する認証情報のタイプを決定します。

アサーション検索サービスがクライアント証明書認証方式で保護される場合は、以下の設定タスクを完了してください。

1. [クライアント証明書を証明書データストアに追加](#) (P. 203) します。
2. [バックチャネル認証のクライアント証明書オプションを選択](#) (P. 204) します。証明書は必須の認証情報です。

注: アサーティング側のポリシー サーバの管理者は、アサーション検索サービスを保護するためのポリシーを設定しておく必要があります。このポリシーのレルムは、X.509 クライアント証明書認証方式を使用する必要があります。

証明書データストアへのクライアント証明書の追加

認証機関から秘密キー/証明書ペアを取得する必要があります。管理 UI を使用して、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。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。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/証明書ペアをインポートする場合、割り当てるエイリアスは、認証方式設定の [アフィリエイト名] フィールドと同じ値であることが必要です。さらに、証明書の件名の CN 属性も [アフィリエイト名] フィールドに一致する必要があります。たとえば、アフィリエイト名が CompanyA であるとしします。すると、エイリアスは CompanyA であることが必要です。また、件名の CN の値に CN=CompanyA, OU=Development, O=CA, L=Islandia, ST=NY, C=US と記述される必要があります。

重要: 認証方式内の [アフィリエイト名] フィールドは、プロデューサでアフィリエイトオブジェクトに割り当てられる名前に一致する必要があります。SiteMinder がプロデューサである場合、認証方式内の [アフィリエイト名] は、オブジェクトの [一般] 設定の [名前] フィールドに一致する必要があります。

バックチャネル認証のクライアント証明書オプションの選択

プロデューサで、コンシューマがアサーション検索サービスにアクセスしようとするときに、認証情報として証明書を示すには、クライアント証明書オプションを選択します。

クライアント証明書オプションを選択する方法

1. [SAML Artifact 認証方式] ダイアログ ボックスの [方式のセットアップ] セクションに移動します。
2. [認証] フィールドで [クライアント証明書] を選択します。

SAML 1.x 認証方式によってリソースを保護する方法

SAML 1.x 認証方式を使用する SiteMinder ポリシーを設定することにより、ターゲット フェデレーション リソースを保護します。

次の手順に従ってください:

1. SAML 認証方式を使用するレلمを作成します。レلمは、ターゲット リソースを集めたものです。
レلمは、以下の方法で作成できます。
 - すでに設定済みの認証方式ごとに[固有なレلمを作成](#) (P. 205)する。
 - 対応する SAML 認証方式へのリクエストの送信にカスタム認証方式を使用する[単一のターゲット レلمを設定](#) (P. 206)する。すべてのプロデューサに対して単一のターゲットを持つレلمを 1 つ設定することにより、SAML 認証のレلم設定が簡単になります。
2. 関連のルールを設定します。また、必要に応じてレスポンスを設定します。
3. レلم、ルール、およびレスポンスを、ターゲット リソースを保護するポリシーとしてグループ化します。

重要: レلم内の各ターゲット URL もサイト間転送 URL で識別されます。サイト間転送 URL により、ユーザがプロデューサからコンシューマにリダイレクトされます。この URL は、URL TARGET 変数で指定します。プロデューササイトにおいて、管理者は、ユーザをコンシューマにリダイレクトするリンクにこの URL を含めます。

各認証方式に対する固有なレルムの設定

SAML または WS フェデレーション認証方式ごとに固有のレルムを設定する手順は、レルムを作成するための標準的な手順に従います。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] の順に移動します。
ドメインを作成するためのページが表示されます。
2. [ドメインの作成] をクリックします。
3. ドメイン名を入力します。
4. ユーザディレクトリをドメインに追加します。このディレクトリは、フェデレーション リソースへのアクセスをリクエストするユーザが含まれるディレクトリです。
5. [レルム] タブを選択し、レルムを作成します。
 - [エージェント] フィールドで、ターゲット リソースが存在する Web サーバを保護する Web エージェントを選択します。
 - [認証方式] フィールドで該当する認証方式を選択します。
6. レルムのルールを作成します。
ルールの一部として、ユーザの認証時の処理を制御するために使用可能なアクション (Get、Post、または Put) を選択します。
7. [ポリシー] タブを選択し、ターゲット フェデレーション リソースを保護するポリシーを設定します。先に作成したレルムをこのポリシーと関連付けます。

以上で、固有のレルムを持つポリシーがフェデレーション リソースを保護します。

すべての認証方式に対する単一のターゲットレールの設定

認証方式に対するレールの設定を簡略化するには、アサーションを生成する複数のサイトに対して単一のターゲットレールを作成します。

このタスクを実行するには、以下のコンポーネントをセットアップします

- 単一のカスタム認証方式

このカスタム方式は、すでに各アサーティングパーティに対して設定済みの対応する SAML または WS フェデレーション認証方式にリクエストを転送します。

- 1つのターゲット URL を持つ単一のレール

単一のターゲットレールに対する認証方式の作成

単一のターゲットレールに対するカスタム認証方式を定義するには、以下の作業を実行する必要があります。

- 認証方式を設定する方法
- リソースリクエストに適用する認証方式をポリシーサーバに指定するためのカスタム方式のパラメータを定義します。

まず、SAML または WS フェデレーション認証方式が設定済みであることを確認します。設定されていない場合は、これらの方式を設定して、カスタム方式が参照できるようにします。

認証方式を作成する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。
2. 使用しているプロトコルの手順に従って 1 つまたは複数の認証方式を作成します。
3. [OK] をクリックして終了します。

詳細情報:

[SAML 1.x 認証方式 \(P. 179\)](#)

[WS フェデレーション認証方式の概要 \(P. 372\)](#)

[SAML 2.0 認証方式を設定する方法 \(P. 291\)](#)

カスタム認証方式の設定

単一のターゲット レルムが正しく動作するためには、特定のカスタム認証方式に依存します。

単一のターゲットレルムに対するカスタム認証方式を設定する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。

2. フィールドには以下のように入力します。

名前

カスタム認証方式のわかりやすい名前を入力します (SAML Custom Auth Scheme など)。

3. [各方式共通セットアップ] セクションで、以下のフィールドに入力します。

認証方式タイプ

カスタム テンプレート

保護レベル

デフォルトをそのまま使用するか、新しいレベルを設定します。

4. [方式のセットアップ] セクションで、以下のフィールドに入力します。

ライブラリ

smauthsinglefed

秘密キー

このフィールドは空のままにします。

秘密キーの確認入力

このフィールドは空のままにします。

パラメータ

以下のパラメータのいずれかを指定します。

- SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

使用する SAML 認証方式名のリストを指定します。

artifact_producer1 という名前の Artifact と samlpost_producer2 という名前の POST プロファイル方式を設定した場合は、これらの方式を入力します。例：

SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

設定済みの方式をすべて指定します。カスタム認証方式によって、SAML 認証方式がすべて列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_POST;

設定したすべての SAML POST プロファイル方式を指定します。カスタム認証方式によって、POST プロファイル方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_ART;

設定したすべての SAML Artifact 方式を指定します。カスタム認証方式によって、Artifact 方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=WSFED_PASSIVE;

正しいアカウント パートナー ID を持つ WS フェデレーション認証方式を検索するために、すべての WS フェデレーション認証方式指定します。

この方式を SiteMinder 管理者用に有効にする

チェック マークはオフのままにします。

5. [サブミット] をクリックします。

カスタム認証方式が完成しました。

単一のターゲットレルムの設定

認証方式を設定し、それらをカスタム方式と関連付けた後、フェデレーション リソースに対する単一のターゲット レルムを設定します。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
2. 単一のターゲット レルムのポリシー ドメインを変更します。
3. [レルム] タブを選択し、[作成] をクリックします。
[レルムの作成] ダイアログが表示されます。
4. 以下の値を入力して、単一のターゲット レルムを作成します。

名前

この単一のターゲット レルムの名前を入力します。

5. [リソース] オプションの以下のフィールドに入力します。

エージェント

ターゲット リソースのある Web サーバを保護する Web エージェントを選択します。

リソース フィルタ

ターゲット リソースの場所を指定します。この場所は、フェデレーション リソースをリクエストするすべてのユーザがリダイレクトされる場所です。

例: /FederatedResources など。

6. [デフォルト リソース保護] セクションで [保護] オプションを選択します。
7. 以前に設定したカスタム認証方式を [認証方式] フィールドで選択します。
たとえば、カスタム方式が **Fed Custom Scheme** という名前の場合は、この方式を選択します。
8. [OK] をクリックします。

単一のターゲット レルムのタスクが完了しました。

単一のターゲット レルムに関するルールの設定

単一のターゲット レルムを設定した後、リソースを保護するためのルールを設定します。

1. 単一のターゲット レルムの [変更] ページに移動します。
2. [ルール] セクションの [作成] をクリックします。
[ルールの作成] ページが表示されます。
3. ルール ページのフィールドに値を入力します。
4. [OK] をクリックします。

単一のターゲット レルム設定に新しいルールが含まれました。

単一のターゲット レルムを使用するポリシーの作成

単一のターゲット レルムを参照するポリシーを作成します。単一のターゲット レルムでは、リクエストを適切な SAML 認証方式に届けるためのカスタム認証方式を使用することに注意してください。

注: この手順は、ドメイン、カスタム認証方式、単一のターゲット レルム、関連ルールがすでに設定済みであることが前提です。

次の手順に従ってください:

1. すでに設定済みのドメインに移動します。
2. [ポリシー] タブを選択し、[作成] をクリックします。
[ポリシーの作成] ページが開きます。
3. [一般] セクションで、ポリシーの名前と説明を入力します。
4. [ユーザ] セクションからユーザをポリシーに追加します。
5. [ルール] タブから単一のターゲット レルムに対して作成したルールを追加します。

残りのタブは省略可能です。

6. [OK] をクリックします。
7. [サブミット] をクリックします。

ポリシー タスクが完了しました。 リクエストがこのポリシーのトリガすると、このポリシーは、ユーザを認証するために、単一のレルムおよび関連付けられた認証方式に依存します。

第 14 章: SAML 2.0 アイデンティティプロバイダの設定

アサーティング パートナーの前提条件(レガシー)

アサーティング パートナーを設定するには、以下の条件を確認します。

- ポリシー サーバがインストールされています。
- 以下のオプションのいずれかがインストールされています。
 - **Web エージェント**および **Web エージェント オプション パック**。
Web エージェントは、ユーザを認証し、**SiteMinder セッション**を確立します。 **オプション パック**はフェデレーション **Web サービス アプリケーション**を提供します。 ネットワーク内の適切なシステムに必ず **FWS アプリケーション**を展開してください。
 - **SPS フェデレーション ゲートウェイ**では、**Web エージェント**が組み込まれており、また組み込まれた **Tomcat Web サーバ**上にフェデレーション **Web サービス アプリケーション**が配置されています。

詳細については、「**Web エージェント オプション パック ガイド**」を参照してください。

- 秘密キーと証明書は、メッセージの署名および復号を必要とする機能に対してインポートされます。
- 依存パートナーは、フェデレーション ネットワーク内でセットアップされます。

アイデンティティプロバイダを設定する方法

SiteMinder は、アイデンティティ プロバイダとして、そのビジネス パートナー（すなわちサービス プロバイダ）に対するアサーションを生成します。フェデレーション パートナリップを確立するために、アイデンティティ プロバイダでは、各パートナーに関する情報を必要とします。パートナーごとにサービス プロバイダ オブジェクトを作成し、2つのエンティティがどのような通信方法でアサーションを渡し、シングル サインオンなどのプロファイルを満たすかを定義します。

アイデンティティプロバイダを設定する方法

1. サービスプロバイダオブジェクトを作成します。
2. サービスプロバイダをアフィリエイトドメインに追加します。
3. サービスプロバイダの一般的な識別情報を指定します。
4. ユーザストアからユーザを選択します。アイデンティティプロバイダは、これらのユーザに対するアサーションを生成します。
5. 名前IDを指定します。
6. シングルサインオン (SSO) プロファイルを設定します。
完全な SSO プロファイルを設定しなくとも、サービスプロバイダエンティティを保存できます。ただし、SSO 設定が完了しないと、アサーションをサービスプロバイダに渡すことができません。
7. リクエストおよびレスポンスの暗号化と署名を設定します。
8. オプションの設定タスクを実行します。

ヒント：

- 設定が機能するためには、アイデンティティプロバイダとサービスプロバイダの特定のパラメータ値が一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」(Configuration Settings that Must Use the Same Values) にあります。
- フェデレーション Web サービス サブレットの正しい URL を使用してください。URL の一覧は、「[SiteMinder 設定で使用するフェデレーション Web サービス URL \(P. 457\)](#)」(Federation Web Services URLs Used in SiteMinder Configuration) にあります。

サービスプロバイダの識別に関するオプション設定タスク

サービスプロバイダを識別する際に以下のオプションのタスクがあります。

- サービスプロバイダのアクセスに使用されるアドレスを制限するための [IP アドレス制限の設定 \(P. 221\)](#)
- サービスプロバイダ動作の [時間制限の設定 \(P. 220\)](#)
- [機能強化クライアントまたはプロキシプロファイルの有効化 \(P. 233\)](#)
- アサーションに含める属性の設定

- [シングル ログアウト \(SLO\) の設定](#) (P. 275)
- [アイデンティティ プロバイダ ディスカバリ プロファイルの設定](#) (P. 279)
- アサーション内の[名前 ID の暗号化](#) (P. 283) またはアサーション全体の暗号化
- [アサーションの署名](#) (P. 242) またはアサーション レスポンス全体の署名
- [Artifact 解決メッセージの署名](#) (P. 242) または Artifact レスポンスの署名
- アサーション ジェネレータ プラグインを使用した [SAML アサーション レスポンスのカスタマイズ](#) (P. 171)

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。

- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

アフィリエイトドメインへの SAML 2.0 サービス プロバイダの追加

SiteMinder 生成アサーションの利用可能なコンシューマとしてサービス プロバイダを識別するには、そのサービス プロバイダをアイデンティティ プロバイダのアフィリエイト ドメインに追加します。そして、必要なアサーションをアイデンティティ プロバイダが発行できるようにサービス プロバイダの設定を定義します。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [フェデレーション] - [レガシー フェデレーション] - [SAML サービス プロバイダ] をクリックします。
3. [SAML サービス プロバイダの作成] をクリックします。
4. [アフィリエイト ドメイン] を選択した後、[次へ] をクリックします。

[一般] 設定項目を設定します。

サービス プロバイダ オブジェクトの一般情報の設定

[一般] ページを選択して、サービス プロバイダを指定し、SP ID や IDP ID などの詳細を入力します。また、サービス プロバイダにアクセスするための IP アドレスや時間制限を設定できます。

一般設定項目を設定する方法

1. [一般] 設定に移動します。

2. 必須フィールドに注意しながら、フィールドに値を入力します。

フィールドの説明については、[ヘルプ] をクリックしてください。
ただし、以下のフィールドに注意してください。

認証 URL

この URL は `redirect.jsp` ファイルを指します。SiteMinder ポリシーを使用して `redirect.jsp` ファイルを保護します。ポリシーは、保護されたサービスプロバイダリソースをリクエストするが SiteMinder セッションを持たないユーザに対する認証チャレンジをトリガします。

スキュー時間

アイデンティティプロバイダのシステムクロックとサービスプロバイダのシステムクロックの差を秒単位で指定します。スキュー時間は、シングルサインオンおよびシングルログアウトに使用されます。

シングルサインオンの場合、スキュー時間の値およびシングルサインオン有効期間（[SSO] タブの [有効期間] フィールド）によってアサーションが有効な時間が決定されます。スキュー時間についての理解を深めるには、[アサーション有効期間の計算](#) (P. 153) 方法について再確認してください。

シングルログアウトの場合、スキュー時間の値および SLO 有効期間（[SLO] タブの [有効期間] フィールド）によってシングルログアウトリクエストが有効な合計時間が決定されます。スキュー時間についての理解を深めるには、[シングルログアウト有効期間](#) (P. 277) の計算方法について再確認してください。

詳細情報:

[サービスプロバイダの IP アドレス制限の設定 \(オプション\)](#) (P. 221)

[サービスプロバイダ利用可能時間の制限の設定 \(オプション\)](#) (P. 220)

SiteMinder セッションのないユーザの認証

サービス プロバイダをアフィリエイト ドメインに追加する際に設定が必要なパラメータの 1 つが、認証 URL パラメータです。

認証 URL は、**redirect.jsp** ファイルを指します。このファイルは、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 ゲートウェイをインストールするアイデンティティ プロバイダ サイトでインストールされます。SiteMinder ポリシーを使用して **redirect.jsp** ファイルを保護します。ポリシーは、保護されたサービス プロバイダ リソースをリクエストするが SiteMinder セッションを持たないユーザに対する認証チャレンジをトリガします。

以下のバインドには、SiteMinder セッションが必要です。

- 保護されたサービス プロバイダ リソースをリクエストするユーザ
HTTP-Artifact バインドを使用してシングル サインオンを設定する場合、SAML アサーションをセッション ストアに保存するための永続セッションをセットアップします。
- HTTP-POST バインドを使用するシングル サインオン
ユーザはセッションを持つ必要がありますが、セッションは永続的である必要はありません。アサーションは、ブラウザ経由でサービス プロバイダに直接配信されます。アサーションは、セッション ストアに保存される必要はありません。
- シングル ログアウト
シングル ログアウトを有効にする場合、永続セッションが必要です。ユーザが最初にサービス プロバイダ リソースをリクエストする際、セッションがセッション ストアに保存されます。セッション情報は、後でシングル ログアウトが実行されたときに必要になります。

ユーザが認証され、正常に **redirect.jsp** ファイルにアクセスした後、セッションが確立されます。**redirect.jsp** ファイルは、ユーザをリダイレクトしてアイデンティティ プロバイダ Web エージェントまたは SPS フェデレーション ゲートウェイに戻します。その後、SiteMinder がリクエストを処理します。

認証 URL を保護するための手順は、以下の展開に無関係に同じです。

- Web エージェントと同じシステムにインストールされた Web エージェント オプションパック
- Web サーバプロキシにインストールされた Web エージェントのあるアプリケーションサーバ
- アプリケーションサーバ エージェントのあるアプリケーションサーバ
- アイデンティティ プロバイダでインストールされた SPS フェデレーションゲートウェイ

認証 URL を保護するポリシーの設定

認証 URL を保護する方法

1. 管理 UI にログインします。
2. アサーティング パーティ Web サーバに対して定義したレルムにバインドする Web エージェントを作成します。Web サーバと FWS アプリケーションに個別のエージェント名を割り当てるか、両方に同じエージェント名を使用します。
3. コンシューマ リソースへのアクセス試行時に認証情報が要求されるユーザに対してポリシー ドメインを作成します。
4. ポリシー ドメインに含まれるリソースへのアクセス権が必要なユーザを選択します。
5. ポリシー ドメインに対して以下の値を持つレルムを定義します。

エージェント

アサーティング パーティ Web サーバのエージェント

リソースフィルタ

Web エージェント r6.x QMR 6、r12.0 SP2、r12.0 SP3、および SPS フェデレーションゲートウェイによって次の値が入力されます。

/siteminderagent/redirectjsp/

リソース フィルタ `/siteminderagent/redirectjsp/` は、FWS アプリケーションが自動的にセットアップするエイリアスです。エイリアス参照には次の内容が含まれます。

- Web エージェント

`web_agent_home/affwebservices/redirectjsp`

- SPS フェデレーション ゲートウェイ

`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`

永続セッション

SAML Artifact プロファイルの場合のみ、[レルム] ダイアログ ボックスの [セッション] セクションで [永続] チェック ボックスをオンにします。永続セッションを設定しない場合、ユーザはコンシューマ リソースにアクセスできません。

残りの設定項目は、デフォルトのままにするか、必要に応じ変更します。

6. [OK] をクリックしてレルムを保存します。
7. レルムのルールを作成します。レルムに対応するすべてのリソースを保護するには、[レルム] フィールドでデフォルト値のアスタリスク (*) をそのまま使用します。
8. 前の手順で作成したルールが含まれるアサーティング パーティ Web サーバに対するポリシーを作成します。
9. 「[アサーションが生成される対象のユーザの選択](#) (P. 148)」のタスクを実行します。

サービスプロバイダ利用可能時間の制限の設定(オプション)

サービスプロバイダ リソースの利用可能な時間に制限を指定できます。時間制限を指定すると、リソースへのアクセスは指定された期間中のみ可能となります。ユーザが指定の期間外にリソースへのアクセスを試みても、アイデンティティプロバイダは SAML アサーションを生成しません。

注: 時間制限は、ポリシー サーバがインストールされたサーバのシステムクロックに基づきます。

時間制限を指定する方法

1. まず、[一般] 設定に移動します。
ページの [制限] セクションで、[時間を設定] セクションを選択します。
[時間制限] ページが表示されます。
2. スケジュールを設定します。このスケジュール グリッドは、ルール オブジェクトの [時間制限] グリッドと同一です。詳細については、「[ポリシー サーバ設定ガイド](#)」を参照してください。
3. [OK] をクリックします。

時間制限のスケジュールが設定されました。

サービスプロバイダの IP アドレス制限の設定(オプション)

サービスプロバイダにアクセスするブラウザが実行されている Web サーバの IP アドレス、範囲アドレス、またはサブネット マスクを指定できます。サービスプロバイダの IP アドレスが指定される場合、サービスプロバイダでは、適切な IP アドレスからのユーザのみを受け付けます。

IP アドレスを指定する方法

1. まず、[一般] 設定に移動します。
ページの [制限] セクションで、[IP アドレス] 領域の [追加] をクリックします。
[IP 制限] ページが表示されます。
2. 追加する IP アドレス タイプのオプションを選択し、そのアドレス タイプに関連するフィールドに入力します。

注: IP アドレスが不明でも、アドレスのドメイン名が既知の場合は、[DNS の検索] ボタンをクリックしてください。このボタンにより、[DNS の検索] ページが表示されます。[ホスト名] フィールドに完全修飾ホスト名を入力し、[OK] をクリックします。

- [単一ホスト] -- ブラウザをホストする単一の IP アドレスを指定します。単一の IP アドレスを指定する場合、ユーザは指定された IP アドレスからのみサービスプロバイダにアクセスできます。
- [ホスト名] -- Web サーバをそのホスト名で指定します。ホスト名を指定する場合、サービスプロバイダは指定されたホストからのユーザのみにアクセス可能です。

- [サブネットマスク] -- Web サーバのサブネットマスクを指定します。サブネットマスクを指定する場合、サービスプロバイダは指定されたサブネットマスクからのユーザのみにアクセス可能です。このボタンを選択する場合、[アドレスとサブネットマスクの追加] ダイアログボックスが表示されます。左矢印ボタンと右矢印ボタンを使用するか、スライダーバーをクリックしてからドラッグして、サブネットマスクを選択してください。
 - [範囲] -- IP アドレス範囲を指定します。IP アドレスの範囲を指定する場合、サービスプロバイダは、アドレスの範囲内の IP アドレスからのユーザのみを許可します。範囲の開始アドレス（[FROM]）および終了アドレス（[TO]）を入力します。
3. [OK] をクリックして設定を保存します。

プロキシ サーバの識別(オプション)

ネットワークでクライアントとフェデレーション Web サービスのあるシステムとの間にプロキシサーバが存在する場合は、URL の **protocol** と **authority** の部分を指定します。構文は、*protocol:authority* です。

protocol

http: または https:

authority

//host.domain.com または //host.domain.com:port

例 : http://example.ca.com

プロキシ サーバを識別する方法

1. まず、設定ウィザードの [一般] 手順に移動します。
ページの [詳細] セクションで、[サーバ] フィールドに URL を入力します。
2. [サブミット] をクリックします。

アサーションが生成される対象のユーザの選択

アサーティング パーティでの設定の一部として、アサーション ジェネレータが SAML アサーションを生成する対象のユーザおよびグループのリストを含めます。アサーティング パーティは、SAML 1.x プロデューサ、SAML 2.0 アイデンティティ プロバイダ、または WS フェデレーション アカウント パートナーのいずれかです。

アフィリエイト ドメインにあるディレクトリからのユーザおよびグループのみを追加できます。

フェデレーション トランザクションのユーザおよびグループを指定する方法

1. 設定するパートナーの [ユーザ] 設定に移動します。

[ユーザ ディレクトリ] ページには、ポリシー ドメインのユーザ ディレクトリごとにエントリが表示されます。

2. ユーザ ディレクトリからポリシーにユーザまたはグループを追加します。

各ユーザ ディレクトリ テーブルで、[メンバーの追加] - [エントリの追加] - [すべて追加] を選択できます。選択したメソッドに応じて、ダイアログ ボックスが表示され、ユーザの追加が可能になります。

- [メンバーの追加] を選択すると、[ユーザ/グループ] ウィンドウが表示されます。個々のユーザは、自動的に表示されません。検索ユーティリティを使用して、ディレクトリの 1 つに含まれる特定のユーザを見つけることができます。
- [エントリの追加] を選択する場合は、[ユーザ ディレクトリ 検索式編集] ダイアログ ボックスで[手動設定](#) (P. 150)によってユーザを選択してください。

ユーザやグループの編集または削除を行うには、それぞれ右向き矢印 (>) またはマイナス記号 (-) をクリックします。

3. 好きな方法で個別のユーザ、ユーザ グループ、または両方を選択して [OK] をクリックします。

[ユーザ ディレクトリ] ページが再度開き、新しいユーザがユーザ ディレクトリ テーブルに表示されます。

詳細情報:

[リソースへのアクセスからのユーザまたはグループの除外](#) (P. 149)

[ネストされたグループによるリソースへのアクセスの許可](#) (P. 149)

[手動設定によるユーザの追加](#) (P. 150)

リソースへのアクセスからのユーザまたはグループの除外

ユーザやユーザのグループをアサーションの取得から除外できます。

次の手順に従ってください:

1. [ユーザ] 設定に移動します。
2. 特定のユーザ ディレクトリのリストからユーザまたはグループを選択します。
3. 選択したユーザまたはグループを除外するには、[除外] をクリックします。

選択内容が 管理 UI に反映されます。

4. [OK] をクリックして、変更を保存します。

ネストされたグループによるリソースへのアクセスの許可

LDAP ユーザ ディレクトリには、サブグループを持つグループが含まれることがあります。複雑なディレクトリの場合、大量のユーザ情報を組織化する方法の 1 つとして、他のグループの階層内にグループをネストすることがあります。

ネストされたグループ内のユーザの検索を有効にする場合、リクエストされたユーザ レコードがすべてのネストされたグループで検索されます。ネストされたグループを有効にしない場合、指定したグループのみをポリシー サーバが検索します。

ネストされたグループでの検索を有効にする方法

1. [ユーザ] 設定に移動します。

関連するアフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、それぞれのユーザ ディレクトリが独自のセクションに表示されます。

2. ネストされたグループ内の検索を有効にするには、[ネストされたグループの許可] チェック ボックスをオンにします。

手動設定によるユーザの追加

アサーション生成のユーザを指定する場合の選択肢の 1 つとして、手動設定によりユーザを識別します。

次の手順に従ってください:

1. 設定するパートナーの [ユーザ] 設定に移動します。

アフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、すべてのディレクトリが [ユーザ ディレクトリ] ページに表示されます。

2. [エントリの追加] をクリックします。

[ユーザ ディレクトリ 検索式編集] ページが表示されます。

3. 検索オプションを選択した後、その検索オプションのフィールドに入力します。

検索する場所

LDAP ディレクトリの場合、ドロップダウン リストからオプションを選択します。

DN の確認

LDAP 検索はディレクトリ内でこの DN を検索します。

ユーザの検索

LDAP 検索の実行範囲は、ユーザ エントリ内の一致に限定されます。

グループの検索

LDAP 検索の実行範囲は、グループ エントリ内の一致に限定されます。

組織の検索

LDAP 検索の実行範囲は、組織エントリ内の一致に限定されます。

エントリの検索

LDAP 検索の実行範囲は、ユーザ エントリ、グループ エントリ、および組織エントリ内の一致に制限されています。

- Microsoft SQL Server、Oracle、および Windows NT ディレクトリの場合は、[手動設定] フィールドにユーザ名を入力できます。
- Microsoft SQL Server または Oracle では、SQL クエリを代わりに入力することもできます。例：

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

ポリシー サーバは、ユーザ ディレクトリの [認証情報と接続] タブの [ユーザ名] フィールドに指定されたデータベース ユーザとしてクエリを実行します。[手動設定] フィールドに入力する SQL 文を作成する場合には、ユーザ ディレクトリのデータベーススキーマに関する知識が必要です。たとえば、SmSampleUsers スキーマを使用している場合に特定のユーザを追加するには、SmUser テーブルからユーザ エントリを選択します。

- LDAP ディレクトリの場合は、[手動設定] フィールドに「all」と入力して、すべてのディレクトリ エントリを追加します。

4. [OK] をクリックして、変更を保存します。

SAML 2.0 アサーションの名前 ID の指定

名前 ID は、独自の方法でアサーション内のユーザを指定します。名前 ID は、サービス プロバイダに送信されるアサーションに追加されます。

名前 ID 形式により、その ID に使用されるコンテンツのタイプが確定されます。たとえば、形式が **User DN** の場合、コンテンツは **uid** です。

名前 ID を暗号化できます。ただし、**Artifact** バインドによるシングルサインオンの場合、アサーション内の他のデータをと共に **NameID** を暗号化すると、アサーションのサイズが大きくなります。

注: **NameID** はアサーションに必須です。

名前 ID を指定する方法

1. まず、設定ウィザードの [名前 ID] 手順に移動します。
2. [名前 ID 形式] を選択します。

各形式の説明については、「*OASIS Security Assertion Markup Language (SAML) V2.0*」仕様を参照してください。

3. [名前 ID タイプ] に以下のオプションから選択します。

- スタティック値
- ユーザ属性
- DN 属性（ネストされたグループの有無）

選択した [名前 ID タイプ] に従って [名前 ID フィールド] セクションの内容が変化します。

4. 選択した [名前 ID タイプ] の各種フィールドに入力します。

注: [名前 ID] を設定する場合は、[SAML アフィリエーション] フィールドでアフィリエーションを選択しないでください。名前 ID とアフィリエーションは相互に排他的です。

詳細情報:

[NameID とアサーションの暗号化 \(P. 283\)](#)

SAML アサーション レスポンスのカスタマイズ(オプション)

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を使用して、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次のとおりです。

1. SiteMinder 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（SDK に含まれています）を実装します。
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. 管理 UI でアサーション ジェネレータ プラグイン パラメータを有効にします。

アサーション ジェネレータ プラグインに関する追加の情報は、以下を参照してください。

- 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 クラスの新規コンストラクタが「*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。
- 認証および認可の API に関する概要および概念情報については、「Java 用 SiteMinder プログラミング ガイド」を参照してください。

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。

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。

- 要件を満たすようにインターフェースにメソッドを実装します。

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。既存の実装を上書きできます。例については、以下のサンプルクラスを参照してください。

SAML 1.x / WS フェデレーション

AssertionSample.java

SAML 2.0

SAML2AssertionSample.java

サンプルクラスは、`/sdk/samples/assertiongeneratorplugin` ディレクトリにあります。

実装によって `customizeAssertion` メソッド内に渡されるパラメータ文字列のコンテンツは、カスタム オブジェクトから提供されます。

アサーション ジェネレータ プラグインの展開

AssertionGeneratorPlugin 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、SiteMinder 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

- アサーション プラグイン Java ファイルをコンパイルします。

コンパイルには以下の `.jar` ファイルが必要となります。これらのファイルは、ポリシー サーバと共にインストールされています。

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

- 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。この変更により、変更されたクラスパスによるプラグインのロードが可能になります。
`installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

注: `xercesImpl.jar`、`xalan.jar`、`SMJavaApi.jar` のクラスパスを変更しないでください。

- プラグインを有効化します。

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、管理 UI 内で設定することにより、このプラグインを有効にします。UI パラメータにより、SiteMinder 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. 172)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. 管理 UI にログインします。
2. [フェデレーション] - [レガシー フェデレーション] - [SAML サービス プロバイダ] をクリックします。
3. 既存のサービス プロバイダ エントリを選択するか、新規に作成します。
4. [一般] 設定に移動します。
5. [アサーション ジェネレータ 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。

Java クラス名

既存のプラグインの Java クラス名を指定します

プラグイン クラスは、アサーションの解析および変更を行った後、最終的な処理のために結果をアサーション ジェネレータに返すことができます。

各サービス プロバイダには 1 つのプラグインのみが対応します。
たとえば、次のようになります。

`com.mycompany.assertiongenerator.AssertionSample`

パラメータ

(オプション) [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。

注: 管理 UI によってアサーション プラグインを有効にする代わりに、ポリシー管理 API (C または Perl) を使用して、プラグインを統合することができます。詳細については、「C 言語用 SiteMinder プログラミング ガイド」または「Java 言語用 SiteMinder プログラミング ガイド」を参照してください。

6. ポリシー サーバを再起動します。

ポリシー サーバを再起動することにより、再コンパイル後に最新のアサーション プラグインのバージョンが選択されることが保証されます。

Web アプリケーションの属性によるアサーションのカスタマイズ

アサーション ジェネレータ プラグインを使用して、Web アプリケーション属性をアサーションに追加できます。これは、アサーションのカスタマイズ方法の 1 つです。

Web アプリケーション属性をアサーション含める方法

1. アサーション プラグイン Java ファイルをコンパイルします。

コンパイルには以下の .jar ファイルが必要となります。これらのファイルは、ポリシー サーバと共にインストールされています。

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

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。この変更により、変更されたクラスパスによるプラグインのロードが可能になります。

`installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

注: `xercesImpl.jar`、`xalan.jar`、`SMJavaApi.jar` のクラスパスを変更しないでください。

3. サンプル プラグインを設定します。

SMJavaAPI の `APIContext` クラスには、`getAttrMap()` という新しいメソッドが含まれています。このメソッドは、アサーションに含まれた Web アプリケーションの属性が含まれるマップ オブジェクトを返します。SiteMinder SDK には、このマップ オブジェクトの使用方法を示すサンプル アサーション ジェネレータ プラグインが 2 つ含まれています。

- `SAML2AppAttrPlugin.java` (SAML 2.0)
- `WSFedAppAttrPlugin.java` (WS フェデレーション)

これらのサンプルは、`sdk/samples/assertiongeneratorplugin` ディレクトリにあります。これらを使用して、アサーション ジェネレータが Web アプリケーションの属性をアサーションに追加することができます。

4. 管理 UI にログインします。
5. [フェデレーション] - [レガシー フェデレーション] - [SAML サービス プロバイダ] または [リソース パートナー] を選択します。
6. 既存のエントリを選択するか、新規に作成します。
7. [一般] 設定に移動します。
8. [アサーション ジェネレータ 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。

Java クラス名

プラグインの Java クラスを指定します。たとえば、SiteMinder SDK に付属のサンプル クラスは次のとおりです。

- `com.ca.assertiongenerator.SAML2AppAttrPlugin`
(SAML 2.0)
- `com.ca.assertiongenerator.WSFedAppAttrPlugin`
(WS フェデレーション)

パラメータ

[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の文字列を指定します。これらのパラメータがアサーションに含める属性です。

注: 管理 UI によって設定する代わりに、ポリシー管理 API (C または Perl) を使用して、プラグインを統合することができます。手順については、「C 言語用 SiteMinder プログラミング ガイド」または「Java 言語用 SiteMinder プログラミング ガイド」を参照してください。

9. ポリシー サーバを再起動します。

ポリシー サーバを再起動することにより、再コンパイル後に最新のアサーション プラグインのバージョンが選択されることが確定されます。

SAML 2.0 の場合のシングル サインオンの設定

シングル サインオン設定の一環として、アイデンティティ プロバイダがアサーションをサービス プロバイダに配信する方法を決定します。

次の手順に従ってください:

1. 管理 UI で、サービス プロバイダ オブジェクトの [SAML プロファイル] 設定に移動します。
2. ページの [SSO] セクションのフィールドに入力します。通信に使用する SAML バインドを選択します。

HTTP-Artifact バインドの場合、保護された Artifact 解決サービスにユーザがアクセスできるようにバックチャネルを設定します。バックチャネル設定は、設定ウィザードの [属性] 手順にあります。

3. [サブミット] をクリックして、変更を保存します。

詳細情報:

[Artifact サービスを保護する認証方式の設定 \(P. 160\)](#)

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、スキュー時間と有効期間の値によって、アサーションの合計有効時間を SiteMinder が計算する方法が決定されます。SiteMinder 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。

注: この説明におけるアサーティング パーティは、SAML 1.x プロデューサ、SAML 2.0 アイデンティティ プロバイダ、または WS フェデレーション アカウント パートナーです。依存パーティは、SAML 1.x コンシューマ、SAML 2.0 サービス プロバイダ、または WS フェデレーション リソース パートナーです。

アサーション ドキュメントで、**NotBefore** と **NotOnOrAfter** の値は、有効期間の開始と終了を表します。

アサーティング パーティで、**SiteMinder** はアサーション有効期間を設定します。有効期間は、アサーションが生成されたときのシステム時間です。**SiteMinder** は、この時間を使用してアサーションに **IssueInstant** 値を設定した後、**IssueInstant** 値からスキュー時間の値を引き算します。その結果の時間が **NotBefore** 値になります。

NotBefore=IssueInstant - スキュー時間

有効期間の終了を決定するために、**SiteMinder** 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、**SiteMinder** は、アサーティング パーティで受信したアサーションが有効かどうか判別するために実行する計算と同じ計算を実行します。

パートナーシップの両側での **SiteMinder** によるアサーション有効期間の計算

SiteMinder がパートナーシップの両側にある場合、アサーション有効期間は、スキュー時間の **2 倍**と有効期間の総和になります。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) + 有効期間 + 2 x スキュー時間 (依存パーティ)

式の前半部分 ($2 \times$ スキュー時間 + 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 ($2 \times$ 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: レガシー フェデレーション の場合、有効期間はアサーティング パーティのみで設定されます。

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
- 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、アサーションから **NotBefore** と **NotOnOrAfter** の値を使用 し、それらの値にスキュー時間を適用します。この数式は、依存パーティ が新しい **NotBefore** と **NotOnOrAfter** の値を計算する方法です。

- スキュー時間 = 180 秒 (3 分)
- **NotBefore** = 4:56PM
- **NotOnOrAfter**=5:05PM

アサーション有効期間

この例の値を使用したアサーションの合計有効期間の計算は、次のとおり です。

$$120 \text{ 秒 } (2 \times 60) + 60 \text{ 秒 } + 360 \text{ 秒 } (2 \times 180) = 540 \text{ 秒 } (9 \text{ 分})$$

さまざまなシングル サインオン バインドに対するインデックス付きエンドポイントの定義

フェデレーション通信に対してインデックス付きのエンドポイントを設定できます。インデックス付きのエンドポイントは、アサーションが消費されるサイトです。**SiteMinder** のコンテキストでは、このエンドポイントはアサーション コンシューマ サービスが存在するサービス プロバイダです。

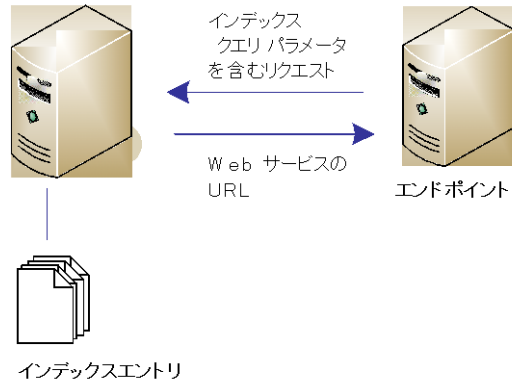
設定したエンドポイントのそれぞれには、アサーション コンシューマ サービス URL への明示的な単一の参照の代わりに、一意のインデックス値が割り当てられます。割り当てられたインデックスは、サービス プロバイダがアイデンティティ プロバイダに送信するアサーション リクエストに追加されます。

インデックス付きのエンドポイントをサポートするサードパーティ アイデンティティ プロバイダとのフェデレーション関係を持つ **SiteMinder** サービス プロバイダに対して、インデックス付きのエンドポイントを設定できます。また、複数のエンドポイントをサービスに割り当てることにより、アサーション コンシューマ サービスに対してさまざまなプロトコル バインド（Artifact、POST）を設定できます。

注: 使用しているネットワークに別の **SiteMinder** バージョンが含まれる場合、インデックス付きのエンドポイントは設定できません。たとえば、サービス プロバイダが **r12.0 SP 2** で、アイデンティティ プロバイダが **r12.0 SP3** である場合、インデックス付きのエンドポイントを設定できません。両方の HTTP バインドに対して、アサーション コンシューマ サービスを 1 つだけ設定します。

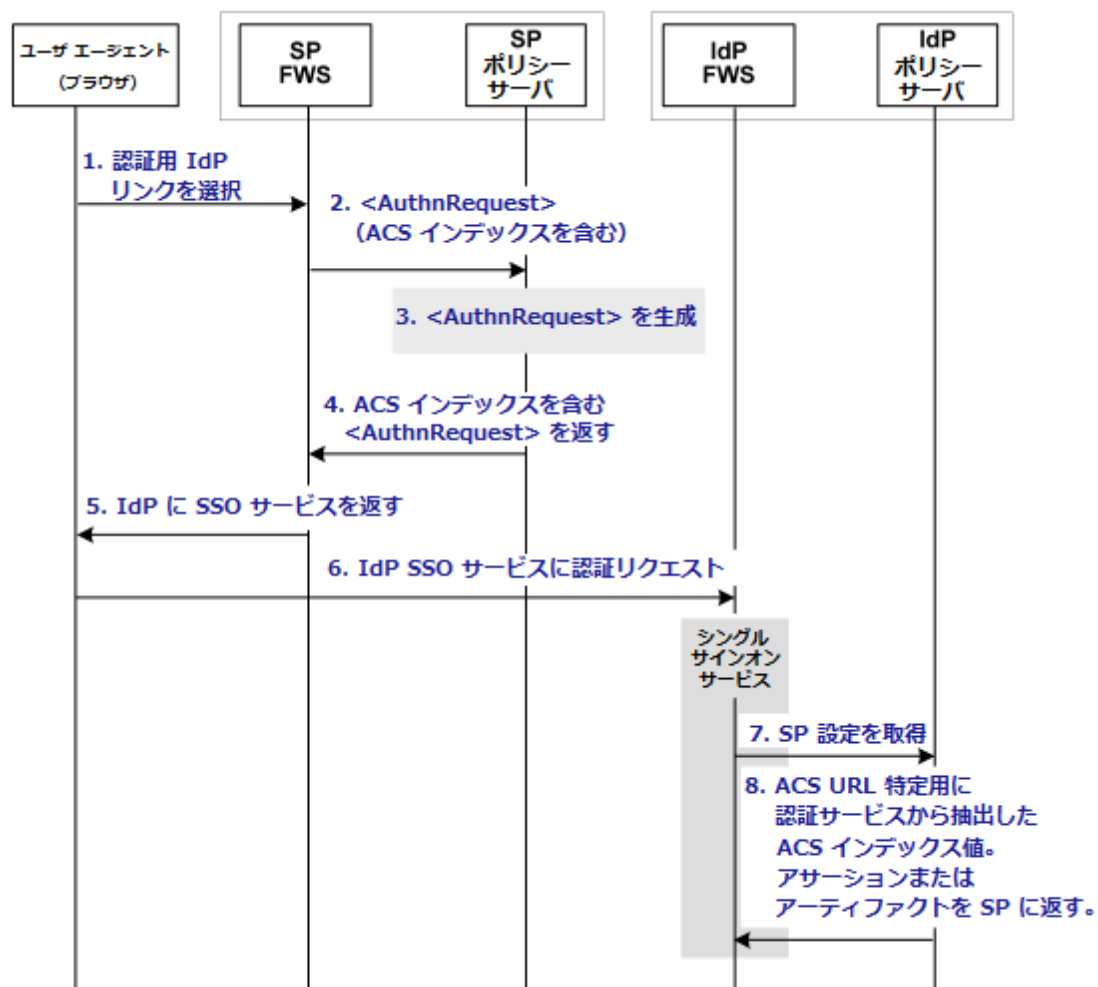
次の図は、インデックス付きのエンドポイントによる利点があるネットワークを示しています。

ID プロバイダ



インデックス付けされたエンドポイントフロー図

次の図は、インデックス付けされたエンドポイントを使用して、シングルサインオンがどのように動作するかを示します。



注: 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より FWS 機能を提供できます。

インデックス付けされたエンドポイントを使用したイベントのシーケンスは以下のとおりです。

1. ユーザは、固有の **IdP** を使用して認証するリンクを選択します。インデックス機能が有効であるため、リンクには **IdP ID** および **AssertionConsumerServiceIndex** クエリ パラメータ インデックスがクエリ パラメータとして含まれます。
2. **SP フェデレーション Web サービス (FWS)** アプリケーションは、そのローカルポリシー サーバから 認証リクエストを要求します。このアプリケーションが送信するリクエストには、**IdP ID** のほか、必要に応じてクエリ パラメータ **AssertionConsumerServiceIndex** および **ForceAuthn** が含まれます。

ACS Index パラメータと **Protocol Binding** パラメータは相互に排他的であるため、プロトコルバインドはリクエストに含まれません。

AssertionConsumerServiceIndex は、すでにバインドに関連付けられているため、**Protocol Binding** 値を指定する必要はありません。プロトコルバインドおよび **AssertionConsumerServiceIndex** がクエリ パラメータとして渡される場合、ローカル ポリシー サーバはエラーの応答を出力し、リクエストを拒否します。

3. 認証リクエスト サービスは、**IdP** 情報を **SP** ポリシー サーバから抽出し、認証リクエスト メッセージを生成します。このメッセージには **AssertionConsumerServiceIndex** が含まれます。
AssertionConsumerServiceIndex はクエリ パラメータの 1 つであるため、その値は **IdP** 記述子ドキュメントの **IdP** に照らして確認されます。このドキュメントは、事前に **IdP** から **SP** に送信されています。

認証リクエスト サービスは以下のように動作します。

- **Artifact** バインドのインデックスが **IdP** メタデータ内で設定されている場合、このインデックスは **AssertionConsumerServiceIndex** 値と比較されます。値が一致する場合、インデックス値は認証リクエストの一部としてそのまま使用されます。インデックス値が一致しない場合、**IdP** メタデータが確認されます。

AssertionConsumerServiceIndex は、**POST** バインドに対応する必要があります。

- インデックスが **HTTP-POST** バインドに対応する場合、このインデックス値は認証リクエスト内の **AssertionConsumerServiceIndex** と再度比較されます。**AssertionConsumerServiceIndex** パラメータの値が **POST** バインドに一致しない場合、認証リクエスト サービスはエラーを生成します。エラーには、**AssertionConsumerServiceIndex** が **IdP** メタデータ内のインデックスに一致しないという説明が記述されます。

4. **IdP** メタデータ インデックスおよび **AssertionConsumerServiceIndex** 値が一致する場合、**SP** ポリシー サーバは認証リクエストを生成します。
5. **SP** ポリシー サーバは、**HTTP** リダイレクト バインドで認証リクエストを返します。
6. **SP FWS** アプリケーションは、認証リクエストを **IdP** のシングルサインオン サービスにリダイレクトします。**SP** がシングルサインオン サービスの **URL** を既知であるのは、この **URL** が認証リクエスト内の設定情報の一部であるためです。
7. ブラウザは、シングルサインオン サービスをリクエストします。
8. シングルサインオン サービスは、認証リクエストから **AssertionConsumerServiceIndex** 値を抽出します。このサービスでは、**AssertionConsumerServiceIndex** を使用して、アサーション コンシューマ サービス **URL** を判別します。インデックスがメタデータに含まれていない場合、サービスはエラーを生成します。エラー メッセージには、無効な **AssertionConsumerServiceIndex** が認証リクエスト メッセージに存在することが示されます。

アサーションや **Artifact** を **SP** に送信するアサーション コンシューマ **URL** は、使用しているシングルサインオン プロファイルに応じて異なります。

注: **AssertionConsumerServiceIndex** パラメータが認証リクエストに存在しない場合、アサーション コンシューマ サービスの値および対応するバインドがデフォルトで使用されます。

アサーション コンシューマ サービスにおけるインデックス付きエンドポイントの設定

シングルサインオン サービスは、認証リクエストから ACS インデックス値を抽出します。サービスでは、インデックス値をそのインデックス エントリのリストと比較し、そのインデックス値に関連付けられたアサーション コンシューマ サービス URL を判別します。その後、シングルサインオン サービスは、インデックス値に関連付けられたバインドに応じて、アサーションまたは Artifact の送信先を認識します。

アイデンティティ プロバイダでインデックス エントリを設定する方法

1. 管理 UI にログインします。
2. 変更するサービス プロバイダ エントリを選択か、新規に作成します。
3. [SAML プロファイル] ページに移動します。
4. ページの [SSO] セクションで、[アサーション コンシューマ サービス] フィールドの端にある省略記号ボタンをクリックします。
[アサーション コンシューマ サービス] ページが表示されます。
5. [追加] をクリックします。
[アサーション コンシューマ サービスの追加] ページが表示されます。
6. ページの各種フィールドに入力します。
同じアサーション コンシューマ サービス URL に別のインデックス値を割り当てることができます。
7. [OK] をクリックして、変更を保存します。

注: サービス プロバイダで SAML 2.0 認証方式のインデックス エントリを必ず設定してください。

SSO に対する認証方式保護レベルの適用

ユーザは、フェデレーション リソースをリクエストする場合、SiteMinder セッションがあることが必要です。ユーザにセッションがない場合、そのユーザはセッションの確立のために認証 URL にリダイレクトされます。認証 URL を保護する認証方式は、特定の保護レベルに設定されます。この保護レベルは、SAML サービス プロバイダ設定で設定する認証レベルと同じかそれ以上であることが必要です。

認証 URL の保護レベルが 管理 UI で設定した認証レベルより低い場合、ポリシー サーバはアサーションを生成しません。

デジタル署名オプションの決定

フェデレーションでは、秘密キー/証明書ペアを使用して、フェデレーション通信のさまざまなデジタル署名タスクを実行できます。秘密キー/証明書ペアは、以下のメッセージに署名できます。

- アサーション
- SAML レスポンス
- Artifact レスポンス
- シングル ログアウト リクエストおよびレスポンス

シングル ログアウトの場合、ログアウトを開始する側がリクエストに署名します。リクエストを受信する側は署名を検証します。逆に、受信する側は **SLO** レスポンスに署名する必要があります。またログアウトを開始する側は、レスポンスの署名を検証する必要があります。

- 属性レスポンス（ユーザ属性に基づく認可に対する）

署名を担当するパートナーは、秘密キーに関連付けられた証明書（公開キー）を署名検証担当のパートナーに渡します。この交換は、あらゆるトランザクションが行われる前に、独立した通信で実行されます。

IdP は、SP にアサーションを送信する際、デフォルトで証明書をアサーションに含めます。ただし、SP では、自身のサイトに保存した証明書を使用して署名を検証します。

デジタル署名の設定オプションには、以下のものがあります。

- 署名エイリアス
- 署名アルゴリズム（RSAwithSHA1 または RSAwithSHA256）
- HTTP-Artifact アサーション、SAML レスポンス、および Artifact レスポンス オプション
- HTTP-POST アサーションおよび SAML レスポンス オプション

[一般]タブまたは[SSO]タブからの署名オプションを指定する方法

1. 既存の SAML サービス プロバイダ オブジェクトを変更するか、新規に作成します。
2. [SAML プロファイル] に移動します。
3. ダイアログ ボックスの [署名オプション] セクションで、フィールドに入力します。
4. [サブミット] をクリックします。

機能強化クライアントまたはプロキシ プロファイルの概要

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。

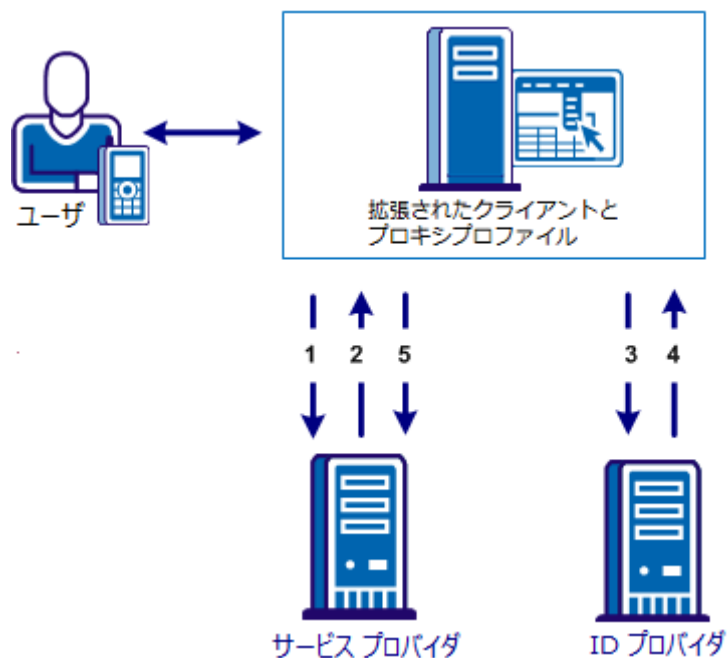
ECP プロファイルは、アイデンティティ プロバイダとサービス プロバイダが直接通信できない場合に、シングル サインオンを有効にします。ECP は、サービス プロバイダとアイデンティティ プロバイダの間で仲介する機能を果たします。

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。ポリシー サーバは ECP リクエストのみを処理し、SAML 要件に準拠する 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。

ECP 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。
ID プロバイダとは異なり、ECP エンティティは常に直接アクセスできます。
2. サービス プロバイダは、認証リクエストを ECP アプリケーションに送り返します。
3. ECP アプリケーションは、認証リクエストを処理および変更し、アイデンティティ プロバイダに送信します。

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

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

アイデンティティ プロバイダでの ECP の設定

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

次の手順に従ってください:

1. アイデンティティ プロバイダで 管理 UI にログインします。
2. 変更する **SAML** サービス プロバイダの [**SAML** プロファイル] タブに移動します。
3. ダイアログ ボックスで必要なシングル サインオン設定を完了します。
4. **SSO** セクションの[拡張されたクライアントとプロキシプロファイル] チェック ボックスをオンにします。
5. [サブミット] をクリックします。

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。

フェデレーション パートナーも ECP を有効にする必要があります。
SiteMinder の場合は、[SAML 2.0 認証方式で ECP を有効にします](#) (P. 303)。

注: 単一の **SAML** 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。

Allow/Create の有効化によるユーザ識別子の作成

アイデンティティ プロバイダでは、**SAML 2.0 Allow/Create** 機能を使用して、サービス プロバイダのリクエスト時にユーザ識別子を作成できます。この機能が動作するためには、サービス プロバイダ リクエストに **Allow/Create** 属性が含まれる必要があります。また、管理者は、識別子を作成するようにアイデンティティ プロバイダを設定する必要があります。アイデンティティ プロバイダは、**NameID** の一部として使用される一意の値を生成します。この値は、サービス プロバイダに返されるアサーション内に配置されます。

サービス プロバイダがアサーションを受信すると、**SAML 2.0** 認証方式がレスポンスを処理します。認証方式は、そのローカル ユーザ ストアでユーザを検索します。ユーザ レコードが検出される場合、そのユーザはアクセスが付与されます。

新規ユーザ識別子の作成を有効にする方法

1. 管理 UI にログインします。
2. サービス プロバイダ オブジェクトの **[SAML プロファイル]** 設定に移動します。
3. ページの **[SSO]** セクションで、**[新規識別子の作成を許可する]** を選択します。
4. **[サブミット]** をクリックします。

SP からの認証コンテキストの無視

ユーザ認証コンテキストに関する情報を交換することにより、フェデレーションパートナーシップの各側に対し、認証処理について通信する方法が提供されます。

サービス プロバイダがアイデンティティ プロバイダへのリクエストで認証コンテキストをリクエストする場合、コンテキストを無視するようにアイデンティティ プロバイダを設定することができます。

認証コンテキストを無視する方法

1. サービス プロバイダ オブジェクトの [一般] 設定に移動します。
2. [詳細 SSO 設定] セクションで、[リクエストされた AuthnContext を無視] を選択します。
3. [サブミット] をクリックします。

アサーションの単一使用の設定

SAML 2.0 の仕様に準拠して、ポリシー サーバはアサーションの単一使用を適用できます。単一使用のアサーションを生成することにより、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ことが依存パーティで認識されます。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。

アサーションの単一使用を設定する方法

1. サービス プロバイダ オブジェクトの [一般] 設定に移動します。
2. [詳細 SSO 設定] セクションで、[OneTimeUse 条件の設定] を選択します。
3. [サブミット] をクリックします。

IdP での HTTP エラー処理

アサーション ベースのシングル サインオンは、さまざまな理由によってアイデンティティ プロバイダで失敗する可能性があります。HTTP エラーが発生した場合、後続処理用の別のアプリケーション (URL) にユーザをリダイレクトすることができます。カスタマイズされたエラー ページへのリダイレクトが実行されるのは、サービス プロバイダに関する必要な情報がアイデンティティ プロバイダに提供されている場合のみです。情報が利用可能でない場合、HTTP エラー コードのみがブラウザに返されます。リダイレクトは発生しません。

HTTP 処理用のリダイレクト URL を設定できますが、それらは必須ではありません。

エラー処理用にオプションのリダイレクト URL を設定する方法

1. [一般] 設定に移動します。
2. [詳細 SSO 設定] セクションで、有効にする URL を選択した後、その URL を入力します。以下のエラーの 1 つまたは複数に対して URL を指定できます。
 - サーバエラー URL の有効化
 - 無効なリクエスト URL の有効化
 - 不正なアクセス URL の有効化
3. モードに関して以下のいずれかのオプションを選択します。
 - 302 データなし
 - HTTP POST
4. [サブミット] をクリックします。

注: これらのリダイレクト URL を SiteMinder アサーション コンシューマ プラグインと共に使用して、より詳細なアサーション処理を実行できます。アサーション リクエストが失敗した場合、プラグインは、指定されたりダイレクト URL の 1 つにユーザをリダイレクトできます。

アサーションでのセッション期間のカスタマイズ

ポリシー サーバ IdP は、アサーションを送信する際、デフォルトでアサーションの **Authentication** ステートメントに **SessionNotOnOrAfter** パラメータを含めます。サードパーティ SP は、**SessionNotOnOrAfter** の値を使用して、独自のタイムアウト値を設定することができます。タイムアウト値により、ユーザセッションが無効になるまでの期間が決定され、無効になると、IdP で再認証するようにユーザが送信されます。

重要: ポリシー サーバは、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、ターゲット リソースを保護する設定済みの SAML 認証方式に相当するレルム タイムアウトに基づき、セッション タイムアウトを設定します。

SessionNotOnOrAfter パラメータは、アサーション有効期間およびスキュー時間の決定に使用される **NotOnOrAfter** パラメータとは異なります。

SessionNotOnOrAfter パラメータをカスタマイズする方法

1. UI にログオンします。
2. 変更するサービス プロバイダ エントリを選択します。
3. [詳細] タブに移動します。
4. ダイアログ ボックスの [詳細 SSO 設定] セクションで [検証期間のカスタマイズ] を選択します。

[検証期間のカスタマイズ] ダイアログ ボックスが表示されます。

5. [SP セッション有効期間] の値を選択します。入力する値がアサーションの **SessionNotOnOrAfter** パラメータの値になります。

オプションを以下に示します。

アサーション有効期間の使用

アサーション有効期間に基づく **SessionNotOnOrAfter** 値を計算します。

省略

アサーションに **SessionNotOnOrAfter** パラメータを含めないように IdP に命じます。

IDP セッション

IdP セッション タイムアウトに基づく **SessionNotOnOrAfter** 値を計算します。タイムアウトは認証 URL 用の IdP レルムで設定されます。このオプションを使用すると、IdP および SP セッション タイムアウト値を同期することができます。

カスタム

アサーションに **SessionNotOnOrAfter** パラメータのカスタム値を指定できます。このオプションを選択する場合は、[アサーション有効期間のカスタマイズ] フィールドに時間を入力します。

6. [OK] をクリックして変更内容を保存します。

アサーションを取得するサービスへのアクセスの許可 (Artifact SSO)

HTTP Artifact シングル サインオンの場合、依存パーティは、アサーションを取得するための FWS サービスを保護するポリシーへのアクセスを許可する必要があります。

アクセスを許可するには、以下の手順に従います。

- **FederationWebServicesAgentGroup** エージェント グループに、FWS アプリケーションを保護する **Web** エージェントを追加します。
- 特定のサービスへのアクセスを許可された[ユーザとして依存パートナーを追加](#) (P. 158) します。

ユーザを所定のポリシーに追加する以外、その他すべてのポリシー オブジェクトは自動的にセットアップされます。

フェデレーション エージェント グループへの Web エージェントの追加

FederationWebServicesAgentGroup エージェント グループに、FWS アプリケーションを保護する Web エージェントを追加します。

- ServletExec の場合、このエージェントは、Web エージェント オプションパックがインストールされている Web サーバにあります。
- WebLogic や JBOSS などのアプリケーション サーバの場合、この Web エージェントは、アプリケーション サーバ プロキシがインストールされている場所にインストールされています。Web エージェント オプションパックは別のシステムに存在することもあります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [エージェント] - [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. 展開内の Web エージェントの名前を指定します。[サブミット] をクリックします。
5. [インフラストラクチャ] - [エージェント] - [エージェント グループ] をクリックします。
6. [FederationWebServicesAgentGroup] エントリを選択します。
7. [追加/削除] をクリックします。[エージェント グループのメンバ] ダイアログ ボックスが表示されます。
8. Web エージェントを [使用可能なメンバ] リストから [選択されたメンバ] リストに移動します。
9. [OK] ボタンをクリックして、[エージェント グループ] ダイアログに戻ります。
10. [サブミット] クリックした後、[閉じる] クリックしてメイン ページに戻ります。

アサーションを取得するための FWS ポリシーへの依存パートナーの追加

シングルサインオンに HTTP-Artifact バインドを使用している場合、パートナーシップ内の依存パーティはアサーション検索サービスへのアクセス許可を必要とします。SiteMinder は、ポリシーを使用して SAML 1.x および 2.0 検索サービスを保護します。

ポリシー サーバをインストールする際に、デフォルトで FederationWebServicesDomain がインストールされます。このドメインには、SiteMinder がアサーションを取得するサービスに対応して以下のポリシーが含まれます。

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

注: WS フェデレーションは、HTTP-Artifact プロファイルを使用しません。そのため、この手順はリソース プロバイダには適用されません。

これらのポリシーが任意の関連依存パートナーにアクセスするための権限を付与します。

次の手順に従ってください:

1. 管理 UI で、[ポリシー] - [ドメイン] - [ドメイン ポリシー] の順に移動します。

ドメイン ポリシーのリストが表示されます。

2. SAML プロファイルのポリシーを選択します。

SAML 1.x

FederationWSAssertionRetrievalServicePolicy

SAML 2.0

SAML2FWSArtifactResolutionServicePolicy

[ドメイン ポリシー] ページが表示されます。

3. [変更] をクリックしてポリシーを変更します。
4. [ユーザ] タブを選択します。

5. 該当するユーザディレクトリのダイアログボックスで、[メンバーの追加] をクリックします。

SAML 1.x

FederationWSCustomUserStore

SAML 2.0

SAML2FederationCustomUserStore

[ユーザ/グループ] ページが表示されます。

先に設定したアフィリエイト ドメインが [ユーザ/グループ] ダイアログボックスにリスト表示されます。たとえば、アフィリエイト ドメインが **fedpartners** という名前の場合、エントリーは **affiliate:fedpartners** となります。

6. サービスへのアクセスを必要とするパートナーが存在するアフィリエイト ドメインの横のチェックボックスをオンにします。[OK] をクリックします。

ユーザディレクトリのリストに戻ります。

7. [サブミット] をクリックします。

ポリシーのリストに戻ります。

Artifact サービスを保護する認証方式の設定

HTTP-Artifact プロファイルの場合、アサーション検索サービス (SAML 1.x) および Artifact 解決サービス (SAML 2.0) は、アサーティングパーティでアサーションを取得します。これらのサービスでは、アサーション レスポンスを依存パーティに送信する際、セキュリティで保護されたバックチャネルを使用します。これらのサービスおよびバックチャネル上の通信を不正なアクセスから保護することを強くお勧めします。

注: WS フェデレーションは HTTP-Artifact プロファイルをサポートしません。

これらのサービスを保護するには、アサーティングパーティでサービスが含まれるレムに対する認証方式を指定します。認証方式により、依存パーティで消費サービスがバックチャネル上の関連サービスにアクセスするために提供する必要がある認証情報のタイプが指定されます。

以下のいずれかの認証方式を選択できます。

- [基本](#) (P. 161)
- [SSL を介した基本](#) (P. 162)
- [X.509 クライアント証明書](#) (P. 162)

アサーションを取得するサービスを保護するベーシック認証

HTTP Artifact シングル サインオンの場合、アサーティング パーティは、安全なバックチャネルでアサーションを依存パーティに送信します。ベーシック認証の場合、Artifact を解決し、アサーションを取得するサービスにアクセスするためのパスワードを設定します。すると、サービスは、バックチャネルでアサーションを依存パーティに送信します。

SSL を有効にしてベーシック認証を使用できます。ただし、SSL は必須ではありません。

注: バックチャネル全域でベーシックまたは Basic over SSL を認証方式として使用する場合にのみパスワードが関連します。

SAML 1.x アサーション検索サービスの場合、次の手順に従ってください:

1. 管理 UI にログインします。
2. プロデューサに関する [一般] 設定に移動します。
3. 以下のフィールドに値を入力します。
 - パスワード
 - パスワードの確認入力
4. [サブミット] をクリックして変更内容を保存します。

SAML 2.0 Artifact 解決サービスの場合、次の手順に従ってください:

1. 管理 UI にログインします。
2. アイデンティティ プロバイダに関する [属性] 設定に移動します。
3. [バックチャネル] セクションで、以下のフィールドに値を入力します。
 - パスワード
 - パスワードの確認入力
4. [サブミット] をクリックして変更内容を保存します。

Basic over SSL によるアサーション検索サービスの保護

アサーション検索サービス (SAML 1.x) や Artifact 解決サービス (SAML 2.0) の保護として、Basic over SSL 認証方式を使用できます。アサーティングパーティにおいて、ポリシー サーバのインストール時に、サービスを保護するための一連のデフォルト ポリシーがすでに設定されています。

必要な設定は、各パートナーで SSL を有効にすることだけです。アサーティングパーティや依存パーティでの他の設定は不要です。依存パーティでは、証明書データ ストアのデフォルト ルート認証機関 (CA) の 1 つを使用して、SSL 接続を確立できます。デフォルト CA の代わりに独自のルート CA を使用する場合は、CA 証明書をデータ ストアにインポートします。

Basic over SSL 認証方式を使用する場合には、すべてエンドポイント URL で SSL 通信が使用される必要があります。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。

アサーションを検索するサービスを保護ためのクライアント証明書認証

クライアント証明書認証方式を使用して、アサーション検索サービス (SAML 1.x) および Artifact 解決サービス (SAML 2.0) を保護できます。アサーティングパーティがクライアント証明書認証を必要とするように設定されている場合、依存パーティは、接続をアサーティングパーティに戻し、クライアント証明書の表示を試みます。

クライアント証明書認証方式の使用方法

1. 関連するサービスを保護するためのポリシーをアサーティングパーティで作成します。このポリシーは、クライアント証明書認証方式使用します。
2. 依存パーティでのバックチャネル設定用にクライアント証明書認証を有効化します。
3. パートナーシップのそれぞれの側で SSL を有効化します。

クライアント証明書認証を使用する場合、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。すなわち、URL が **https://** で始まる必要があります。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。

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)。

検索サービスを保護するポリシーの作成

アサーティングパーティがアサーションを取得するサービスを保護するためのポリシーをアサーティングパーティで作成します。

次の手順に従ってください:

1. アサーションをリクエストするアフィリエイトごとに、個別のエントリをユーザディレクトリに追加します。ユーザディレクトリを作成するか既存のディレクトリを使用します。

管理 UI のアフィリエイト一般設定の [名前] フィールドで指定されている値と同じ値をユーザレコードに入力します。たとえば、アフィリエイトの [名前] フィールドの値が **Company A** の場合、ユーザディレクトリのエントリは次のようになります。

`uid=CompanyA, ou=Development, o=CA`

ポリシーサーバは、アフィリエイトクライアント証明書の Subject DN 値をこのディレクトリエントリにマップします。

2. 設定したユーザディレクトリを **FederationWebServicesDomain** に追加します。
3. 証明書マッピング エントリを作成します。

属性名を、アフィリエイトのユーザディレクトリ エントリにマップします。属性は、アフィリエイトの証明書における **Subject DN** エントリを表します。たとえば、**CN** を属性名として選択すると、この値は **cn=CompanyA,ou=Development,o=partner** というアフィリエイトを表します。

マップを設定するには、[インフラストラクチャ] - [ディレクトリ] - [証明書マッピング] の順に移動します。

4. X509 クライアント証明書認証方式を設定します。
5. 以下のエントリが含まれるレルムを **FederationWebServicesDomain** の下に作成します。

名前

any_name

例：cert assertion retrieval

エージェント

FederationWebServicesAgentGroup

リソース フィルタ

/affwebservices/certassertionretriever (SAML 1.x)

/affwebservices/saml2certartifactresolution (SAML 2.0)

認証方式

前の手順で作成されたクライアント証明書認証方式。

6. 以下の情報が含まれるルールを cert assertion retrieval レルムの下に作成します。

名前

any_name

例：cert assertion retrieval rule

リソース

*

Web エージェント アクション

GET、POST、PUT

7. FederationWebServicesDomain の下に Web エージェント レスポンス ヘッダを作成します。

アサーション検索サービスは、この HTTP ヘッダを使用して、アフィリエイトがアサーションを取得するサイトであることを確認します。

以下の値が含まれるレスポンスを作成します。

名前

any_name

属性

WebAgent-HTTP-Header-Variable

属性の種類

ユーザ属性

変数名

consumer_name

属性名

アフィリエイト名の値が含まれる使用ディレクトリ属性を入力します。

例：uid=CompanyA

以下のエントリに基づいて、Web エージェントは、HTTP_CONSUMER_NAME という名前のレスポンスを返します。

8. 以下の値が含まれるポリシーを FederationWebServicesDomain の下に作成します。

名前

any_name

ユーザ

この手順で先に作成したユーザ ディレクトリからユーザを追加します。

ルール

rule_created_earlier_in_this_procedure

Response

response_created_earlier_in_this_procedure

Artifact 解決サービスを保護するポリシーが完成しました。

依存パーティにおいて、管理者は、以下のように関連するアサーションサービスに接続するバックチャネルでクライアント証明書認証を有効にする必要があります

SAML 1.x: アサーション検索サービスに対して [クライアント証明書認証を有効化](#) (P. 202) します

SAML 2.0: Artifact 解決サービスに対して [クライアント証明書認証を有効化](#) (P. 328) します

バックチャネル認証に必要な WebLogic 設定

アイデンティティ プロバイダでは、Web エージェント オプション パックが WebLogic 9.2.x アプリケーション サーバにインストールされています。バックチャネルでのベーシック認証がこのサーバで動作するように、WebLogic config.xml ファイルを変更します。

アプリケーション ドメインの WebLogic config.xml ファイルで、以下のよう
に <security-configuration> 要素内に <enforce-valid-basic-auth-credentials> を
設定します。

```
<enforce-valid-basic-auth-credentials>>false</enforce-valid-basic-auth-credentials>
```

IdP または SP からシングル サインオンを開始します。

ユーザは、アイデンティティ プロバイダまたはサービス プロバイダでシングルサインオンを開始できます。シングルサインオン動作をトリガするリンクを、各サイトで設定するかアプリケーションの一部として設定します。

詳細情報:

[アイデンティティ プロバイダで開始された SSO \(POST または Artifact バインド\)](#) (P. 260)

[サービス プロバイダで開始された SSO \(POST バインドまたは Artifact バインド\)](#) (P. 263)

アイデンティティプロバイダで開始された SSO (POST または Artifact バインド)

ユーザがサービス プロバイダに移動する前にアイデンティティ プロバイダにアクセスする場合、アイデンティティ プロバイダでは未承認応答を生成する必要があります。未承認応答を開始するには、そのサービス プロバイダ ID に関連したクエリ パラメータが含まれる HTTP Get リクエストを生成するハードコードリンクを作成します。アイデンティティ プロバイダは、この ID に対するアサーション レスポンスを生成します。フェデレーション Web サービス アプリケーションおよびアサーション ジェネレータは、GET リクエストを受理する必要があります。

未承認応答を開始するために作成するリンクは、ユーザがクリックします。

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
```

idp_server:port

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する Web サーバおよびポートを識別します。

SP_ID

サービス プロバイダ ID の値。エンティティ ID は大文字と小文字を区別します。管理 UI に表示される通りに正確に入力してください。

URI_for_binding

ProtocolBinding 要素用の POST バインディングまたは Artifact バインディングの URI を識別します。SAML 2.0 仕様によりこの URI が定義されます。

また、未承認応答が機能するように、SAML サービス プロバイダ プロパティ内でバインドを指定します。

以下の情報に注意してください。

- リンク内に **ProtocolBinding** パラメータがなく、サービス プロバイダ プロパティ内にバインドが 1 つだけ存在する場合、サービス プロバイダ ではそのバインドを使用します。
- **Artifact** バインドおよび **POST** バインドがサービス プロバイダ プロパティ内で有効化されている場合、**POST** がデフォルトになります。そのため、**Artifact** バインドのみを使用する場合には、リンクに **ProtocolBinding** クエリ パラメータを含めます。
- アサーション コンシューマ サービスに対してインデックス付きのエンドポイントを設定する場合、**ProtocolBinding** クエリ パラメータによりアサーション コンシューマ サービスのバインドがオーバーライドされます。

詳細情報:

[IdP によって使用される未承認応答のクエリ パラメータ](#) (P. 261)

IdP によって使用される未承認応答のクエリ パラメータ

ポリシー サーバ 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
- ProtocolBinding
- RelayState

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際に使用される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。

ProtocolBinding クエリ パラメータの必須使用

ProtocolBinding パラメータの使用が必要となるのは、Artifact バインドおよび POST バインドがサービス プロバイダ プロパティ内で有効化されている場合のみです。両方のプロファイルが有効化されている場合、Artifact バインドの使用時のみクエリ パラメータを使用します。

- SAML 2.0 仕様による Artifact バインドの URI は次のとおりです。

urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact

- SAML 2.0 仕様による POST バインドの URI は次のとおりです。

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 HTTP エンコードしないでください。

例: ProtocolBinding が含まれる未承認応答

このリンクは、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。このリンクには、サービス プロバイダの ID が含まれています。SPID クエリ パラメータは、この ID を示します。さらに、バインド クエリ パラメータは、Artifact バインドが使用中であることを示しています。ユーザは、このハードコードされたリンクをクリックすると、ローカルのシングル サインオン サービスにリダイレクトされます。

```
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
```

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、以下の条件が適用されます。

- ProtocolBinding が未承認応答で指定されない場合、サービス プロバイダに対するプロファイルが使用されます。
- サービス プロバイダに対して両方のプロファイルを有効化できます。ProtocolBinding が未承認応答に含まれない場合、サービス プロバイダはデフォルトで POST プロファイルを使用します。

例：ProtocolBinding のない未承認応答

このリンクは、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。このリンクには、サービス プロバイダ ID が含まれています。SPID クエリ パラメータは、この ID を示します。

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すると、ローカルのシングル サインオン サービスにリダイレクトされます。

```
http://fedsrv.fedsite.com:82/affwebservices/public/saml2sso?SPID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90
```

RelayState

サービス プロバイダにおけるターゲットを指定します。RelayState クエリ パラメータは、ターゲットの宛先を示すために使用します。ただし、この方法はオプションです。サービス プロバイダでターゲットを示す設定メカニズムも使用可能です。

RelayState の値を URL エンコードします。

例

```
http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=
http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90&
RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp
```

サービス プロバイダで開始された SSO (POST バインドまたは Artifact バインド)

ユーザは、サービス プロバイダにアクセスした後、アイデンティティ プロバイダに移動することがあります。そのため、サービス プロバイダの認証リクエスト サービスまでのハードコードリンクが含まれる HTML ページをサービス プロバイダで作成します。HTML ページ内のリンクにより、ユーザがアイデンティティ プロバイダにリダイレクトされ、認証されます。また、リンクには、認証リクエストに含まれる内容も示されます。

ユーザが選択するハードコードリンクには、特定のクエリ パラメータが含まれる必要があります。これらのパラメータは、サービス プロバイダにおける認証リクエスト サービスに対する HTTP GET リクエストの一部です。

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`

sp_server:port

サービスプロバイダで 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する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。

リクエストが機能するためには、SAML 認証方式に関するバインドを有効にする必要があります。

以下の情報に注意してください。

- 認証リクエストに ProtocolBinding クエリ パラメータが含まれない場合、認証方式に対して定義されたバインドがデフォルトのバインドになります。認証方式に両方のバインドが定義されている場合、認証リクエストにバインドは渡されません。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。
- SAML 認証方式に対して Artifact および POST が有効化されている場合があります。Artifact バインド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。

SiteMinder SP で使用される認証リクエスト クエリ パラメータ

SiteMinder サービス プロバイダは、認証リクエスト サービスへのリンクでクエリ パラメータを使用できます。許可されるクエリ パラメータを以下に示します。

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 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。

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 プロバイダが **SAML** レスポンスを返すために使用する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。

ProtocolBinding クエリ パラメータの必須使用

Artifact および **POST** バインドを認証方式として有効化できます。**Artifact** バインドのみを使用する場合、**ProtocolBinding** パラメータが必要です。

- SAML 2.0 仕様による **Artifact** バインドの URI は次のとおりです。
`urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact`
- SAML 2.0 仕様に基づいた **POST** バインドの URI は次のとおりです。

`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。

例：ProtocolBinding が含まれる認証リクエスト 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
```

ユーザは、サービス プロバイダのリンクをクリックします。フェデレーション Web サービス アプリケーションは、ローカル ポリシー サーバからの認証リクエスト メッセージを要求します。

ProtocolBinding の任意使用

ユーザが ProtocolBinding クエリ パラメータを使用しない場合、以下の条件が適用されます。

- 認証方式に対してバインドを 1 つのだけ有効化し、ProtocolBinding クエリ パラメータは指定しない場合は、認証方式で有効化されたバインドが使用されます。
- 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 エンコードしないでください。

例: ProtocolBinding が含まれない認証リクエスト リンク

このサンプル リンクは、認証リクエスト サービスに移動します。このリンクは、ProviderID クエリ パラメータのアイデンティティ プロバイダを指定します。

`http://ca.sp.com:90/affwebservices/public/saml2authnrequest?ProviderID=http%3A%2F%2Ffedsrv.acme.com%2Fsmidp2for90`

ユーザは、サービス プロバイダのリンクをクリックします。フェデレーション Web サービス アプリケーションは、ローカル ポリシー サーバからの認証リクエスト メッセージを要求します。

ForceAuthn

ユーザに関する既存のセキュリティ コンテキストが存在する場合であっても、アイデンティティ プロバイダがそのユーザを認証するように SP が強制するかどうかを示します。

- 認証リクエスト メッセージに **ForceAuthn=True** と指定されており、特定のユーザに対して **SiteMinder** セッションが存在する場合、**IdP** はユーザに認証情報を再要求します。ユーザが正常に認証される場合、**IdP** は、既存のセッションからの ID 情報をアサーションに含めます。**IdP** は、再認証用に生成したセッションを破棄します。

注: ユーザは、既存のセッションとは異なる認証情報を使用して再認証を試みることができます。すると、**IdP** は、現在と既存のセッションの **userDN** ユーザ ディレクトリ **OID** を比較します。セッションの対象が同じユーザではない場合、**IdP** は **SAML 2.0** レスポンスを返します。このレスポンスは、認証が失敗したことを示します。

- SP が認証リクエスト メッセージで **ForceAuthn=True** を設定し、**SiteMinder** セッションが存在しない場合、**SiteMinder 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://www.sp.demo:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo&ForceAuthn=yes`

RelayState

サービス プロバイダにおけるターゲットを指定します。**RelayState** クエリ パラメータを使用して、ターゲットの宛先を示すことができます。ただし、この方法はオプションです。代わりに、**SAML 2.0** 認証方式で設定されたターゲットを指定することができます。この認証方式には、ターゲットを **RelayState** クエリ パラメータでオーバーライドするオプションがあります。

RelayState の値を URL エンコードします。

例

`http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%2Fapps%2Fapp.jsp`

IsPassive

アイデンティティ プロバイダがユーザと対話できるかどうかを決定します。このクエリ パラメータが **true** に設定されている場合、アイデンティティ プロバイダはユーザとの対話が禁止されます。また、**IsPassive** パラメータは、アイデンティティ プロバイダに送信された 認証リクエストに付属しています。このクエリ パラメータが **false** に設定されている場合、アイデンティティ プロバイダはユーザと対話できます。

例

```
http://www.spdemo.com:81/affwebservices/public/saml2authnrequest?
ProviderID=idp.demo&RelayState=http%3A%2F%2Fwww.spdemo.com%
2Fapps%2Fapp.jsp&IsPassive=true
```

AssertionConsumerServiceIndex

アサーション コンシューマ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。

このパラメータを認証リクエストで使用する場合、**ProtocolBinding** 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。

SiteMinder IdP によるクエリ パラメータの処理

サービス プロバイダがシングル サインオンを開始する場合、そのサービス プロバイダは **ForceAuthn** クエリ パラメータまたは **IsPassive** クエリ パラメータを認証リクエスト メッセージに含めることができます。サービス プロバイダがこれら 2 つのクエリ パラメータを認証リクエスト メッセージに含める場合、**SiteMinder** アイデンティティ プロバイダではこれらのクエリ パラメータが以下のように処理されます。

ForceAuthn の処理

サービス プロバイダが **ForceAuthn=True** を認証リクエストに含めている場合、**SiteMinder** アイデンティティ プロバイダでは以下のアクションを実行します。

- 認証リクエスト メッセージに **ForceAuthn=True** があり、**SiteMinder** セッションが特定のユーザに対して存在する場合。**SiteMinder IdP** は、再度認証情報の入力をユーザに要求します。ユーザが正常に認証された場合、**IdP** では、その識別情報を既存のセッションからアサーション内に送信します。**IdP** は、再確認のために生成したセッションを破棄します。

ユーザは、元のセッションとは異なる認証情報を使用して再認証を試みることができます。**SiteMinder IdP** では、現在と既存のセッションの **userDN** およびユーザ ディレクトリ **OID** を比較します。セッションの対象が同じユーザではない場合、**SAML 2.0** レスポンスを返します。このレスポンスは、認証が失敗したことを示します。

- 認証リクエスト メッセージに **ForceAuthn=True** があり、**SiteMinder** セッションが存在しない場合。**SiteMinder IdP** は、認証情報の入力をユーザに要求します。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。

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めており、**IdP** 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** があり、**SiteMinder** セッションが存在しない場合。**SiteMinder** アイデンティティ プロバイダは、**SAML** レスポンスを返します。**SiteMinder** にはセッションが必要であるため、このレスポンスにはエラー メッセージが含まれます。

- 認証リクエストメッセージに `IsPassive=True` があり、**SiteMinder** セッションが存在する場合。**SiteMinder** 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` と `ForceAuthn` があり、両方が `True` に設定されている場合。リクエストが無効であるため、**SiteMinder** アイデンティティ プロバイダはエラーを返します。`IsPassive` と `ForceAuthn` は相互に排他的です。

アサーションの属性の設定(オプション)

属性は、サービス プロバイダ リソースへのアクセスをリクエストしているユーザに関する追加の情報を提供できます。属性ステートメントにより、**SAML** アサーション内でユーザ属性、**DN** 属性、または静的なデータがアイデンティティ プロバイダからサービス プロバイダに渡されます。すべての設定済みの属性は、1 つの `<AttributeStatement>` 要素内のアサーションまたはアサーション内の `<EncryptedAttribute>` 要素に含まれています。

注: アサーション内には属性ステートメントは不要です。

サーブレット、**Web** アプリケーション、その他のカスタム アプリケーションは、属性を使用して、カスタマイズされたコンテンツを表示したり、他のカスタム機能を実現したりします。属性を **Web** アプリケーションで使用する場合、属性を使用して、サービス プロバイダでのユーザのアクティビティを制限できます。たとえば、**Authorized Amount** という名前の属性変数を最高額設定して送信するなどです。この額は、ユーザがサービス プロバイダで費やすことができる限度額です。

注: **SiteMinder** がアサーション クエリ/リクエスト プロファイルの一部として **SAML 2.0** 属性機関として機能する場合、属性は認可プロセスの一部となります。この実装については、「[ユーザを認可するための属性機関の使用](#) (P. 407)」を参照してください。

属性の形式は、名前/値のペアになっています。サービス プロバイダでは、アサーションを受信すると、属性値をアプリケーションが利用できるようにします。

属性は、HTTP ヘッダまたは HTTP Cookie として利用できるようになります。

HTTP ヘッダおよび HTTP Cookie にはサイズ制限があり、アサーション属性でそれらを超えるはできません。サイズ制限は以下のとおりです。

- **HTTP ヘッダ**：SiteMinder は、Web サーバのヘッダに関するサイズ制限までの属性をヘッダで送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダ サイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。
- **HTTP Cookie**：SiteMinder は、Cookie のサイズ制限までの Cookie を送信できます。それぞれのアサーション属性は、それ自身の Cookie として送信されます。Cookie のサイズ制限はブラウザ固有です。また、その制限は、属性ごとではなく、アプリケーションに渡されるすべての属性に適用されます。Cookie のサイズ制限を調べるには、使用する Web ブラウザのドキュメントを参照してください。

SSO アサーションの属性の指定

属性は、サービス プロバイダ リソースへのアクセスを要求しているユーザに関する情報を提供できます。属性ステートメントにより、SAML アサーション内でユーザ属性、DN 属性、または静的なデータがアイデンティティ プロバイダからサービス プロバイダに渡されます。

属性を設定する方法

1. 編集するエンティティの [属性] 設定に移動します。
2. [追加] をクリックします。
[属性の追加] ページが表示されます。

3. [属性タイプ] ドロップダウン リストから、名前形式タイプを選択します。このエントリは、アサーション属性ステートメントの **<Attribute>** 要素の **<NameFormat>** 属性に一致する必要があります。サービスプロバイダが名前を解釈できるように、タイプによって属性名が分類されます。

オプションを以下に示します。

unspecified

名前の解釈方法は実装が決定します。

basic

名前形式が許容可能な値を使用する必要があることを示します。
許容可能な値は、プリミティブ タイプの **xs:Name** に属する値です。

URI

名前形式が **URI** 参照の基準に従う必要があることを示します。**URI** の解釈方法は、その属性値を使用するアプリケーションに固有です。

4. [属性のセットアップ] セクションから、以下のいずれかのオプションを [属性の種類] セクションで選択します。
 - スタティック
 - ユーザ属性
 - DN 属性

[属性の種類] の選択内容によって、[属性フィールド] セクション内の利用可能なフィールドが決定されます。
5. ページの [属性フィールド] セクションを設定します。設定項目は、[属性の種類] の選択内容に応じて異なります。オプションを以下に示します。
 - 変数名
 - 変数値
 - 属性名
 - DN 仕様
6. (オプション) 属性がネストされたグループを持つ **LDAP** ユーザディレクトリから取得される場合、ポリシー サーバはネストされたグループから **DN** 属性を取得できます。ネストされたグループを使用するには、[属性の種類] セクションの [ネストされたグループの許可] チェック ボックスをオンにします。

7. (オプション) 属性値を暗号化するには、[暗号化] チェック ボックスをオンにします。
8. [取得方法] では、属性がシングル サインオン アサーション向けのみであることを確定するために、デフォルト値の SSO を使用します。
9. [OK] をクリックして変更内容を保存します。

属性の新規作成スクリプトの使用

[属性] ダイアログ ボックスの [詳細] セクションには [スクリプト] フィールドが含まれています。このフィールドには、[属性のセットアップ] セクションの入力に基づいて SiteMinder が生成したスクリプトが表示されます。このフィールドの内容をコピーし、別のレスポンス属性の [スクリプト] フィールドに貼り付けることができます。

注: 別の属性の [スクリプト] フィールドの内容をコピーして貼り付ける場合は、[属性の種類] セクションで適切なオプションを選択してください。

アサーション属性の最大文字数を指定します

ユーザ アサーション属性の最大長は設定可能です。アサーション属性の最大長を変更するには、EntitlementGenerator.properties ファイルの設定を変更します。

ファイル内のプロパティ名は、設定のプロトコルに固有です。

次の手順に従ってください:

1. ポリシー サーバがインストールされているシステムで、
`policy_server_home¥config¥properties¥EntitlementGenerator.properties` に移動します。
2. テキスト エディタでファイルを開きます。

- 現在の環境で使用中のプロトコルでのユーザ属性の最大長を調節します。各プロトコルの設定を以下に示します。

WS-フェデレーション

プロパティ名：

`com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength`

プロパティタイプ：正の整数値

デフォルト値：1024

説明：WS-FED アサーション属性の最大属性長を示します。

SAML 1.x

プロパティ名：

`com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength`

プロパティタイプ：正の整数値

デフォルト値：1024

説明：SAML1.1 アサーション属性の最大属性長を示します。

SAML 2.0

プロパティ名：

`com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength`

プロパティタイプ：正の整数値

デフォルト値：1024

説明：SAML2.0 アサーション属性用の最大属性長を示します。

- これらのパラメータのいずれかを変更した後は、ポリシー サーバを再起動します。

SSO の属性および属性クエリリクエスト

設定する属性がシングルサインオン リクエストに関する属性であるか、または属性クエリ リクエストに関する属性であることを示します。設定する取得方法によって、属性の機能が決定されます。

両方のサービスに同じ属性を使用するには、同じ属性名と変数を使用する属性ステートメントを2つ作成します。ただし、一方の属性は取得方法としてSSOを使用し、もう一方の属性は取得方法として属性サービスを使用します。

シングル ログアウトの設定(オプション)

シングル ログアウト プロトコル (SLO) は、特定のユーザに関するすべてのセッションの同時終了を実行し、セキュリティの保証に役立ちます。これらのセッションは、ログアウトを開始したブラウザセッションの一部である必要があります。

シングル ログアウトにより、特定ユーザに関するすべてのセッションが必ずしも終了するとは限りません。たとえば、ユーザが 2 つのブラウザを開いている場合、そのユーザは 2 つの独立したセッションを確立できます。シングル ログ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。ユーザが開始したログアウトによりシングル ログアウトがトリガされます。

注: SiteMinder は、シングル ログアウト プロトコルの HTTP リダイレクト バインドのみをサポートします。

SLO を設定することにより、サービス プロバイダがシングル ログアウト プロトコルをサポートするかどうかと、その場合にサービス プロバイダがシングル ログアウトを処理する方法がアイデンティティ プロバイダに通知されます。

シングル ログアウトを有効にする場合には、以下の操作も必要です。

- ポリシー サーバ管理コンソールを使用して、アイデンティティ プロバイダでセッション 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。

- サービス プロバイダで保護されたリソースが含まれるレルムに対して永続セッションを設定します。管理 UI 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。

シングル ログアウトを設定する方法

1. 管理 UI にログオンします。
2. 設定する SAML サービス プロバイダの [SAML プロファイル] ページに移動します。
3. ページの [SLO] セクションで、[HTTP リダイレクト] チェック ボックスをオンにします。この設定により、シングル ログアウトが有効になります。
4. 以下のフィールドに注意して、残りのフィールドに値を入力します。

有効期間

シングル ログアウト リクエストが有効な秒数を指定します。このプロパティは、シングル サインオンの有効期間とは異なります。シングル サインオンの有効期間はアサーションが対象です。有効期間が期限切れになると、IdP は、ログアウトを開始したエンティティにシングル ログアウト レスポンスを送信します。また、有効期間は、シングル ログアウト メッセージ期間の計算でスキュー時間（[一般] で設定）にも依存します。

SLO ロケーション URL、SLO レスポンス ロケーション URL、SLO 確認 URL

これらのフィールドのエントリは <https://> または <http://> で始まる必要があります。

5. （オプション）1つのブラウザセッション中で同じパートナーに送信されたアサーションのセッションインデックスと同じセッションインデックスを使用するには、[セッションインデックスの再利用] フィールドを選択します。このオプションを使用することで、すべてのサードパーティ パートナーでのシングル ログアウトの成功を保証できます。

フェデレーション Web サービスは、ユーザセッションがアイデンティティ プロバイダおよびすべてのサービス プロバイダ サイトで削除された後、ユーザをログアウト確認ページにリダイレクトします。

詳細情報:

[シングル ログアウト リクエスト有効期間 \(P. 277\)](#)

[シングル ログアウト確認ページのガイドライン \(P. 278\)](#)

シングル ログアウト リクエスト有効期間

SLO 有効期間およびスキュー時間により、シングル ログアウト リクエストが有効となる総時間の計算方法がポリシー サーバに指定されます。

注: SLO 有効期間は、SSO 有効期間とは別の値です。

ログアウト リクエスト期間の計算において関連する 2 つの値は、**IssueInstant** 値および **NotOnOrAfter** 値と呼ばれます。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 リクエストが生成される際、ポリシー サーバはそのシステム時刻を使用します。結果として得られる時間が **IssueInstant** 値になります。この値は、リクエスト メッセージで設定されます。

ポリシー サーバは、ログアウト リクエストがいつ無効でなくなるかを判別します。ポリシー サーバは、その現在のシステム時刻を使用し、スキュー時間および SLO 有効期間を加算します。その結果の時間が **NotOnOrAfter** 値になります。時間は GMT が基準になります。

たとえば、アイデンティティ プロバイダでログアウト リクエストが 1:00 GMT に生成されたとします。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 ログアウト確認ページのガイドライン

シングル ログアウトをサポートするには、任意のサイトにログアウト確認ページを用意する必要があります。 ユーザは、このページによってログアウトを認識できます。

ログアウト確認ページは、以下の基準を満たす必要があります。

- シングル ログアウトがサービス プロバイダで開始される場合、ログアウト確認ページは、サービス プロバイダサイトの非保護のローカル リソースであることが必要です。
- シングル ログアウトがアイデンティティ プロバイダ サイトで開始される場合、ログアウト確認ページはアイデンティティ プロバイダ サイトの非保護のローカル リソースであることが必要です。
- このページは、フェデレーション パートナー ドメイン内のリソースであってはなりません。たとえば、ローカル ドメインが **ca.com** の場合、SLO 確認ページを **example.com** ドメインに配置することはできません。

ログアウト失敗に関するフィードバックを受信するには、ログアウト確認ページが以下の要件もサポートすることが必要です。

- **Base 64** エンコード データの処理ができ、**Cookie** を読み取ることができること。
- **SIGNOUTFAILURE Cookie** を検索するコードが含まれていること。この条件は、IdP および SP のログアウト ページが満たす必要があります。シングル ログアウトの失敗が発生すると、**Cookie** がブラウザで設定されます。**Cookie** には、ログアウトが失敗したフェデレーション サイトのパートナー ID が含まれます。これらの ID は、**Base 64** でエンコードされています。複数の ID が列挙される場合、それらはスペース文字によって区切られます。

この **Cookie** を検索するようにログアウト確認ページを設定することによって、ログアウトが失敗した場所を確認ページからユーザに通知できます。この情報は、ユーザが複数のパートナー サイトからログアウトするネットワークで便利です。

また、**SIGNOUTFAILURE Cookie** が検出された場合、ログアウト確認ページは、**Web** ブラウザを閉じてセッション データをすべて削除するようにユーザ通知する必要があります。

IdP でのアイデンティティプロバイダ ディスカバリの設定

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。

IDP ディスカバリ プロファイルの場合、SP は、認証リクエストの送信先となる IdP を決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。

IdP では、アイデンティティ プロバイダ ディスカバリ機能の有効化のみを行います。追加の設定は不要です。この機能を有効にすると、結果的に IDP ディスカバリ サービスの共通のドメインに Cookie が設定されます。この処理は、ユーザには見えません。

アイデンティティプロバイダ ディスカバリ プロファイルの有効化(オプション)

フェデレーション ネットワークには、アサーションを生成するアイデンティティプロバイダが複数存在する場合があります。アイデンティティプロバイダ ディスカバリ プロファイルを使用することにより、ユーザが特定のアイデンティティ プロバイダを認証用を選択することができます。

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. 管理 UI にログオンします。
2. 変更する SP の [SAML プロファイル] ページに移動します。

3. [IPD] セクションで、[有効] チェック ボックスをオンにします。
4. 必要なフィールドに入力し、必要な設定を選択します。

注: [サービス URL] フィールドに以下のアイデンティティ プロバイダ ディスカバリ プロファイル サブレットを設定してください。

`https://host:port/affwebservices/public/saml2ipd`

5. [サブミット] をクリックして、変更を保存します。

IdP ディスカバリ ターゲットの攻撃からの保護

SiteMinder 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 を配置することができます。この URL により、悪意のあるサイトにリダイレクトされ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、この設定は特に連携に対して定義されます。

IPD サービスは、**IPDTarget** クエリ パラメータを検査します。このサービスは、クエリ パラメータによって指定される 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、サービスは検証を一切実行せずに、ユーザをターゲット URL にリダイレクトします。

署名済みリクエストおよびレスポンスの検証

ポリシー サーバでは、次の署名済みメッセージを確認できます。

- SSO 認証。
- シングル ログアウト リクエストおよびレスポンス。

デフォルトでは、署名の処理は有効化されています。これは、SAML 2.0 仕様で必要とされるためです。実稼働環境では常に署名処理を有効にしてください。

ポリシー サーバは、常に SAML 2.0 POST レスポンスおよびシングル ログアウト リクエストに署名します。署名には、管理 UI を使用した設定は不要です。署名のセットアップに必要な操作は、署名機関の秘密キー/証明書ペアを証明書データ ストアに追加することだけです。

重要: デバッグ目的のみに限り、すべての署名処理（署名および署名の検証の両方）を一時的に無効化できます。[暗号化 & 署名] 設定の [署名] セクションで、[署名の処理を無効にする] を選択してください。

サービス プロバイダからの認証リクエストの署名や、シングル ログアウトのリクエストおよびレスポンスの署名を検証するには、管理 UI の設定手順を完了する必要があります。

検証をセットアップする方法

1. アイデンティティ プロバイダで公開キーを証明書データ ストアに追加します。

公開キーは、サービス プロバイダが署名に使用した秘密キーと証明書に対応する必要があります。

2. 管理 UI 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ます。

- 署名された認証リクエストが必要です（[暗号化 & 署名] 設定）

このチェック ボックスをオンにすると、アイデンティティ プロバイダは署名された認証リクエストを必要とし、IdP はリクエストの署名を検証します。認証リクエストに署名されていない場合、アイデンティティ プロバイダはそれを拒否します。

重要: AuthnRequests に署名すると、未承認応答がアイデンティティ プロバイダから送信されることはありません。

- HTTP-Redirect（[SAML プロファイル] 設定）

このチェック ボックスをオンにすると、アイデンティティ プロバイダは SLO リクエストおよびレスポンスの署名を検証します。

3. [発行者 DN] および [シリアル番号] フィールド（[暗号化 & 署名] 設定）に入力します。

フィールドの値は、証明書データ ストア内の証明書に一致する必要があります。証明書は、リクエストに署名する機関の秘密キー/証明書ペアに対応するものです。入力値が一致することを確認するには、証明書の DN を表示します。

NameID とアサーションの暗号化

アサーション内で名前 ID を暗号化したり、アサーションそのものを暗号化したりできます。暗号化することにより、アサーションの転送時に新たなレベルの保護が得られます。

暗号化を設定する場合、パートナー証明書を指定します。この証明書はアサーション内に配置します。アサーションがサービス プロバイダに到着すると、サービス プロバイダでは、関連付けられた秘密キーを使用して、暗号化されたデータを復号します。

注: 暗号化を有効にする場合、最初のフェデレーション コールで暗号化ライブラリをロードし、追加のメモリを割り当てるためにポリシー サーバのメモリが増加する可能性があります。

暗号化の有効化

暗号化を実装する方法

1. 管理 UI にログインします。
2. 設定する SAML サービス プロバイダの [暗号化 & 署名] 設定に移動します。
3. アサーション暗号化の設定項目を設定します。

以下の条件に注意してください。

- 暗号化キーアルゴリズムとして `rsa-oaep` を選択する場合、最低限必要なキーサイズは 1024 ビットです。
- 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 Policy Files をインストールします。これらのファイルは、<http://java.sun.com/javase/downloads/index.jsp> からダウンロードできます

- [IssuerDN] および [シリアル番号] フィールドには、証明書発行者の DN およびその関連のシリアル番号をそれぞれ入力します。この情報により、証明書データ ストア内でサービス プロバイダの証明書が検索されます。サービス プロバイダがこのデータを提供します。

入力する IssuerDN およびシリアル番号の値は、アイデンティティ プロバイダの証明書データ ストアに保存されたキー/証明書ペアの IssuerDN およびシリアル番号と一致する必要があります。

4. [サブミット] をクリックして、変更を保存します。

IdP においてプロキシ サーバで処理されるリクエスト

IdP としてのポリシー サーバは、リクエストを処理する前に、フェデレーション Web サービス アプリケーションのローカル URL を使用して、メッセージ属性を検証します。

たとえば、SP からの認証リクエスト メッセージには、以下の属性が含まれることがあります。

```
Destination="http://idp.domain.com:8080/affwebservices/public/saml2sso"
```

この例では、認証リクエスト内の宛先属性とフェデレーション Web サービス アプリケーションのアドレスは同じです。ポリシー サーバは、宛先属性が FWS アプリケーションのローカル URL に一致することを確認します。

ポリシー サーバがプロキシ サーバの後に配置されている場合、ローカル URL と宛先属性 URL は同じにはなりません。宛先属性は、プロキシ サーバの URL です。たとえば、認証リクエストに以下の宛先属性が含まれているとします。

```
Destination="http://proxy.domain.com:9090/affwebservices/public/saml2sso"
```

フェデレーション Web サービスのローカル URL

(<http://idp.domain.com:8080/affwebservices/public/saml2sso>) が宛先属性と一致しないため、ポリシー サーバはリクエストを拒否します。

プロキシ設定の指定により、ポリシー サーバがメッセージ属性の検証に使用するローカル URL の判別方法を変更できます。プロキシを指定すると、ローカル URL の *protocol://authority* の部分がプロキシ サーバ URL と置換されます。その結果、2 つの URL が一致します。

プロキシ サーバを使用して処理するリクエストの設定

ポリシー サーバをプロキシ サーバの後ろに配置することができます。この展開の場合、リクエスト メッセージ属性内の URL とローカル プロキシ URL の間の一致が検索されるように、プロキシを設定します。リクエストを処理するための一致は必ず存在します。ポリシー サーバは、ローカル URL の *protocol://authority* の部分をプロキシ サーバ URL と置換します。その結果、2 つの URL が一致します。

IdP でプロキシ サーバを使用する方法

1. 管理 UI にログインします。
2. 設定する サービス プロバイダの [一般] 設定に移動します。
3. プロキシ サーバの部分 URL を *protocol://authority* の形式で入力します。

たとえば、プロキシ サーバ設定は次ようになります。

`http://proxy.domain.com:9090`

ネットワークに SPS フェデレーション ゲートウェイが含まれる場合、[サーバ] フィールドには、たとえば、SPS フェデレーション ゲートウェイ ホストおよびポートを指定する必要があります。

`http://sps_gateway_server.ca.com:9090`

4. [サブミット] をクリックして、変更を保存します。

[サーバ] フィールドに入力する値は、以下の IdP サービスの URL に影響します。

- シングルサインオン サービス
- シングルログアウト サービス
- Artifact 解決サービス
- 属性サービス
- 認証 URL -- プロキシ サーバ URL を使用します。ポリシー サーバは、ユーザを認証した後、シングルサインオン サービスを使用できるように、ユーザをプロキシ サーバにリダイレクトします。

[サーバ] の値は、宛先属性と同様に、SAML 属性の確認に使用される URL の一部になります。1 つの URL に対して 1 つのプロキシ サーバを使用している場合は、これらのすべての URL に対してそのプロキシ サーバを使用してください。

第 15 章: SAML 2.0 サービスプロバイダの設定

このセクションには、以下のトピックが含まれています。

- [サービスプロバイダのセットアップ](#) (P. 287)
- [依存パートナーの前提条件](#) (P. 291)
- [SAML 2.0 認証方式を設定する方法](#) (P. 291)
- [認証方式タイプの選択](#) (P. 293)
- [SAML 2.0 認証方式に関する一般情報の指定](#) (P. 294)
- [SAML 2.0 認証用のユーザレコードの検索](#) (P. 294)
- [SPでのシングルサインオンの設定](#) (P. 298)
- [シングルログアウトの有効化](#) (P. 304)
- [サービスプロバイダでのデジタル署名オプション](#) (P. 306)
- [シングルサインオンのアサーション暗号化要件の適用](#) (P. 307)
- [カスタム SAML 2.0 認証方式の作成 \(オプション\)](#) (P. 308)
- [サービスプロバイダでの IDP ディスカバリ設定](#) (P. 309)
- [メッセージコンシューマプラグインによるアサーション処理のカスタマイズ](#) (P. 312)
- [HTTP ヘッダとしての SAML 属性の供給](#) (P. 316)
- [SAML 2.0 認証が失敗した場合のリダイレクト URL の指定](#) (P. 325)
- [SPにおいてプロキシサーバで処理されるリクエスト](#) (P. 326)
- [バックチャネル対応のクライアント証明書認証の有効化 \(オプション\)](#) (P. 328)
- [SAML 2.0 認証方式でターゲットフェデレーションリソースを保護する方法](#) (P. 331)

サービスプロバイダのセットアップ

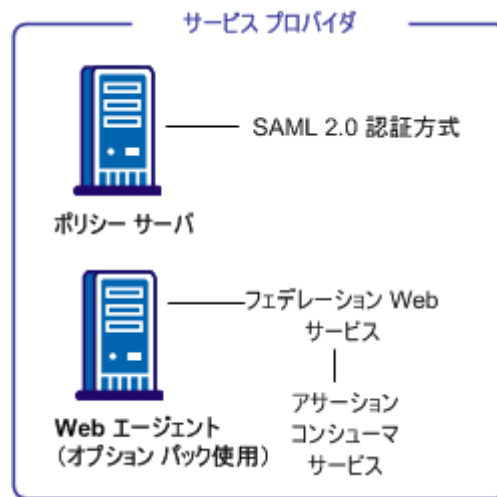
SiteMinder または SPS フェデレーションゲートウェイは、SAML 2.0 サービスプロバイダとして機能することができます。サービスプロバイダでは、アイデンティティプロバイダから受信するアサーションを使用して、ユーザを認証した後、リクエストされたフェデレーションリソースへのアクセスを提供します。SiteMinder サービスプロバイダがそのサイトのユーザストアへのアクセス権を持つと仮定し、サービスプロバイダは、SiteMinder SAML 2.0 認証方式を使用してユーザを認証します。

SAML 2.0 認証方式の場合、クロスドメイン シングル サインオンが可能になります。サービスプロバイダは、アイデンティティプロバイダからのアサーションを消費し、ユーザを識別して、**SiteMinder** セッションを確立できます。セッションが確立された後、サービスプロバイダは、特定のリソースに対してユーザに認可を与えることができます。

次の図は、サービスプロバイダでの認証のためのコンポーネントを示しています。

注: 1つのサイトがアイデンティティプロバイダとサービスプロバイダの両方として機能できます。

SAML 2.0 認証の主要コンポーネントを次の図に示します。



注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション 機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシ サーバ 管理 ガイド*」 (Secure Proxy Server Administration Guide) を参照してください。

SAML 2.0 認証方式は、サービスプロバイダサイトに存在するポリシーサーバで設定されます。認証方式は、サービスプロバイダサイトの Web エージェントまたは SPS フェデレーション ゲートウェイにインストールされたアサーション コンシューマ サービス (フェデレーション Web サービス アプリケーションのコンポーネント) を呼び出します。サービスは、SAML 認証方式からの情報を取得し、その情報を使用して、SAML アサーションから必要な情報を抽出します。

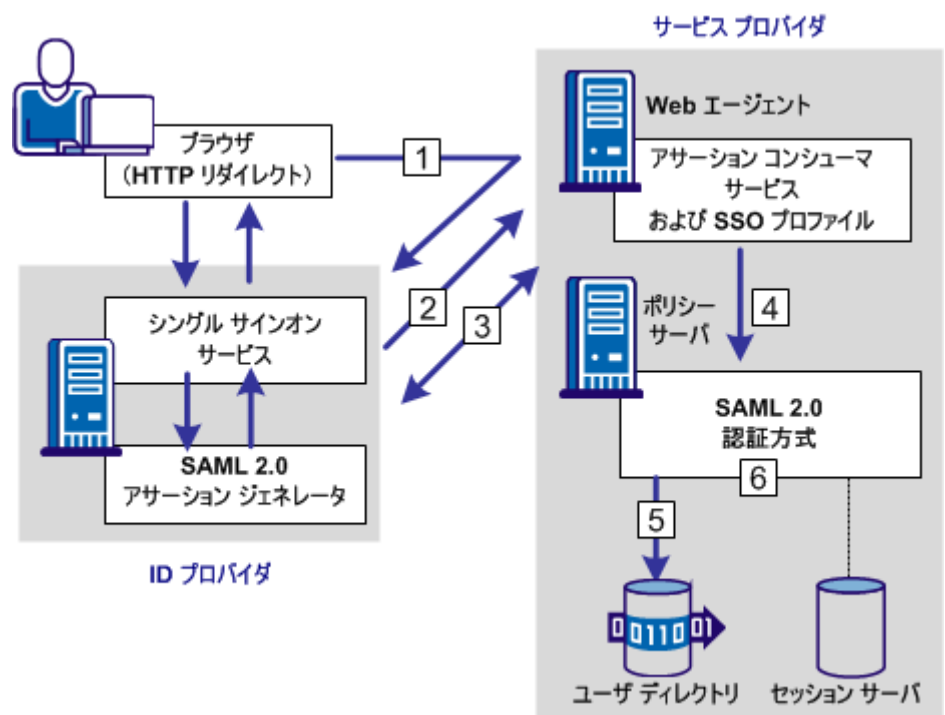
SAML アサーションは、サービス プロバイダポリシー サーバにログインするためのユーザ認証情報になります。ユーザの認証および認可が行われ、認可が成功すると、このユーザはターゲット リソースにリダイレクトされます。

アサーション コンシューマ サービスは、値が 0 の

AssertionConsumerServiceIndex が含まれる AuthnRequests を受理します。この設定の他のすべての値は拒否されます。

SAML 認証リクエストのプロセス

次の図は、SAML 2.0 認証方式がどのようにリクエストを処理するかを示しています。



注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション 機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシ サーバ 管理 ガイド*」 (Secure Proxy Server Administration Guide) を参照してください。

認証の機能的な流れは次のとおりです。

1. ユーザがサービスプロバイダリソースをリクエストします。このリクエストは、サービスプロバイダの認証リクエストサービスに送信されます。そして、このリクエストは **SAML** アサーションを取得するためにアイデンティティプロバイダにリダイレクトされます。
2. アイデンティティプロバイダは、サービスプロバイダにレスポンスを返します。

HTTP-POST バインドの場合、レスポンスにアサーションが含まれます。
HTTP Artifact バインドの場合、レスポンスに **SAML Artifact** が含まれます。

3. サービスプロバイダのアサーションコンシューマサービスがレスポンスメッセージを受信し、**POST** バインドまたは **Artifact** バインドが使用されているかどうかを判別します。

HTTP Artifact バインドの場合、アサーションコンシューマサービスは、アサーションを取得するためにアイデンティティプロバイダに **Artifact** を送信します。アイデンティティプロバイダは、アサーションが含まれるレスポンスを返します。アサーションコンシューマサービスは、アサーションが含まれるレスポンスをポリシーサーバに対する認証情報として使用します。

4. ポリシーサーバは、ユーザ認証情報が含まれる応答メッセージを認証対象方式に渡すことにより、**SAML 2.0** 認証方式をます。
5. ユーザ不明瞭解消プロセスが開始されます。
6. 不明瞭解消フェーズが完了した後、**SAML 2.0** 認証方式はアサーション内の認証情報を検証します。この認証方式は、アサーションの時間妥当性も検証します。該当する場合には、信頼されたアイデンティティプロバイダがアサーションに署名したことを確認します。

注: **POST** バインドの場合、署名が必須です。署名が存在しない場合、認証は失敗します。**Artifact** バインドの場合、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の安全なチャネル上でアサーションが取得されるため、署名されたアサーションはオプションです。

シングルログアウトが有効化されている場合は、**SLO** サブレットによってユーザが **No Access URL** にリダイレクトされます。

依存パートナーの前提条件

SiteMinder が依存パートナーとして機能するには、以下のタスクを完了する必要があります。

- ポリシー サーバをインストールします。
- 以下のコンポーネントのいずれかをインストールします。
 - Web エージェントおよび Web エージェント オプション パック。
Web エージェントは、ユーザを認証し、セッションを確立します。オプション パックはフェデレーション Web サービス アプリケーションを提供します。ネットワーク内の適切なシステムに必ず FWS アプリケーションを展開してください。
 - SPS フェデレーション ゲートウェイ (Web エージェントが組み込まれており、組み込まれた Tomcat Web サーバ上にフェデレーション Web サービス アプリケーションが配置されています)。

詳細については、「Web エージェント オプション パック ガイド」を参照してください。

- 検証およびメッセージの暗号化を必要とする機能のために秘密キーと証明書をインポートします。
- フェデレーション ネットワーク内でアサーティング パートナーをセットアップします。

SAML 2.0 認証方式を設定する方法

サービス プロバイダを設定するには、以下のタスクが必要となります。

1. SAML 2.0 認証方式の前提条件を実行します。
2. [認証方式タイプ](#) (P. 293) を選択します。
3. ユーザを認証するための不明瞭解消を設定します。
4. [シングル サインオンを設定](#) (P. 298) します。

フェデレーション パートナーとしてアサーションを生成するアイデンティティ プロバイダごとに SAML 認証方式を設定します。各方式をレルムにバインドします。レルムは、ユーザに要求されたターゲット リソースのすべての URL で構成されます。これらのリソースをポリシーで保護します。

ヒント：

- 設定が機能するためには、アイデンティティプロバイダとサービスプロバイダの特定のパラメータ値が一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」 (Configuration Settings that Must Use the Same Values) に記載されています。
- フェデレーション Web サービス サブレットの正しい URL を使用していることを確認してください。これらの URL は、「[SiteMinder 設定で使用されるフェデレーション Web サービス URL \(P. 457\)](#)」 (Federation Web Services URLs Used in SiteMinder Configuration) に一覧表示されています。

サービスプロバイダに関するオプション設定タスク

SiteMinder をサービスプロバイダとして設定する際に以下のオプションタスクがあります。

- [シングルログアウトの有効化 \(P. 304\)](#)
- [名前 ID またはアサーションの暗号化 \(P. 307\)](#)
- [Artifact 解決メッセージの署名 \(P. 307\)](#)
- [署名された Artifact レスポンスの要求 \(P. 307\)](#)
- [メッセージコンシューマプラグインを使用したアサーションのカスタマイズ \(P. 190\)](#)

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。

- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

認証方式タイプの選択

サービス プロバイダは、アサーション内の ID 情報を使用して、保護されたフェデレーション リソースへのアクセスを認可します。このプロセスでは SAML 認証方式が使用されます。

SAML 2.0 認証方式を割り当ててリソースを保護するには、先に方式を設定する必要があります。

次の手順に従ってください：

1. SAML 2.0 認証方式の前提条件を再度確認しておいてください。
2. 管理 UI にログインします。
3. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[一般] 設定に [認証方式] ページが表示されます。
4. 認証方式に名前を付けます。
5. [認証方式タイプ] ドロップダウン リストで、[SAML 2.0 テンプレート] を選択します。この方式の保護レベルも選択できます。
[認証方式] ダイアログ ボックスの内容が、SAML 2.0 をサポートするように変化します。
6. [方式のセットアップ] セクションで、[SAML 2.0 設定] をクリックして、認証方式の詳細を定義します。

認証方式をはじめて設定する場合は、設定ウィザードに従って認証方式をセットアップしてください。

SAML 2.0 認証方式に関する一般情報の指定

SAML 2.0 認証方式の [一般] 設定にサービス プロバイダおよびアイデンティティ プロバイダを指定します。

次の手順に従ってください:

1. 認証方式のメイン ページから、[SAML 2.0 認証方式] をクリックします。

既存の方式を変更する場合は、[変更] をクリックした後、[SAML 2.0 設定] をクリックします。

その方式に関する詳細な設定が表示されます。

2. [一般] 設定の必須フィールドに入力します。
3. [ユーザの特定] セクションに移動します。

SAML 2.0 認証用のユーザレコードの検索

認証スキームを設定する際に、認証スキームがローカル ユーザ ストアでユーザを検索する方法を定義します。正しいユーザが検索された後、システムはこのユーザのセッションを生成します。ユーザ ストアでのユーザの検索は、不明瞭解消のプロセスです。ポリシー サーバがユーザの不明瞭解消を行う方法は、認証方式の設定により異なります。

不明瞭解消を正しく行うために、認証スキームでは、最初のアサーションから LoginID を判別します。LoginID は、ユーザを識別に使用する SiteMinder 固有の用語です。デフォルトでは、LoginID はアサーション内の名前 ID から抽出されます。また、Xpath クエリを使用することによっても LoginID を取得できます。

認証方式が **LoginID** を判別した後、ポリシー サーバは認証方式に対して検索仕様が設定されているかどうかを確認します。認証方式に対して検索仕様が定義されていない場合、**LoginID** がポリシー サーバに渡されます。ポリシー サーバでは、ユーザストア検索仕様と共に **LoginID** を使用してユーザを検索します。たとえば、**LoginID** の値が **Username** で、LDAP 検索仕様が **uid** 属性に設定されているとします。すると、ポリシー サーバは、**uid** の値 (**Username=uid**) を使用してユーザを特定します。

認証スキームに対して検索仕様が設定されている場合、**LoginID** はポリシー サーバに渡されません。その代わりに、検索仕様を使用してユーザが検索されます。

以下の 2 つの方法のいずれかを使用してユーザの不明瞭解消を設定できます。

- ローカルで、認証方式の一部として設定する
- 設定済みの **SAML** アフィリエーションを選択することによって設定する

認証方式の一部としての不明瞭解消のローカル設定

不明瞭解消をローカルで選択する場合、そのプロセスには以下の 2 つの手順があります。

1. デフォルト動作または **Xpath** クエリの使用によって **LoginID** を取得する。
2. デフォルト動作またはユーザ ルックアップを定義することによって、ユーザストアでユーザを検索する。

注: **Xpath** および検索仕様の使用はオプションです。

LoginID の取得

LoginID は、以下の 2 つの方法で検索できます。

- **LoginID** がアサーションの **NameID** から抽出されるというデフォルト動作を利用する。この場合、設定は不要です。
- デフォルト動作の代わりに、**Xpath** クエリを使用して **LoginID** を検索する。

Xpath クエリを使用して LoginID を判別する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] をクリックします。
3. [SAML 2.0 プロパティ] ページから、認証方式が LoginID を取得するために使用する Xpath クエリを入力します。[OK] をクリックして、変更を保存します。

Xpath クエリにはネームスペースプレフィックスを含めることはできません。以下の例は無効な Xpath クエリです。

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

有効な Xpath クエリは次のとおりです。

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

ユーザの検索の設定

LoginID を取得した後、デフォルト動作 (LoginID がポリシー サーバに渡される) 以外の方法でユーザを検索するように設定できます。

検索仕様を使用してユーザを検索する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] をクリックします。
3. [ユーザの検索] セクションで、適切なネームスペース フィールドに検索仕様を入力します。検索仕様は、認証方式がネームスペースを検索するために使用する属性を定義します。LoginID を表すエントリとして %s を使用します。

たとえば、LoginID の値が user1 であるとしします。[検索仕様] フィールドで Username=%s と指定すると、文字列は Username=user1 となります。この文字列は、ユーザストアに照らして確認され、認証のための正しいレコードが検索されます。

4. [OK] をクリックして、設定の変更内容を保存します。

SAML アフィリエーションを使用したユーザレコードの検索(オプション)

サービスプロバイダのグループは、アフィリエーションを形成できます。サービスプロバイダをグループ化することで、フェデレーションネットワーク全域での関連付けが確立され、アフィリエーションの1つのメンバとのリレーションシップによって、アフィリエーションのすべてのメンバとのリレーションシップが確立されます。

アフィリエーション内のすべてのサービスプロバイダが、単一のプリンシパルの名前識別子を共有します。1つのアイデンティティプロバイダがユーザを認証し、そのユーザにIDを割り当てると、アフィリエーションのすべてのメンバがその同じ名前IDを使用します。単一の名前IDを使用することで、それぞれのサービスプロバイダで必要とされる設定が削減されます。さらに、1つのプリンシパルに対して1つの名前IDを使用することにより、アイデンティティプロバイダでのストレージ容量が節約されます。

ユーザ不明瞭解消に、オプションのXpathクエリおよび検索仕様を使用できます。これらのオプションは、認証方式の一部ではなくアフィリエーション自体の一部として定義されます。

注: アフィリエーションを認証方式設定で使用するには、先にアフィリエーション定義してください。

アフィリエーションを選択する方法

1. SAML 2.0 認証方式ページに移動します。
2. [SAML 2.0 設定] をクリックします。
3. [一般] 設定。
4. [ユーザの特定] セクションで、[SAML アフィリエーション] ドロップダウンフィールドから事前定義済みのアフィリエーションを選択します。これらのアフィリエーションは、アイデンティティプロバイダで設定されています。

SP でのシングル サインオンの設定

アイデンティティ プロバイダとサービス プロバイダの間のシングル サインオンを確立するには、SSO バインディングを指定します。

SSO 設定により、Artifact または POST バインドを使用するシングル サインオンを設定します。

シングル サインオン設定の一環として、リダイレクト モード設定を定義します。リダイレクト モードにより、SiteMinder がどのようにアサーション属性（利用可能な場合）をターゲット アプリケーションに送信するかが指定されます。アサーション属性は HTTP ヘッダまたは HTTP Cookie として送信できます。

HTTP ヘッダおよび HTTP Cookie にはサイズ制限があり、アサーション属性でそれらを超えるはできません。サイズ制限は以下のとおりです。

- HTTP ヘッダ：SiteMinder は、Web サーバのヘッダに関するサイズ制限までの属性をヘッダで送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダ サイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。
- HTTP Cookie：SiteMinder は、Cookie のサイズ制限までの Cookie を送信できます。それぞれのアサーション属性は、それ自身の Cookie として送信されます。Cookie のサイズ制限はブラウザ固有です。また、その制限は、属性ごとではなく、アプリケーションに渡されるすべての属性に適用されます。Cookie のサイズ制限を調べるには、使用する Web ブラウザのドキュメントを参照してください。

シングル サインオンを設定する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [SSO] を選択します。
3. SSO フィールドに入力します。
4. （オプション）シングル サインオンが動作するターゲット リソースを指定します。ターゲットとしては、宛先のサービス プロバイダ サイトにおける要求対象リソースを指定します。

サービス プロバイダは、デフォルトのターゲットを使用する必要はありません。シングル サインオンを開始するリンクには、ターゲットを指定するクエリ パラメータを含めることができます。

5. [バインド] セクションで、HTTP-Artifact および HTTP-POST を選択できます。

HTTP-POST が選択されており、Artifact が選択されていない場合、POST バインディングのみがアイデンティティ プロバイダから受理されます。バインディングが指定されない場合、デフォルトは HTTP-artifact です。

HTTP-Artifact バインドを選択する場合、以下のように操作してください。

- [セッション サーバを有効化 \(P. 117\)](#) して、アサーションが取得される前にアサーションを保存するようにします。
- [バックチャネルを設定 \(P. 302\)](#) します。Artifact 解決サービスとの通信を保護する認証方式のタイプを選択します。このサービスは、アイデンティティ プロバイダにおいてアサーションを取得します。
- オプションで、各バインドのインデックス エントリとして整数を指定します。

複数のエンドポイントがある場合、インデックス付きのエンドポイントを設定できます。サービス プロバイダには、指定されたエンドポイント エントリが認証リクエストのクエリ パラメータとして含まれます。認証リクエストは、アイデンティティ プロバイダのシングル サインオン サービスに送信されます。

詳細情報:

[Artifact サービスを保護する認証方式の設定 \(P. 160\)](#)

使い捨てポリシーの適用によるセキュリティ強化

使い捨てポリシーは、サービス プロバイダで別のセッションを確立するために SAML 2.0 アサーションが再利用されるのを防止します。この機能は、POST バインドの目的で送信されてくるアサーションに適用されます。

注: HTTP-POST バインドを選択すると、使い捨てポリシー機能がデフォルトで有効化されます。

アサーションを単一使用に指定することで、シングル サインオン環境全域における認証のセキュリティを強化できます。ブラウザから、SiteMinder セッションを確立するために使用された SAML アサーションを攻撃者が取得する可能性があります。この攻撃者は、別のセッションを確立するために、サービス プロバイダのアサーション コンシューマ サービスにアサーションをポストする可能性があります。ただし、アサーションが 1 回使用に指定されている場合には、この種のリスクが軽減されます。

SiteMinder では、有効期限データを使用して、使い捨てポリシーを適用します。有効期限データは、アサーションに関する時間ベースのデータです。SAML 2.0 認証方式では、有効期限データがセッション ストアに保存されています。有効期限データにより、SAML 2.0 POST アサーションが 1 回だけ使用されることが確認されます。

使い捨てポリシーが適用される仕組み

SAML 2.0 アサーションの検証に成功すると、認証方式は、アサーションデータを有効期限データ テーブルに書き込みます。このデータには、アサーション ID キーおよび有効期限が含まれます。ポリシー サーバ内のセッション ストア管理スレッドにより、期限切れデータが有効期限データ テーブルから削除されます。

認証方式がアサーション データを検証し、有効期限データ エントリに同じアサーション ID キーが存在する場合、アサーション データの書き込みは失敗します。認証方式が有効期限データ テーブルに書き込むことができない場合、SAML 2.0 認証方式は、無効なアサーションと同じ方法で認証を拒否します。

データベースが利用できない場合、アサーションの使い捨ては適用できません。その結果、認証方式がリクエストを拒否し、アサーションは再利用されません。

使い捨てポリシーの設定

使い捨てポリシーを設定する方法

1. SAML 2.0 認証方式に移動します。
[変更] - [SAML 2.0 設定] をクリックします。
2. [SSO] タブを選択します。

3. [HTTP-POST] セクションの[使い捨てポリシーを構成する] チェックボックスはデフォルトで選択されています。
4. セッションストアを有効にします。
セッションストアを有効化する手順については、「ポリシー サーバ管理ガイド」を参照してください。

詳細情報:

[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 \(P. 117\)](#)
[使い捨てポリシーの適用によるセキュリティ強化 \(P. 299\)](#)

SSO の名前識別子の作成の許可

シングルサインオン リクエストの一部として、サービス プロバイダは、**AllowCreate** という名前の属性 (**true** に設定) が含まれる認証リクエストを生成できます。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、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 プロバイダは、**NameID** 属性の値が見つからない場合、永続的な識別番号を生成します。Allow/Create 機能により、この識別番号の作成が有効化されます。

永続的な識別番号は、ランダムに生成された ID です。アイデンティティ プロバイダでは、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、アイデンティティ プロバイダは、アサーションをサービス プロバイダに返します。たとえば、**NameID** 属性が電話に設定されており、ユーザ レコードに電話の値が存在しない場合、**NameID** は、ランダムに生成された識別番号に設定されます。

サービス プロバイダがアサーションを受信すると、**SAML 2.0** 認証方式がレスポンスを処理します。その後、認証方式は、そのローカルユーザストアでユーザの検索を実行します。サービス プロバイダは、ユーザ レコードを検出した場合、そのユーザにアクセス権を付与します。

アイデンティティ プロバイダが固有の識別番号を生成するには、アイデンティティ プロバイダで **Allow/Create** 機能を有効にします。アイデンティティ プロバイダでは、機能が有効化されている場合のみ識別番号を生成します。固有の識別番号が作成されなかったというエントリがアイデンティティ プロバイダのログ ファイルに作成された後、アサーション生成の通常フローが続行されます。

認証リクエストへの Allow/Create 属性の追加

アイデンティティ プロバイダに名前 ID の識別子の作成を許可するには、認証リクエスト メッセージに **Allow/Create** 属性を含めます。

注: アイデンティティ プロバイダの管理者は、生成される識別子に対して **Allow/Create** 機能を有効にする必要があります。

次の手順に従ってください:

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [SSO] を選択します。
3. [IDP による新規識別子の作成を許可する] チェック ボックスをオンにします。
4. [OK] をクリックします。

HTTP-Artifact SSO に関するバックチャネルの設定

シングル サインオンに HTTP-Artifact バインドを選択する場合、Artifact 解決サービスまでのバックチャネルのセキュリティを保護する認証方式を選択します。このサービスは、アイデンティティ プロバイダにおいてアサーションを取得します。

バックチャネルを設定する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [暗号化] - [署名] を選択します。

3. [バックチャネル]セクションで、すべてのフィールドに入力します。

重要: バックチャネル認証方式にベーシック認証を使用している場合、[SP 名] フィールドの値はサービス プロバイダの名前になります。追加の設定は不要です。クライアント証明書認証を使用している場合、[SP 名] フィールドは、証明書データ ストアに保存されたクライアント証明書のエイリアスであることが必要です。SP は、Artifact 解決サービスへのアクセスを獲得取得するための認証情報として証明書を使用します。

4. [OK] をクリックして設定を保存します。

詳細情報:

[バックチャネル対応のクライアント証明書認証の有効化 \(オプション\) \(P. 328\)](#)

サービス プロバイダでの ECP の設定

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

ECP の詳細については、「[概要](#) (P. 243)」を参照してください。

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. サービス プロバイダで 管理 UI にログインします。
3. 関連する SAML 2.0 認証方式オブジェクトを変更します。
4. [方式のセットアップ] セクションの [SAML 2.0 設定] をクリックします。

方式の設定タブが表示されます。

5. [SSO] タブを選択します。

6. [拡張されたクライアントとプロキシプロファイル] チェック ボックスをオンにして、[OK] をクリックします。
7. [サブミット] をクリックして変更内容を保存します。

SiteMinder 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 SAML 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。

シングル ログアウトの有効化

シングル ログアウト (SLO) プロファイルを使用することにより、特定のセッション 機関によって提供され、特定のユーザと関連付けられているすべてのセッションのほぼ同時のログアウトが可能になります。ユーザは、ログアウトを直接開始します。セッション機関は、最初にユーザを認証した認証エンティティです。ほとんどの場合、セッション機関はアイデンティティ プロバイダです。

シングル ログアウトは、セッションを確実に閉じるために役立ち、サービス プロバイダで不正なユーザがリソースのアクセスを獲得するのを防止できます。

ユーザは、サービス プロバイダまたはアイデンティティ プロバイダでリンクをクリックすることにより、ブラウザからシングル ログアウト サービスを開始できます。ユーザは、SLO サブレットを指すログアウト リンクをクリックします。このサブレットは、フェデレーション Web サービスのコンポーネントであり、サービス プロバイダやアイデンティティ プロバイダから送信されたログアウト リクエストおよびレスポンスを処理します。サブレットでは、リクエストやレスポンスの発信元が既知である必要はありません。このサブレットは、SiteMinder セッション Cookie を使用して、ログアウトするセッションを判別します。

シングル ログアウトのバインディング

シングル ログアウト機能は、HTTP リダイレクト バインディングを使用して、メッセージをトランスポートします。このバインディングは、HTTP リダイレクト メッセージ（302 のステータス コード レスポンス）を使用して、SAML プロトコル メッセージがどのように転送されるか決定します。

シングル ログアウトの設定

サービス プロバイダでシングル ログアウトを有効にする場合は、サービス プロバイダで保護されたリソースが含まれるレルムに対する永続セッションを設定します。管理 UI で永続セッションを設定します。

シングル ログアウトを設定する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [SLO] を選択します。
3. ページの [SLO] セクションで、[HTTP リダイレクト] チェック ボックスをオンにします。他のシングル ログアウト設定がアクティブになります。
4. 以下の情報に注意して、残りのフィールドに値を入力します。

有効期間

シングル ログアウト リクエストが有効な秒数を指定します。有効期間が期限切れになると、シングル ログアウト レスポンスが生成されます。レスポンスは、ログアウトを開始したエンティティに送信されます。また、有効期間は、シングル ログアウト メッセージ期間の計算でスキュー時間にも依存します。

SLO ロケーション URL、SLO レスポンス ロケーション URL、SLO 確認 URL

これらのフィールドのエントリは <https://> または <http://> で始まる必要があります。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ] をクリックします。

シングル ログアウトが開始された後、アイデンティティ プロバイダおよびすべてのサービス プロバイダ サイトでのユーザセッションが削除されます。その後、フェデレーション Web サービスによってユーザがログアウト確認ページにリダイレクトされます。

詳細情報:

[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 \(P. 117\)](#)

サービス プロバイダでのデジタル署名オプション

SAML 2.0 認証方式設定には、以下のトランザクションに対するデジタル署名オプションが含まれています。

- 認証リクエスト
- シングル ログアウト リクエストおよびレスポンス
- Artifact 解決メッセージ -- アサーションを取得するための SAML Artifact の解決
- 属性クエリ -- 属性機関 (IdP) と SAML リクエスタ (SP) の間で実行される認可

デフォルトでは、署名の処理は有効化されています。これは、SAML 2.0 仕様で署名が必要とされるためです。最初のフェデレーションセットアップのデバッグのみに限り、[署名の処理を無効にする] オプションを選択することにより、サービス プロバイダに対するすべての署名処理（署名および署名の検証）を一時的に無効化できます。デバッグが完了した後は、署名処理を再度有効にしてください。

重要: 実稼働環境で処理する署名を無効にすると、必須のセキュリティ機能を無効化していることになります。

署名オプションを指定する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [暗号化 & 署名] をクリックします。

3. [D-Sig 情報] セクションで、フィールドに入力します。以下の情報に注意してください。
 - HTTP-POST（シングル サインオン）の場合、ポストされたアサーションの署名を検証する証明書に関する情報を入力します。発行者 DN と シリアル番号の組み合わせにより、IdP がアサーションに署名するために使用した秘密キーに対応する証明書が識別されます。

[発行者 DN] フィールドに入力する値は、証明書データ ストアの証明書の発行者 DN と一致する必要があります。
 - HTTP リダイレクト（シングル ログアウト）の場合、SLO リクエストの署名を検証する証明書に関する情報を入力します。
4. ダイアログ ボックスの [署名処理] セクション内の設定を完了します。
5. HTTP-Artifact シングル サインオンの場合のみ、バックチャネルを設定します。
6. [OK] をクリックします。

シングル サインオンのアサーション暗号化要件の適用

暗号化機能により、認証方式がアサーション内の暗号化されたアサーションまたは名前 ID のみを処理することが指定されます。

セキュリティを強化する場合は、アイデンティティ プロバイダにおいて名前 ID、ユーザ属性、またはアサーション全体を暗号化することができます。暗号化することにより、アサーションの転送時に新たなレベルの保護が得られます。アイデンティティ プロバイダにおいて暗号化が有効化されている場合、データの暗号化に証明書（公開キー）が使用されます。アサーションがサービス プロバイダに到着すると、サービス プロバイダでは関連付けられた秘密キーを使用して、暗号化されたデータを復号します。

セッション プロバイダで暗号化を設定する場合、アサーションには暗号化された名前 ID またはアサーションが含まれる必要があります。そうでない場合、サービス プロバイダはアサーションを拒否します。

SSO の暗号化のセットアップ

アサーションの暗号化要件を適用できます。

暗号化要件を適用する方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [暗号化 & 署名] をクリックします。
暗号化と署名の設定ページが表示されます。
3. 暗号化された名前 ID を要求するには、[暗号化された名前 ID が必要] チェック ボックスをオンにします。
4. 暗号化されたアサーションを要求するには、[暗号化されたアサーションが必要] チェック ボックスをオンにします。
名前 ID およびアサーションを選択できます。
5. (オプション) アイデンティティ プロバイダから受信したアサーション内の暗号化されたデータを復号する秘密キーのエイリアスを指定します。
6. [OK] をクリックして、変更を保存します。

暗号化を要求しない場合、サービス プロバイダは、名前 ID やアサーションが暗号化されていても、クリア テキストでも、これらを受理します。

カスタム SAML 2.0 認証方式の作成(オプション)

既存の SAML 2.0 認証テンプレートの代わりに SiteMinder 認証 API で記述したカスタム SAML 2.0 方式を使用できます。

認証方式のメイン ページには、[方式のセットアップ] セクションに [ライブラリ] フィールドが含まれます。このフィールドには、SAML Artifact 認証を処理する共有ライブラリの名前が含まれています。カスタム認証方式を作成していない場合は、この値を変更しないでください。

デフォルトの共有ライブラリは `smauthhtml` です。

サービスプロバイダでの IDP ディスカバリ設定

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

注: 認証を要求するユーザは、以前にアイデンティティ プロバイダにアクセスし、認証を受けていることが必要です。

IDP ディスカバリは、以下のように行われます。

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. サイト選択ページが共通のドメインの IDP ディスカバリ サービス URL にユーザをリダイレクトします。リダイレクト URL には、共通ドメイン Cookie を必要とすることを示すクエリ パラメータが含まれます。
3. IDP ディスカバリ サービスは、共通ドメイン Cookie の値を取得し、これをクエリ パラメータとして設定します。その後、IDP ディスカバリ サービスは、SP のサイト選択ページにユーザをリダイレクトして戻します。
4. SP は、サイト選択ページに IdP ID を入力します。これらは、そのユーザが以前に認証を受けた URI です。
5. ユーザは IdP を選択してユーザ認証を実行します。

詳細情報:

[IdP でのアイデンティティプロバイダ ディスカバリの設定 \(P. 279\)](#)

SP でのアイデンティティプロバイダ ディスカバリの設定

サービスプロバイダでのアイデンティティプロバイダ ディスカバリ プロファイルの設定に 管理 UI は関与しません。このプロファイルは、アイデンティティプロバイダの 管理 UI で有効化されます。

プロセスの最初の手順は、サイト選択ページを作成することです。ポリシー サーバには、SP で使用可能な `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。

サイト選択ページの最初のリンクは、任意のドメインから共通ドメイン内の IdP ディスカバリ サービスにブラウザをリダイレクトします。サービスは、`_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 サービスは、リクエストを受信すると、共通ドメイン Cookie を取得します。このサービスは、共通ドメイン Cookie をクエリ パラメータとしてリンクに追加します。その後、サービスは、通常ドメインの `IdPDiscovery.jsp` サイト選択ページにユーザをリダイレクトして戻します。

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。リストは静的のものであり、リストに関連付けられた HTML リンクはないため、リンクから関連する IdP との通信を開始するようにはなっていません。

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。

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

`web_agent_home/affwebservices/public`

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/saml2idp/?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 ディスカバリが動作すると、サイト選択ページには、選択対象の IdP のリストが表示されます。

IdP ディスカバリ ターゲットの攻撃からの保護

SiteMinder 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 を配置することができます。この URL により、悪意のあるサイトにリダイレクトされ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、この設定は特に連携に対して定義されます。

IPD サービスは、**IPDTarget** クエリ パラメータを検査します。このサービスは、クエリ パラメータによって指定される 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、サービスは検証を一切実行せずに、ユーザをターゲット URL にリダイレクトします。

メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ

メッセージ コンシューマ プラグインは、メッセージ コンシューマ プラグインを実装する 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。

注: 認証と不明瞭解消に関するステータス コードの詳細については、「Java 用 SiteMinder 用プログラミング ガイド」を参照してください。

認証時、**SiteMinder** 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、**SiteMinder** 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、**SiteMinder** 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、**SiteMinder** 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、**SiteMinder** 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。

メッセージ コンシューマ プラグインに関する追加の情報については、以下を参照してください。

- 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「*Java Developer Reference*」にあります。**MessageConsumerPlugin** インターフェースを参照してください。
- 認証および認可の API に関する概要および概念情報については、「Java 用 **SiteMinder** プログラミング ガイド」を参照してください。

プラグインを設定する方法

1. **SiteMinder SDK** をインストールします (未インストールの場合)。
2. **MessageconsumerPlugin.java** インターフェースを実装します (**SiteMinder SDK** に含まれています)。
3. メッセージ コンシューマ プラグイン実装クラスを展開します。
4. 管理 UI でメッセージ コンシューマ プラグインを有効にします。

MessageConsumerPlugin の実装

`MessageConsumerPlugin.java` インターフェースを実装するにより、カスタム メッセージ コンシューマ プラグインを作成します。実装クラスの最小要件は、以下の手順に示されています。

次の手順に従ってください:

1. パラメータが含まれない公のデフォルト コンストラクタ メソッドを提供します。
2. 実装がステートレスになるように、コードを提供します。多数のスレッドが 1 つのプラグイン クラスを使用できる必要があります。
3. 現実の要件に応じて、インターフェース内のメソッドを実装します。

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

`init()`

プラグインが必要とする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、`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`」の下に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最終結果を決定する追加のコードを提供します。

SiteMinder から、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
(installation_home¥sdk¥samples¥messageconsumerplugin)

MessageConsumerSAML20.java
(installation_home¥sdk¥samples¥authextensionsaml20)
```

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、SiteMinder が実行可能ファイルを検索できることを確認します。

メッセージ コンシューマ プラグインを展開するには、以下の手順を実行します。

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、ポリシー サーバと共にインストールされています。

```
installation_home¥siteminder¥bin¥jars¥SmJavaApi.jar
```

SmJavaApi.jar の同一のコピーが SiteMinder SDK と共にインストールされています。このファイルは、`installation_home¥sdk¥Java¥SmJavaApi.jar` ディレクトリにあります。

開発時にそれらのいずれも使用できます。

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。 `installation_home¥siteminder¥config` ディレクトリの JVMOptions.txt ファイルの場所を特定します。

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ポリシー サーバの再起動 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。

SAML 2.0 におけるメッセージ コンシューマ プラグインの有効化

メッセージ コンシューマ プラグインを作成してコンパイルした後に、管理 UI 内で設定することにより、このプラグインを有効にします。UI 設定により、SiteMinder にプラグインの検索場所が指定されます。

[プラグインを展開](#) (P. 192)するまで、プラグイン設定を実行しないでください。

メッセージ コンシューマ プラグインを有効にする方法

1. 管理 UI にログオンします。
2. [SAML 2.0 認証方法] ダイアログ ボックスに移動します。
3. [詳細] をクリックします。
4. [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。

完全 Java クラス名

プラグインの Java クラス名を指定します。たとえば、SiteMinder 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`

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。

管理 UI でプラグインを設定する代わりに、ポリシー管理 API (C または Perl) を使用して、`IdpPluginClass` および `IdpPluginParameters` を設定します。

5. ポリシー サーバを再起動します。

HTTP ヘッダとしての SAML 属性の供給

アサーション レスポンスには、アサーションの属性を含めることができます。これらの属性は HTTP ヘッダ変数として提供できるため、クライアント アプリケーションでは、これらを使用して、より細かい粒度のアクセス制御を実行できます。

属性を HTTP ヘッダに含める場合には、以下のような利点があります。

- HTTP ヘッダは永続的ではありません。これらのヘッダは、それらが含まれるリクエストまたはレスポンス内のみに存在します。
- HTTP ヘッダは、SiteMinder Web エージェントによって提供される場合と同様に、ブラウザ内で非表示であるため、セキュリティ上の心配が軽減されます。

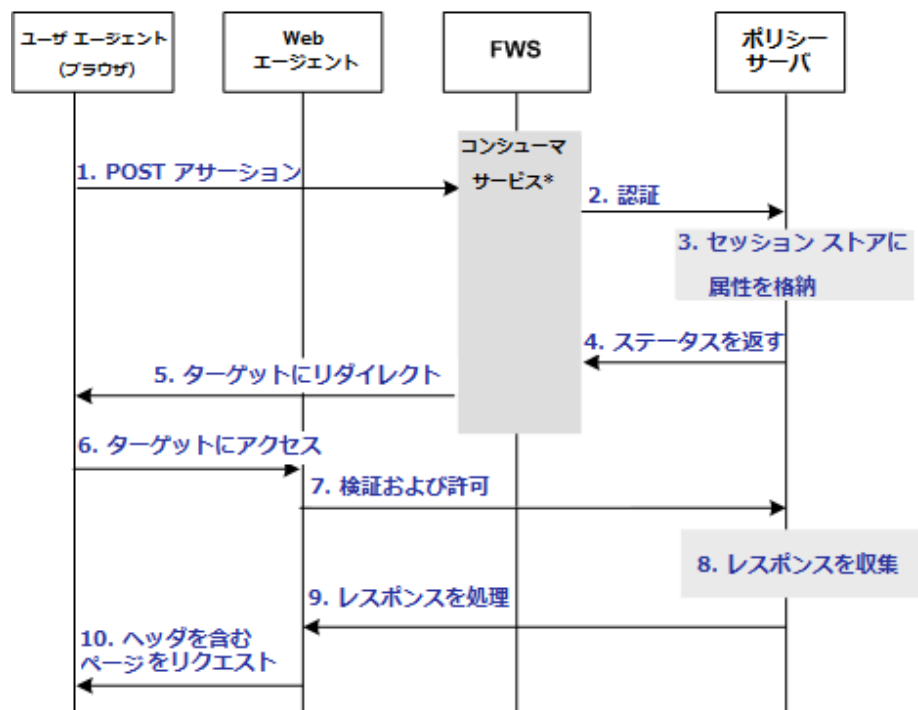
注: HTTP ヘッダにはサイズの制限があり、属性はその制限を超えることができません。SiteMinder では、ヘッダに関する Web サーバのサイズ制限に達するまで、属性をヘッダに送信できます。1つのヘッダごとに1つのアサーション属性のみが許可されます。ヘッダサイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。

HTTP ヘッダとしての SAML 属性のユース ケース

認証時、一連の SAML 属性がアサーションから抽出され、HTTP ヘッダとして提供されます。認可プロセス時、これらのヘッダはカスタマ アプリケーションに返されます。

次のフロー図は、ランタイムのイベントのシーケンスを表します。

コンシューマでの属性としてのヘッダの処理



*コンシューマ サービスは以下のいずれかです:

- SAML 認証情報コレクタ (SAML 1.x)
- アサーション コンシューマ サービス (SAML 2.0)
- セキュリティ トークン コンシューマ サービス (WS-Federation)

注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション 機能を提供できます。フロー図では、Web エージェント ブロックは SPS フェデレーション ゲートウェイ に組み込まれた Web エージェント になります。SPS フェデレーション ゲートウェイ をインストール する および 設定 する 詳細 については、「*Secure プロキシ サーバ 管理 ガイド*」 (Secure Proxy Server Administration Guide) を参照 してください。

属性を HTTP ヘッダとして処理する場合のイベントのシーケンスは以下のとおりです。

1. アサーションがアサーティング パーティで生成された後、アサーションが依存パーティの適切なコンシューマサービスに送信されます。配信方法 (POST または Artifact または WS フェデレーション) には無関係です。

注: コンシューマ サービスは、SAML 認証情報コレクタ (SAML 1.x の場合)、アサーション コンシューマ サービス (SAML 2.0 の場合)、またはセキュリティ トークン コンシューマ サービス (WS フェデレーションの場合) です。

2. コンシューマサービスは、そのローカル ポリシー サーバをコールし、設定済みの認証方式を使用してアサーションによるユーザ認証を実行します。
3. 認証方式のリダイレクト モードパラメータが `PersistAttributes` に設定されている場合、ポリシー サーバはセッション変数として属性をセッションストアにキャッシュします。
4. 認証の結果はコンシューマ サービスに返されます。
5. コンシューマ サービスは、保護されたターゲット リソースにブラウザをリダイレクトします。
6. ブラウザは、ターゲット リソースへのアクセスを試みます。
7. Web エージェントは、ポリシー サーバをコールして、ユーザセッションを検証すると共に、ユーザのターゲット リソースへのアクセスが許可されていることを確認します。
8. ポリシー サーバは、設定済みのレスポンスによって属性を取得します。
9. ポリシー サーバは、レスポンスを処理し、属性を Web エージェントに送信します。
10. Web エージェントは、必要に応じて HTTP ヘッダを設定します。

属性を HTTP ヘッダとして提供するため設定の概要

セッションストアにキャッシュされた SAML 属性を取得し、それらを HTTP ヘッダとして提供するには、複数の設定手順が必要となります。

次の手順に従ってください:

1. SAML 認証方式のリダイレクトモードとして **PersistAttributes** を選択します。これにより、SAML 属性が HTTP ヘッダとして返されるようになります。
2. ターゲット リソースが含まれるレルムに対する認可ルールを設定します。
3. ターゲット リソースを保護するレルムで **PersistentRealm** を設定します。
4. ヘッダとして提供される SAML 属性ごとに、アクティブなレスポンスタイプを使用するレスポンスを設定します。
5. 認可ルールとアクティブなレスポンスをバインドして属性のユーザを HTTP ヘッダとして実装するポリシーを作成します。

SAML 属性を保存するためのリダイレクトモードの設定

依存パーティが SAML アサーションでユーザを認証した後、SAML 属性はセッションストアに書き込まれます。その後、ブラウザはターゲット リソースにリダイレクトされます。

属性データを持つブラウザをリダイレクトする方法

1. 管理 UI にログインします。
2. SAML 認証方式の設定ページに移動します。

3. [リダイレクトモード] パラメータを[属性を保持する]に設定します。[リダイレクトモード] フィールドの場所は以下のように特定されます。

SAML 1.x

[リダイレクトモード] は、設定のメインページの[方式のセットアップ] セクションにあります。

SAML 2.0

[SAML 2.0 設定] - [SSO] を選択します。[リダイレクトモード] は、ページの[SSO] セクションにあります。

WS-フェデレーション

[WS-Federation の設定] - [SAML プロファイル] をクリックします。[リダイレクトモード] は、ページの[SSO] セクションにあります。

4. [サブミット] をクリックして、変更を保存します。

以上で、属性データを渡すようにリダイレクトモードが設定されました。

ユーザを検証するための認可ルールの作成

保護されたターゲット リソースが含まれるレルムの場合、セッションストアから SAML 属性を取得するルールを作成します。

ルールは、認可イベント (onAccessAccept) に基づきます。ユーザは、FWS アプリケーションによってすでに認証されています。Web エージェントは、ユーザを再認証して HTTP ヘッダを渡すことはできません。属性の取得は、認可の段階で行われます。

レルムの onAccessAccept ルールを作成する方法

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [レルム] と順に移動します。
3. ターゲット リソースが含まれるレルムを選択します。
4. [ルール] セクションの[作成] をクリックします。
[ルールの作成] ページが表示されます。
5. 名前、および必要に応じて説明を入力します。
6. [リソース] フィールドにアスタリスク (*) を入力します。

7. [アクション] セッションで [許可イベント] および [OnAccessAccept] を選択します。
8. [許可/拒否と有効/無効] セクションで [有効] を選択します。
9. [OK] ボタンをクリックしてルールを保存します。

以上で、保護されたリソースのあるレルムに対して認可ルールが定義されました。

属性を HTTP ヘッダとして送信するレスポンスの設定

SAML 属性を HTTP ヘッダとして Web エージェントに送信するレスポンスを設定します。Web エージェントは、レスポンスを処理して、ヘッダ変数がクライアント アプリケーションに利用可能になるようにします。

次の手順に従ってください:

1. 管理 UI にログオンします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
3. ターゲット リソースのドメインを選択し、[変更] をクリックします。
4. [レスポンス] タブを選択します。
5. [作成] をクリックします。
[レスポンス] ダイアログが表示されます。
6. 名前を入力します。
7. エージェントのタイプが **SiteMinder Web** エージェントであることを確認します。
8. [レスポンス属性の作成] をクリックします。
[レスポンス属性] ダイアログが表示されます。
9. [属性] フィールドで [WebAgent-HTTP-Header-Variable] を選択します。
10. [属性の種類] の [アクティブ レスポンス] を選択します。

11. フィールドには以下のように入力します。

変数名

ヘッダ変数の名前を指定します。この名前が割り当てられます。

ライブラリ名

`smfedattrresponse`

この値がこのフィールドに入力される必要があります。

関数名

`getAttributeValue`

この値がこのフィールドに入力される必要があります。

パラメータ

アサーションに表示される属性名を指定します。

アサーション内の属性は、フェデレーションパートナーとの合意に基づいて決定してください。

12. [OK] をクリックして属性を保存します。
13. HTTP ヘッダ変数の各属性について手順を繰り返します。1つのレスポンスに対して複数の属性を設定できます。
[レスポンス] タブに戻ります。作成して属性は、[属性リスト] セクションに一覧表示されます。
14. [OK] をクリックしてレスポンスを保存します。
[レスポンス] タブに戻ります。
15. [サブミット] をクリックしてドメインを保存します。

レスポンスは、HTTP ヘッダになるように属性を Web エージェント上に送信します。

属性を HTTP ヘッダとして実装するためのポリシーの作成

SAML 属性の使用を HTTP ヘッダとして実装するには、認可イベントルールおよびアクティブレスポンスを 1 つのポリシーにまとめます。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。

3. ターゲット リソースが含まれるドメインを選択し、[変更] をクリックします。
4. [ポリシー] タブを選択し、[ポリシー] セクションの [作成] をクリックします。
[ポリシーの作成] ダイアログが表示されます。
5. [名前] フィールドにわかりやすい名前を入力します。
6. 保護されたリソースへのアクセスを許可するユーザを [ユーザ] タブで選択します。
7. 先に [ルール] タブで作成した認可ルールを追加します。
8. 認可ルールを選択し、[レスポンスの追加] をクリックします。
[使用可能なレスポンス] ダイアログ ボックスが表示されます。
9. 先に作成したアクティブ レスポンスを選択し、[OK] をクリックします。
[ルール] タブに戻ります。レスポンスが認証ルールと共に表示されます。
10. [サブミット] をクリックしてポリシーを保存します。

SAML 属性を HTTP ヘッダとして使用できるようにするためのポリシーが作成されました。

SAML 2.0 認証が失敗した場合のリダイレクト URL の指定

サービス プロバイダがシングル サインオン トランザクション中にユーザを認証できない場合、そのユーザを後続の処理用にカスタマイズされた URL にリダイレクトできます。

認証が失敗した場合の処理として複数のリダイレクト URL をオプションで設定できます。アサーションが無効な場合、リダイレクト URL を使用することで、ユーザのリダイレクトをより詳細に制御できます。たとえば、ユーザディレクトリでユーザを検索できない場合は、ユーザが見つからない場合のリダイレクト URL を指定します。この URL はユーザを登録ページにリダイレクトできます。

以下の URL を設定できます。

- ステータス リダイレクト URL
- HTTP エラー リダイレクト URL

注: リダイレクト URL の設定は必須ではありません。

一部のリダイレクト URL は、特定のステータス条件に対応します。これらの条件としては、ユーザが見つからない、シングル サインオン メッセージが無効、ユーザ認証情報が受理されない、などがあります。他のリダイレクト URL は、HTTP 500、400、405、403 のエラー状態を処理します。いずれかの状態が発生すると、リダイレクト URL は、後続アクション用のアプリケーションまたはカスタマイズされたエラー ページにユーザをリダイレクトできます。

これらのカスタマイズされた URL へのリダイレクトが実行されるのは、アイデンティティ プロバイダに関する十分な情報がサービス プロバイダに提供されている場合のみです。たとえば、リクエスト中、証明書情報の取得に問題がある場合、ユーザは指定されたサーバエラー URL にリダイレクトされます。一方、リクエストに無効な IdP ID が含まれる場合、リダイレクトは発生せずに、HTTP エラー コード 400 がブラウザに返されます。

オプションのリダイレクト URL を設定する方法

1. 変更する SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [詳細] を選択します。

3. [ステータス リダイレクト URL とモード] セクションで、1 つまたは複数のフィールドに URL に入力します。

フィールドの説明については、[ヘルプ] をクリックしてください。

フェデレーション Web サービスでは、設定されたリダイレクト URL の 1 つに認証理由をマッピングすることにより、エラーを処理します。ユーザをそのリダイレクト URL にリダイレクトして、エラーを報告することができます。

4. 以下のモードのいずれかを選択します。

- 302 データなし
- HTTP POST

5. [OK] をクリックして、変更を保存します。

注: これらのリダイレクト URL をメッセージ コンシューマ プラグインと共に使用して、より詳細なアサーション処理を実行できます。認証が失敗した場合、プラグインは、指定されたリダイレクト URL の 1 つにユーザをリダイレクトできます。

SP においてプロキシ サーバで処理されるリクエスト

SiteMinder では、特定のリクエストを SP で受信すると、メッセージ属性を検証します。SiteMinder は、フェデレーション Web サービス アプリケーションのローカル URL を使用して属性を検証します。検証後、SiteMinder はリクエストを処理します。

たとえば、ログアウト リクエスト メッセージに以下の属性が含まれているとします。

Destination="http://sp.domain.com:8080/affwebservices/public/saml2slo"

この例では、ログアウト メッセージ内の宛先属性と、フェデレーション Web サービス アプリケーションのアドレスは同じです。SiteMinder は、宛先属性が FWS アプリケーションのローカル URL に一致することを検証します。

SiteMinder がプロキシ サーバの後に配置されている場合、ローカル URL と宛先属性 URL は同じにはなりません。宛先属性は、プロキシ サーバの URL です。たとえば、ログアウト メッセージに以下の宛先属性が含まれているとします。

Destination=" http://proxy.domain.com:9090/affwebservices/public/saml2slo"

フェデレーション Web サービスのローカル URL

(http://sp.domain.com:8080/affwebservices/public/saml2slo) が宛先属性に一致しないため、リクエストは拒否されます。

プロキシ設定の指定により、SiteMinder がリクエストのメッセージ属性の検証に使用するローカル URL の判別方法を変更できます。プロキシ設定において、SiteMinder によってローカル URL の `<protocol>://<authority>` の部分がプロキシ サーバ URL と置換されます。この置換の結果、2 つの URL が一致します。

SP においてプロキシ サーバを使用して処理するリクエストの設定

プロキシ設定の指定により、SiteMinder がリクエストのメッセージ属性の検証に使用するローカル URL の判別方法を変更できます。

サービス プロバイダでプロキシ サーバを使用する方法

1. 変更する SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [詳細] を選択します。
3. [プロキシ] セクションで、[サーバ] フィールドに部分的な URL を入力します。形式は、`<protocol>://<authority>` です。

たとえば、プロキシサーバ設定は次ようになります。

http://proxy.domain.com:9090

ネットワークに SPS フェデレーション ゲートウェイが含まれる場合、[サーバ] フィールドには、たとえば、SPS フェデレーション ゲートウェイ ホストおよびポートを指定する必要があります。

http://sps_federation_gateway.domain.com:9090

4. [OK] をクリックして、変更を保存します。

サーバ設定は、SP の以下のサービスの URL に影響します。

- アサーション コンシューマ サービス
- シングル ログアウト サービス

サーバの値は、宛先属性と同様に、SiteMinder が SAML 属性の確認に使用する URL の一部になります。

注: 1つの URL に対して 1つのプロキシサーバを使用している場合は、これらのすべての URL に対してそのプロキシサーバを使用してください。

バックチャネル対応のクライアント証明書認証の有効化(オプション)

この手順は、Artifact バインドによるシングル サインオンのみが対象です。

アサーション コンシューマ サービスは、認証方式から情報を収集して、アイデンティティ プロバイダからアサーションを取得します。方式は、アサーションを取得するためにアイデンティティ プロバイダに提供する認証情報のタイプをアサーション コンシューマ サービスに通知します。アサーションが取得された後、アイデンティティ プロバイダは、セキュリティで保護されたバックチャネルでアサーションをサービス プロバイダに送信します。クライアント証明書認証を使用して、バックチャネルセキュリティのセキュリティを保護することができます。

バックチャネルに対する証明書認証はオプションです。代わりに Basic 認証も使用できます。

バックチャネルにクライアント証明書認証を使用するには、以下のように操作します。

1. [クライアント証明書を証明書データストアに追加](#) (P. 329) します。
2. [バックチャネルに対するクライアント証明書認証を選択](#) (P. 330) します。この方式は、証明書がサービス プロバイダの認証情報として機能することを示します。

ポリシー サーバが FIPS 専用モードで作動している場合にも、FIPS 140 以外で暗号化された証明書を使用してバックチャネルセキュリティのセキュリティを保護することができます。ただし、厳密な FIPS のみのインストールの場合、FIPS 140 互換のアルゴリズムで暗号化された証明書のみを使用してください。

アサーティング側のポリシー サーバの管理者は、アサーション検索サービスを保護するためのポリシーを設定しておく必要があります。このポリシーのレلمは、X.509 クライアント証明書認証方式を使用する必要があります。

詳細情報:

[Artifact サービスを保護する認証方式の設定](#) (P. 160)

証明書データストアへのクライアント証明書の追加

認証機関から秘密キー/証明書ペアを取得する必要があります。管理 UI を使用して、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。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。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/証明書ペアペアをインポートする場合、割り当てるエイリアスは、認証方式設定の [名前] フィールドと同じ値である必要があります。さらに、証明書の件名の CN 属性も [名前] フィールドに一致する必要があります。たとえば、名前が **CompanyA** であるとします。すると、エイリアスは **CompanyA** である必要があります。また、件名の CN の値に **CN=CompanyA, OU=Development, O=CA, L=Islandia, ST=NY, C=US** と記述される必要があります。

重要: 認証方式内の [名前] フィールドは、アイデンティティプロバイダでサービスプロバイダオブジェクトに割り当てられる名前に一致する必要があります。**SiteMinder** がアイデンティティプロバイダである場合、認証方式内の [名前] は、オブジェクトの [一般] 設定の [名前] フィールドに一致する必要があります。

バックチャネルに関するクライアント証明書オプションの設定

バックチャネルに関するクライアント証明書認証を有効にする場合、証明書は認証情報として機能します。

認証情報としてクライアント証明書を示す方法

1. SAML 2.0 認証方式に移動します。
2. [SAML 2.0 設定] - [SSO] を選択します。
[SSO] ページが表示されます。
3. [バインディング] セクションで [HTTP-Artifact] を選択します。
4. [OK] をクリックします。
5. [暗号化 & 署名] ページに移動します。
6. [バックチャネル] セクションの [認証] フィールドで [クライアント証明書] を選択します。
7. [SP 名] の値を入力します。
8. [OK] をクリックします。

SAML 2.0 認証方式でターゲット フェデレーション リソースを保護する方法

SAML 2.0 認証方式を使用する SiteMinder ポリシーを設定することにより、ターゲット フェデレーション リソースを保護します。

SAML 認証方式でフェデレーション リソースを保護するには、以下のよう
に操作します。

1. SAML 認証方式を使用するレルムを作成します。レルムは、ユーザがリクエストするターゲット リソースを集めたものです。

レルムは、以下のいずれかの方法で作成します。

- すでに設定済みの認証方式ごとに[固有なレルムを作成](#) (P. 205)する。
- 対応する SAML 認証方式へのリクエストの送信にカスタム認証方式を使用する[単一のターゲット レルムを設定](#) (P. 206)する。すべてのアイデンティティ プロバイダに対して単一のターゲットを持つレルムを 1 つ設定することにより、SAML 認証のレルム設定が簡単になります。

2. レルムを設定した後、関連のルールおよび必要に応じてレスポンスを作成します。
3. レルム、ルール、およびレスポンスを、ターゲット リソースを保護するポリシーとしてグループ化します。

重要: レルム内の各ターゲット URL も未承認応答 URL で識別されます。未承認応答は、サービス プロバイダからの初期リクエストなしに、アイデンティティ プロバイダからサービス プロバイダに送信されます。未承認応答にはターゲットが含まれます。アイデンティティ プロバイダにおいて、管理者がこのレスポンスをリンクに含め、アイデンティティ プロバイダがサービス プロバイダにユーザをリダイレクトできるようにする必要があります。

各認証方式に対する固有なレルムの設定

SAML または WS フェデレーション認証方式ごとに固有のレルムを設定する手順は、レルムを作成するための標準的な手順に従います。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] の順に移動します。
ドメインを作成するためのページが表示されます。
2. [ドメインの作成] をクリックします。
3. ドメイン名を入力します。
4. ユーザディレクトリをドメインに追加します。このディレクトリは、フェデレーション リソースへのアクセスをリクエストするユーザが含まれるディレクトリです。
5. [レルム] タブを選択し、レルムを作成します。
 - [エージェント] フィールドで、ターゲット リソースが存在する Web サーバを保護する Web エージェントを選択します。
 - [認証方式] フィールドで該当する認証方式を選択します。
6. レルムのルールを作成します。
ルールの一部として、ユーザの認証時の処理を制御するために使用可能なアクション (Get、Post、または Put) を選択します。
7. [ポリシー] タブを選択し、ターゲット フェデレーション リソースを保護するポリシーを設定します。先に作成したレルムをこのポリシーと関連付けます。

以上で、固有のレルムを持つポリシーがフェデレーション リソースを保護します。

すべての認証方式に対する単一のターゲット レルムの設定

認証方式に対するレルムの設定を簡略化するには、アサーションを生成する複数のサイトに対して単一のターゲット レルムを作成します。

このタスクを実行するには、以下のコンポーネントをセットアップします

- 単一のカスタム認証方式

このカスタム方式は、すでに各アサーティング パーティに対して設定済みの対応する SAML または WS フェデレーション認証方式にリクエストを転送します。

- 1 つのターゲット URL を持つ単一のレルム

単一のターゲット レルムに対する認証方式の作成

単一のターゲット レルムに対するカスタム認証方式を定義するには、以下の作業を実行する必要があります。

- 認証方式を設定する方法
- リソース リクエストに適用する認証方式をポリシー サーバに指定するためのカスタム方式のパラメータを定義します。

まず、SAML または WS フェデレーション認証方式が設定済みであることを確認します。設定されていない場合は、これらの方式を設定して、カスタム方式が参照できるようにします。

認証方式を作成する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。
2. 使用しているプロトコルの手順に従って 1 つまたは複数の認証方式を作成します。
3. [OK] をクリックして終了します。

詳細情報:

[SAML 1.x 認証方式 \(P. 179\)](#)

[WS フェデレーション認証方式の概要 \(P. 372\)](#)

[SAML 2.0 認証方式を設定する方法 \(P. 291\)](#)

カスタム認証方式の設定

単一のターゲット レルムが正しく動作するためには、特定のカスタム認証方式に依存します。

単一のターゲットレルムに対するカスタム認証方式を設定する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。
2. フィールドには以下のように入力します。

名前

カスタム認証方式のわかりやすい名前を入力します (SAML Custom Auth Scheme など)。

3. [各方式共通セットアップ] セクションで、以下のフィールドに入力します。

認証方式タイプ

カスタム テンプレート

保護レベル

デフォルトをそのまま使用するか、新しいレベルを設定します。

4. [方式のセットアップ] セクションで、以下のフィールドに入力します。

ライブラリ

smauthsinglefed

秘密キー

このフィールドは空のままにします。

秘密キーの確認入力

このフィールドは空のままにします。

パラメータ

以下のパラメータのいずれかを指定します。

- SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

使用する SAML 認証方式名のリストを指定します。

artifact_producer1 という名前の Artifact と samlpost_producer2 という名前の POST プロファイル方式を設定した場合は、これらの方式を入力します。例：

SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

設定済みの方式をすべて指定します。カスタム認証方式によって、SAML 認証方式がすべて列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_POST;

設定したすべての SAML POST プロファイル方式を指定します。カスタム認証方式によって、POST プロファイル方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_ART;

設定したすべての SAML Artifact 方式を指定します。カスタム認証方式によって、Artifact 方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=WSFED_PASSIVE;

正しいアカウント パートナー ID を持つ WS フェデレーション 認証方式を検索するために、すべての WS フェデレーション 認証方式指定します。

この方式を SiteMinder 管理者用に有効にする

チェック マークはオフのままにします。

5. [サブミット] をクリックします。

カスタム認証方式が完成しました。

単一のターゲット レルムの設定

認証方式を設定し、それらをカスタム方式と関連付けた後、フェデレーション リソースに対する単一のターゲット レルムを設定します。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
2. 単一のターゲット レルムのポリシー ドメインを変更します。
3. [レルム] タブを選択し、[作成] をクリックします。
[レルムの作成] ダイアログが表示されます。
4. 以下の値を入力して、単一のターゲット レルムを作成します。

名前

この単一のターゲット レルムの名前を入力します。

5. [リソース] オプションの以下のフィールドに入力します。

エージェント

ターゲット リソースのある Web サーバを保護する Web エージェントを選択します。

リソース フィルタ

ターゲット リソースの場所を指定します。この場所は、フェデレーション リソースをリクエストするすべてのユーザがリダイレクトされる場所です。

例: /FederatedResources など。

6. [デフォルト リソース保護] セクションで [保護] オプションを選択します。
7. 以前に設定したカスタム認証方式を [認証方式] フィールドで選択します。
たとえば、カスタム方式が **Fed Custom Scheme** という名前の場合は、この方式を選択します。
8. [OK] をクリックします。

単一のターゲット レルムのタスクが完了しました。

単一のターゲット レルムに関するルールの設定

単一のターゲット レルムを設定した後、リソースを保護するためのルールを設定します。

1. 単一のターゲット レルムの [変更] ページに移動します。
2. [ルール] セクションの [作成] をクリックします。
[ルールの作成] ページが表示されます。
3. ルール ページのフィールドに値を入力します。
4. [OK] をクリックします。

単一のターゲット レルム設定に新しいルールが含まれました。

単一のターゲット レルムを使用するポリシーの作成

単一のターゲット レルムを参照するポリシーを作成します。単一のターゲット レルムでは、リクエストを適切な SAML 認証方式に届けるためのカスタム認証方式を使用することに注意してください。

注: この手順は、ドメイン、カスタム認証方式、単一のターゲット レルム、関連ルールがすでに設定済みであることが前提です。

次の手順に従ってください:

1. すでに設定済みのドメインに移動します。
2. [ポリシー] タブを選択し、[作成] をクリックします。
[ポリシーの作成] ページが開きます。
3. [一般] セクションで、ポリシーの名前と説明を入力します。
4. [ユーザ] セクションからユーザをポリシーに追加します。
5. [ルール] タブから単一のターゲット レルムに対して作成したルールを追加します。

残りのタブは省略可能です。

6. [OK] をクリックします。
7. [サブミット] をクリックします。

ポリシー タスクが完了しました。 リクエストがこのポリシーのトリガすると、このポリシーは、ユーザを認証するために、単一のレルムおよび関連付けられた認証方式に依存します。

第 16 章: WS-フェデレーション アカウント パートナーの設定

アサーティング パートナーの前提条件(レガシー)

アサーティング パートナーを設定するには、以下の条件を確認します。

- ポリシー サーバがインストールされています。
- 以下のオプションのいずれかがインストールされています。
 - **Web エージェント**および**Web エージェント オプション パック**。
Web エージェントは、ユーザを認証し、**SiteMinder セッション**を確立します。 **オプション パック**は**フェデレーション Web サービス アプリケーション**を提供します。 ネットワーク内の適切なシステムに必ず**FWS アプリケーション**を展開してください。
 - **SPS フェデレーション ゲートウェイ**では、**Web エージェント**が組み込まれており、また組み込まれた**Tomcat Web**サーバ上に**フェデレーション Web サービス アプリケーション**が配置されています。

詳細については、「**Web エージェント オプション パック ガイド**」を参照してください。

- 秘密キーと証明書は、メッセージの署名および復号を必要とする機能に対してインポートされます。
- 依存パートナーは、フェデレーション ネットワーク内でセットアップされます。

アカウント パートナーを設定する方法

SiteMinder は、アカウント パートナーとして、そのビジネス パートナー（すなわちリソース パートナー）に対するアサーションを生成します。フェデレーション パートナーシップを確立するために、アカウント パートナーでは、各リソース パートナーに関する情報を必要とします。パートナーごとにリソース パートナー オブジェクトを作成します。2つのエンティティがどのように通信してアサーションを渡し、シングルサインオンなどのプロファイルを満たすかを定義してください。

次の手順に従ってください：

1. リソース パートナー オブジェクトを作成します。
2. リソース パートナーをアフィリエイト ドメインに追加します。
3. リソース パートナーの一般的な識別情報を指定します。
4. ユーザストアからユーザを選択します。アカウント パートナーは、選択したユーザに対してアサーションを生成します。
5. アサーションに含める名前 ID を指定します。
6. シングルサインオン プロファイルを設定します。

完全な SSO プロファイルを設定しなくても、リソース パートナー エンティティを保存できます。ただし、SSO を設定しないと、アサーションをリソース パートナーに渡すことができません。

7. [オプションの設定タスク](#) (P. 341) を実行します。

ヒント：

- 設定が動作するためには、特定パラメータ値がアカウント パートナーとリソース パートナーで一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定](#) (P. 451)」(Configuration Settings that Must Use the Same Values) にあります。
- フェデレーション Web サービス サブレットの正しい URL を使用してください。URL の一覧は、「[SiteMinder 設定で使用するフェデレーション Web サービス URL](#) (P. 457)」(Federation Web Services URLs Used in SiteMinder Configuration) にあります。

アカウント パートナーに関するオプション設定タスク

アカウント パートナーの設定には、以下のようなオプションのタスクがあります。

- シングルサインオンの制限の設定
 - リソース パートナーの[時間制限の設定](#) (P. 348)
 - リソース パートナーのアクセスに使用されるアドレスを制限するための[IP アドレス制限の設定](#) (P. 349)
- アサーションに含める属性の設定
- サインアウトの設定
- アサーション ジェネレータ プラグインを使用した[SAML レスポンスのカスタマイズ](#) (P. 171)

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。
- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

アフィリエイトドメインへのリソース パートナーの追加

リソース パートナーをアサーションの利用可能なコンシューマとして識別するには、アカウント パートナーでリソース パートナーをアフィリエイト ドメインに追加します。そして、アサーションが含まれるセキュリティ トークン応答メッセージをアカウント パートナーが発行できるように、リソース パートナーを設定します。

次の手順に従ってください:

1. [フェデレーション] - [レガシー フェデレーション] - [リソース パートナー] の順に移動します。
2. [リソース パートナーの作成] をクリックします。
[リソース パートナーの作成] ページが表示されます。
3. アフィリエイト ドメインを選択した後、[次へ] をクリックします。
[一般] ページが表示されます。
4. ダイアログ ボックスの上部のフィールドに入力します。
フィールドの説明については、[ヘルプ] をクリックしてください。
5. 設定済みのリソース パートナーをアカウント パートナーが認識できるように、[有効] を選択します。

リソース パートナーオブジェクトに関する一般情報の設定

リソース パートナーを指定し、リソース パートナーやアカウント パートナーの ID などの詳細を指定するには、[一般] ページを選択します。また、サービス プロバイダにアクセスするための IP アドレスや時間制限を設定できます。

一般設定項目を設定する方法

1. [一般] 設定に移動します。

2. 必須フィールドに注意しながら、フィールドに値を入力します。

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。

[スキュー時間] フィールドについては、以下の情報に注意してください。

スキュー時間

現在のシステム時刻から引いた秒数を指定します。この計算は、アカウント パートナーと同期していないクロックを持つリソース パートナーを補償するためのものです。

シングルサインオンの場合、スキュー時間の値とシングルサインオン有効期間によって、アサーションが有効な時間が決定されます。スキュー時間についての理解を深めるには、[アサーション有効期間の計算 \(P. 153\)](#)方法について再確認してください。

3. デバッグ目的のみに限り、[署名の処理を無効にする] チェックボックスをオンにすることにより、すべての署名処理（署名および署名の検証の両方）を一時的に無効化できます。

重要: 署名処理はデフォルトで有効化されています。これは、シングルサインオンにおける WS フェデレーション パッシブ リクエスト プロファイルで署名処理が必須であるためです。

SiteMinder セッションのないユーザの認証

リソース パートナーをアフィリエイト ドメインに追加する際、設定が必要なパラメータの 1 つに認証 URL パラメータがあります。

認証 URL は、`redirect.jsp` ファイルを指します。このファイルは、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 ゲートウェイをインストールするアカウント パートナー サイトでインストールされます。SiteMinder ポリシーを使用して `redirect.jsp` ファイルを保護します。ポリシーは、保護されたリソース パートナー ソースをリクエストするが SiteMinder セッションを持たないユーザに対する認証チャレンジをトリガします。

以下のバインドには、**SiteMinder** セッションが必要です。

- **HTTP-POST** バインドを使用するシングル サインオン

ユーザはセッションを持つ必要がありますが、セッションは永続的である必要はありません。アサーションは、ブラウザ経由でリソース パートナーに直接配信されます。アサーションは、セッションストアに保存される必要はありません。

- サインアウト

シングル ログアウトを有効にする場合、永続セッションが必要です。ユーザが最初にリソースをリクエストする際、アカウント パートナーはセッションをセッションストアに保存します。セッション情報は、後でシングル ログアウトが実行されたときに必要になります。

ユーザが認証され、正常に **redirect.jsp** ファイルにアクセスした後、セッションが確立されます。**redirect.jsp** ファイルは、ユーザをリダイレクトしてアカウント パートナー **Web** エージェントまたは **SPS** フェデレーション ゲートウェイに戻します。その後、**SiteMinder** がリクエストを処理します。

認証 URL を保護するための手順は、以下の展開に無関係に同じです。

- **Web** エージェントと同じシステムにインストールされた **Web** エージェント オプションパック
- **Web** サーバ プロキシにインストールされた **Web** エージェントのあるアプリケーション サーバ
- アプリケーション サーバ エージェントのあるアプリケーション サーバ
- アイデンティティ プロバイダでインストールされた **SPS** フェデレーション ゲートウェイ

認証 URL を保護するポリシーの設定

認証 URL を保護する方法

1. 管理 UI にログインします。
2. アサーティング パーティ **Web** サーバに対して定義したレルムにバインドする **Web** エージェントを作成します。**Web** サーバと **FWS** アプリケーションに個別のエージェント名を割り当てるか、両方に同じエージェント名を使用します。
3. コンシューマ リソースへのアクセス試行時に認証情報が要求されるユーザに対してポリシー ドメインを作成します。

4. ポリシー ドメインに含まれるリソースへのアクセス権が必要なユーザを選択します。
5. ポリシー ドメインに対して以下の値を持つレلمを定義します。

エージェント

アサーティング パーティ Web サーバのエージェント

リソース フィルタ

Web エージェント r6.x QMR 6、r12.0 SP2、r12.0 SP3、および SPS フェデレーション ゲートウェイによって次の値が入力されます。

`/siteminderagent/redirectjsp/`

リソース フィルタ `/siteminderagent/redirectjsp/` は、FWS アプリケーションが自動的にセットアップするエイリアスです。エイリアス参照には次の内容が含まれます。

- Web エージェント

`web_agent_home/affwebservices/redirectjsp`

- SPS フェデレーション ゲートウェイ

`sps_home/secure-proxy/Tomcat/webapps/affwebservices/redirectjsp`

永続セッション

SAML Artifact プロファイルの場合のみ、[レلم] ダイアログ ボックスの [セッション] セクションで [永続] チェック ボックスをオンにします。永続セッションを設定しない場合、ユーザはコンシューマ リソースにアクセスできません。

残りの設定項目は、デフォルトのままにするか、必要に応じ変更します。

6. [OK] をクリックしてレلمを保存します。
7. レلمのルールを作成します。レلمに対応するすべてのリソースを保護するには、[レلم] フィールドでデフォルト値のアスタリスク (*) をそのまま使用します。
8. 前の手順で作成したルールが含まれるアサーティング パーティ Web サーバに対するポリシーを作成します。
9. 「[アサーションが生成される対象のユーザの選択](#) (P. 148)」のタスクを実行します。

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、スキュー時間と有効期間の値によって、アサーションの合計有効時間を **SiteMinder** が計算する方法が決定されます。**SiteMinder** 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。

注: この説明におけるアサーティング パーティは、**SAML 1.x** プロデューサ、**SAML 2.0** アイデンティティ プロバイダ、または **WS フェデレーション** アカウント パートナーです。依存パーティは、**SAML 1.x** コンシューマ、**SAML 2.0** サービス プロバイダ、または **WS フェデレーション** リソース パートナーです。

アサーション ドキュメントで、**NotBefore** と **NotOnOrAfter** の値は、有効期間の開始と終了を表します。

アサーティング パーティで、**SiteMinder** はアサーション有効期間を設定します。有効期間は、アサーションが生成されたときのシステム時間です。**SiteMinder** は、この時間を使用してアサーションに **IssueInstant** 値を設定した後、**IssueInstant** 値からスキュー時間の値を引き算します。その結果の時間が **NotBefore** 値になります。

NotBefore=IssueInstant - スキュー時間

有効期間の終了を決定するために、**SiteMinder** 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、**SiteMinder** は、アサーティング パーティで受信したアサーションが有効かどうか判別するために実行する計算と同じ計算を実行します。

パートナーシップの両側での SiteMinder によるアサーション有効期間の計算

SiteMinder がパートナーシップの両側にある場合、アサーション有効期間は、スキュー時間の 2 倍と有効期間の総和になります。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) + 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 + 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: レガシー フェデレーション の場合、有効期間はアサーティング パーティのみで設定されます。

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
- 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、アサーションから **NotBefore** と **NotOnOrAfter** の値を使用し、それらの値にスキュー時間を適用します。この数式は、依存パーティが新しい **NotBefore** と **NotOnOrAfter** の値を計算する方法です。

- スキュー時間 = 180 秒 (3 分)
- NotBefore = 4:56PM
- NotOnOrAfter=5:05PM

アサーション有効期間

この例の値を使用したアサーションの合計有効期間の計算は、次のとおりです。

$$120 \text{ 秒 (2x60)} + 60 \text{ 秒} + 360 \text{ 秒 (2x180)} = 540 \text{ 秒 (9 分)}$$

リソース パートナー利用可能時間の制限の設定(オプション)

リソース パートナー リソースの利用可能な時間に制限を指定できます。時間制限を指定すると、リソース パートナーのリソースへのアクセスは指定された期間中のみ可能となります。ユーザが指定の期間外にリソースへのアクセスを試みても、アカウント パートナーは **SAML** アサーションを生成しません。

注: 時間制限は、ポリシー サーバがインストールされたサーバのシステムクロックに基づきます。

時間制限を指定する方法

1. まず、[一般] 設定に移動します。
ページの [制限] セクションで、[時間を設定] セクションを選択します。
[時間制限] ページが表示されます。
2. スケジュールを設定します。このスケジュール グリッドは、ルール オブジェクトの [時間制限] グリッドと同一です。詳細については、「[ポリシー サーバ設定ガイド](#)」を参照してください。
3. [OK] をクリックします。

時間制限のスケジュールが設定されました。

リソース パートナーの IP アドレス制限の設定(オプション)

リソース パートナーにアクセスするための Web サーバの IP アドレス、範囲アドレス、またはサブネット マスクを指定できます。リソース パートナーの IP アドレスが指定される場合、リソース パートナーでは、適切な IP アドレスからのユーザのみを受け付けます。

IP アドレスを指定する方法

1. まず、[一般] 設定に移動します。
ページの [制限] セクションで、[IP アドレス] 領域の [追加] をクリックします。
[IP 制限] ページが表示されます。
2. 追加する IP アドレス タイプのオプションを選択し、そのアドレス タイプに関連するフィールドに入力します。
注: IP アドレスが不明でも、アドレスのドメイン名が既知の場合は、[DNS の検索] ボタンをクリックしてください。このボタンにより、[DNS の検索] ページが表示されます。[ホスト名] フィールドに完全修飾ホスト名を入力し、[OK] をクリックします。
 - [単一ホスト] -- ブラウザをホストする単一の IP アドレスを指定します。単一の IP アドレスを指定する場合、ユーザは指定された IP アドレスからのみリソース パートナーにアクセスできます。
 - [ホスト名] -- Web サーバをそのホスト名で指定します。ホスト名を指定する場合、リソース パートナーは指定されたホストからのユーザのみにアクセス可能です。

- [サブネットマスク] -- Web サーバのサブネットマスクを指定します。サブネットマスクを指定する場合、リソース パートナーは指定されたサブネットマスクからのユーザのみにアクセス可能です。このボタンを選択する場合、[アドレスとサブネットマスクの追加] ダイアログ ボックスが表示されます。左矢印ボタンと右矢印ボタンを使用するか、スライダーバーをクリックしてからドラッグして、サブネットマスクを選択してください。
 - [範囲] -- IP アドレス範囲を指定します。IP アドレスの範囲を指定する場合、リソース パートナーは、アドレスの範囲内の IP アドレスからのユーザのみを許可します。範囲の開始アドレス（[FROM]）および終了アドレス（[TO]）を入力します。
3. [OK] をクリックして設定を保存します。

アサーションが生成される対象のユーザの選択

アサーティング パーティでの設定の一部として、アサーション ジェネレータが SAML アサーションを生成する対象のユーザおよびグループのリストを含めます。アサーティング パーティは、SAML 1.x プロデューサ、SAML 2.0 アイデンティティ プロバイダ、または WS フェデレーション アカウント パートナーのいずれかです。

アフィリエイト ドメインにあるディレクトリからのユーザおよびグループのみを追加できます。

フェデレーショントランザクションのユーザおよびグループを指定する方法

1. 設定するパートナーの [ユーザ] 設定に移動します。

[ユーザディレクトリ] ページには、ポリシー ドメインのユーザディレクトリごとにエントリが表示されます。

2. ユーザディレクトリからポリシーにユーザまたはグループを追加します。

各ユーザディレクトリ テーブルで、[メンバーの追加] - [エントリの追加] - [すべて追加] を選択できます。選択したメソッドに応じて、ダイアログ ボックスが表示され、ユーザの追加が可能になります。

- [メンバーの追加] を選択すると、[ユーザ/グループ] ウィンドウが表示されます。個々のユーザは、自動的に表示されません。検索ユーティリティを使用して、ディレクトリの 1 つに含まれる特定のユーザを見つけることができます。
- [エントリの追加] を選択する場合は、[ユーザディレクトリ検索式編集] ダイアログ ボックスで[手動設定](#) (P. 150)によってユーザを選択してください。

ユーザやグループの編集または削除を行うには、それぞれ右向き矢印 (>) またはマイナス記号 (-) をクリックします。

3. 好きな方法で個別のユーザ、ユーザ グループ、または両方を選択して [OK] をクリックします。

[ユーザディレクトリ] ページが再度開き、新しいユーザがユーザディレクトリ テーブルに表示されます。

詳細情報:

[リソースへのアクセスからのユーザまたはグループの除外](#) (P. 149)

[ネストされたグループによるリソースへのアクセスの許可](#) (P. 149)

[手動設定によるユーザの追加](#) (P. 150)

リソースへのアクセスからのユーザまたはグループの除外

ユーザやユーザのグループをアサーションの取得から除外できます。

次の手順に従ってください:

1. [ユーザ] 設定に移動します。
2. 特定のユーザ ディレクトリのリストからユーザまたはグループを選択します。
3. 選択したユーザまたはグループを除外するには、[除外] をクリックします。

選択内容が 管理 UI に反映されます。

4. [OK] をクリックして、変更を保存します。

ネストされたグループによるリソースへのアクセスの許可

LDAP ユーザ ディレクトリには、サブグループを持つグループが含まれることがあります。複雑なディレクトリの場合、大量のユーザ情報を組織化する方法の 1 つとして、他のグループの階層内にグループをネストすることがあります。

ネストされたグループ内のユーザの検索を有効にする場合、リクエストされたユーザ レコードがすべてのネストされたグループで検索されます。ネストされたグループを有効にしない場合、指定したグループのみをポリシー サーバが検索します。

ネストされたグループでの検索を有効にする方法

1. [ユーザ] 設定に移動します。
関連するアフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、それぞれのユーザ ディレクトリが独自のセクションに表示されます。
2. ネストされたグループ内の検索を有効にするには、[ネストされたグループの許可] チェック ボックスをオンにします。

手動設定によるユーザの追加

アサーション生成のユーザを指定する場合の選択肢の 1 つとして、手動設定によりユーザを識別します。

次の手順に従ってください:

1. 設定するパートナーの [ユーザ] 設定に移動します。
アフィリエイト ドメインに複数のユーザ ディレクトリが含まれる場合、すべてのディレクトリが [ユーザ ディレクトリ] ページに表示されます。
2. [エントリの追加] をクリックします。
[ユーザ ディレクトリ 検索式編集] ページが表示されます。
3. 検索オプションを選択した後、その検索オプションのフィールドに入力します。

検索する場所

LDAP ディレクトリの場合、ドロップダウン リストからオプションを選択します。

DN の確認

LDAP 検索はディレクトリ内でこの DN を検索します。

ユーザの検索

LDAP 検索の実行範囲は、ユーザ エントリ内の一致に限定されます。

グループの検索

LDAP 検索の実行範囲は、グループ エントリ内の一致に限定されます。

組織の検索

LDAP 検索の実行範囲は、組織 エントリ内の一致に限定されます。

エントリの検索

LDAP 検索の実行範囲は、ユーザ エントリ、グループ エントリ、および組織 エントリ内の一致に制限されています。

- Microsoft SQL Server、Oracle、および Windows NT ディレクトリの場合は、[手動設定] フィールドにユーザ名を入力できます。
- Microsoft SQL Server または Oracle では、SQL クエリを代わりに入力することもできます。例：

```
SELECT NAME FROM EMPLOYEE WHERE JOB ='MGR';
```

ポリシー サーバは、ユーザ ディレクトリの [認証情報と接続] タブの [ユーザ名] フィールドに指定されたデータベース ユーザとしてクエリを実行します。[手動設定] フィールドに入力する SQL 文を作成する場合には、ユーザ ディレクトリのデータベース スキーマに関する知識が必要です。たとえば、SmSampleUsers スキーマを使用している場合に特定のユーザを追加するには、SmUser テーブルからユーザ エントリを選択します。

- LDAP ディレクトリの場合は、[手動設定] フィールドに「all」と入力して、すべてのディレクトリ エントリを追加します。

4. [OK] をクリックして、変更を保存します。

WS フェデレーションアサーションの名前 ID の設定

名前 ID は、独自の方法でアサーション内のユーザを指定します。管理 UI で設定する値は、リソース パートナーに送信されるアサーションに含まれます。

名前 ID 形式により、その ID に使用されるコンテンツのタイプが確定されます。たとえば、形式が User DN の場合、コンテンツは UID です。

名前 ID を指定する方法

1. 設定するリソース パートナーオブジェクトに移動します。
2. [名前 ID] 設定を選択します。
3. [名前 ID 形式] を選択します。

各形式の説明については、「*Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*」仕様を参照してください。

4. [名前 ID タイプ] に以下のオプションから選択します。

- スタティック値
- ユーザ属性
- DN 属性（ネストされたグループの有無）

[名前 ID タイプ] に応じて、[名前 ID フィールド] の内容が変化します。

5. 選択した [名前 ID タイプ] の各種名前 ID フィールドに入力します。

WS フェデレーションの場合のシングル サインオンの設定

アサーションは、リソース パートナーでのシングル サインオンを推進するために必要な ID 情報を提供します。アカウント パートナーは、セッションが確立されたユーザに対して SAML 1.1 アサーションを生成します。アカウント パートナーは、アサーションを WS フェデレーションの RequestSecurityTokenResponse メッセージに配置した後、トークンをリソース パートナーに配信します。リソース パートナーでは、セキュリティ トークンを消費し、WS フェデレーション セキュリティ トークンの内容に基づくセッションを確立します。

シングル サインオン設定の一環として、アカウント パートナーがアサーションをリソース パートナーに配信する方法を決定します。

アカウント パートナーでのシングル サインオンを設定する方法

1. リソース パートナーオブジェクトの [SAML プロファイル] 設定に移動します。
2. ページの [SSO] セクションのフィールドに入力します。
フィールドの説明については、[ヘルプ] をクリックしてください。
3. [サブミット] をクリックして、変更を保存します。

アカウント パートナーでのシングル サインオンの開始

ユーザは、リソース パートナーに移動する前に、アカウント パートナーをアクセスすることがあります。ユーザが最初にアカウント パートナーをアクセスする場合、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、アカウント パートナーのシングル サインオン サービスを指します。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。

シングル サインオン サービスへのリンクの構文は次のとおりです。

```
https://ap_server:port/affwebservices/public/wsfedsso?wa=wsignin1.0&wtrealm=RP_ID
```

ap_server:port

アカウント 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

リソース パートナー アイデンティティ、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。

リソース パートナーでのシングル サインオンの開始

ユーザは、リソース パートナーでシングル サインオンを開始する場合、通常はアカウント パートナーの一覧から選択します。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。

サイト選択ページ上のリンクは、アカウント パートナーのシングル サインオン サービスを指します。リンクが選択された後、リソース パートナーは、アサーションを取得するためにユーザをアカウント パートナーにリダイレクトします。

SAML アサーション レスポンスのカスタマイズ(オプション)

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を使用して、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次のとおりです。

1. SiteMinder 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（SDK に含まれています）を実装します。
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. 管理 UI でアサーション ジェネレータ プラグイン パラメータを有効にします。

アサーション ジェネレータ プラグインに関する追加の情報は、以下を参照してください。

- 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 クラスの新規コンストラクタが「*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。
- 認証および認可の API に関する概要および概念情報については、「Java 用 SiteMinder プログラミング ガイド」を参照してください。

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。

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。

- 要件を満たすようにインターフェースにメソッドを実装します。

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。既存の実装を上書きできます。例については、以下のサンプルクラスを参照してください。

SAML 1.x / WS フェデレーション

AssertionSample.java

SAML 2.0

SAML2AssertionSample.java

サンプルクラスは、`/sdk/samples/assertiongeneratorplugin` ディレクトリにあります。

実装によって `customizeAssertion` メソッド内に渡されるパラメータ文字列のコンテンツは、カスタム オブジェクトから提供されます。

アサーション ジェネレータ プラグインの展開

AssertionGeneratorPlugin 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、SiteMinder 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

- アサーション プラグイン Java ファイルをコンパイルします。

コンパイルには以下の `.jar` ファイルが必要となります。これらのファイルは、ポリシー サーバと共にインストールされています。

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

- 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。この変更により、変更されたクラスパスによるプラグインのロードが可能になります。
`installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

注: `xercesImpl.jar`、`xalan.jar`、`SMJavaApi.jar` のクラスパスを変更しないでください。

- プラグインを有効化します。

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、管理 UI 内で設定することにより、このプラグインを有効にします。UI パラメータにより、SiteMinder 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. 172)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. 管理 UI にログオンします。
2. [フェデレーション] - [レガシー フェデレーション] - [リソース パートナー] をクリックします。
3. 既存のリソース パートナー エントリを選択するか、新規に作成します。
4. [一般] 設定に移動します。
5. [詳細] セクションで、以下のフィールドに入力します。

Java クラス名

既存のプラグインの Java クラス名を指定します

プラグイン クラスは、アサーションの解析および変更を行った後、最終的な処理のために結果をアサーション ジェネレータに返すことができます。

各パートナーには 1 つのプラグインのみが対応します。たとえば、次のようになります。

`com.mycompany.assertiongenerator.AssertionSample`

パラメータ

(オプション) [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。

注: 管理 UI によってアサーション プラグインを有効にする代わりに、ポリシー管理 API (C または Perl) を使用して、プラグインを統合することができます。詳細については、「C 言語用 SiteMinder プログラミング ガイド」または「Java 言語用 SiteMinder プログラミング ガイド」を参照してください。

6. ポリシー サーバを再起動します。

ポリシー サーバを再起動することにより、再コンパイル後に最新のアサーション プラグインのバージョンが選択されることが保証されます。

Web アプリケーションの属性によるアサーションのカスタマイズ

アサーション ジェネレータ プラグインを使用して、**Web** アプリケーション属性をアサーションに追加できます。これは、アサーションのカスタマイズ方法の 1 つです。

Web アプリケーション属性をアサーション含める方法

1. アサーション プラグイン Java ファイルをコンパイルします。

コンパイルには以下の .jar ファイルが必要となります。これらのファイルは、ポリシー サーバと共にインストールされています。

- `policy_server_home/bin/jars/SmJavaApi.jar`
- `policy_server_home/bin/thirdparty/xercesImpl.jar`
- `policy_server_home/bin/endorsed/xalan.jar`

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。この変更により、変更されたクラスパスによるプラグインのロードが可能になります。

`installation_home¥siteminder¥config` ディレクトリの `JVMOptions.txt` ファイルの場所を特定します。

注: `xercesImpl.jar`、`xalan.jar`、`SMJavaApi.jar` のクラスパスを変更しないでください。

3. サンプル プラグインを設定します。

SMJavaAPI の `APIContext` クラスには、`getAttrMap()` という新しいメソッドが含まれています。このメソッドは、アサーションに含まれた Web アプリケーションの属性が含まれるマップ オブジェクトを返します。SiteMinder SDK には、このマップ オブジェクトの使用方法を示すサンプル アサーション ジェネレータ プラグインが 2 つ含まれています。

- `SAML2AppAttrPlugin.java` (SAML 2.0)
- `WSFedAppAttrPlugin.java` (WS フェデレーション)

これらのサンプルは、`sdk/samples/assertiongeneratorplugin` ディレクトリにあります。これらを使用して、アサーション ジェネレータが Web アプリケーションの属性をアサーションに追加することができます。

4. 管理 UI にログインします。
5. [フェデレーション] - [レガシー フェデレーション] - [SAML サービス プロバイダ] または [リソース パートナー] を選択します。
6. 既存のエントリを選択するか、新規に作成します。
7. [一般] 設定に移動します。
8. [アサーション ジェネレータ 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。

Java クラス名

プラグインの Java クラスを指定します。たとえば、SiteMinder SDK に付属のサンプル クラスは次のとおりです。

- `com.ca.assertiongenerator.SAML2AppAttrPlugin`
(SAML 2.0)
- `com.ca.assertiongenerator.WSFedAppAttrPlugin`
(WS フェデレーション)

パラメータ

[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の文字列を指定します。これらのパラメータがアサーションに含める属性です。

注: 管理 UI によって設定する代わりに、ポリシー管理 API (C または Perl) を使用して、プラグインを統合することができます。手順については、「C 言語用 SiteMinder プログラミング ガイド」または「Java 言語用 SiteMinder プログラミング ガイド」を参照してください。

9. ポリシー サーバを再起動します。

ポリシー サーバを再起動することにより、再コンパイル後に最新のアプリケーション プラグインのバージョンが選択されることが確定されます。

WS-Federation のサインアウトの設定

サインアウトは、ログアウトを開始したブラウザに関するすべてのセッションからログアウトするユーザのプロセスです。サインアウトにより、必ずしもユーザのすべてのセッションが終了するとは限りません。たとえば、ユーザが 2 つのブラウザを開いている場合、そのユーザは 2 つの独立したセッションを確立できます。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。

ユーザは、アカウント パートナーまたはリソース パートナーからサインアウト 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。

注: サインアウトに関して **WS-フェデレーション** パッシブ リクエストのみがサポートされます。

[サインアウト] セクションを設定することにより、リソース パートナーがサインアウトをサポートする方法をアカウント パートナーに指定します。

サインアウトを有効にする場合、以下の操作も必要となります。

- ポリシー サーバ管理コンソールを使用して、アカウント パートナーのセッション 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。

サインアウトを設定する方法

1. 設定するリソース パートナーの [SAML プロファイル] ページに移動します。
2. [サインアウト] セクションで、[サインアウトの有効化] を選択します。
3. 以下の URL フィールドに値を入力します。
 - サインアウト クリーンアップ URL
 - サインアウト確認 URLこれらのフィールドのそれぞれに、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。
フィールドの説明については、[ヘルプ] をクリックしてください。
4. [OK] をクリックします。

WS フェデレーション アサーションの属性の設定(オプション)

属性は、リソース パートナーのリソースへのアクセスをリクエストしているユーザに関する情報を提供することができます。属性ステートメントにより、SAML アサーション内でユーザ属性、DN 属性、または静的なデータがアカウント パートナーからリソース パートナーに渡されます。すべての設定済みの属性は、1 つの <AttributeStatement> 要素内のアサーションまたはアサーション内の <EncryptedAttribute> 要素に含まれています。

注: アサーション内には属性ステートメントは不要です。

サーブレット、Web アプリケーション、その他のカスタム アプリケーションは、属性を使用して、カスタマイズされたコンテンツを表示したり、他のカスタム機能を実現したりします。Web アプリケーションでは、属性を使用してリソース パートナーでのユーザ アクティビティを制限することにより、きめ細かいアクセス制御を実装できます。たとえば、Authorized Amount という名前の属性変数を最高額設定して送信するなどです。この額は、ユーザがリソース パートナーで費やすことができる限度額です。

属性の形式は、名前/値のペアになっています。リソース パートナーは、アサーションを受信すると、アプリケーションが属性値を利用できるようにします。

属性は、HTTP ヘッダまたは HTTP Cookie として利用できるようになります。

HTTP ヘッダおよび HTTP Cookie にはサイズ制限があり、アサーション属性でそれらを超えるはできません。サイズ制限は以下のとおりです。

- **HTTP ヘッダ** : SiteMinder は、Web サーバのヘッダに関するサイズ制限までの属性をヘッダで送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダ サイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。
- **HTTP Cookie** : SiteMinder は、Cookie のサイズ制限までの Cookie を送信できます。それぞれのアサーション属性は、それ自身の Cookie として送信されます。Cookie のサイズ制限はブラウザ固有です。また、その制限は、属性ごとではなく、アプリケーションに渡されるすべての属性に適用されます。Cookie のサイズ制限を調べるには、使用する Web ブラウザのドキュメントを参照してください。

WS-Federation のアサーション属性の設定

アサーション属性を設定する方法

1. 設定するリソース パートナー オブジェクトの [属性] ページに移動します。
2. [属性] セクションの [追加] をクリックします。
[属性の追加] ダイアログ ボックスが表示されます。

3. [属性] ドロップダウンから、名前形式識別子を選択します。アサーションの <Attribute> 要素内の <NameFormat> 属性によって識別子が指定されます。リソース パートナーが名前を解釈できるように、この値によって属性名が分類されます。

オプションを以下に示します。

- EmailAddress
- UPN
- CommonName
- Group
- NameValue

これらのオプションの詳細については、WS フェデレーションの仕様を参照してください。

4. [属性のセットアップ] セクションで、以下のオプションのいずれかを選択します。

- スタティック
- ユーザ属性
- DN 属性

以下のオプションの選択により、[属性フィールド] セクションで利用可能なフィールドが決定されます。

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. 任意です。属性は、ネストされたグループを持つ LDAP ユーザディレクトリから取得できます。ネストされたグループからポリシー サーバが DN 属性を取得するには、[属性の種類] セクション内の [ネストされたグループの許可] チェック ボックスをオンにします。
6. [属性の種類] の必要なフィールドに入力し、変更を保存します。

アサーション属性の最大文字数を指定します

ユーザ アサーション属性の最大長は設定可能です。アサーション属性の最大長を変更するには、`EntitlementGenerator.properties` ファイルの設定を変更します。

ファイル内のプロパティ名は、設定のプロトコルに固有です。

次の手順に従ってください:

1. ポリシー サーバがインストールされているシステムで、`policy_server_home¥config¥properties¥EntitlementGenerator.properties` に移動します。
2. テキスト エディタでファイルを開きます。
3. 現在の環境で使用中のプロトコルでのユーザ属性の最大長を調節します。各プロトコルの設定を以下に示します。

WS-フェデレーション

プロパティ名:

`com.netegrity.assertiongenerator.wsfed.MaxUserAttributeLength`

プロパティ タイプ: 正の整数値

デフォルト値: 1024

説明: WS-FED アサーション属性の最大属性長を示します。

SAML 1.x

プロパティ名:

`com.netegrity.assertiongenerator.saml1.MaxUserAttributeLength`

プロパティ タイプ: 正の整数値

デフォルト値: 1024

説明: SAML1.1 アサーション属性の最大属性長を示します。

SAML 2.0

プロパティ名:

`com.netegrity.assertiongenerator.saml2.MaxUserAttributeLength`

プロパティ タイプ: 正の整数値

デフォルト値: 1024

説明: SAML2.0 アサーション属性用の最大属性長を示します。

4. これらのパラメータのいずれかを変更した後は、ポリシー サーバを再起動します。

属性の新規作成スクリプトの使用

[属性] ダイアログ ボックスの [詳細] セクションには [スクリプト] フィールドが含まれています。このフィールドには、[属性のセットアップ] セクションの入力に基づいて SiteMinder が生成したスクリプトが表示されます。このフィールドの内容をコピーし、別のレスポンス属性の [スクリプト] フィールドに貼り付けることができます。

注: 別の属性の [スクリプト] フィールドの内容をコピーして貼り付ける場合は、[属性の種類] セクションで適切なオプションを選択してください。

第 17 章: SiteMinder の WS-フェデレーション リソース パートナーとしての設定

このセクションには、以下のトピックが含まれています。

[依存パートナーの前提条件 \(P. 369\)](#)

[リソース パートナーを設定する方法 \(P. 370\)](#)

[WS フェデレーション認証方式の概要 \(P. 372\)](#)

[WS フェデレーション認証方式タイプの選択 \(P. 373\)](#)

[WS フェデレーション認証方式の一般情報の指定 \(P. 374\)](#)

[認証用のユーザレコードの検索 \(P. 374\)](#)

[リソース パートナーでの WS フェデレーション シングル サインオンの設定 \(P. 377\)](#)

[WS フェデレーション サインアウトの実装 \(P. 378\)](#)

[カスタム WS フェデレーション認証方式の作成 \(P. 379\)](#)

[メッセージコンシューマプラグインによるアサーション処理のカスタマイズ \(P. 380\)](#)

[WS フェデレーションの認証試行が失敗した後のユーザのリダイレクト \(P. 385\)](#)

[HTTP ヘッダとしての SAML 属性の供給 \(P. 386\)](#)

[WS-Federation 認証方式でターゲット リソースを保護する方法 \(P. 395\)](#)

依存パートナーの前提条件

SiteMinder が依存パートナーとして機能するには、以下のタスクを完了する必要があります。

- ポリシー サーバをインストールします。

- 以下のコンポーネントのいずれかをインストールします。
 - **Web エージェント**および**Web エージェント オプション パック**。
Web エージェントは、ユーザを認証し、セッションを確立します。
オプション パックはフェデレーション **Web サービス アプリケーション**を提供します。 ネットワーク内の適切なシステムに必ず **FWS アプリケーション**を展開してください。
 - **SPS フェデレーション ゲートウェイ** (**Web エージェント**が組み込まれており、組み込まれた **Tomcat Web** サーバ上にフェデレーション **Web サービス アプリケーション**が配置されています)。
- 詳細については、「**Web エージェント オプション パック ガイド**」を参照してください。
- 検証およびメッセージの暗号化を必要とする機能のために秘密キーと証明書をインポートします。
- フェデレーション ネットワーク内でアサーティング パートナーをセットアップします。

リソース パートナーを設定する方法

WS フェデレーション リソース パートナーを設定するには、以下のタスクを実行する必要があります。

1. **SAML 1.x** 認証方式の前提条件を実行します。
2. 認証方式タイプを選択し、これに名前を割り当てます。
3. **SAML 1.x** 認証方式で認証されるユーザのネームスペースを指定します。
4. このコンシューマがサポートするシングル サインオン プロファイル (**Artifact** または **POST**) を選択します。

フェデレーション パートナーであり、アサーションを生成するアカウント パートナーごとに **SAML** 認証方式を設定します。各方式をレルムにバインドします。レルムには、ユーザがリクエストしているターゲット リソースの **URL** が含まれます。このタスクをアカウント パートナー単位で実行できます。また、単一のカスタム認証方式と単一のレルムを作成することもできます。これらのリソースを **SiteMinder** ポリシーで保護します。

ヒント：

- 設定が動作するためには、特定パラメータ値がアカウント パートナーとリソース パートナーで一致する必要があります。それらのパラメータの一覧は、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」(Configuration Settings that Must Use the Same Values) に記載されています。
- フェデレーション Web サービス サブレットの正しい URL を使用していることを確認してください。これらの URL は、「[SiteMinder 設定で使用されるフェデレーション Web サービス URL \(P. 457\)](#)」(Federation Web Services URLs Used in SiteMinder Configuration) に一覧表示されています。

リソース パートナーに関するオプション設定タスク

リソース パートナーの設定には、以下のようなオプションのタスクがあります。

- メッセージ コンシューマ プラグインを使用したアサーションのカスタマイズ
- 失敗した認証の試行のリダイレクト

レガシー フェデレーション ダイアログ ボックスへの移動

管理 UI では、レガシー フェデレーション設定ダイアログ ボックスに移動する方法が 2 つ用意されています。

以下の 2 つの方法のいずれかを使用して移動できます。

- 新しいレガシー フェデレーション オブジェクトを設定するためのウィザードに従う。

オブジェクトを作成すると、設定ウィザードのあるページが表示されます。設定ウィザードでオブジェクトを作成する手順に従います。

- タブを選択して既存のレガシー フェデレーションオブジェクトを変更する。

既存のオブジェクトを変更すると、一連のタブを持つページが表示されます。これらのタブから設定を変更します。これらのタブは、設定ウィザードの手順と同じです。

WS フェデレーション認証方式の概要

任意の SiteMinder サイトまたは SPS フェデレーション ゲートウェイが <RequestSecurityTokenResponse> メッセージを消費でき、レスポンス内のアサーションを使用してユーザの認証および認可を行えます。 フェデレーション ネットワーク内のサイトにユーザ ストアがある場合、WS フェデレーション認証を使用できます。

WS フェデレーション認証方式では、リソース パートナーがユーザを認証します。 この認証方式を使用する場合、SAML アサーションを消費し、SiteMinder セッションを確立することにより、クロスドメインシングルサインオンを実現できます。 ユーザが識別された後、リソース パートナーサイトでは、特定のリソースに対してユーザに認可を与えることができます。

1 つのサイトが WS フェデレーション リソース パートナーとアカウント パートナーの両方として機能できます。



注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。 SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「**Secure プロキシ サーバ管理ガイド**」 (Secure Proxy Server Administration Guide) を参照してください。

WS フェデレーション認証方式は、リソース パートナー側のポリシー サーバで設定されます。WS フェデレーションセキュリティ トークン コンシューマ サービスは、この認証方式を呼び出します。セキュリティ トークン コンシューマ サービスは、フェデレーション Web サービス アプリケーションのコンポーネントであり、リソース パートナー側の Web エージェントにインストールされます。このサービスは、ポリシー サーバで WS フェデレーション認証方式からの情報を取得します。FWS は、その情報を使用してアサーションから必要な情報を抽出し、ユーザを認証します。

SAML アサーションは、リソース パートナー サイトでポリシー サーバにログインするためのユーザ認証情報になります。ユーザの認証および認可が行われ、認可が成功すると、このユーザはターゲット リソースにリダイレクトされます。

WS フェデレーション認証方式タイプの選択

WS フェデレーション認証方式は、リソース パートナーのアサーションを生成するアカウント パートナーに関する情報を提供します。この認証方式により、リソース パートナーが認証処理をサポートする方法が指定されます。

認証方式を設定した後、保護するリソースが含まれるレルムと認証方式を関連付けます。

WS フェデレーション認証方式を設定する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ダイアログ ボックスが表示されます。
2. [認証方式タイプ] ドロップダウン リストから、[WS-Federation テンプレート] を選択します。
[認証方式] ダイアログ ボックスの内容が方式に合わせて変化します。

注:それぞれの要件および制限など、設定とコントロールの説明を参照するには、[ヘルプ] をクリックします。

認証方式テンプレートを選択した後、認証方式の詳細を設定できます。

[WS-Federation の設定] をクリックすることにより、設定ダイアログ ボックスの残りの部分にアクセスできます。

詳細情報:

[WS-Federation 認証方式でターゲット リソースを保護する方法 \(P. 395\)](#)

WS フェデレーション認証方式の一般情報の指定

WS フェデレーション認証方式の [一般] 設定にリソース パートナーおよびアカウント パートナーを指定します。

次の手順に従ってください:

1. 認証方式のメイン ページから、[WS-Federation の設定] をクリックします。

既存の方式を変更する場合は、[変更] をクリックしてから [WS-Federation の設定] をクリックします。

その方式に関する詳細な設定が表示されます。

2. [一般] 設定の必須フィールドに入力します。
3. シングルサインオンの [署名の処理を無効にする] オプションが適切に設定されることを確認します。

重要: デバッグ目的のみに限り、[署名の処理を無効にする] オプションを有効にすることにより、すべての署名処理（署名および署名の検証の両方）を一時的に無効化できます。

一般設定が終了しました。

認証用のユーザ レコードの検索

認証スキームを設定する際に、認証スキームがローカル ユーザ ストアでユーザを検索する方法を定義します。正しいユーザが検索された後、システムはこのユーザのセッションを生成します。ユーザ ストアでのユーザの検索は、不明瞭解消のプロセスです。SiteMinder がユーザの不明瞭解消を実行する方法は、認証スキームの設定によって異なります。

不明瞭解消を正しく行うために、認証スキームでは、最初にアサーションから LoginID を判別します。デフォルトでは、LoginID はアサーション内の名前 ID の値から抽出されます。また、Xpath クエリを指定することによっても LoginID を取得できます。

認証方式が LoginID を判別した後、SiteMinder は認証方式に対して検索仕様が設定されているかどうかを確認します。認証方式に対して検索仕様が定義されていない場合、LoginID がポリシー サーバに渡されます。ポリシー サーバでは、ユーザストア検索仕様と共に LoginID を使用してユーザを検索します。たとえば、LoginID の値が Username で、LDAP 検索仕様が uid 属性に設定されているとします。すると、ポリシー サーバは、UID の値 (Username=uid) を使用してユーザを検索します。

認証方式に対して検索仕様が設定されている場合、LoginID はポリシー サーバに渡されません。その代わりに、検索仕様を使用してユーザが検索されます。

不明瞭解消プロセスには、以下の 2 手順が必要とされます。

1. デフォルト動作または Xpath クエリの使用によって LoginID を取得する。
2. デフォルト動作または検索仕様によってユーザストアでユーザを検索する。

注: Xpath の使用および検索仕様はオプションです。

WS フェデレーション ユーザの LoginID の取得

LoginID は、以下の 2 つの方法で検索できます。

- LoginID がアサーションの NameID から抽出されるというデフォルト動作を利用する。この場合、設定は不要です。
- デフォルト動作の代わりに、Xpath クエリを使用して LoginID を検索する。

Xpath クエリを指定する方法

1. 設定するリソース パートナーの認証方式に移動します。
2. [WS-Federation の設定] - [SAML プロファイル] を選択します。既存の認証方式を変更する場合は、最初に [変更] クリックします。
[SAML プロファイル] ダイアログ ボックスが表示されます。

3. [ユーザの特定]セクションで認証方式が **LoginID** を取得するために使用する Xpath クエリを入力して、[OK] をクリックします。

Xpath クエリにはネームスペース プレフィックスを含めることはできません。以下の例は無効な Xpath クエリです。

```
/saml:Response/saml:Assertion/saml:AuthenticationStatement/  
saml:Subject/saml:NameIdentifier/text()
```

有効な Xpath クエリは次のとおりです。

```
//Response/Assertion/AuthenticationStatement/Subject/  
NameIdentifier/text()
```

検索仕様を使用した WS フェデレーション ユーザの検索

検索仕様を使用して、デフォルトの動作（LoginID がポリシー サーバに渡される）以外の方法でユーザを検索できます。

検索仕様を使用してユーザを検索する方法

1. 設定するリソース パートナーの認証方式に移動します。
2. [WS-Federation の設定] - [SAML プロファイル] を選択します。既存の認証方式を変更する場合は、最初に [変更] をクリックします。

[SAML プロファイル] ダイアログ ボックスが表示されます。

3. [ユーザの特定] セクションで、該当するネームスペース フィールドに検索仕様を入力します。検索仕様は、認証方式がネームスペースを検索するために使用する属性を定義します。ログイン ID を表す変数としてエントリ内に %s を使用します。

たとえば、LoginID の値が **user1** であるとしします。[検索仕様] フィールドで **Username=%s** と指定すると、文字列は **Username=user1** となります。この文字列は、ユーザ ストアに照らして確認され、認証のための正しいレコードが検索されます。

4. [OK] をクリックします。

リソース パートナーでの WS フェデレーション シングル サインオンの設定

認証のための WS フェデレーション シングル サインオン バインドの設定は、SAML プロファイル ページの [SSO] セクションで行います。また、このセクション内での有効なアサーションのを再実行を防ぐために使い捨てアサーション ポリシーを適用することもできます。

シングル サインオン設定の一環として、リダイレクト モード設定を定義します。リダイレクト モードでは、ポリシー サーバがアサーション属性（存在する場合）をターゲット アプリケーションに送信する方法が指定されます。アサーション属性は HTTP ヘッダまたは HTTP Cookie として送信できます。

HTTP ヘッダおよび HTTP Cookie にはサイズ制限があり、アサーション属性でそれらを超えるはできません。サイズ制限は以下のとおりです。

- **HTTP ヘッダ** : SiteMinder は、Web サーバのヘッダに関するサイズ制限までの属性をヘッダで送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダ サイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。
- **HTTP Cookie** : SiteMinder は、Cookie のサイズ制限までの Cookie を送信できます。それぞれのアサーション属性は、それ自身の Cookie として送信されます。Cookie のサイズ制限はブラウザ固有です。また、その制限は、属性ごとではなく、アプリケーションに渡されるすべての属性に適用されます。Cookie のサイズ制限を調べるには、使用する Web ブラウザのドキュメントを参照してください。

WS フェデレーション シングル サインオンの設定方法

1. 設定するリソース パートナーの認証方式に移動します。
2. [WS-Federation の設定] - [SAML プロファイル] を選択します。既存の認証方式を変更する場合は、最初に [変更] クリックします。
[SAML プロファイル] ダイアログ ボックスが表示されます。
3. [SSO] セクションのフィールドに入力します。
フィールドの説明については、[ヘルプ] をクリックしてください。
4. [サブミット] をクリックします。

WS フェデレーション サインアウトの実装

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 リクエスト プロファイルのみをサポートします。

サインアウトの有効化

WS フェデレーション サインアウトを設定する方法

1. 変更する認証方式に移動します。
2. [WS-Federation の設定] - [SAML プロファイル] を選択します。既存の認証方式を変更する場合は、最初に [変更] クリックします。
[SAML プロファイル] ダイアログ ボックスが表示されます。
3. [サインアウト] セクションで、[サインアウトの有効化] チェックボックスをオンにします。
4. サインアウト URL の値を入力します。URL は、https:// または http:// で始まる必要があります。
5. [OK] をクリックします。

詳細情報:

[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 \(P. 117\)](#)

カスタム WS フェデレーション認証方式の作成

既存の WS フェデレーション認証テンプレートの代わりに SiteMinder 認証 API で記述されているカスタム WS フェデレーション認証方式を使用できます。

認証方式のメイン ページには、[方式のセットアップ] セクションに [ライブラリ] フィールドが含まれます。このフィールドには、SAML Artifact 認証を処理する共有ライブラリの名前が含まれています。カスタム認証方式の作成が完了するまで、この値を変更しないでください。

デフォルトの共有ライブラリは `smauthhtml` です。

メッセージ コンシューマ プラグインによるアサーション処理のカスタマイズ

メッセージ コンシューマ プラグインは、メッセージ コンシューマ プラグインを実装する 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。

注: 認証と不明瞭解消に関するステータス コードの詳細については、「Java 用 SiteMinder 用プログラミング ガイド」を参照してください。

認証時、SiteMinder 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、SiteMinder 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、SiteMinder 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、SiteMinder 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、SiteMinder 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。

メッセージ コンシューマ プラグインに関する追加の情報については、以下を参照してください。

- 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「*Java Developer Reference*」にあります。 `MessageConsumerPlugin` インターフェースを参照してください。
- 認証および認可の API に関する概要および概念情報については、「Java 用 SiteMinder プログラミング ガイド」を参照してください。

プラグインを設定する方法

1. SiteMinder SDK をインストールします（未インストールの場合）。
2. MessageconsumerPlugin.java インターフェースを実装します（SiteMinder SDK に含まれています）。
3. メッセージ コンシューマ プラグイン実装クラスを展開します。
4. 管理 UI でメッセージ コンシューマ プラグインを有効にします。

MessageConsumerPlugin の実装

MessageConsumerPlugin.java インターフェースを実装するにより、カスタム メッセージ コンシューマ プラグインを作成します。実装クラスの最小要件は、以下の手順に示されています。

次の手順に従ってください:

1. パラメータが含まれない公のデフォルト コンストラクタ メソッドを提供します。
2. 実装がステートレスになるように、コードを提供します。多数のスレッドが 1 つのプラグイン クラスを使用できる必要があります。
3. 現実の要件に応じて、インターフェース内のメソッドを実装します。

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

`init()`

プラグインが必要とする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、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最終結果を決定する追加のコードを提供します。

SiteMinder から、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
(*installation_home*¥sdk¥samples¥messageconsumerplugin)

MessageConsumerSAML20.java
(*installation_home*¥sdk¥samples¥authextensionsaml20)

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、SiteMinder が実行可能ファイルを検索できることを確認します。

メッセージ コンシューマ プラグインを展開するには、以下の手順を実行します。

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、ポリシー サーバと共にインストールされています。

installation_home¥siteminder¥bin¥jars¥SmJavaApi.jar

SmJavaApi.jar の同一のコピーが SiteMinder SDK と共にインストールされています。このファイルは、*installation_home*¥sdk¥Java¥SmJavaApi.jar ディレクトリにあります。

開発時にそれらのいずれも使用できます。

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。 `installation_home¥siteminder¥config` ディレクトリの JVMOptions.txt ファイルの場所を特定します。

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ポリシー サーバの再起動 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ことに必要です。
4. プラグインを有効化します。

WS フェデレーションでのメッセージ コンシューマ プラグインの有効化

メッセージ コンシューマ プラグインを作成してコンパイルした後に、管理 UI 内で設定することにより、このプラグインを有効にします。UI 設定により、SiteMinder にプラグインの検索場所が指定されます。

[プラグインを展開](#) (P. 192)するまで、プラグイン設定を実行しないでください。

メッセージ コンシューマ プラグインを有効にする方法

1. 管理 UI にログインします。

2. 該当する WS フェデレーション方式の「認証方式」ダイアログ ボックスに移動します。「一般」設定で、「詳細」セクションに移動し、以下のフィールドに入力します。

完全 Java クラス名

プラグインの Java クラス名を指定します。たとえば、SiteMinder 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`

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。

管理 UI でプラグインを設定する代わりに、ポリシー管理 API (C または Perl) を使用して、`IdpPluginClass` および `IdpPluginParameters` を設定します。

3. ポリシー サーバを再起動します。

WS フェデレーションの認証試行が失敗した後のユーザのリダイレクト

シングルサインオン処理において、ユーザがリソース パートナーで認証できない場合のリダイレクト URL をいくつかオプションで設定できます。リダイレクト URL を使用することで、ユーザのリダイレクトをより詳細に制御できます。たとえば、ユーザストアでユーザを検索できない場合、[ユーザが見つからなかった状態のためのリダイレクト URL] の指定により、ユーザを適切な場所にリダイレクトできます。

注: これらの URL は必須ではありません。

リダイレクト URL を設定しない場合、SiteMinder の標準の処理が行われます。失敗した認証を SiteMinder が処理する方法は、設定によって異なります。

リソース パートナーがシングルサインオン トランザクション中にユーザを認証できない場合、リソース パートナーは、後続の処理用にカスタマイズされた URL にそのユーザをリダイレクトできます。

認証が失敗した場合の処理として複数のリダイレクト URL をオプションで設定できます。アサーションが無効な場合、リダイレクト URL を使用することで、ユーザのリダイレクトをより詳細に制御できます。たとえば、ユーザストアでユーザを検索できない場合、ユーザが見つからない場合のリダイレクト URL に入力することができます。

[ステータス リダイレクト URL とモード] は、[認証ダイアログ ボックス] の [追加構成] セクションにあります。リダイレクト URL は、特定のステータス条件に対応します。

- ユーザが見つかりません
- シングルサインオン メッセージが無効です
- ユーザ認証情報は受理されません

いずれかの状態が発生すると、リダイレクト URL は、後続アクション用のアプリケーションまたはカスタマイズされたエラー ページにユーザをリダイレクトできます。

注: リダイレクト URL の設定は必須ではありません。

オプションのリダイレクト URL を設定する方法

1. 変更する フェデレーション認証方法に移動します。
2. [WS-Federation の設定] を選択します。
3. [詳細] セクションで、以下の 1 つまたは複数のフィールドに URL を入力します。
 - ユーザが見つからなかった状態のためのリダイレクト URL
 - 無効な SSO メッセージステータスのためのリダイレクト URL
 - 承認されなかったユーザ認証情報 (SSO メッセージ) ステータスのリダイレクト URL

[無効な SSO メッセージステータスのためのリダイレクト URL] に値を入力する場合は、モードを選択します。

フェデレーション Web サービスでは、設定されたリダイレクト URL の 1 つに認証理由をマッピングすることにより、エラーを処理します。エラーを報告するために、ユーザがその URL にリダイレクトされることがあります。

注: これらのリダイレクト URL を SiteMinder メッセージコンシューマプラグインと共に使用して、より詳細なアサーション処理を実行できます。認証が失敗した場合、プラグインは、指定されたリダイレクト URL の 1 つにユーザをリダイレクトできます。

HTTP ヘッダとしての SAML 属性の供給

アサーション レスポンスには、アサーションの属性を含めることができます。これらの属性は HTTP ヘッダ変数として提供できるため、クライアントアプリケーションでは、これらを使用して、より細かい粒度のアクセス制御を実行できます。

属性を HTTP ヘッダに含める場合には、以下のような利点があります。

- HTTP ヘッダは永続的ではありません。これらのヘッダは、それらが含まれるリクエストまたはレスポンス内のみに存在します。
- HTTP ヘッダは、SiteMinder Web エージェントによって提供される場合と同様に、ブラウザ内で非表示であるため、セキュリティ上の心配が軽減されます。

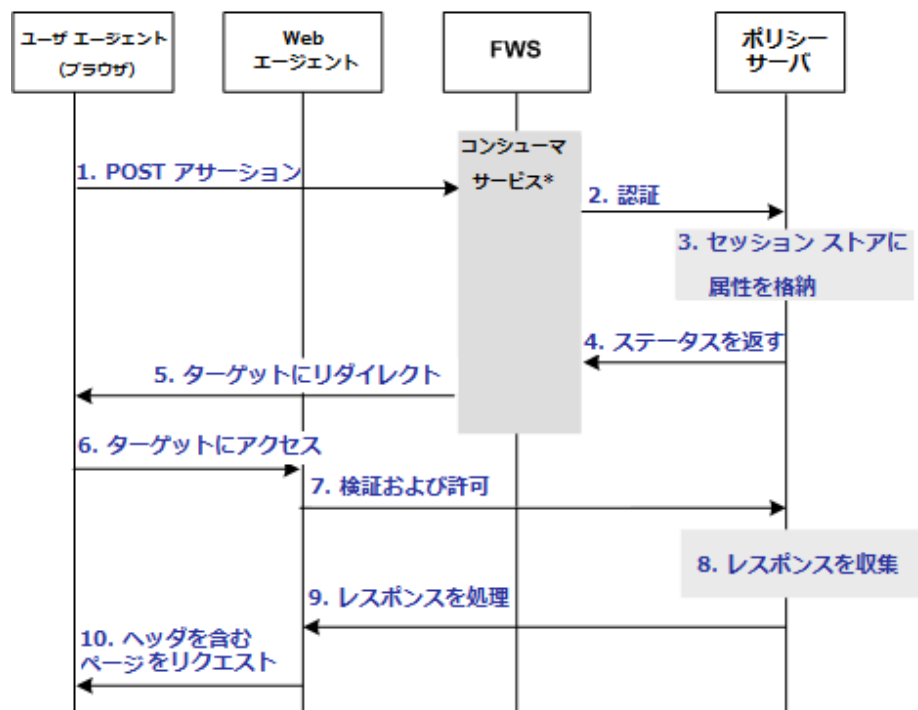
注: HTTP ヘッダにはサイズの制限があり、属性はその制限を超えることができません。SiteMinder では、ヘッダに関する Web サーバのサイズ制限に達するまで、属性をヘッダに送信できます。1 つのヘッダごとに 1 つのアサーション属性のみが許可されます。ヘッダサイズの制限を調べるには、使用している Web サーバのドキュメントを参照してください。

HTTP ヘッダとしての SAML 属性のユース ケース

認証時、一連の SAML 属性がアサーションから抽出され、HTTP ヘッダとして提供されます。認可プロセス時、これらのヘッダはカスタマ アプリケーションに返されます。

次のフロー図は、ランタイムのイベントのシーケンスを表します。

コンシューマでの属性としてのヘッダの処理



*コンシューマ サービスは以下のいずれかです:

- SAML 認証情報コレクタ (SAML 1.x)
- アサーション コンシューマ サービス (SAML 2.0)
- セキュリティ トークン コンシューマ サービス (WS-Federation)

注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプション パックを置き換えて、フェデレーション Web サービス アプリケーション 機能を提供できます。フロー図では、Web エージェント ブロックは SPS フェデレーション ゲートウェイ に組み込まれた Web エージェント になります。SPS フェデレーション ゲートウェイ をインストール する および 設定 する 詳細 については、「*Secure プロキシ サーバ 管理 ガイド*」 (Secure Proxy Server Administration Guide) を参照 してください。

属性を HTTP ヘッダとして処理する場合のイベントのシーケンスは以下のとおりです。

1. アサーションがアサーティング パーティで生成された後、アサーションが依存パーティの適切なコンシューマサービスに送信されます。配信方法 (POST または **Artifact** または **WS フェデレーション**) には無関係です。
注: コンシューマ サービスは、**SAML 認証情報コレクタ (SAML 1.x の場合)**、**アサーション コンシューマ サービス (SAML 2.0 の場合)**、または**セキュリティ トークン コンシューマ サービス (WS フェデレーションの場合)** です。
2. コンシューマサービスは、そのローカル ポリシー サーバをコールし、設定済みの認証方式を使用してアサーションによるユーザ認証を実行します。
3. 認証方式のリダイレクト モードパラメータが **PersistAttributes** に設定されている場合、ポリシー サーバはセッション変数として属性をセッションストアにキャッシュします。
4. 認証の結果はコンシューマ サービスに返されます。
5. コンシューマサービスは、保護されたターゲット リソースにブラウザをリダイレクトします。
6. ブラウザは、ターゲット リソースへのアクセスを試みます。
7. **Web** エージェントは、ポリシー サーバをコールして、ユーザセッションを検証すると共に、ユーザのターゲット リソースへのアクセスが許可されていることを確認します。
8. ポリシー サーバは、設定済みのレスポンスによって属性を取得します。
9. ポリシー サーバは、レスポンスを処理し、属性を **Web** エージェントに送信します。
10. **Web** エージェントは、必要に応じて HTTP ヘッダを設定します。

属性を HTTP ヘッダとして提供するため設定の概要

セッションストアにキャッシュされた SAML 属性を取得し、それらを HTTP ヘッダとして提供するには、複数の設定手順が必要となります。

次の手順に従ってください:

1. SAML 認証方式のリダイレクトモードとして **PersistAttributes** を選択します。これにより、SAML 属性が HTTP ヘッダとして返されるようになります。
2. ターゲット リソースが含まれるレルムに対する認可ルールを設定します。
3. ターゲット リソースを保護するレルムで **PersistentRealm** を設定します。
4. ヘッダとして提供される SAML 属性ごとに、アクティブなレスポンスタイプを使用するレスポンスを設定します。
5. 認可ルールとアクティブなレスポンスをバインドして属性のユーザを HTTP ヘッダとして実装するポリシーを作成します。

SAML 属性を保存するためのリダイレクトモードの設定

依存パーティが SAML アサーションでユーザを認証した後、SAML 属性はセッションストアに書き込まれます。その後、ブラウザはターゲット リソースにリダイレクトされます。

属性データを持つブラウザをリダイレクトする方法

1. 管理 UI にログインします。
2. SAML 認証方式の設定ページに移動します。

3. [リダイレクトモード] パラメータを[属性を保持する]に設定します。[リダイレクトモード] フィールドの場所は以下のように特定されます。

SAML 1.x

[リダイレクトモード] は、設定のメインページの [方式のセットアップ] セクションにあります。

SAML 2.0

[SAML 2.0 設定] - [SSO] を選択します。[リダイレクトモード] は、ページの [SSO] セクションにあります。

WS-フェデレーション

[WS-Federation の設定] - [SAML プロファイル] をクリックします。[リダイレクトモード] は、ページの [SSO] セクションにあります。

4. [サブミット] をクリックして、変更を保存します。

以上で、属性データを渡すようにリダイレクトモードが設定されました。

ユーザを検証するための認可ルールの作成

保護されたターゲット リソースが含まれるレルムの場合、セッションストアから SAML 属性を取得するルールを作成します。

ルールは、認可イベント (onAccessAccept) に基づきます。ユーザは、FWS アプリケーションによってすでに認証されています。Web エージェントは、ユーザを再認証して HTTP ヘッダを渡すことはできません。属性の取得は、認可の段階で行われます。

レルムの OnAccessAccept ルールを作成する方法

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [レルム] と順に移動します。
3. ターゲット リソースが含まれるレルムを選択します。
4. [ルール] セクションの [作成] をクリックします。
[ルールの作成] ページが表示されます。
5. 名前、および必要に応じて説明を入力します。
6. [リソース] フィールドにアスタリスク (*) を入力します。

7. [アクション] セッションで [許可イベント] および [OnAccessAccept] を選択します。
8. [許可/拒否と有効/無効] セクションで [有効] を選択します。
9. [OK] ボタンをクリックしてルールを保存します。

以上で、保護されたリソースのあるレルムに対して認可ルールが定義されました。

属性を HTTP ヘッダとして送信するレスポンスの設定

SAML 属性を HTTP ヘッダとして Web エージェントに送信するレスポンスを設定します。Web エージェントは、レスポンスを処理して、ヘッダ変数がクライアント アプリケーションに利用可能になるようにします。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
3. ターゲット リソースのドメインを選択し、[変更] をクリックします。
4. [レスポンス] タブを選択します。
5. [作成] をクリックします。
[レスポンス] ダイアログが表示されます。
6. 名前を入力します。
7. エージェントのタイプが **SiteMinder Web** エージェントであることを確認します。
8. [レスポンス属性の作成] をクリックします。
[レスポンス属性] ダイアログが表示されます。
9. [属性] フィールドで [WebAgent-HTTP-Header-Variable] を選択します。
10. [属性の種類] の [アクティブ レスポンス] を選択します。

11. フィールドには以下のように入力します。

変数名

ヘッダ変数の名前を指定します。この名前が割り当てられます。

ライブラリ名

`smfedattrresponse`

この値がこのフィールドに入力される必要があります。

関数名

`getAttributeValue`

この値がこのフィールドに入力される必要があります。

パラメータ

アサーションに表示される属性名を指定します。

アサーション内の属性は、フェデレーション パートナーとの合意に基づいて決定してください。

12. [OK] をクリックして属性を保存します。
13. HTTP ヘッダ変数の各属性について手順を繰り返します。1 つのレスポンスに対して複数の属性を設定できます。
- [レスポンス] タブに戻ります。作成して属性は、[属性リスト] セクションに一覧表示されます。
14. [OK] をクリックしてレスポンスを保存します。
- [レスポンス] タブに戻ります。
15. [サブミット] をクリックしてドメインを保存します。

レスポンスは、HTTP ヘッダになるように属性を Web エージェント上に送信します。

属性を HTTP ヘッダとして実装するためのポリシーの作成

SAML 属性の使用を HTTP ヘッダとして実装するには、認可イベント ルールおよびアクティブ レスポンスを 1 つのポリシーにまとめます。

次の手順に従ってください:

1. 管理 UI にログオンします。
2. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。

3. ターゲット リソースが含まれるドメインを選択し、[変更] をクリックします。
4. [ポリシー] タブを選択し、[ポリシー] セクションの [作成] をクリックします。
[ポリシーの作成] ダイアログが表示されます。
5. [名前] フィールドにわかりやすい名前を入力します。
6. 保護されたリソースへのアクセスを許可するユーザを [ユーザ] タブで選択します。
7. 先に [ルール] タブで作成した認可ルールを追加します。
8. 認可ルールを選択し、[レスポンスの追加] をクリックします。
[使用可能なレスポンス] ダイアログ ボックスが表示されます。
9. 先に作成したアクティブ レスポンスを選択し、[OK] をクリックします。
[ルール] タブに戻ります。レスポンスが認証ルールと共に表示されます。
10. [サブミット] をクリックしてポリシーを保存します。

SAML 属性を HTTP ヘッダとして使用できるようにするためのポリシーが作成されました。

WS-Federation 認証方式でターゲット リソースを保護する方法

WS フェデレーション認証方式を使用する SiteMinder ポリシーを設定することにより、ターゲットのフェデレーション リソースを保護します。

次の手順に従ってください:

1. WS フェデレーション認証方式を使用するレルムを作成します。レルムは、ターゲット リソースを集めたものです。
レルムは、以下の方法で作成できます。
 - すでに設定済みの認証方式ごとに[固有なレルムを作成](#) (P. 205)する。
 - 対応する WS フェデレーション認証方式へのリクエストの送信にカスタム認証方式を使用する[単一のターゲット レルムを設定](#) (P. 206)する。すべてのプロデューサに対して単一のターゲットを持つレルムを 1 つ設定することにより、認証のレルム設定が簡単になります。
2. 関連のルールを設定します。また、必要に応じてレスポンスを設定します。
3. レルム、ルール、およびレスポンスを、ターゲット リソースを保護するポリシーとしてグループ化します。

重要: レルム内の各ターゲット URL も未承認応答 URL で識別されます。未承認応答は、リソース パートナーからの初期リクエストなしに、アカウント パートナーからリソース パートナーに送信されます。このレスポンスにターゲットが含まれます。アカウント パートナーにおいて、管理がこのレスポンスをリンクに含めます。このリンクにより、ユーザがリソース パートナーにリダイレクトされます。

各認証方式に対する固有なレルムの設定

SAML または WS フェデレーション認証方式ごとに固有のレルムを設定する手順は、レルムを作成するための標準的な手順に従います。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] の順に移動します。
ドメインを作成するためのページが表示されます。
2. [ドメインの作成] をクリックします。

3. ドメイン名を入力します。
4. ユーザディレクトリをドメインに追加します。このディレクトリは、フェデレーション リソースへのアクセスをリクエストするユーザが含まれるディレクトリです。
5. [レلم] タブを選択し、レلمを作成します。
 - [エージェント] フィールドで、ターゲット リソースが存在する **Web** サーバを保護する **Web** エージェントを選択します。
 - [認証方式] フィールドで該当する認証方式を選択します。
6. レلمのルールを作成します。

ルールの一部として、ユーザの認証時の処理を制御するために使用可能なアクション (**Get**、**Post**、または **Put**) を選択します。
7. [ポリシー] タブを選択し、ターゲット フェデレーション リソースを保護するポリシーを設定します。先に作成したレلمをこのポリシーと関連付けます。

以上で、固有のレلمを持つポリシーがフェデレーション リソースを保護します。

すべての認証方式に対する単一のターゲットレلمの設定

認証方式に対するレلمの設定を簡略化するには、アサーションを生成する複数のサイトに対して単一のターゲットレلمを作成します。

このタスクを実行するには、以下のコンポーネントをセットアップします

- 単一のカスタム認証方式

このカスタム方式は、すでに各アサーティングパーティに対して設定済みの対応する **SAML** または **WS フェデレーション** 認証方式にリクエストを転送します。
- 1つのターゲット URL を持つ単一のレلم

単一のターゲットレルムに対する認証方式の作成

単一のターゲットレルムに対するカスタム認証方式を定義するには、以下の作業を実行する必要があります。

- 認証方式を設定する方法
- リソース リクエストに適用する認証方式をポリシー サーバに指定するためのカスタム方式のパラメータを定義します。

まず、**SAML** または **WS フェデレーション** 認証方式が設定済みであることを確認します。設定されていない場合は、これらの方式を設定して、カスタム方式が参照できるようにします。

認証方式を作成する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。
2. 使用しているプロトコルの手順に従って 1 つまたは複数の認証方式を作成します。
3. [OK] をクリックして終了します。

詳細情報:

[SAML 1.x 認証方式 \(P. 179\)](#)

[WS フェデレーション認証方式の概要 \(P. 372\)](#)

[SAML 2.0 認証方式を設定する方法 \(P. 291\)](#)

カスタム認証方式の設定

単一のターゲットレルムが正しく動作するためには、特定のカスタム認証方式に依存します。

単一のターゲットレルムに対するカスタム認証方式を設定する方法

1. [インフラストラクチャ] - [認証] - [認証方式] の順に移動します。
[認証方式の作成] ページが表示されます。
2. フィールドには以下のように入力します。

名前

カスタム認証方式のわかりやすい名前を入力します (**SAML Custom Auth Scheme** など)。

3. [各方式共通セットアップ] セクションで、以下のフィールドに入力します。

認証方式タイプ

カスタム テンプレート

保護レベル

デフォルトをそのまま使用するか、新しいレベルを設定します。

4. [方式のセットアップ] セクションで、以下のフィールドに入力します。

ライブラリ

smauthsinglefed

秘密キー

このフィールドは空のままにします。

秘密キーの確認入力

このフィールドは空のままにします。

パラメータ

以下のパラメータのいずれかを指定します。

- SCHEMESET=LIST; <saml-scheme1>;<saml_scheme2>

使用する SAML 認証方式名のリストを指定します。

artifact_producer1 という名前の Artifact と samlpost_producer2 という名前の POST プロファイル方式を設定した場合は、これらの方式を入力します。例：

SCHEMESET=LIST;artifact_producer1;samlpost_producer2

- SCHEMESET=SAML_ALL;

設定済みの方式をすべて指定します。カスタム認証方式によって、SAML 認証方式がすべて列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_POST;

設定したすべての SAML POST プロファイル方式を指定します。カスタム認証方式によって、POST プロファイル方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=SAML_ART;

設定したすべての SAML Artifact 方式を指定します。カスタム認証方式によって、Artifact 方式が列挙され、リクエストに対して適切なプロバイダ ソース ID を持つものが特定されます。

- SCHEMESET=WSFED_PASSIVE;

正しいアカウント パートナー ID を持つ WS フェデレーション認証方式を検索するために、すべての WS フェデレーション認証方式指定します。

この方式を SiteMinder 管理者用に有効にする

チェック マークはオフのままにします。

5. [サブミット] をクリックします。

カスタム認証方式が完成しました。

単一のターゲット レルムの設定

認証方式を設定し、それらをカスタム方式と関連付けた後、フェデレーション リソースに対する単一のターゲット レルムを設定します。

次の手順に従ってください:

1. [ポリシー] - [ドメイン] - [ドメイン] と順に移動します。
2. 単一のターゲット レルムのポリシー ドメインを変更します。
3. [レルム] タブを選択し、[作成] をクリックします。
[レルムの作成] ダイアログが表示されます。
4. 以下の値を入力して、単一のターゲット レルムを作成します。

名前

この単一のターゲット レルムの名前を入力します。

5. [リソース] オプションの以下のフィールドに入力します。

エージェント

ターゲット リソースのある Web サーバを保護する Web エージェントを選択します。

リソース フィルタ

ターゲット リソースの場所を指定します。この場所は、フェデレーション リソースをリクエストするすべてのユーザがリダイレクトされる場所です。

例： /FederatedResources など。

6. [デフォルト リソース保護] セクションで [保護] オプションを選択します。
7. 以前に設定したカスタム認証方式を [認証方式] フィールドで選択します。
たとえば、カスタム方式が Fed Custom Scheme という名前の場合は、この方式を選択します。
8. [OK] をクリックします。

単一のターゲット レルムのタスクが完了しました。

単一のターゲット レルムに関するルールの設定

単一のターゲット レルムを設定した後、リソースを保護するためのルールを設定します。

1. 単一のターゲット レルムの [変更] ページに移動します。
2. [ルール] セクションの [作成] をクリックします。
[ルールの作成] ページが表示されます。
3. ルール ページのフィールドに値を入力します。
4. [OK] をクリックします。

単一のターゲット レルム設定に新しいルールが含まれました。

単一のターゲットレلمを使用するポリシーの作成

単一のターゲットレلمを参照するポリシーを作成します。単一のターゲットレلمでは、リクエストを適切な SAML 認証方式に届けるためのカスタム認証方式を使用することに注意してください。

注: この手順は、ドメイン、カスタム認証方式、単一のターゲットレلم、関連ルールがすでに設定済みであることが前提です。

次の手順に従ってください:

1. すでに設定済みのドメインに移動します。
2. [ポリシー] タブを選択し、[作成] をクリックします。
[ポリシーの作成] ページが開きます。
3. [一般] セクションで、ポリシーの名前と説明を入力します。
4. [ユーザ] セクションからユーザをポリシーに追加します。
5. [ルール] タブから単一のターゲットレلمに対して作成したルールを追加します。
残りのタブは省略可能です。
6. [OK] をクリックします。
7. [サブミット] をクリックします。

ポリシータスクが完了しました。リクエストがこのポリシーのトリガすると、このポリシーは、ユーザを認証するために、単一のレلمおよび関連付けられた認証方式に依存します。

第 18 章: SAML 2.0 アフィリエーションの設定

このセクションには、以下のトピックが含まれています。

[アフィリエーションの概要 \(P. 403\)](#)

[SAML 2.0 アフィリエーションの設定 \(P. 404\)](#)

アフィリエーションの概要

SAML アフィリエーションは、単一のプリンシパルの名前識別子を共有する SAML エンティティのグループです。

サービス プロバイダおよびアイデンティティ プロバイダがアフィリエーションに所属できます。ただし、1 つのエンティティが所属できるのは 1 つのアフィリエーションのみです。各種サービス プロバイダは、アフィリエーションの全域で名前 ID 定義を共有します。アイデンティティ プロバイダは、アフィリエーションにわたってユーザの不明瞭解消を共有します。

アフィリエーションにより、各サービス プロバイダで必要とされる設定が少なくなります。さらに、1 つのプリンシパルに対して 1 つの名前 ID を使用することにより、アイデンティティ プロバイダでのストレージ容量が節約されます。

アフィリエーションにより、以下の機能が提供されます。

- シングル サインオン
- シングル ログアウト

注: アフィリエーションの設定はオプションです。

アフィリエーションによるシングル サインオン

シングル サインオンを使用する場合、サービス プロバイダはアサーションのリクエストをアイデンティティ プロバイダに送信します。認証リクエストには、アフィリエーション識別子を指定する属性が含まれます。

アイデンティティ プロバイダでは、リクエストを受信すると、以下のアクションを実行します。

- サービス プロバイダが 認証リクエストで識別されたアフィリエーションのメンバであることを確認する。
- アフィリエーションによって共有される名前 ID のアサーションを生成する。
- このアサーションをサービス プロバイダに返信する。

アサーションを受信すると、認証がサービス プロバイダで実行されます。

アフィリエーションによるシングル ログアウト

サービス プロバイダは、ログアウト リクエストを生成する際に、アイデンティティ プロバイダがアフィリエーションのメンバかどうかを確認します。サービス プロバイダは、属性をリクエストに含め、これをアフィリエーション ID に設定します。アイデンティティ プロバイダでは、リクエストを受信し、サービス プロバイダが属性で識別されたアフィリエーションに属することを確認します。

アイデンティティ プロバイダは、セッションストアからアフィリエーション名前 ID を取得します。アイデンティティ プロバイダは、すべてのセッション参加者にログアウト リクエスト メッセージを発行する際、アフィリエーションのメンバのアフィリエーション名前 ID を含めます。

SAML 2.0 アフィリエーションの設定

SAML アフィリエーションを使用して SAML エンティティをグループに追加することにより、このエンティティが単一のプリンシパルの名前識別子を共有できるようになります。フェデレーション ネットワークのいずれかのパートナーでアフィリエーションを設定できます。

アイデンティティ プロバイダの場合は、アフィリエーションと関連付けられた名前 ID を割り当てます。共有された名前 ID プロパティは、アフィリエーションに属するすべてのサービス プロバイダに適用されます。

サービスプロバイダの場合、アフィリエーションは、認証のユーザ不明瞭解消プロセスを提供します。サービスプロバイダは、アサーションを受信すると、ユーザ ID 情報をアサーションから抽出します。ユーザ不明瞭解消設定に基づいて、サービスプロバイダは、ローカルユーザディレクトリに照らして ID 情報を比較し、適切なユーザレコードを検索します。

次の手順に従ってください:

1. [フェデレーション] - [レガシーフェデレーション] - [SAML アフィリエーション] の順に移動します。
[SAML アフィリエーションの作成] ページが表示されます。
2. 必要なフィールドに入力します。以下の情報に注意してください。
 - アイデンティティプロバイダの場合、[ユーザ] 設定は機能しません。これらの設定は無視してください。
 - サービスプロバイダの場合、[名前 ID] 設定は機能しません。これらの設定は無視してください。
3. [サブミット] をクリックします。

アフィリエーションのメンバのサービスプロバイダのリストが、アフィリエイトダイアログボックスの [SAML サービスプロバイダの関連付け] セクションに表示されます。このサービスプロバイダのリストは、読み取り専用です。このリストを編集するには、サービスプロバイダオブジェクトを変更する必要があります。

ユーザ不明瞭解消にアフィリエーションを使用する SAML 2.0 認証方式のリストが、[SAML 認証方式の関連付け] セクションに表示されます。この認証方式のリストは、読み取り専用リストです。このリストを編集するには、特定の方式を変更する必要があります。

アイデンティティプロバイダでのアフィリエーションの割り当て

アイデンティティプロバイダに対し、アフィリエーションはアサーションで名前 ID を提供します。また、アイデンティティプロバイダは、アフィリエーション ID をアサーションに含めます。サービスプロバイダオブジェクトの設定時、アフィリエーションを選択します。

ランタイム時、アイデンティティ プロバイダは、アフィリエーションの NameID を使用し、サービス プロバイダ オブジェクトに対して定義された 名前 ID 設定は無視します。

次の手順に従ってください:

1. 変更するサービス プロバイダオブジェクトに移動します。
2. [名前 ID] ページに移動します。
3. プルダウン リストから **SAML アフィリエーション**を選択します。

アフィリエーションは、あらかじめリスト内に設定されている必要があります。

サービス プロバイダでのアフィリエーションの割り当て

サービス プロバイダでは、アフィリエーションによってユーザ情報が決定されます。サービス プロバイダで認証方式を設定する際に、アフィリエーションを選択します。

実行時、サービス プロバイダはアフィリエーションのユーザ設定に依存します。認証方式のユーザ設定は無視されます。

次の手順に従ってください:

1. 変更する **SAML 2.0** 認証方式に移動します。
2. [一般] ページに移動します。
3. [ユーザの特定] セクションで、プルダウン リストから **SAML アフィリエーション**を選択します。

アフィリエーションは、あらかじめリスト内に設定されている必要があります。

第 19 章: アサーション クエリの属性を使用したユーザの認可

このセクションには、以下のトピックが含まれています。

[属性機関による認可の実行 \(P. 407\)](#)

[ユーザ属性を持つユーザを認可するフロー図 \(P. 410\)](#)

[属性機関および SAML リクエストの設定方法 \(P. 411\)](#)

[属性クエリを生成する SAML リクエストをセットアップする方法 \(P. 417\)](#)

属性機関による認可の実行

ポリシー サーバは、以下の情報を使用してユーザを認可します。

- ポリシー設定で指定されたユーザ
- ポリシー式
- アクティブなポリシー
- IP アドレス制限
- 時間制限

また、ポリシー サーバは、**SAML 2.0** 属性機関が提供するユーザ属性を使用してユーザを認可します。ユーザが保護されたリソースへのアクセスを要求すると、ポリシー サーバは認可エンティティとして、追加のユーザ属性を要求することができます。ポリシー サーバは、リソースへのアクセスを許可する前にこれらの属性を評価します。

SAML 2.0 アサーション クエリ/リクエスト プロファイルは、**SAML 属性機関** および **SAML リクエスト** の 2 つのエンティティを使用します。

SAML 属性機関

SAML 属性機関は、クエリ メッセージの処理およびアサーションへの属性の追加を属性サービスに依存しています。これらのアサーションには、保護されたリソースへのアクセスを認可するために **SAML リクエスト** が使用するユーザ属性が含まれます。属性サービスは、フェデレーション **Web** サービス アプリケーションの一部です。

エンティティが属性機関に対してリクエストする際、メッセージにはリクエストが取得したいユーザ属性が含まれます。メッセージには、名前 ID およびリクエストの発行者も含まれます。属性サービスは、NameID を使用してユーザを不明瞭解消するため、リクエストされた属性に関して返す値の内容を認識しています。属性サービスは、SOAP メッセージにラップされた属性アサーションが含まれるレスポンスメッセージを返します。このレスポンスには、ユーザ属性が含まれます。

注: ユーザは属性機関で認証される必要はありません。また、属性機関とリクエストの間のシングルサインオンの関係も必要はありません。

SAML リクエスト

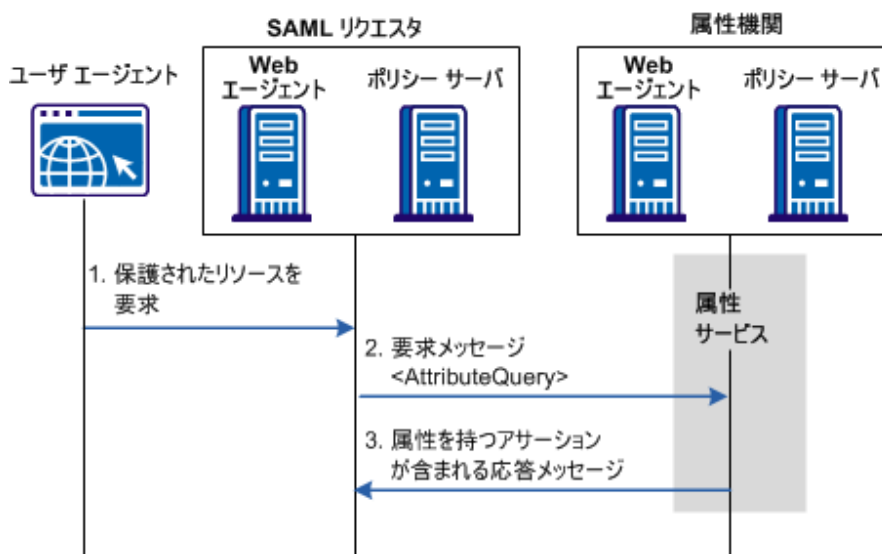
SAML リクエストは、SAML 2.0 アサーション クエリ/リクエスト プロファイルを使用してユーザに関する属性をリクエストする SAML エンティティです。SiteMinder では、SAML リクエストは特定のサービスではなく、<AttributeQuery> メッセージの作成および処理を行うことのできる一連のポリシー サーバ機能です。保護されたターゲット リソースは常に SAML 要求者に存在するため、リクエストは属性機関からユーザ属性を要求します。リクエストでは、これらの属性をポリシー式で使用される変数に解決します。

注: SiteMinder のフェデレーション環境では、SAML 属性機関がアイデンティティ プロバイダ、SAML リクエストがサービス プロバイダとなります。ただし、必ずしもこのとおりであるとは限りません。

SAML 2.0 ユーザ属性に基づいた許可リクエストを評価するには、**フェデレーション属性変数**という名前の属性タイプをポリシー式に追加します。ターゲット リソースを保護するポリシーは、この変数を使用します。SAML リクエストは、ポリシー変数に基づいて、クエリ メッセージを属性機関に送信します。このクエリ メッセージには、属性がリクエストされている SAML エンティティの名前 ID が含まれます。SAML 属性機関は、属性ステートメントを使用したアサーションが含まれるレスポンス メッセージを返します。

ユーザは SAML リクエストでセッションを持つ必要があります。ただし、ユーザは属性機関でログインしたり、認証したりする必要はありません。

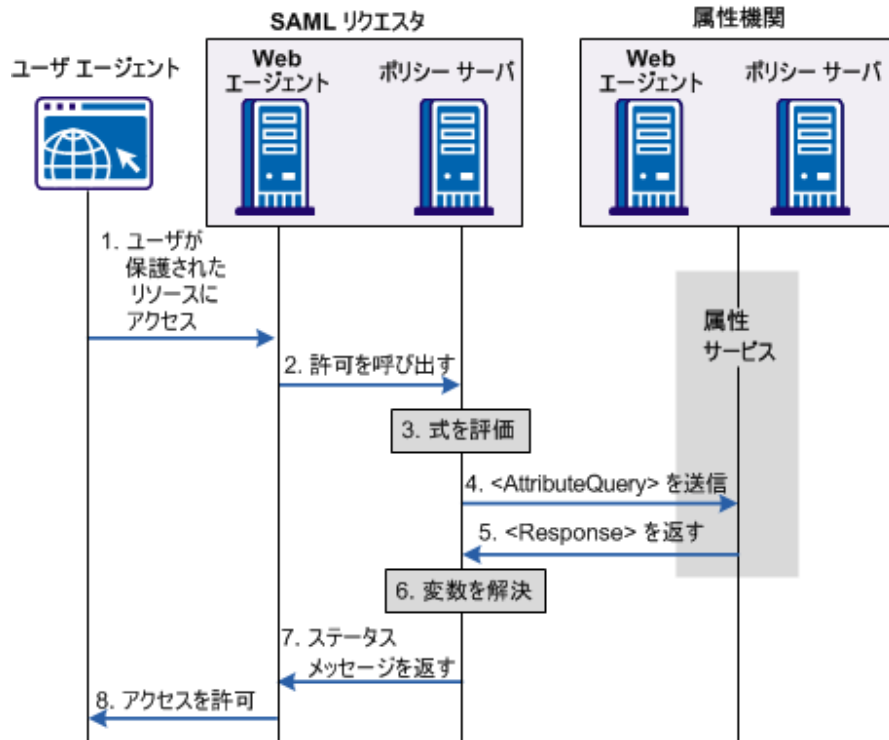
次の図は、属性クエリがどのように処理されるかを示しています。



注: SPS フェデレーション ゲートウェイは、Web エージェント Web エージェント オプションパックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーション ゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシ サーバ管理ガイド*」 (Secure Proxy Server Administration Guide) を参照してください。

ユーザ属性を持つユーザを認可するフロー図

次のフロー図は、属性機関を使用した認可プロセスを示します。



注: SPS フェデレーションゲートウェイは、Web エージェント Web エージェント オプションパックを置き換えて、フェデレーション Web サービス アプリケーション機能を提供できます。SPS フェデレーションゲートウェイをインストールするおよび設定する詳細については、「*Secure プロキシサーバ管理ガイド*」(Secure Proxy Server Administration Guide)を参照してください。

ユーザ属性リクエストのシーケンスは以下のとおりです。

1. ユーザが保護されたリソースにアクセスします。ユーザは、ローカルにログインできます。または SAML アサーションを使用して認証を受けることができます。
2. SAML リクエストの Web エージェントは、ユーザがリソースへのアクセスを許可されているかどうかを判別するために、ローカルのポリシーサーバをコールします。リソースを保護するポリシーは、連携した属性変数を使用した認可にポリシー式を使用します。

3. ポリシー サーバは、これらの変数を解決しようとしませんができません。ポリシー サーバは、ユーザの **NameID** を取得するためにローカル ユーザ ストアでユーザを検索します。
4. 属性機関の **AttributeService URL** に属性クエリが送信されます。**AttributeQuery** には、ユーザの **NameID** および要求された属性が含まれます。
5. 属性機関は、要求された属性のアサーションが含まれる **SAML** レスポンスを返します。
6. **SAML** リクエストは、変数の解決を完了し、ポリシー式を評価します。
7. 認可ステータス メッセージが **Web** エージェントに返されます。
8. 認可ステータスに応じて、**Web** エージェントは、要求されたリソースへのアクセスの許可または拒否を行います。

属性機関および SAML リクエストの設定方法

SiteMinder のコンテキストでは、属性機関はアイデンティティ プロバイダになります。

SiteMinder を SAML 属性機関として設定する方法

1. ユーザを特定するための検索仕様を定義します。 **NameID** を検索仕様に入力します。
2. 属性機関がクエリにレスポンスを送信する対象となるバックチャネルを設定します。
3. クエリに応じて返される属性を定義します。
4. 属性機関サービスに対するアクセス権限をユーザに付与します。

SiteMinder コンテキストでは、**SAML** リクエストはサービス プロバイダになります。

SiteMinder を SAML リクエストとして設定する方法

1. 属性クエリ機能を有効にします。
2. リクエストが属性機関からのレスポンスを受信する対象となるバックチャネルを設定します。
3. 属性クエリで要求された属性の一覧を定義します。

4. フェデレーション属性変数を設定します。
5. 属性クエリに包める NameID を設定します。

属性機関のセットアップ

SiteMinder コンテキストでは、属性機関は、属性機関サービス対応のアイデンティティプロバイダになります。

注: アイデンティティプロバイダを属性機関として機能させる場合に、シングルサインオンなど、他のアイデンティティプロバイダ機能を設定する必要はありません。

属性機関を設定する方法

1. 管理 UI にログインします。
2. SAML リクエストを表すサービス プロバイダ オブジェクトに移動します。SAML リクエストは、ユーザ属性をリクエストします。
3. [Modify] を選択します。
[SAML サービス プロバイダ] ページが表示されます。
4. [属性] タブを選択します。
5. [属性サービス] セクションで、[有効] を選択します。このチェックボックスにより、属性機関機能を有効になります。
6. (オプション) 有効期間の値を変更します。デフォルトの 60 秒を使用してもかまいません。

この設定は、アサーションを 60 秒より長い時間有効にする場合のみ変更します。

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。

7. (オプション) 署名設定項目のいずれかまたは両方を設定します。いずれの設定も必須ではありません。

署名された属性クエリが必要

このオプションは、属性機関が **SAML** リクエストからの署名済みクエリのみを受け付けるようにする場合に選択します。

署名オプション

SAML リクエストに返されるときに、属性アサーションを署名する、SAML レスポンスを署名する、両方を署名する、またはどちらも署名しない、という選択肢の 1 つを選択します。

8. [ユーザの検索] セクションで、使用するネームスペース用の検索仕様を指定します。

認証方式が検索文字列として使用するネームスペース属性を入力してください。

NameID を表す変数としてエントリ内に **%s** を使用します。たとえば、NameID の値が **user1** であるとします。[検索仕様] フィールドで **Username=%s** と指定すると、文字列は **Username=user1** となります。この文字列は、ユーザストアに照らして確認され、認証のための正しいレコードが検索されます。

9. [バックチャネル] セクションで、以下のフィールドに入力します。

- パスワード
- パスワードの確認入力

SAML 2.0 Artifact 認証の設定が完了している場合、バックチャネルのパスワードがすでに設定されています。このパスワードは、SSO および属性機関サービスの両方に使用できます。

10. [サブミット] をクリックして、変更を保存します。
11. 「[属性機関の属性の設定 \(P. 413\)](#)」に進みます。

属性機関での属性の設定

設定している属性がシングルサインオン リクエストの一部であるか、属性クエリ リクエストであるかを示します。[SAML サービスプロバイダ属性] ダイアログ ボックスの [取得方法] フィールドにより、属性の機能が決定されます。

両方のサービスに同じ属性を使用するには、同じ属性名と変数を使用する属性ステートメントを 2 つ作成します。一方の属性は取得方法として SSO を使用し、もう一方の属性は取得方法として属性サービスを使用します。

属性を設定する方法

1. [SSO アサーションの属性を設定](#) (P. 271) します。

属性機関で属性を設定するための設定プロセスは、シングル サインオン アサーションの属性を設定する場合と同じです。

2. SAML リクエストを表すサービス プロバイダ オブジェクトの [属性] ダイアログ ボックスに移動します。
3. [属性] ダイアログ ボックスから、[属性] セクションの [追加] を選択します。

[属性の追加] ダイアログ ボックスが表示されます。

4. ページの [属性のセットアップ] セクションにある [取得方法] フィールドで [属性サービス] を選択します。

属性クエリがこの属性をリクエストする場合、取得方法として属性サービスを選択することにより、この属性が属性アサーションに含まれることがマークされます。

依存パートナーの属性機関サービスへのアクセスの許可

属性機関サービスがリクエストに応答するには、依存パートナーにサービスへのアクセス権を与える必要があります。これは 2 段階のプロセスです。

1. [フェデレーション エージェント グループへの Web エージェントの追加](#) (P. 157)
2. [属性機関サービスのポリシーへの依存パートナーの追加](#) (P. 416)

フェデレーション エージェントグループへの Web エージェントの追加

FederationWebServicesAgentGroup エージェントグループに、FWS アプリケーションを保護する Web エージェントを追加します。

- ServletExec の場合、このエージェントは、Web エージェント オプションパックがインストールされている Web サーバにあります。
- WebLogic や JBOSS などのアプリケーション サーバの場合、この Web エージェントは、アプリケーション サーバプロキシがインストールされている場所にインストールされています。Web エージェント オプションパックは別のシステムに存在することもあります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [インフラストラクチャ] - [エージェント] - [エージェント] をクリックします。
3. [エージェントの作成] をクリックします。
4. 展開内の Web エージェントの名前を指定します。[サブミット] をクリックします。
5. [インフラストラクチャ] - [エージェント] - [エージェントグループ] をクリックします。
6. [FederationWebServicesAgentGroup] エントリを選択します。
7. [追加/削除] をクリックします。[エージェントグループのメンバ] ダイアログボックスが表示されます。
8. Web エージェントを [使用可能なメンバ] リストから [選択されたメンバ] リストに移動します。
9. [OK] ボタンをクリックして、[エージェントグループ] ダイアログに戻ります。
10. [サブミット] クリックした後、[閉じる] クリックしてメインページに戻ります。

属性機関サービスのポリシーへの依存パートナーの追加

属性機関による認可を実装する場合、パートナーシップ内の依存パーティは属性機関サービスにアクセスする許可を必要とします。SiteMinder は、ポリシーによって SAML 2.0 属性機関を保護します。

ポリシー サーバをインストールする際に、デフォルトで FederationWebServicesDomain がインストールされます。このドメインには、属性サービスの SAML2FWSAttributeServicePolicy が含まれます。

属性サービス ポリシーに関するアクセス権をすべての関連する依存パートナーに付与します。

次の手順に従ってください:

1. 管理 UI で、[ポリシー] - [ドメイン] - [ドメイン ポリシー] の順に移動します。

ドメイン ポリシーのリストが表示されます。

2. [SAML2FWSAttributeServicePolicy] を選択します。

[ドメイン ポリシー] ページが表示されます。

3. [変更] をクリックしてポリシーを変更します。

4. [ユーザ] タブを選択します。

5. SAML2FederationCustomUserStore ユーザディレクトリのダイアログボックスで、[メンバーの追加] をクリックします。

[ユーザ/グループ] ページが表示されます。

先に設定したアフィリエイト ドメインが [ユーザ/グループ] ダイアログボックスにリスト表示されます。たとえば、アフィリエイト ドメインが **fedpartners** という名前の場合、エントリーは **affiliate:fedpartners** となります。

6. サービスへのアクセスを必要とするパートナーが存在するアフィリエイト ドメインの横のチェック ボックスをオンにします。[OK] をクリックします。

ユーザディレクトリのリストに戻ります。

7. [サブミット] をクリックします。
ポリシーのリストに戻ります。

以上で、必要なパートナーに属性機関サービスへのアクセス権が付与されました。

属性クエリを生成する SAML リクエストをセットアップする方法

SiteMinder サービス プロバイダが SAML リクエストとして機能するためには、属性クエリを生成できるように SAML 2.0 認証方式を設定する必要があります。この設定は、サービス プロバイダサイトで行います。

次の手順に従ってください:

1. 管理 UI にログインします。
2. SAML 2.0 認証方式設定に移動します。
3. [属性クエリを有効にし、属性を指定](#) (P. 417) します。
4. [属性クエリの名前 ID を設定](#) (P. 418) します。
5. [属性クエリ用のバックチャネルを設定します。](#) (P. 419)
6. [フェデレーション属性変数を設定します。](#) (P. 419)
7. [フェデレーション属性変数を使用してポリシー式を作成します。](#) (P. 420)

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。

属性クエリの有効化と属性の指定

SAML リクエストが属性クエリを生成するには、属性クエリ機能を有効にする必要があります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. SAML 2.0 認証方式の認証設定にアクセスします。
3. [属性] タブを選択します。
4. [属性クエリ] セクションで、[有効] を選択します。

5. (オプション) 以下のチェック ボックスをオンにします。
 - 属性クエリに署名
 - 署名された属性クエリが必要
 - すべての属性の取得
6. [属性サービス] フィールドに値を入力します。
7. ページの [属性] セクションで、[追加] をクリックします。
[属性の追加] ページが表示されます。
8. ページでフィールドに値を入力します。
9. [OK] をクリックして、変更を保存します。
[属性] ページに戻ります。
10. [NameID を設定](#) (P. 418) します。この NameID は、属性機関で使用する属性クエリに含められます。

属性クエリに関する NameID の設定

属性機関に送信されるクエリ メッセージには、属性機関が属性をリクエストしているユーザの名前 ID が含まれます。名前 ID 設定では、SAML リクエストが名前 ID を取得する方法を指定します。リクエストは、名前 ID を属性クエリに配置します。

名前 ID を指定する方法

1. SAML リクエスト認証方式の [属性] ダイアログ ボックスに移動します。
2. ページの [名前 ID] セクションで、以下の設定を定義します。
 - 名前 ID 形式
 - 名前 ID タイプ
 - 名前 ID フィールド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. [OK] をクリックして、変更を保存します。
4. バックチャネルが未設定の場合は、バックチャネルを設定します。

属性クエリに関するバックチャネルの設定

属性クエリは、安全なバックチャネル上で属性機関に送信されます。

サービス プロバイダとアイデンティティ プロバイダの間で 1 つのバックチャネルのみを利用できます。そのため、属性クエリに関するバックチャネル設定は、SAML Artifact プロファイルに使用されるバックチャネル設定と同じです。

バックチャネルを設定する方法

1. SAML リクエストの [認証方式] ページに移動します。
2. [暗号化 & 署名] タブをクリックします。
3. [バックチャネル] セクションで、以下のフィールドに入力します。
 - 認証
 - SP 名
 - パスワード
 - パスワードの確認入力フィールドの説明については、[ヘルプ] をクリックしてください。
4. [OK] をクリックします。

フェデレーション属性変数の作成

ポリシー式でフェデレーション属性変数を使用するには、先に属性変数を作成する必要があります。

フェデレーション属性変数を定義する方法

1. [ポリシー] - [ドメイン] - [変数] と順に移動します。
[変数] ダイアログ ボックスが表示されます。
2. [変数の作成] を選択します。
設定ウィザードの最初の手順で [ドメイン] セクションが表示されます。
3. 変数の追加先のフェデレーション ポリシー ドメインを選択し、[次へ] をクリックします。

4. [変数の定義] の手順で、[一般] セクションの 2 つのフィールドに入力します。

名前

変数を識別します。

変数のタイプ

フェデレーション属性

5. [定義] セクションのフィールドに入力します。
フィールドの説明については、[ヘルプ] をクリックしてください。
6. [終了] をクリックして変数を保存します。
7. [この変数をポリシー式に追加します。](#) (P. 420) フェデレーション リソースを保護するポリシーは、このポリシー式を使用します。

注: ポリシー式には複数のフェデレーション属性変数を使用できます。それぞれの変数が **SAML 2.0** 認証方式に関連付けられます。そのため、単一の表現によって、多数の属性リクエストが多数の属性権限に送信される可能性があります。

フェデレーション属性変数を使用したポリシー式の作成

フェデレーション属性変数を認可プロセスの一部として使用するには、属性変数をポリシー式に追加します。このポリシー式を、**SAML** リクエストでターゲットリソースを保護するポリシーに関連付けます。

ポリシー式の作成に関する詳細については、「[ポリシー サーバ設定ガイド](#)」の「ポリシー」の章を参照してください。

第 20 章：設定を簡略化する SAML 2.0 プロバイダ メタデータの使用

このセクションには、以下のトピックが含まれています。

[SAML 2.0 のメタデータ ツール \(P. 421\)](#)

[メタデータのエクスポート ツール \(P. 422\)](#)

[メタデータのインポート ツール \(P. 432\)](#)

SAML 2.0 のメタデータ ツール

ポリシー サーバでは、SAML 2.0 メタデータのインポートおよびエクスポートをプログラム上で行うためのメタデータ ツールを提供しています。メタデータを使用することで、SiteMinder を使用するサイトと、サードパーティまたは SiteMinder を使用するパートナーとの間のフェデレーション設定の交換を効率的に行えます。SAML 2.0 メタデータをプログラム上で使用することにより、実行する設定の規模を抑制できます。

メタデータ ツールは、smfedexport と smfedimport という 2 つのコマンドラインユーティリティで構成されます。

エクスポートされるメタデータには、以下のような入力タイプがあります。

- ユーザ入力
- KeyInfo をメタデータに含むための証明書データ ストアへのアクセス
- 署名のための証明書データ ストアへのアクセス
- テンプレートとして使用可能な類似のメタデータを参照するためのポリシー ストアへのアクセス

インポートされるメタデータには、以下のようなタイプがあります。

- ユーザ入力
- ポリシー ストアへのアクセス
- 証明書が設定されている場合に署名を検証するための証明書データ ストアへのアクセス
- メタデータ ドキュメント内の XML メタデータの解析
- 関連するメタデータのポリシー ストアへの保存
- メタデータからの PKI 情報の証明書データ ストアへの保存

メタデータのエクスポート ツール

以下の状況には、エクスポート ツールを使用できます。

- サービス プロバイダで使用するアイデンティティ プロバイダ メタデータ ファイルを作成する。

ツールを使用して、アイデンティティ プロバイダがサポートするプロファイルに関する情報が含まれるメタデータ ファイルを作成します。この XML 出力は、アイデンティティ プロバイダについての記述であり、エクスポート ツールによって生成されます。 サービス プロバイダとして機能するサイトでは、このメタデータ ファイルをインポートして、アイデンティティ プロバイダとのリレーションシップを確立できます。

- 既存のサービス プロバイダからアイデンティティ プロバイダ メタデータ ファイルを作成する。

SiteMinder アイデンティティ プロバイダは、既存の サービス プロバイダ オブジェクトからメタデータ ファイルを生成します。 サービス プロバイダ オブジェクトを使用することにより、ユーザ設定の必要な必須データの量が削減されます。 アイデンティティ プロバイダ メタデータ ファイルの設定の多くは、既存のサービス プロバイダから得られます。 また、SiteMinder は、サーブレットのデフォルトの名前を提供します。

メタデータ ファイルを使用するには、アイデンティティ プロバイダとサービス プロバイダ間の既存のリレーションシップが、確立しようとしているリレーションシップと同様であることが必要です。

SSO および SLO のサーブレット URL は、フェデレーション Web サービス アプリケーションの IP アドレスおよびポートが先頭に付けられたデフォルト サーブレット名です。

サーブレット名は以下のとおりです。

- `http://idp_server:port/affwebservices/public/saml2sso`
- `http://idp_server:port/affwebservices/public/saml2slo`

`idp_server:port`

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する Web サーバおよびポートを識別します。

- アイデンティティ プロバイダで使用するサービス プロバイダ メタデータ ファイルを作成する。

SiteMinder サービス プロバイダでは、そのサポートするプロファイルに関する情報が含まれるメタデータ ファイルを作成するにより、アイデンティティ プロバイダとして機能するサイトとの連携を促進できます。アイデンティティ プロバイダは、そのメタデータ ファイルをインポートして、サービス プロバイダとのリレーションシップを確立できます。

- 既存の SAML 2.0 認証方式からサービス プロバイダ メタデータ ファイルを作成する。

SiteMinder サービス プロバイダは、既存の SAML 2.0 認証方式 オブジェクトからメタデータ ファイルを生成します。サービス プロバイダ オブジェクトを使用することにより、ユーザ設定の必要な必須データの量が削減されます。SP メタデータ ファイルの設定の多くは、既存の SAML 2.0 認証方式から得られます。SiteMinder は、サーブレットのデフォルトの名前を提供します。

メタデータ ファイルを使用するには、サービス プロバイダとアイデンティティ プロバイダ間の既存のリレーションシップが、確立しようとしているリレーションシップと同様であることが必要です。SSO および SLO のサーブレット URL は、フェデレーション Web サービス アプリケーションの IP アドレスおよびポートが先頭に付けられたデフォルト サーブレット名です。

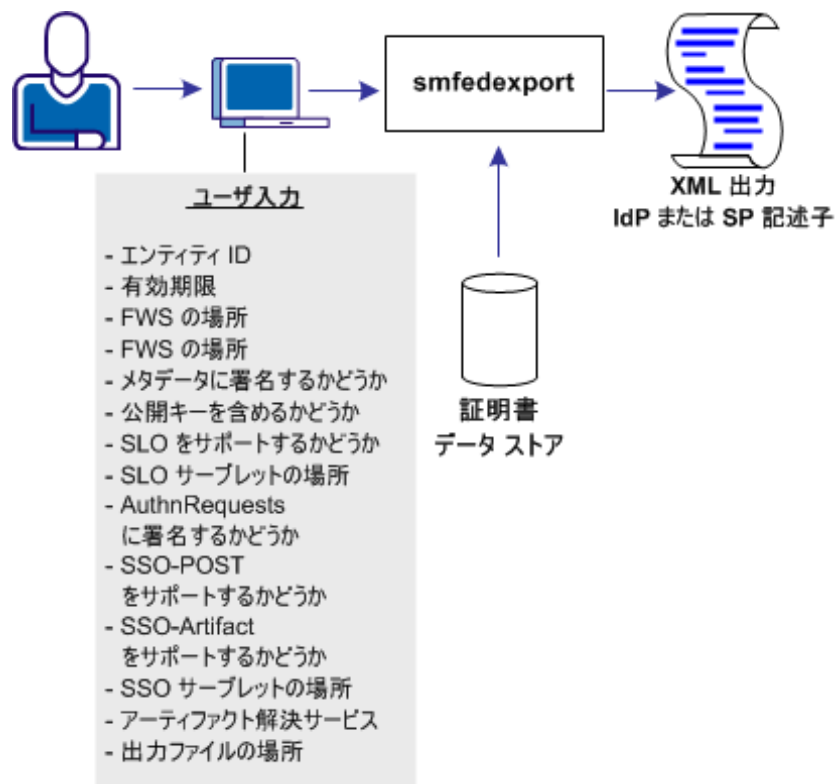
サーブレットは以下のとおりです。

- `http://idp_server:port/affwebservices/public/saml2sso`
- `http://idp_server:port/affwebservices/public/saml2slo`

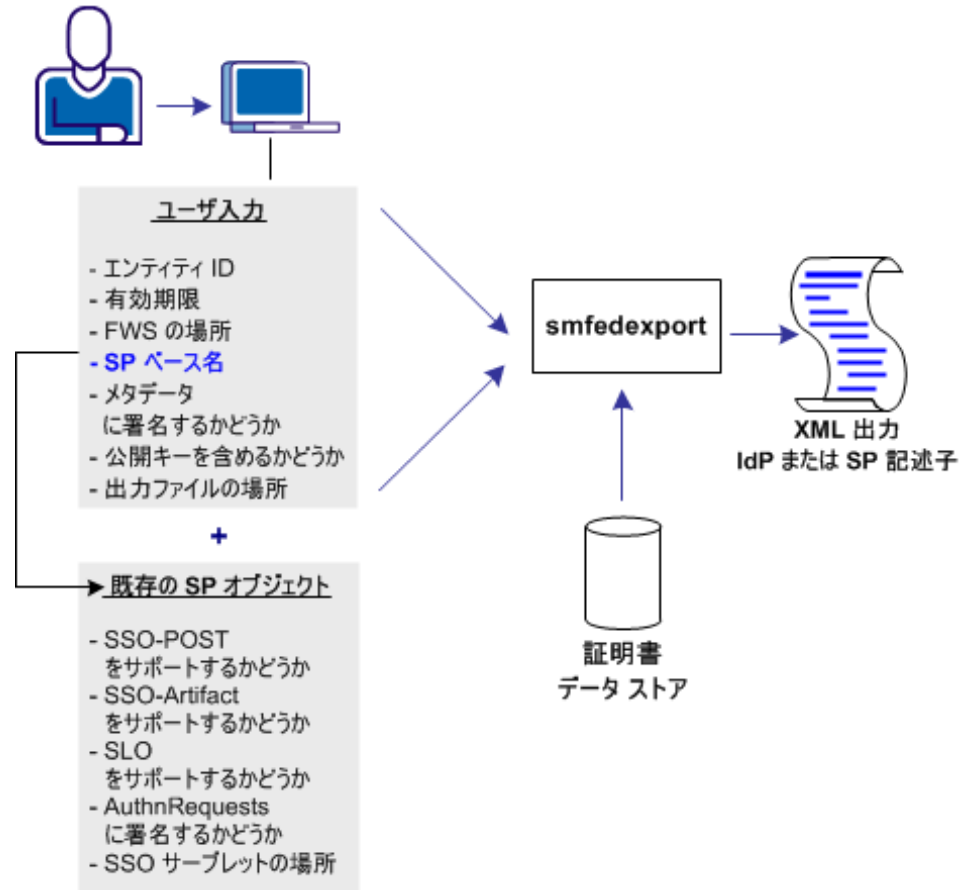
`idp_server:port`

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する Web サーバおよびポートを識別します。

次の図は、ユーザ入力のみから生成されるメタデータ ファイルを示しています。



次の図は、ユーザ入力および既存のサービス プロバイダ オブジェクトからのデータの組み合わせから生成されるメタデータ ファイルを示しています。



smfedexport ツールの実行

smfedexport ツールでは、SAML 2.0 メタデータを XML ファイルにエクスポートできます。

コマンド引数を指定せずに **smfedexport** を入力すると、すべてのコマンド引数およびそれらの用法が表示されます。

smfedexport ツールを実行する方法

1. ポリシー サーバ をインストールしたシステムで、コマンド ウィンドウを開きます。
2. 実行するタスクに応じた構文を使用して、**smfedexport** コマンドを入力します。

注: 角かっこ [] で囲まれたコマンド引数はオプションです。

SAML 2.0 アイデンティティ プロバイダ メタデータ ファイルをエクスポートする方法

```
smfedexport -type saml2idp [-entityid <entityid>] [-expiredays <num>]
[-fwsurl <FWS Location> [-spbase <spname>] -username <SiteMinder Admin Name>
-password <SiteMinder Admin Password>]][-sign][[-pubkey]
[-slo <SLO Service Location> -slobinding <REDIR>] [-reqsignauthr]
[-sso <SSO Service Location> -ssobinding <REDIR|SOAP>]
[-ars <Artifact Resolution Service Location>][[-output <file>]
```

SAML 2.0 サービス プロバイダ メタデータ ファイルをエクスポートするには、以下の手順に従います。

```
smfedexport -type saml2sp [-entityid <entityid>] [-expiredays <num>]
[-fwsurl <FWS Location> [-schemebase <Auth Scheme name>
-username <SiteMinder Admin Name> -password <SiteMinder Admin Password>]]
[-sign][[-pubkey][[-slo <SLO Service Location> -slobinding <REDIR>]
[-signauthr][[-acs <Assertion Consumer Service> -acsbinding <ART|POST|PAOS>
-acsindex <num>][[-acsisdef]]][[-output <file>]
```

既存の Metadata ドキュメントの署名するには、以下の手順に従います。

```
smfedexport -type (saml2sp|saml2idp) -sign -input <file> -output <file>
```

ツールを実行した後、XML ファイルが作成されます。 **-type** オプションが **saml2idp** に設定されている場合、デフォルトの出力ファイル名は **IDPSSODescriptor.xml** になります。 **-type** オプションが **saml2sp** に設定されている場合、デフォルトの出力ファイル名は **SPSSODescriptor.xml** になります。

smfedexport は、初期コマンド オプションを処理した後、生成しているエクスポート ファイルのタイプに関する追加のデータの入力を要求します。入力していないオプション引数には、すべてデフォルト値が使用されます。

注: IdP メタデータ ファイルを作成する場合は、**smfedexport** コマンドに少なくとも 1 つのシングル サインオン サービスを定義する必要があります。
SP メタデータ ファイルを作成する場合は、**smfedexport** コマンドに少なくとも 1 つのアサーション コンシューマ サービスを定義する必要があります。

smfedexport のコマンド オプション

smfedexport コマンド ライン オプションの一覧を以下の表に示します。

オプション	説明	値
-acs	アサーション コンシューマ サービス URL	URL
-acsindex	アサーション コンシューマ サービスのインデックス値	整数
-acsisdef	直前のアサーション コンシューマ サービスをデフォルトにします。	なし
-acsbinding	アサーション コンシューマ サービスの SAML プロトコル バインド。	■ ART (Artifact) ■ POST (POST) ■ PAOS (Reverse SOAP - ECP)
-ars	Artifact 解決サービス	URL
-decryptionkeyalias	証明書 (公開キー) をメタデータに含めるようにポリシー サーバに指示します。この証明書はメタデータを暗号化します。SP は対応する秘密キーを使用して、メタデータを復号化します。	エイリアス名

オプション	説明	値
-entityid	エクスポートしているメタデータを保有している SP または IDP の ID を表します。	URI
-expiredays	メタデータ ドキュメントが無効になるまでの日数。	整数 (0 がデフォルト) 値 0 は、メタデータ ドキュメントに期限がないことを示します。エクスポート済みの XML で「validUntil」要素は生成されません。
-fwsurl	FWS アプリケーションを指す URL です。	以下の形式の URL <code>http://host:port</code>
-input	既存の XML ファイルへの完全パス	文字列 (デフォルトなし)
-output	出力 XML ファイルへの完全パス	デフォルト値: IDPSSODescriptor.xml SPSSODescriptor.xml
-password	SiteMinder 管理者名	文字列 (デフォルトなし)
	-username オプションが必須	
-pubkey	証明書 (公開キー) をメタデータに含めるようにポリシー サーバに指示します。パートナー サイトは、署名の暗号化および検証に証明書を使用します。メタデータは署名の必要性がないため、この設定はオプションです。	true (該当する場合) false (その他の場合)
-reqsignauthr	署名された認証リクエストが必要	true (該当する場合) false (その他の場合)
-schemebase	既存のサービス プロバイダを指します。プロファイル/バインドの設定はこのプロバイダから取得されます。 以下のオプションが必須 -fwsurl -username -password	認証方式名

オプション	説明	値
-spbase	既存のサービス プロバイダを指します。プロファイル/バインドの設定はこのプロバイダから取得されます。 以下のオプションが必須 -fwsurl -username -password	サービス プロバイダ名
-sign	ポリシー サーバがメタデータを署名するかどうかを示します。この手順はオプションです。	true (該当する場合) false (その他の場合)
-sigalg	アサーションとアサーション レスポンス、シングル ログアウト リクエストとレスポンスの署名に署名ハッシュ アルゴリズムを使用するように SiteMinder に指定します。	rsawithsha1 rsawithsha256
-signauthr	SP が認証リクエストに署名するかどうかを示します	true (該当する場合) false (その他の場合)
-signingcertalias	メタデータに署名するキー/証明書ペアのエイリアスを指定します。ペアを証明書データ ストアに保存します。 この設定は、デフォルト エイリアスである defaultenterpriseprivatekey の代わりです。このオプションに値を入力しない場合、ポリシー サーバは、defaultenterpriseprivatekey エイリアスを使用してメタデータに署名します。	エイリアス名
-slo	シングル ログアウト サービス URL	URL
-slobinding	シングル ログアウトに使用される HTTP バインド。選択肢は HTTP リダイレクト バインドのみです。	
-sso	シングル サインオン サービス URL	URL

オプション	説明	値
-ssobinding	SSO サービス URL プロトコル バインド	■ REDIR (Web SSO) ■ SOAP (ECP)
-type (必須)	エクスポート ファイルのエンティティタイプ	saml2idp saml2sp
-username	SiteMinder 管理者名 (-password オプションが必須です)。	文字列 (デフォルトなし)

smfedexport ツールの例

例: アイデンティティプロバイダのエクスポート

```
smfedexport -type saml2idp -entityid http://www.myidp.com/idp1  
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com/  
/affwebservices/public/saml2slo -reqsignauthr  
-ssoart http://www.mysite.com/affwebservices/public/saml2sso  
-artressvc http://www.mysite.com/affwebservices/  
saml2artifactresolution -output myidpdescription.xml
```

例: サービスプロバイダのエクスポート

```
smfedexport -type saml2sp -entityid http://www.myidp.com/sp1  
-expiredays 30 -sign -pubkey -slohttpredir http://www.mysite.com/  
affwebservices/public/saml2slo -signauthr -aconsvcpost  
http://www.mysite.com/affwebservices/public/saml2assertionconsumer  
-aconsvcpostindex 12345 -output myidpdescription.xml
```

例: エクスポートされたデータファイルの変更および署名

この例では、`smfedexport` を使用して、XML ファイルの変更およびデジタル署名を行います。

メタデータ ファイルの変更および署名を行う方法

1. XML エディタを使用して、既存の XML ファイルを編集します。
2. 以下のコマンドを入力します。

```
smfedexport -sign -input file -output file
```

例 :

```
smfedexport -sign -input myspdescription.xml -output newspdescription.xml
```

すでにデジタル署名されたエクスポートファイルを変更する方法

1. 必要に応じて XML エディタを使用して、既存の XML ファイルを編集します。
2. ファイルから `<Signature>` 要素を削除します。
3. 以下のコマンドを入力します。

```
smfedexport -sign -input file -output file
```

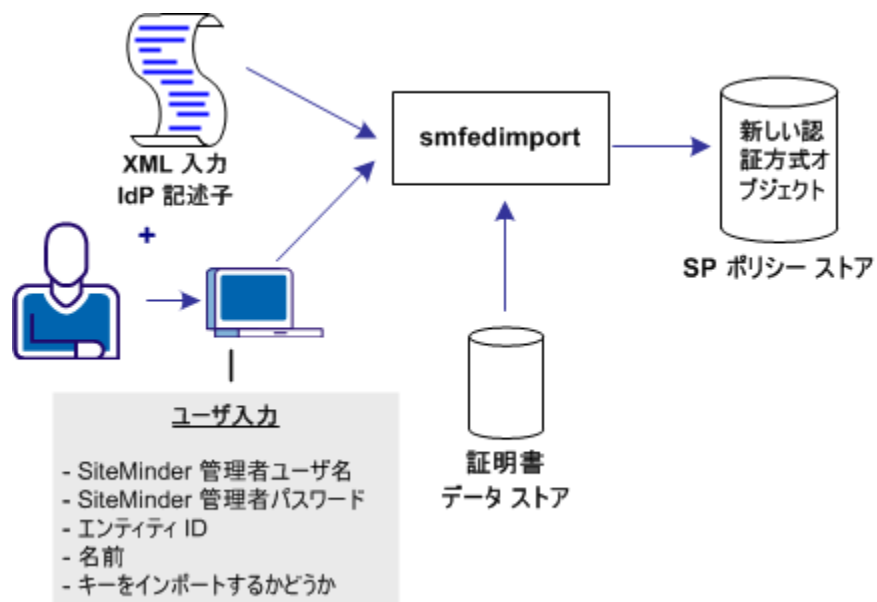
例 :

```
smfedexport -sign -input myspdescription.xml -output newspdescription.xml
```

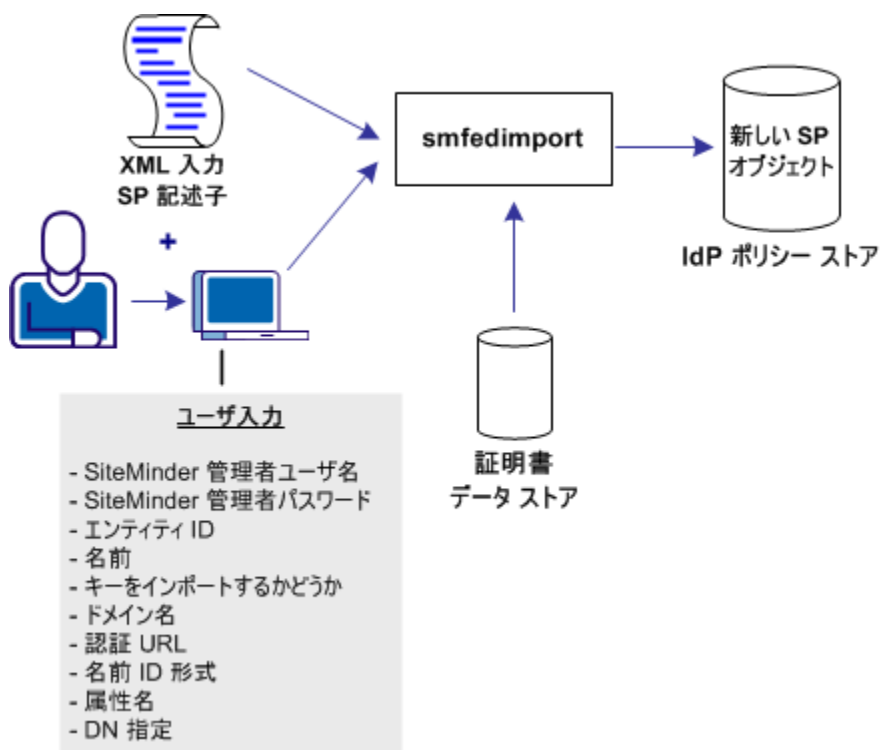
メタデータのインポート ツール

以下の状況には、インポート ツールを使用できます。

- 次の図に示すサービス プロバイダの SAML 2.0 認証方式を作成する。



- アイデンティティ プロバイダの SAML 2.0 サービス プロバイダ オブジェクトを作成する。



smfedimport ツールの実行

`smfedimport` ユーティリティは、アイデンティティ プロバイダおよびサービス プロバイダを SiteMinder ポリシー ストアおよび証明書データ ストアにインポートできます。サービス プロバイダ入力ファイルをインポートすると、その結果は、既存のアフィリエイト ドメイン内の新しい SiteMinder サービス プロバイダ オブジェクトになります。アイデンティティ プロバイダ入力ファイルをインポートすると、その結果は、SiteMinder SAML 2.0 テンプレートに基づく認証方式になります。

`smfedimport` コマンドライン ユーティリティが実行される場合、1 番目と 2 番目パラメータが SiteMinder 管理者のユーザ名およびパスワードです。3 番目と最後の引数は、入力 XML ファイルへのパスです。

smfedimport ツールを実行する方法

1. ポリシー サーバ をインストールしたシステムで、コマンド ウィンドウを開きます。
2. 以下の構文を使用して、コマンドを入力します。

SAML2 アイデンティティ プロバイダ メタデータ ファイルをポリシー ストアにインポートする方法

```
smfedimport -type saml2idp -username <username>  
-password <password> -entityid <entityid> -name <name>  
[-importkeys <name>] [-silent] -input <file>
```

サービス プロバイダ メタデータ ファイルをポリシー ストアにインポートするには、以下の手順に従います。

```
smfedimport -type saml2sp -username <username>  
-password <password> -entityid <entityid> -domainname <name>  
-authurl <URL> -nameidformat (U|E|X|W|K|N|P|T|U)  
-nameidtype (S | U | D) -attrname <name> -dnspec <spec>  
-name <name>[-importkeys <name>] [-silent] -input <file>
```

注: 角かっこ [] 内のスイッチはオプションです。

smfedimport は、初期コマンド オプションを処理した後、インポートしているファイルのタイプに基づいて追加のデータの入力を要求します。コマンドラインに入力していないオプション引数には、すべてデフォルト値が使用されます。

smfedimport ツールの例

例: アイデンティティ プロバイダ メタデータのインポート

```
smfedimport -type saml2idp -username Siteminder  
-password siteminderpassword -entityid http://www.myidp.com  
-name mynewauthscheme -importkeys keyaliasname -input mypartnersidpinfo.xml
```

例: サービス プロバイダ メタデータのインポート

```
smfedimport -type saml2sp -username Siteminder -password siteminderpassword  
-entityid http://www.mysp.com -name mynewsaml2sp -importkeys  
keyalisname -domainname myaffiliatedomain  
-authurl http://www.mysite.com/login.html -nameidformat U  
-nameidtype S -attrname attrname -input mypartnersspsinfo.xml
```

smfedimport のコマンド オプション

コマンドライン オプションの一覧を以下の表に示します。

オプション	説明	値
-attrname	nameID の属性名	文字列
-authurl	認証 URL	URL
-dnspec	名前 ID タイプのみに関する DN 指定	文字列
-domainname	アフィリエイト ドメイン名	文字列
-entityid	エンティティ ID	インポート用サービス プロバイダ ID またはインポート用アイデンティティ プロバイダ ID
-importkeys	メタデータ内の証明書が証明書データ ストアにインポートされるかどうかを示します。	文字列 証明書データ ストア内の証明書のエイリアスになる名前を入力します。複数の証明書が存在する場合、エイリアスは name 、 name1 、 name2 として追加されます。
-input	入力ファイル	文字列
-name	SiteMinder オブジェクトの名前を示します (サービス プロバイダ名、SAML 認証方式名など)。	文字列

オプション	説明	値
-nameidformat	名前 ID 形式	(U)nspecified (未設定) -- デフォルト (E)mail address (電子メール アドレス) (X)509 Subject name (X509 件名) (W)indows domain name (Windows ドメイン名) (K)erberos Principal Name (Kerberos プリンシパル名) E(n)tity Identifier (エンティティ識別子) (P)ersistent Identifier (永続的な識別番号) (T)ransient Identifier (一時的な識別番号)
-nameidtype	名前 ID タイプ	(S)tatic (静的) (U)ser attribute (ユーザ属性) (D)N attribute (DN 属性)
-password	SiteMinder 管理者パスワード	文字列 (デフォルトなし)
-type (必須)	インポート ファイルのエンティティ種類	saml2idp sam2sp

オプション	説明	値
-silent	ツールが対話形式でユーザとやり取りするかどうかが決まります。	true (該当する場合) false (その他の場合)
	このオプションを使用する場合、ツールはサイレントモードで作動します。ツールは、欠落している入力を対話形式でユーザに要求することはありません。また、ツールは、入力ファイル内の各エンティティの別個インポートを受け入れるようにユーザに促すこともありません。ツールは、入力ファイル内のすべてのエンティティがインポートされると仮定します。	
-username	SiteMinder 管理者名	文字列 (デフォルトなし)

複数の SAML 2.0 プロバイダを使用したインポート ファイルの処理

1 つのインポート ファイルに複数のプロバイダが指定されている場合、ツールはそれらのプロバイダを同じアフィリエイト ドメインにインポートします。各プロバイダの名前は、`smfedimport` コマンドの **-name** オプションに対して指定した値に基づきます。

たとえば、インポート ファイルに 3 つのサービス プロバイダがあり、次のように指定するとします。

```
-name mySP
```

ツールは、インポートされたプロバイダを `mysp`、`mysp_1`、`mysp_2` として登録します。整数は、以降のプロバイダごとに 1 ずつ増加します。インポート ファイルにアイデンティティ プロバイダとサービス プロバイダが混在している場合も、この命名規則は適用されます。

複数の証明書エイリアスを使用したインポート ファイルの処理

インポート ファイルに複数の証明書が含まれる場合、ツールはそれらを証明書データ ストアにインポートします。 ツールでは、その `smfedimport` コマンド オプション **-importkeys** を使用して指定された値からのエイリアス名を割り当てます。

たとえば、インポート ファイルに 3 つの証明書があり、次のように指定するとします。

```
-importkeys myalias
```

ツールは、インポートされた証明書を `myalias`、`myalias_1`、および `myalias_2` として登録します。 整数は、以降の証明書ごとに 1 ずつ増加します。

第 21 章: レガシー フェデレーション トレース 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。

[追跡ロギング \(P. 439\)](#)

[トレース ログの FWS キャッシュのクリア \(P. 440\)](#)

[Fed Client コンポーネントのログ メッセージ \(P. 441\)](#)

[Fed Server コンポーネントのログ メッセージ \(P. 443\)](#)

[ログ内の FWS データの更新 \(P. 445\)](#)

[トレース設定テンプレートによるロギングの簡略化 \(P. 446\)](#)

追跡ロギング

SiteMinder では、Web エージェント トレース ログ機能およびポリシー サーバ プロファイラを使用して、Web エージェントおよびポリシー サーバのパフォーマンスを監視できます。これらのロギングメカニズムは、SiteMinder プロセスに関する包括的な情報を提供するため、パフォーマンスの分析と問題のトラブルシューティングが可能になります。

レガシー フェデレーション では、フェデレーション通信に関するトレース メッセージを収集するためのロギング コンポーネントがいくつか利用可能です。トレース メッセージは、追跡、デバッグ、または両方を目的としてプログラム動作に関する詳細情報を提供します。トレース メッセージは一般的に、通常の動作時は無効にしておきます。トレース メッセージを有効にすることで、トレース メッセージ自体に加えて詳細な情報を抽出することができます。たとえば、FWSTrace.log を調べることによって、SiteMinder が生成する SAML アサーションを確認したり、現在のユーザの名前を収集したりできます。

収集されたトレース メッセージは、トレース ログに書き込まれます。FWSTrace.log は、web_agent_home/log ディレクトリにあります。

注: IIS 6.0 サーバ上で動作する Web エージェントの場合、最初のユーザ リクエストがサブミットされた後のみ、ログ ファイルが作成されます。ログ ファイル内で設定を確認するには、任意のユーザがリクエストをサブミットする必要があります。

Web エージェントおよびポリシー サーバでトレース ログを確立して、SiteMinder 動作を監視することができます。

トレース ログの FWS キャッシュのクリア

アサーション パーティまたは依存パーティでフェデレーション設定を変更する場合には、変更がトレース ログに表示されるように、フェデレーション Web サービス キャッシュをクリアしてください。設定の変更とは、たとえば SAML バインドの有効化や無効化などです。

次の手順に従ってください:

1. 管理 UI にログインします。
2. [管理] - [ポリシー サーバ] - [キャッシュ管理] をクリックします。
3. [すべてのキャッシュ] セクションで [すべてクリア] をクリックします。
4. [閉じる] をクリックします。

以上で、すべてのキャッシュがクリアされました。

Fed_Client コンポーネントのログ メッセージ

Web エージェント オプション パックは、フェデレーション Web サービス (FWS) アプリケーションをインストールします。FWS アプリケーションは、フェデレーション クライアントを表します。トレース メッセージを制御し、FWS アクティビティを監視するコンポーネントは、Fed_Client コンポーネントです。

FWS は、Web エージェントがトレース メッセージをログ記録するために使用する共通のトレース機能を使用します。以下のファイルがトレース ログをセットアップします。

トレース設定ファイル

FWS が監視するコンポーネントおよびイベントの内容を判別する設定 ファイルを指定します。デフォルトのファイルは、`fwstrace.conf` です。

トレース ログ ファイル

すべてのログ記録されたメッセージの出力ファイルを指定します。このファイルの名前と場所を Web エージェント設定ファイルに入力してください。

Web エージェント設定ファイルまたはエージェント設定オブジェクト

ロギングを有効化し、ログ形式を設定するロギング パラメータが含まれています。このファイルは、メッセージ コンテンツを定義するものではありません。

Fed_Client コンポーネントには、以下のサブコンポーネントが含まれます。

シングル サインオン

シングル サインオンの動作を監視します。

シングル ログアウト

シングル ログアウトのリクエストを監視します。

ディスカバリ プロファイル

アイデンティティ プロバイダ ディスカバリ プロファイルの動作を監視します。

管理

管理関連のメッセージを監視します。

リクエスト

リクエストおよび認証の動作を監視します。

一般

他のサブコンポーネントが監視していない動作を監視します。

設定

SAML 2.0 サービス プロバイダ設定メッセージを監視します。

トレース ロギングの設定

フェデレーション Web サービス アプリケーションのトレース メッセージを収集するには、FWS トレース ロギングを設定します。

次の手順に従ってください:

1. 以下のタスクのいずれかを実行します。
 - デフォルトテンプレート (FWSTrace.conf) のコピーを作成し、監視するデータのみが含まれるようにファイルを変更する。
 - 事前設定済みのテンプレートの 1 つをコピーし、これに新しい名前を割り当てる。

注: テンプレートを直接編集しないでください。
2. `web_agent_home/affwebservices/WEB-INF/classes` ディレクトリの `LoggerConfig.properties` ファイルを開き、以下のパラメータを設定します。
 - `TracingOn` を `Yes` に設定します。このオプションにより、メッセージをファイルに書き込むようにトレース機能に指定されます。
 - `TraceFileName` パラメータにトレース ログ ファイルの完全パスを設定します。デフォルトの場所は、`web_agent_home/config/FWSTrace.log` 内です。
 - `TraceConfigFile` パラメータにトレース設定ファイル (デフォルトテンプレートの `FWSTrace.conf` または別のテンプレートのいずれか) の完全パスを設定します。テンプレートは `web_agent_home/config` にあります。

3. オプションで、トレース ログ ファイル (ログ出力が含まれるファイル) の形式を指定できます。以下のパラメータは、トレース ログ ファイルの形式を指定する Web エージェント設定パラメータです。

- TraceRollover
- TraceSize
- TraceCount
- TraceFormat
- TraceDelim

LoggerConfig.properties ファイルには、これらすべての設定の説明が含まれています。

Fed_Server コンポーネントのログ メッセージ

ポリシー サーバにおいてフェデレーション サービスに関するトレース メッセージを制御するコンポーネントは、Fed_Server コンポーネントです。このコンポーネントは、アサーション ジェネレータおよび SAML 認証方式の動作を監視します。たとえば、smtracedefault.log ファイルに生成されたアサーションを見ることができます。

ポリシー サーバでロギングを設定するには、ポリシー サーバ プロファイラを使用します。プロファイラには、トレース ログ コンポーネントを指定できます。これには以下が含まれます。

プロファイラの使用に関する詳細については、「SiteMinder ポリシー サーバ管理ガイド」を参照してください。

フェデレーション サービス トレース ログ (smtracedefault.log)

プロファイラは、ログを記録するためのポリシー サーバ機能です。プロファイラを使用してフェデレーション サービスのトレース メッセージを収集し、smtracedefault.log ファイルに書き込むことができます。

ポリシー サーバにおいてフェデレーション サービスに関するトレース メッセージを制御するコンポーネントは、Fed_Server コンポーネントです。

ポリシー サーバ プロファイラを使用すると、ポリシー サーバの内部診断と処理機能をトレースできます。

次の手順に従ってください:

1. ポリシーサーバ管理コンソールを起動します。

重要: Windows Server 2008 上でこのグラフィカル ユーザ インターフェースにアクセスする場合は、管理者権限でショートカットを開きます。管理者としてシステムにログインしている場合でも、管理者権限を使用します。詳細については、お使いの SiteMinder コンポーネントのリリース ノートを参照してください。

2. [プロファイラ] タブをクリックします。
3. [プロファイリングの有効化] オプションを設定して、プロファイリングを有効にします。
4. プロファイラの設定を選択するには、以下のいずれかを実行します。
 - [設定ファイル] ドロップダウン リストに示されるデフォルトの `smtracedefault.txt` ファイルによって指定されるプロファイラ設定を受け入れます。
 - この管理セッションですでに選択されている別の設定ファイルを [設定ファイル] ドロップダウン リストから選択します。
 - [参照] ボタンをクリックして、別の設定ファイルを選択します。
5. プロファイラの設定ファイルに格納されているプロファイラ設定を変更し、その変更内容を同じファイルまたは新しいファイルに保存するには、[環境設定] ボタンをクリックして [ポリシー サーバ プロファイラ] ダイアログ ボックスを開きます。
6. [出力] グループ ボックスに示されている設定を調整して、ポリシー サーバ プロファイラによって生成される情報の出力形式を指定します。
7. [適用] をクリックして、変更内容を保存します。

注:

プロファイラ設定に対する変更は自動的に有効になります。ただし、ポリシー サーバを再起動すると、新しい出力ファイル (プロファイラでファイル出力が設定されている場合) が作成されます。既存のプロファイラ出力ファイルは、バージョン番号と共に自動的に保存されます。例:

`smtracedefault.log.1`

ロギング機能またはトレース機能の設定に対する変更がプロファイラ出力ファイルに関係がない場合（Windows でのコンソール ロギングの有効化または無効化など）、既存のファイルには新しい出力が追加され、そのバージョンは保存されません。

ポリシー サーバはデフォルトで、最大 10 個の出力ファイルを保持します（現在のファイルと 9 個のバックアップ ファイル）。10 個のファイル制限を超えると、古いファイルは新しいファイルに自動的に置き換えられます。保持するファイルの数を変更するには、TraceFilesToKeep DWORD レジストリ設定で希望する 10 進数を指定します。TraceFilesToKeep レジストリ設定は、以下の場所で作成される必要があります。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\
LogConfig\TraceFilesToKeep
```

「プロファイラ」タブには「バッファトレース」オプションがあります。このオプションは、デフォルトでポリシー サーバのパフォーマンスを向上させるように設定されています。。このオプションは、Solaris システムのみにあります。

ログ内の FWS データの更新

フェデレーション設定を一部でも変更する場合には、変更がトレース ログに表示されるように、フェデレーション Web サービス キャッシュをクリアしてください。

注: 変更が実施され、フェデレーション Web サービスが情報を受信するまでに、わずかな遅延が生じることがあります。

次の手順に従ってください:

1. 管理 UI にログインします。
2. 「管理」 - 「ポリシー サーバ」 - 「キャッシュ管理」をクリックします。
3. ページの「すべてのキャッシュ」セクションで「すべてクリア」をクリックします。
4. 「閉じる」をクリックします。

トレース設定テンプレートによるログギングの簡略化

トレース データの収集タスクを簡単にするために、一連の事前設定済みのテンプレートがポリシー サーバおよび **Web** エージェント オプション パックと共にインストールされます。独自のトレース設定ファイルを作成する代わりにこれらのテンプレートを使用して、トレース ログに書き込まれるデータを収集することができます。

FWS のトレース ログギング テンプレート

以下のテンプレートがフェデレーション **Web** サービスで利用可能です。

テンプレート	収集されるトレース メッセージ
WebAgentTrace.conf	デフォルトのテンプレートです。 ユーザ指定のデータを収集します。
FWS_SSOTrace.conf	シングル サインオン メッセージを収集します
FWS_SLOTTrace.conf	シングル ログアウト メッセージを収集します
FWS_IPDTrace.conf	アイデンティティ プロバイダ ディスカバリ プロファイル メッセージの収集

これらすべてのテンプレートには、追跡される特定データに対応した **Fed_Client** コンポーネントおよびサブコンポーネントが含まれます。 詳細な内容については、それぞれのテンプレートをご覧ください。 これらのテンプレートは、**web_agent_home/config** にあります。

トレース ログギングのテンプレートを使用する方法

1. 使用するテンプレートのコピーを作成し、コピーの名前を変更します。
注: テンプレートを直接編集しないでください。
2. エージェント設定ファイルまたはエージェント設定オブジェクトを開きます。
3. **TraceFile** パラメータを **Yes** に設定します。
4. **TraceFileName** パラメータにトレース ログ ファイルの完全パスを設定します。 このファイルにはログ出力が含まれます。

5. `TraceConfigFile` パラメータに、新しく名前を付けたテンプレート ファイルの完全パスを設定します。
6. トレース ログ ファイルの形式を設定します。以下のパラメータは、トレース ログ ファイルの形式を指定する **Web** エージェント設定パラメータです。

- `TraceAppend`
- `TraceFormat`
- `TraceDelimiter`
- `TraceFileSize`
- `LogLocalTime`

それぞれのログギングパラメータの説明については、「**Web** エージェント設定ガイド」を参照してください。

注: IIS 6.0 および Apache 2.0 サーバ上の **Web** エージェントは、エージェント設定ファイルでローカルに設定されるログパラメータの動的設定をサポートしません。したがって、パラメータを変更した場合、エージェントを再起動するまでは、その変更結果は有効になりません。エージェント設定オブジェクト内でログパラメータを設定する場合には、これらのログ設定の動的な保存および更新ができます。

FWS テンプレートのサンプル

以下のテキストは、`FWS_SLOTTrace.conf` テンプレートからの抜粋です。ほとんどのファイルにはコメントが含まれるほか、ファイル、コマンド構文、および **Fed_Client** コンポーネントに対して利用可能なサブコンポーネントの使用方法についての指示が含まれています。

この抜粋では、**Fed_Client** コンポーネントおよび監視されるサブコンポーネント (**Single_Logout** および **Configuration**) が示されています。また、各メッセージの必要なコンテンツ (日付、時刻、**Pid**、**Tid**、**TransactionId**、**SrcFile**、機能、メッセージ) を示すデータ フィールドも示されています。

```
components: Fed_Client/Single_Logout, Fed_Client/Configuration
data: Date, Time, Pid, Tid, TransactionID, SrcFile, Function, Message
```

IdP および SP に関するトレース ロギング テンプレート

トレース データの収集タスクを簡単にするために、事前設定済みの一連のテンプレートがポリシー サーバと共にインストールされます。独自のトレース設定ファイルを作成する代わりにこれらのテンプレートを使用して、トレース ログに書き込まれるデータを収集することができます。

アサーション生成や SAML 認証など、アイデンティティ プロバイダおよびサービス プロバイダに関連したトレース ロギングに、以下のテンプレートが利用可能です。

テンプレート	収集されるトレース メッセージ
samlidp_trace.template	アイデンティティ プロバイダのアクティビティに関するメッセージを収集します
samlsp_trace.template	サービス プロバイダのアクティビティに関するメッセージを収集します

詳細な内容については、それぞれのテンプレートをご覧ください。これらのテンプレートは、`siteminder_home/config/profiler_templates`にあります。

サービス プロバイダに関するテンプレートのサンプル

以下のテキストは、samlsp_trace.template ファイルです。

```
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/Auth_Scheme,
Fed_Server/Configuration
data: Date, Time, Tid, TransactionID, SrcFile, Function, Domain, Resource, Action,
User, Message
```

レガシー フェデレーションの場合、ファイルには Fed_Server コンポーネントと共にサブコンポーネント Auth_Scheme および Configuration が含まれます。

データ フィールドは、各メッセージの必要なコンテンツを示すものであり、以下の内容です。

日付、時刻、Tid、TransactionId、SrcFile、機能、ドメイン、リソース、アクションユーザ、メッセージ。

アイデンティティ プロバイダ プロファイラのサンプル

アイデンティティ プロバイダでは、ポリシー サーバ管理コンソールの [プロファイラ] タブにより、[設定ファイル] フィールドのテンプレートが指定されています。以下のテキストは、[構成ファイル] フィールドのサンプル エントリです。

```
c:¥program
files¥ca¥siteminder¥config¥profile_templates¥samlidp_template.trace
```

プロファイラの使用に関する詳細については、「ポリシー サーバ管理ガイド」を参照してください。

第 22 章：同じ値を使用する必要がある設定

このセクションには、以下のトピックが含まれています。

[設定テーブルを使用する方法](#) (P. 451)

[SAML 1.x の一致設定](#) (P. 452)

[SAML 2.0 設定項目の一致設定](#) (P. 454)

[WS フェデレーションの設定](#) (P. 455)

設定テーブルを使用する方法

フェデレーション環境を設定する際、トランザクションの両側で一致するパラメータ値の設定が必要なインスタンスが多数存在します。

明示的に後続するテーブルでは、パラメータの一致するセットのそれぞれが示されます。1 行に含まれる各セルは、その行の他のセルの対応する値に一致する必要がある設定を示します。

注：この情報は、アサーティング パーティと依存パーティが SiteMinder システムである環境のみに適用されます。

SAML 1.x の一致設定

以下の表は、SAML 1.x プロデューサとコンシューマで同じ値に設定する必要がある SiteMinder 設定を示しています。表には、これらの設定があるダイアログボックスまたはファイルも示されています。これらの設定のほとんどは管理 UI にありますが、いくつかのパラメータはプロパティファイルやリンク中に含まれています。

- 左側の列は、コンシューマで管理 UI に設定する必要がある項目を示します。
- 右側の列は、プロデューサで管理 UI に設定する必要がある項目、コンシューマの設定と一致する必要がある項目を示します。

重要: ユーザが URL を入力する必要がある場合、コロンの後の URL 文字列は大文字と小文字が区別されます。たとえば、**http:** の後のテキストは大文字と小文字が区別されます。そのため、すべてのオーディエンス関連の設定の URL と、アサーション コンシューマ URL 関連設定の URL で大文字と小文字が一致する必要があります。

SAML 1.x コンシューマ設定項目	一致する必要がある SAML 1.x プロデューサ設定項目
アフィリエイト名	[名前] フィールド
認証方式ページの [方式のセットアップ] セクション (Artifact プロファイルおよび POST プロファイル)	アフィリエイトオブジェクトの一般設定 値は小文字であることが必要 プロデューサのサイト間転送 URL リンク内の NAME クエリ パラメータ
[パスワード] フィールド (SAML Artifact 認証 方式のみ)	[パスワード] および [パスワードの確認] フィールド
認証方式ページの [方式のセットアップ] セクション	アフィリエイトオブジェクトの一般設定
[オーディエンス] フィールド	[オーディエンス] フィールド
他の SAML コンシューマすべて (認証方式ページの [方式のセットアップ] セクション)	アフィリエイトオブジェクトのアサーション設定

SAML 1.x コンシューマ設定項目	一致する必要がある SAML 1.x プロデューサ設定項目
アサーション コンシューマ URL (SAML Artifact 認証 方式のみ) 認証方式ページの [方式のセットアップ] セクション	アサーション コンシューマ URL アフィリエイト オブジェクトのアサーション 設定 SMCONSUMERURL クエリ パラメータ プロデューサのサイト間転送 URL リンク
[発行者] フィールド 認証方式ページの [方式のセットアップ] セクション	AssertionIssuerID パラメータ プロデューサの AMAssertionGenerator.properties ファイル
[SAML バージョン] ドロップダウン リストのバージョン [方式のセットアップ] セクション -- 認証方式ページ (SAML Artifact 認証 方式のみ)	[SAML バージョン] ドロップダウン リストのバージョン アフィリエイト オブジェクトのアサーション 設定
企業ソース ID [方式のセットアップ] セクション -- 認証方式ページ (SAML Artifact 認証 方式のみ)	SourceID パラメータ プロデューサの AMAssertionGenerator.properties ファイル

SAML 2.0 設定項目の一致設定

以下の表は、**SAML 2.0x** アイデンティティ プロバイダとサービス プロバイダで同じ値に設定する必要がある **SiteMinder** 設定を示しています。表には、これらの設定がある場所も示されています。これらの設定のほとんどは管理 UI にありますが、いくつかのパラメータはプロパティ ファイルやリンク中に含まれています。

- 左側の列は、サービス プロバイダで管理 UI に設定する必要がある項目を示します。
- 右側の列は、アイデンティティ プロバイダで管理 UI に設定する必要があり、サービス プロバイダの設定と一致する必要がある項目を示します。

重要: ユーザが **URL** を入力する必要がある場合、コロンの後の **URL** 文字列は大文字と小文字が区別されます。たとえば、**http:** の後のテキストは大文字と小文字が区別されます。そのため、**SP ID** および **IdP ID** 関連のすべての設定が一致する必要があります。

サービス プロバイダでの設定	一致する必要があるアイデンティティ プロバイダでの設定
属性名 SAML 2.0 認証方式の [属性] 設定から [追加/編集] ページ。	変数名 SAML サービス プロバイダ オブジェクトの [属性] 設定から [属性の追加] ページの [属性のセットアップ] セクション。
[オーディエンス] フィールド ■ 他のすべての SAML サービス プロバイダ ■ SAML 2.0 認証方式の [SSO] 設定。	[オーディエンス] フィールド SAML サービス プロバイダ オブジェクトの [SAML プロファイル] 設定の [SSO] セクション。
[IdP ID] フィールド SAML 2.0 認証方式の [一般] 設定	[IdP ID] フィールド ■ SAML サービス プロバイダ オブジェクトの [一般] 設定 ■ アイデンティティ プロバイダで開始された SSO の場合 -- 未承認応答の SPID クエリ パラメータ

サービス プロバイダでの設定	一致する必要があるアイデンティティ プロバイダでの設定
ローカル名 SAML 2.0 認証方式の [属性] 設定から [追加/編集] ページ。 ローカル名 SAML リクエスト (サービス Provider) でフェデレーション属性変数を作成するための [Federation Attribute Variable] ページ。	なし
[SP ID] フィールド ■ SAML 2.0 認証方式の [一般] 設定 ■ サービス プロバイダで開始された SSO の場合 -- アイデンティティ プロバイダへのハードコードされたリンク内の ProviderID クエリ パラメータ	[SP ID] フィールド SAML サービス プロバイダ オブジェクトの [一般] 設定
SP 名 SAML 2.0 認証方式の [暗号化 & 署名] 設定の [バックチャネル] セクション。 この値は、小文字で指定する必要があります。この値は、小文字で指定する必要があります。	[名前] フィールド SAML サービス プロバイダ オブジェクトの [一般] 設定

WS フェデレーションの設定

以下の表は、WS フェデレーション アカウント パートナーおよびリソース パートナーで同じ値に設定する必要がある SiteMinder 設定を示しています。表の説明は次のとおりです。

- 左側の列は、リソース パートナーで 管理 UI に設定する必要がある項目を示します。
- 右側の列は、アカウント パートナーで 管理 UI に設定する必要があり、コンシューマの設定と一致する必要がある項目を示します。

重要: ユーザが URL を入力する必要がある場合、コロンの後の URL 文字列は大文字と小文字が区別されます。たとえば、**http:** の後のテキストはすべて大文字と小文字が区別されます。そのため、RP ID および AP ID 関連のすべての設定が一致する必要があります。

リソース パートナーでの設定	一致する必要があるアカウント パートナーでの設定
リソース パートナー ID WS フェデレーション認証方式の [一般] 設定	リソース パートナー ID リソース パートナーオブジェクトの [一般] 設定 wtrealm クエリ パラメータは、アカウント パートナーで開始される SSO をトリガするハードコードリンクのリソース パートナー ID に設定される必要があります。
アカウント パートナー ID WS フェデレーション認証方式の [一般] 設定	アカウント パートナー ID リソース パートナーオブジェクトの [一般] 設定

第 23 章: SiteMinder によって使用される フェデレーション Web サービス URL

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 サービス URL](#) (P. 457)

[アサーティング パーティのサービスの URL](#) (P. 458)

[依存パーティでのサービスの URL](#) (P. 468)

[Web.xml ファイル](#) (P. 476)

フェデレーション サービス URL

フェデレーション Web サービスには、レガシー フェデレーション を実装するための多数のサービスが含まれています。管理 UI 経由でシングルサインオンやシングルログアウト、またはアイデンティティプロバイダ ディスカバリ プロファイルを設定する場合、さまざまなサービスを参照する URL の指定が要求されます。

以下のようなサービスの記述が必要です。

- メンテナンスの簡単な説明
- サービスの URL
- 管理 UI での URL を入力するフィールド
- 関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

Web.xml ファイルは、フェデレーション Web サービス アプリケーションの展開記述子の 1 つです。このファイルには、サーブレットおよび URL マッピングのリストが示されています。

アサーティング パーティのサービスの URL

以下のサービスがアサーティング パーティ（プロデューサ/アイデンティティ プロバイダ/アカウント パートナー）で提供されます。一方、依存パーティ（コンシューマ/サービス プロバイダ/リソース パートナー）ではサービス URL を入力します。

フェデレーション Web サービス アプリケーションでは、以下のサービスを提供します。

- [サイト間の転送サービス](#) (P. 458) (SAML 1.x プロデューサ)
- [アサーション検索サービス](#) (P. 459) (SAML 1.x プロデューサ)
- [Artifact 解決サービス](#) (P. 460) (SAML 2.0 IdP)
- [シングルサインオン サービス](#) (P. 462) (SAML 2.0 IdP)
- [シングル ログアウト](#) (P. 464) (SAML 2.0 IdP)
- [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 サービス](#) (P. 466) (SAML 2.0)
- [属性サービス](#) (P. 467) (SAML 2.0)
- [シングル サインオン サービス](#) (P. 463) (WS フェデレーション)
- [サインアウト サービス](#) (P. 465) (WS フェデレーション)
- [WSFedDispatcher サービス](#) (P. 468) (WS フェデレーション)

サイト間の転送サービス URL (SAML 1.x)

SAML 1.x POST および Artifact プロファイルの場合、サイト間転送 URL は、ユーザをプロデューサからコンシューマに転送するプロデューサ側コンポーネントです。

このサービスのデフォルト URL

`http://producer_server:port/affwebservices/public/intersitetransfer`

`producer_server:port`

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するプロデューサで、システムの Web サーバおよびポート番号を識別します。

サイト間転送 URL

プロデューサのページ上にハードコードされたリンク内の URL が含まれます。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>intersiteTransferService</servlet-name>
  <display-name>Intersite Transfer Service</display-name>
  <description>This servlet acts as the Intersite Transfer URL.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    IntersiteTransferService
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>intersiteTransferService</servlet-name>
  <url-pattern>/public/intersitetransfer/*</url-pattern>
</servlet-mapping>
```

アサーション検索サービス URL (SAML 1.x)

アサーション検索サービスは、SAML に関するアサーションを取得します。
1.x コンシューマ サイト。

このサービスのデフォルト URL

- Basic または Basic over SSL でこのサービスを保護する場合、URL は次のとおりです。
https://producer_server:port/affwebservices/assertionretriever
- クライアント証明書認証でこのサービスを保護する場合、URL は次のとおりです。
https://producer_server:port/affwebservices/certassertionretriever

producer_server:port

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するプロデューサで、システムの Web サーバおよびポート番号を識別します。

アサーション検索 URL

[アサーション取得 URL] フィールドで指定されます。このフィールドは、SAML 1.x 認証方式ページの [方式のセットアップ] セクションにあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>assertionretriever</servlet-name>
  <display-name>SAML Assertion Retrieval servlet</display-name>
  <description>This servlet processes the HTTP post based SAML requests and
returns the SAML Response elements. Both SAML Request and Response elements are
SOAP encoded.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    AssertionRetriever</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/assertionretriever/*</url-pattern>
</servlet-mapping>
<servlet-mapping>

  <servlet-name>assertionretriever</servlet-name>
  <url-pattern>/certassertionretriever/*</url-pattern>
</servlet-mapping>
```

Artifact 解決サービス URL (SAML 2.0)

Artifact 解決サービスは、サービス プロバイダの SAML 2.0 アサーションを取得します。

このサービスのデフォルト URL

- バージック認証でこのサービスを保護する場合、URL は次のとおりです。

`http://idp_server:port/affwebservices/saml2artifactresolution`

- Basic over SSL または X.509 クライアント証明書認証でこのサービスを保護する場合、URL は次のとおりです。

`https://idp_server:port/affwebservices/saml2certartifactresolution`

`idp_server:port`

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する Web サーバおよびポートを識別します。

解決サービス URL

[解決サービス]フィールドで指定されます。このフィールドは、SAML 2.0 認証方式の [SSO] 設定の [バインディング] セクションにあります。フィールドをアクティブにするには、HTTP-Artifact をバインドとしてを選択します。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2artifactresolution</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Artifact Resolution
    service at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.ArtifactResolution</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2artifactresolution/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
<servlet-name>saml2artifactresolution</servlet-name>
<url-pattern>/saml2certartifactresolution/*</url-pattern>
</servlet-mapping>
```

シングル サインオン サービス URL (SAML 2.0)

シングル サインオン サービスは、SAML 2.0 のシングル サインオンを実装します。

このサービスのデフォルト URL

`http://idp_server:port/affwebservices/public/saml2sso`

`idp_server:port`

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する Web サーバおよびポートを識別します。

SSO サービス URL

[SSO サービス] フィールドで指定されます。このフィールドは SAML 2.0 認証方式の [SSO] 設定にあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2sso</servlet-name>
  <display-name>SAML 2.0 Single Sign-On service</display-name>
  <description>This servlet is the SAML 2.0 Single Sign-On service at an
IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
saml2.SSO</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2sso</servlet-name>
  <url-pattern>/public/saml2sso/*</url-pattern>
</servlet-mapping>
```

シングル サインオン サービス URL (WS-Federation)

WS-Federation シングル サインオン サービスは、WS フェデレーションのシングル サインオンを実装します。

このサービスのデフォルト URL

`http://ap_server:port/affwebservices/public/wsfedsso`

ap_server:port

アカウント 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。

SSO サービス URL

[SSO サービス] フィールドで指定されます。このフィールドは、WS フェデレーション認証方式の [SSO] 設定にあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
<servlet-name>wsfedssso</servlet-name>
<display-name>WSFED Single Sign-On service</display-name>
<description>This servlet is the WSFED Single Sign-On service at an Account
Partner.</description>
<servlet-class>com.netegrity.affiliateminder.webservices.wsfed.SSO
</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>wsfedssso</servlet-name>
<url- パターン> /public/wsfedsso/*</url- パターン>
</servlet-mapping>
```

IdP のシングル ログアウト サービス URL (SAML 2.0)

このサービスは、SAML 2.0 のシングル ログアウトを実装します。

このサービスのデフォルト URL

`http://idp_server:port/affwebservices/public/saml2slo`

`idp_server:port`

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する Web サーバおよびポートを識別します。

SLO ロケーション URL/SLO レスポンス ロケーション URL

アイデンティティ プロバイダの同じ名前のフィールドで指定されています。これらのフィールドは、SAML サービス プロバイダ オブジェクトの [SAML プロファイル] 設定の [SLO] セクションにあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```


AP のサインアウト サービス URL (WS フェデレーション)

このサインアウト サービスは、WS フェデレーション サインアウト機能を実装します。

このサービスのデフォルト URL

`http://ap_server:port/affwebservices/public/wsfedsignout`

ap_server:port

アカウント 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。

サインアウト クリーンアップ URL/サインアウト確認 URL

アカウント パートナーの同じ名前のフィールドで指定されています。これらのフィールドは、リソース パートナー プロパティ オブジェクトの [SAML プロファイル] 設定の [サインアウト] セクションにあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
    at an AP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>wsfedsignout</servlet-name>
  <url-pattern>/public/wsfedsignout/*</url-pattern>
</servlet-mapping>
```

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 サービス URL (SAML 2.0)

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 サービスは、アイデンティティプロバイダ ディスカバリ機能を実装します。

このサービスのデフォルト URL

`https://idp_server:port/affwebservices/public/saml2ipd/*`

idp_server:port

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する Web サーバおよびポートを識別します。

サービス URL

[サービス URL] フィールドで指定されます。このフィールドは、アイデンティティプロバイダにおける SAML サービスプロバイダオブジェクトの [SAML プロファイル] 設定の [IPD] セクションにあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2ipd</servlet-name>
  <display-name>SAML 2.X Identity Provider Discovery Profile
    service</display-name>
  <description>This servlet is the SAML 2.X Identity Provider Discovery Profile
    service at an SP or IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.IPDService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2ipd</servlet-name>
  <url-pattern>/public/saml2ipd/*</url-pattern>
</servlet-mapping>
```

属性サービス URL (SAML 2.0)

属性機関では、属性サービスを使用して、SAML リクエストからの属性クエリに応答できます。

このサービスのデフォルト URL

`http://idp_server:port/affwebservices/saml2attributeservice`

`idp_server:port`

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する Web サーバおよびポートを識別します。

属性サービス URL

[属性サービス] フィールドで指定されます。このフィールドは、サービス プロバイダの SAML 2.0 認証方式の [属性] 設定にあります。

関連サブリットおよび Web.xml ファイルのサブリット マッピング

```
<servlet>
  <servlet-name>saml2attributeservice</servlet-name>
  <display-name>SAML 2.0 Attribute service</display-name>
  <description>This servlet is the SAML 2.0 Attribute Service
    at an IdP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.saml2.
    AttributeService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2attributeservice</servlet-name>
  <url-pattern>/saml2attributeservice/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>saml2attributeservice</servlet-name>
  <url-pattern>/saml2certattributeservice/*</url-pattern>
</servlet-mapping>
```

AP の WSFedDispatcher サービス URL

WSFedDispatcher サービスは、着信 WS フェデレーション メッセージをすべて受信し、クエリ パラメータ データに基づいて、リクエスト処理を他のサービスに転送します。

このサービスのデフォルト URL

`https://ap_server:port/affwebservices/public/wsfeddispatcher`

ap_server:port

アカウント 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。

URL

該当せず

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    dispatcher</servlet-class>
</servlet>

<<servlet-mapping>
  <servlet-name>wsfeddispatcher</servlet-name>
  <url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

依存パーティでのサービスの URL

依存パーティでは、以下のサービスを提供します。ただし、アサーティング パーティでのサービスの URL を入力する必要があります。

SiteMinder の依存パーティで提供されるサービスを以下に示します。

- [SAML 認証情報コレクタ \(SAML 1.x\)](#) (P. 469)
- [認証リクエスト サービス \(SAML 2.0\)](#) (P. 470)

- [アサーション コンシューマ サービス \(SAML 2.0\)](#) (P. 471)
- [セキュリティ トークン コンシューマ サービス \(WS フェデレーション\)](#) (P. 472)
- [シングログアウト サービス \(SAML 2.0\)](#) (P. 473)
- [サインアウト サービス \(WS フェデレーション\)](#) (P. 474)
- [WSFedDispatcher サービス \(WS フェデレーション\)](#) (P. 475)

SAML 認証情報コレクタ サービス URL (SAML 1.x)

SAML 認証情報コレクタ サービスは、SAML 1.x アサーションの消費を支援します。

このサービスのデフォルト URL

`https://consumer_server:port/affwebservices/public/samlcc`

`consumer_server:port`

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する Web サーバおよびポートを識別します。

アサーション コンシューマ URL

[アサーション コンシューマ URL] フィールドで指定されます。このフィールドは、SAML 1.x アフィリエイト オブジェクトの [アサーション] ページにあります。また、このフィールドは、コンシューマの SAML 1.x POST 認証方式の [方式のセットアップ] セクションにもあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>samlcredentialcollector</servlet-name>
  <display-name>SAML Credential Collector</display-name>
  <description>This servlet acts as the SAML Credential Collector.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    SAMLCredentialCollector</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>samlcredentialcollector</servlet-name>
  <url-pattern>/public/samlcc/*</url-pattern>
</servlet-mapping>
```

認証リクエスト サービス (SAML 2.0)

この認証リクエスト サービスは、Artifact または POST プロファイルでのシングルサインオンの実装に役立ちます。

このサービスのデフォルト URL

`https://sp_server:port/affwebservices/public/saml2authnrequest`

sp_server:port

サービス プロバイダで 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するサーバおよびポート番号を指定します。

サービスの URL

該当なし

認証リクエストは、サービス プロバイダにおけるアプリケーション内のリンクです。このリンクは、シングルサインオンを開始するものであり、アプリケーションにある必要があります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2authnrequest</servlet-name>
  <display-name>SAML 2.0 AuthnRequest service</display-name>
  <description>This servlet is the SAML 2.0 AuthnRequest service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.AuthnRequest</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2authnrequest</servlet-name>
  <url-pattern>/public/saml2authnrequest/*</url-pattern>
</servlet-mapping>
```

アサーション コンシューマ サービス URL (SAML 2.0)

アサーション コンシューマ サービスは、アサーションの消費を可能にします。

このサービスのデフォルト URL

`https://sp_server:port/affwebservices/public/saml2assertionconsumer`

`sp_server:port`

サービス プロバイダで **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するサーバおよびポート番号を指定します。

アサーション コンシューマ URL

[アサーション コンシューマ URL] フィールドで指定されます。このフィールドは、アイデンティティ プロバイダにおける **SAML** サービス プロバイダ オブジェクトの **SSO** 設定の一部です。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <display-name>SAML 2.0 Assertion Consumer service</display-name>
  <description>This servlet is the SAML 2.0 Assertion Consumer service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.AssertionConsumer</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2assertionconsumer</servlet-name>
  <url-pattern>/public/saml2assertionconsumer/*</url-pattern>
</servlet-mapping>
```

セキュリティトークン コンシューマ サービス URL (WS フェデレーション)

セキュリティ トークン コンシューマ サービスにより、リソース パートナーでのアサーションの消費が可能になります。

このサービスのデフォルト URL

`https://rp_server:port/affwebservices/public/wsfedsecuritytokenconsumer`

`rp_server:port`

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するリソース パートナーで Web サーバとポートを識別します。

セキュリティトークン サービス URL

[セキュリティ トークン コンシューマ サービス] フィールドで指定されます。このフィールドは、アカウント パートナーにおけるリソース パートナー オブジェクトの [SAML プロファイル] 設定の一部です。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>wsfedsecuritytokenconsumer</servlet-name>
  <display-name>Security Token Consumer service</display-name>
  <description>This servlet is the WS-Federation Security Token
    Consumer service at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SecurityTokenConsumer</servlet-class>
</servlet>

<<servlet-mapping>
  <servlet-name>wsfedsecuritytokenconsumer</servlet-name>
  <url-pattern>/public/wsfedsecuritytokenconsumer/*</url-pattern>
</servlet-mapping>
```


SP のシングル ログアウト サービス URL (SAML 2.0)

シングル ログアウト サービスは、SAML 2.0 のシングル ログアウトを実装します。

このサービスのデフォルト URL

`http://sp_server:port/affwebservices/public/saml2slo`

sp_server:port

サービス プロバイダで 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するサーバおよびポート番号を指定します。

SLO ロケーション URL/SLO レスポンス ロケーション URL

同じ名前のフィールドで指定されています。これらのフィールドは、サービス プロバイダで設定する SAML 2.0 認証方式の [SLO] 設定の一部です。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>saml2slo</servlet-name>
  <display-name>SAML 2.0 Single Logout service</display-name>
  <description>This servlet is the SAML 2.0 Single Logout service at an
SP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.
    saml2.SLOService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>saml2slo</servlet-name>
  <url-pattern>/public/saml2slo/*</url-pattern>
</servlet-mapping>
```

RP のサインアウト サービス URL (WS フェデレーション)

サインアウト サービスは、WS フェデレーションのサインアウト機能を実装します。

このサービスのデフォルト URL

`http://rp_server:port/affwebservices/public/wsfedsignout`

`rp_server:port`

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するリソース パートナーで Web サーバとポートを識別します。

サインアウト クリーンアップ URL/サインアウト URL

同じ名前のフィールドで指定されています。これらのフィールドは、リソース パートナーのフェデレーション認証方法の [サインアウト] セクションにあります。

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>wsfedsignout</servlet-name>
  <display-name>WS-Federation Signout Service</display-name>
  <description>This servlet is the WS-Federation Signout service
    at an RP.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    SignoutService</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>wsfedsignout</servlet-name>
  <url-pattern>/public/wsfedsignout/*</url-pattern>
</servlet-mapping>
```

RP の WSFedDispatcher サービス URL

WSFedDispatcher サービスは、着信するすべての WS フェデレーションメッセージを受け取ります。そして、このサービスは、クエリ パラメータ データに基づいて、リクエスト処理を他のサービスに転送します。

このサービスのデフォルト URL

`https://rp_server:port/affwebservices/public/wsfeddispatcher`

`rp_server:port`

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するリソース パートナーで Web サーバとポートを識別します。

サービスの URL

該当なし

関連サーブレットおよび Web.xml ファイルのサーブレット マッピング

```
<servlet>
  <servlet-name>wsfeddispatcher</servlet-name>
  <display-name>WS-Federation Dispatcher service</display-name>
  <description>This servlet is the WS-Federation Dispatcher service for all
WS-Federation services.</description>
  <servlet-class>com.netegrity.affiliateminder.webservices.wsfed.
    dispatcher</servlet-class>
</servlet>

<<servlet-mapping>
  <servlet-name>wsfeddispatcher</servlet-name>
  <url-pattern>/public/wsfeddispatcher/*</url-pattern>
</servlet-mapping>
```

Web.xml ファイル

Web.xml ファイルには、フェデレーション Web サービス アプリケーションに関するサーブレットと URL のマッピングが一覧表示されています。

このファイルの大部分は変更できませんが、URL マッピングの変更はできます。

Web.xml ファイルを表示するには、以下の該当するファイルの場所に移動してください。

- `web_agent_home/affwebservices/WEB-INF`
- `sps_home/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF`

第 24 章: レガシー フェデレーションのトラブルシューティング

このセクションには、以下のトピックが含まれています。

[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 \(P. 477\)](#)

[一般的な考慮事項 \(P. 479\)](#)

[SAML 1.x のみの問題 \(P. 484\)](#)

[SAML 2.0x のみの問題 \(P. 486\)](#)

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-bbbede33-fe78e2dd-38d][SSO.java][processAuthentication][SAMLTransactionID 2aaf90ec-fdef4897-0ef49d91-63d4031d-f508a3e9-12 maps to TransactionID: 1c2d7650-b006e46a-ed071f41-bbbede33-fe78e2dd-38d.]
```

CA SiteMinder® Federation システムは、アサーティング パーティとして機能している場合にのみ、新しい SAMLTransactionID を生成します。該当する場合は以下のとおりです。

- フェデレーション Web サービスがセッションを確立するために認証 URL にブラウザをリダイレクトする場合。
- 以下の HTTP-Artifact シングル サインオン トランザクションの場合。
 - アサーティング パーティが依存パーティに Artifact を送信するとき。
 - アサーティング パーティがアーティファクトを解決するとき。
- ユーザが Identity Discovery プロファイル URL にリダイレクトされる場合。
- アサーティング パーティのシングル ログアウト中。

依存パーティでは、ログ ファイルによって簡単にトレース可能なリクエスト ID が存在します。リクエスト ID があれば、CA SiteMinder® Federation システムは、依存パーティで SAMLTransactionID を生成する必要がありません。

一意の SAML トランザクション ID ごとに、複数のトランザクション ID を生成できます。新しい HTTP トランザクションが発生すると、新しいトランザクション ID が生成されます。このトランザクション ID は、単一の SAML トランザクション ID にマップされます。たとえば、トレース ログで以下のエントリを参照できます。

```
SamlTransactionID ["xyz"] maps to TransationID["123"]  
["123"] HTTP operation  
["123"] HTTP operation
```

新しいトランザクション ID "456" が生成されます。

```
SamlTransactionID["xyz"] Maps to Transactionid["456"]  
["456"] <some operation>  
["456"] <some operation>
```

トランザクション ID は fwstrace.log および smtracedefault.log に記録されます。1 つのトランザクションに対するトランザクション ID の同じセットは、これらのログのそれぞれに書き込まれます。これらのログ内の ID を使用してトランザクションを追跡できるようになります。失敗した場合は、ID を参照すると、トランザクションに対してどのイベントが失敗したのかを判断するのに役立ちます。

一般的な考慮事項

以下のトラブルシューティング トピックは **SAML 1.x** および **SAML 2.0** に適用します。

無効な smjavaagent.dll のために Web エージェント オプション パックを初期化できない

症状:

Web エージェント オプション パックは、他の CA 製品がインストールされたシステムでは初期化できない。「Java Agent API 初期化に失敗しました」(Java Agent API initialization FAILED) や「リンク エラー」(unsatisfied link error) などのエラー メッセージが表示される。

以下のようなエラー メッセージがフェデレーション Web サービスのログ ファイルに記録される。

```
11:04:46[29959477:E] Exception while reading the WebAgent configuration information:
javaagent_api_getConfig
11:04:46[29959477:E] Java Agent API initialization FAILED.
```

解決方法:

無効なバージョンの smjavaagentapi.dll がシステム パスに存在する可能性があります。 インストール済みのすべての製品が相互に互換性があり、互換性のあるバージョンであることを確認してください。

バージョンを確認する方法

1. [テクニカル サポート サイト](#)にログインします。
2. 12.52 SP1 に関するプラットフォーム サポート マトリックスを検索します。

cookie ドメイン不一致エラー

症状:

コンシューマ/SP サイトでの SAML 認証が完了した後も、Cookie ドメイン不一致のために、コンシューマ または SP Web エージェントが認証情報の入力をユーザに要求する。

解決方法:

プロデューサ IdP およびコンシューマ/SP が同じ Cookie ドメインに存在していないことを確認します。レガシー フェデレーションは、同じ Cookie ドメイン内の連携をサポートしていません。プロデューサ/IdP サイトおよびコンシューマ/SP サイトでは、個別の Cookie ドメインが必須です。さらに、CookieDomainScope パラメータが実際の環境に該当する値に設定されていることを確認してください。このパラメータは、Web エージェント パラメータです（「*SiteMinderWeb* エージェント設定ガイド」のシングルサインオンに関する情報を参照）。

個別の Cookie ドメインが使用されている場合には、エージェント設定内の Cookie ドメインが、リクエストされたターゲット URL 内のドメイン名に一致することを確認します。

コンシューマ/SP での認証完了後のエラー

症状:

コンシューマ サイトにおいて認証が完了した後、HTTP 404 「ページが見つかりません」（Page Not Found）エラー コードがブラウザに返される。

解決方法:

ターゲット ページが Web サーバ ドキュメント ルートに存在することを確認します。FWS トレース ログを調べ、ユーザが正しい URL にリダイレクトされていることを確認します。

コンシューマでアサーションを取得する際の HTTP 404 エラー

症状:

依存パーティがアサーションを取得しようとする、HTTP 404 「ページが見つかりません」（Page Not Found）エラー コードがブラウザに返される。

解決方法:

フェデレーション Web サービス アプリケーションが Web アプリケーションとして展開されていることを確認します。アプリケーションは、サポート対象アプリケーション サーバのいずれかを実行する Web サーバ上に展開する必要があります。SiteMinder プラットフォーム サポート マトリックスには、Web エージェント オプション パックに対してサポートされるプラットフォームが一覧表示されています。

フェデレーション Web サービスがプロデューサ/IdP に SAML リクエストを送信しない

症状:

コンシューマ/SP においてフェデレーション Web サービス アプリケーションがプロデューサ/IdP に SAML リクエスト メッセージを送信しない。コンシューマ側が Web サーバの証明書の信頼に失敗する。

解決方法:

プロデューサ/IdP において Web サーバのキー データベースにクライアント証明書を発行した認証機関の証明書を追加します。

一致パラメータの大文字と小文字の区別による設定問題

症状:

プロデューサ/アイデンティティ プロバイダおよびコンシューマ/サービス プロバイダ上で一致する必要がある設定パラメータ間の競合による問題が発生する。これらのパラメータは一致しているように見えるが。

解決方法:

コロンの後の URL 文字列は大文字と小文字が区別されます。たとえば、**http:** の後のテキストは大文字と小文字が区別されます。そのため、すべての対応する設定内の URL の大文字と小文字が一致する必要があります。

アサーティング パーティと依存パーティの間に一致する必要があるパラメータ値については、「[同じ値を使用する必要がある設定 \(P. 451\)](#)」

(Configuration Settings that Must Use the Same Values) のトピックに記載されています。

ログオフ後のポリシー サーバシステムの障害

症状:

環境によっては、ポリシー サーバの実行中にポリシー サーバをログオフすると、ポリシー サーバで障害が発生します。この障害は、JVM の問題によるものです。

解決方法:

JVMOptions.txt ファイルの所定のコマンドラインに `-Xrs` コマンドを追加します。このコマンドは大文字と小文字が区別されるため、前述のとおり追加する必要があります。このコマンドにより、JVM で使用されるオペレーティング システム シグナルが減少します。

JVMOptions.txt ファイルは、`policy_server_home/config/` にあります。

アサーション内のマルチバイト文字が正しく処理されない

症状:

アサーションでマルチバイト文字が含まれると、問題が発生する可能性があります。

解決方法:

以下のように、オペレーティング システムの `LANG` 設定を `UTF-8` に設定します。

`LANG=xx_xx.UTF-8`

たとえば、日本語の場合のエントリは次のとおりです。

`LANG=ja_JP.UTF-8`

ServletExec を使用する IIS Web サーバでトレース ログが表示されない

症状:

LoggerConfig.properties ファイルでトレース ロギングを有効にしても、affwebservices.log ファイルおよび FWStrace.log ファイルが WEB-INF/classes ディレクトリに書き込まれない。

解決方法:

ServletExec に関連付けられた匿名ユーザ アカウントが Windows ファイル システムに書き込む権限を持つことを確認します。このユーザ アカウントがオペレーティング システムの一部として機能する権限を持たない場合、ServletExec はログ ファイルを書き込むことができません。

JVM の初期化中のエラー

症状:

ポリシー サーバ ログ（該当するログ）で以下のエラー メッセージを受信する。

JVM の初期化中にエラーが発生しました (Error occurred during initialization of JVM)
オブジェクト ヒープのための十分なスペースを予約できませんでした。(Could not reserve enough space for object heap.)

この場合、JVM の初期化に失敗しているため、Web エージェント オプション パック機能は動作していません。

解決方法:

オブジェクト ヒープメモリ サイズを制限します。

メモリ サイズを制限する方法

1. `web_agent_home/WEB-INF/properties` file から `JVMOptions.txt` ファイルを開きます。
2. ここに示されているとおりに、以下のエントリをファイルに追加します。

`-Xms128M`
3. ファイルを保存します。
4. ポリシー サーバを再起動します。

SAML 1.x のみの問題

以下の問題は **SAML 1.x** 機能にのみ適用されます。

SAML 1.x Artifact プロファイル シングル サインオンの失敗

症状:

SAML 1.x Artifact プロファイルによるシングルサインオンが設定されている場合、コンシューマ サイトがプロデューサに **SAML** リクエスト メッセージを送信しない。以下のようなエラー メッセージがフェデレーション Web サービスのログ ファイルに記録される。

```
2012/05/23 4:20:44.234 PM[28349544:E] Dispatcher object thrown unknown exception while processing the request message. Message: java.net.ConnectException: Connection refused: connect.
```

```
2012/05/23 4:20:44.234 PM[28349544:E] Exception caught. Message: com.netegrity.affiliateminder.webservices.m: Exception occurred while message dispatcher(srca) object trying to send SOAP request message to the SAML producer.
```

解決方法:

設定された **SSL** ポートを使用して、アサーション検索サービスをホストする Web サーバが実行されていることを確認します。

アサーション検索サービスへのアクセスの認証に失敗する

症状:

SAML 1.x Artifact シングル サインオンを使用する環境で、コンシューマがプロデューサのアサーション検索サービスにアクセスしようとしたときに、認証に失敗する。

解決方法:

ベーシック認証でアサーション検索サービスを保護する場合は、アフィリエイト設定の名前とパスワードが、**SAML** アフィリエイト認証方式のアフィリエイト名とパスワードに一致することを確認してください。

認証方式を変更した後に認証が失敗する

症状:

SAML 1.x アサーション検索サービスを保護する認証方式を「ベーシック」から「クライアント証明書」に変更すると、以降の認証リクエストが失敗することがある。

ユーザが「クライアント証明書」から基本的に SAML 1.x アサーション検索サービスを保護する認証方式を「クライアント証明書」から「ベーシック」に変更すると、以降の認証リクエストが失敗することがある。

解決方法:

認証方式が変更された後は、Web サーバを再起動します。

SAML Artifact シングル サインオンに対するクライアント認証が失敗する

症状:

SAML 1.x Artifact シングル サインオンに対するクライアント証明書の認証がプロデューサで失敗します。以下のエラーが Web エージェント トレース ログに記録されています。

Setting HTTP response variable HTTP_consumer_name=from SiteMinder

たとえば、レスポンス内の属性名が LDAP ユーザディレクトリの「名前」として設定されている場合、レスポンスは失敗します。

解決方法:

FederationWebServicesDomain ドメインの下で Web エージェント レスポンスを作成していることを確認します。レスポンスは以下のとおりにする必要があります。

属性タイプ

WebAgent HTTP ヘッダ変数

属性の種類

ユーザ属性

変数名

consumer_name

属性名

uid (LDAP の場合) または名前 (ODBC の場合)

SAML 2.0x のみの問題

以下の問題は **SAML 2.0** 機能にのみ適用されます。

アサーション検索サービスへのアクセスの認証に失敗する

症状:

SAML 2.0 Artifact シングル サインオンを設定すると、アイデンティティ プロバイダでアサーション検索サービスにアクセスしても、サービス プロバイダが認証ない。

以下のようなエラー メッセージがフェデレーション **Web** サービスのログ ファイルに記録される。

```
2005/05/23 4:43:51.479 PM[31538514:E] SAML producer returned error http status code.  
HTTP return status: 401. Message: <HTML><HEAD><TITLE>401: Access  
Denied</TITLE></HEAD><BODY><H1>401: Access Denied</H1>  
Proper authorization is required for this area. Either your browser does not perform  
authorization, or your authorization has failed.</BODY></HTML>
```

解決方法:

設定された認証によって異なります。

- ベーシック認証の場合、**SAML** サービス プロバイダの名前およびパスワードが **SAML 2.0** 認証方式のアフィリエイト名およびパスワードに一致していることを確認します。
- クライアント証明書認証で **Artifact** 解決サービスを保護する場合、サービス プロバイダのクライアント証明書が有効であることを確認します。また、証明書が証明書データ ストアにあることを確認します。さらに、クライアント証明書を発行した認証機関がアイデンティティ プロバイダの証明書データ ストアにあることを確認します。
- 認証が設定されていない場合は、**Artifact** 解決サービス URL が非保護であることを確認します。

セッション ストアから有効期限データを削除するときの ODBC エラー

症状:

ポリシー サーバを以前のバージョンからアップグレードする場合、有効期限データをセッション ストアから削除する際に ODBC エラーが発生することがある。

解決方法:

「*SiteMinder* アップグレード ガイド」に従ってセッション ストア スキーマをアップグレードします。

付録 A: パートナースhip モデルでのレガシー設定の再作成

レガシー フェデレーション から [set the pfr variable for your book] への直接の移行パスは存在しません。[set the pfr variable for your book] モデルでのレガシー フェデレーション 設定の再生成には、レガシー エンティティの再作成とパートナースhipの設定が必要です。

レガシー オブジェクトとパートナースhip オブジェクトは 1 対 1 対応を共有しません。レガシー フェデレーション モデルでは、フェデレーションの設定に各パートナーで以下のタスクが必要です。

アサーティング パーティ

- アフィリエイト ドメインの設定。
- アフィリエイト ドメイン内の依存パーティの識別およびそれらの依存パーティとの通信の設定。依存パーティには、SAML 1.x アフィリエイト、SAML 2.0 サービス プロバイダおよび WSFED リソース パートナーなどがあります。

依存パーティ

- 依存パーティを定義する認証方式の設定。
- 認証方式で、依存パーティがどのようにアサーションを消費するか、また、どのようにユーザをターゲット アプリケーションにリダイレクトさせるかを指定。

パートナースhip モデルでは、レガシー設定の再作成には次のことが必要です。

- 取引先企業を表すアサーティング パーティおよび依存パーティの各エンティティの設定。
- エンティティ間のパートナースhipの定義。

以下の表は、レガシー フェデレーション コンポーネントと [set the pfr variable for your book] コンポーネントの関係を示しています。

レガシー コンポーネント (アサーティング パーティ)	パートナーシップ コンポーネント (アサーティング パーティ)
SAML 1.1 アフィリエイト	SAML 1.1 プロデューサ - コンシューマ パートナ シップ [set the pfr variable for your book] は SAML 1.0 をサポー トしません。
SAML 2.0 サービス プロバイダ	SAML2 IdP-SP パートナシップ
WSFED リソース パートナー	WSFED IP-RP パートナシップ
レガシー コンポーネント (依存パーティ)	パートナーシップ コンポーネント (依存パーティ)
認証方式 : SAML Artifact または POST テンプレート	SAML 1.1 コンシューマ - プロデューサ パートナ シップ
認証方式 : SAML 2.0 テンプレート	SAML2 SP-IdP パートナシップ
認証方式 : WS-Federation テンプレート	WSFED RP-IP パートナシップ

パートナーシップ モデルで レガシー フェデレーション オブジェクトの再作成を計画する場合は、以下の設定に注意してください。

アクティブ

(レガシー フェデレーション 用アフィリエイト/サービス プロバイダ プロパティ および SAML 認証方式ダイアログ ボックス)。レガシー フェデレーション設定を使用する場合は、このチェック ボックスがオンになっていることを確認します。ソース ID など ID 設定に対して同様の値を持つパートナーシップ フェデレーション モデルでレガシー設定を再作成する場合は、パートナーシップ フェデレーション オブジェクトをアクティブにする前に、このチェック ボックスをオフにします。

SiteMinder は、同じ ID 値を使用するレガシーとパートナーシップ設定では動作できません。動作した場合は名前衝突が発生します。

Artifact 保護タイプ

([set the pfr variable for your book] の SSO 設定)。バックチャネルが HTTP Artifact シングル サインオンに対してどのように保護されるかを定義します。

[set the pfr variable for your book] モデルで レガシー フェデレーション 設定を再作成する場合は、バック チャネルを保護するレガシー方式を使用します。レガシー オプションでは、設定でアサーション検索サービス (SAML 1.x) または Artifact 解決サービス (SAML 2.0) 用に既存の URL を使用できます。

オプションとしてレガシーを選択することによって、SiteMinder は要求を受け入れます。URL を変更する必要はありません。Artifact サービス URL がレガシー設定から取得されるが、この設定に対してパートナーシップ オプションのみが選択されている場合、SiteMinder はリクエストを拒否します。

重要: レガシー フェデレーション オプションについては、Artifact サービスを保護するポリシーを適用します。Artifact サービスはフェデレーション Web サービスのコンポーネントです。このソフトウェアは、フェデレーション Web サービスのポリシーを自動的に作成します。ただし、どのパートナーシップが Artifact を検索するサービスへのアクセスを許可されているかを示す必要があります。詳細については、「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。

オプション: レガシー、パートナーシップ

注: SiteMinder 12.52 SP1 は、フェデレーション セキュリティ サービス ユーザー インターフェイス (FSS UI) および 管理 UI に同梱されています。FSS UI から設定の 管理 UI に切り替える場合は、設定オブジェクトを変更するために FSS UI に戻らないでください。一旦 管理 UI から始めた場合は、管理 UI のみを続けて使用してください。管理 UI を使用した後に FSS UI に戻した場合、ポリシー ストアのオブジェクトがポリシー サーバの機能に問題が発生する可能性があります。