IBM® Security Access Manager for Enterprise Single Sign-On
バージョン 8.2

資格情報管理のための Web API

SA88-4639-00
(英文原典：SC14-7646-00)

IBM® Security Access Manager for Enterprise Single Sign-On
バージョン 8.2

資格情報管理のための Web API

SA88-4639-00
(英文原典：SC14-7646-00)

お願い

本書および本書で紹介する製品をご使用になる前に、 37 ページの『特記事項』に記載されている情報をお読みください。

注: 本書は、IBM Security Access Manager for Enterprise Single Sign-On (製品番号 5724–V67) のバージョン 8.2、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC14-7646-00
IBM® Security Access Manager for Enterprise Single Sign-On
Version 8.2
Web API for Credential Management

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.3

目次

本書について

IBM® Security Access Manager for Enterprise Single Sign-On では、サインオン/サインオフの自動化、認証管理、およびユーザー・トラッキングを提供することにより、強力なデジタル ID へのシームレスなパスを提供します。「*IBM Security Access Manager for Enterprise Single Sign-On 資格情報管理のための Web API*」ガイドには、Web API のインストール、構成、トラブルシューティングの方法、およびセキュリティー trust チェーンの使用方法が記載されています。

重要: 資格情報管理のための Web API では、IBM パートナーとの統合と、それらのパートナーからのサポートが必要になります。

本書の対象読者

本書は、Web API をインストールして構成する必要があるシステム管理者を対象としています。

読者は、以下の各トピックについて十分に理解している必要があります。

- Web API
- セキュリティー trust チェーン

本書の内容

本書には、以下のセクションが含まれます。

- 1 ページの『Web API の概要』

  Web API と Tivoli Federated Identity Manager セキュリティー・トークン・サービス (STS) のインストール方法および構成方法を含む、IBM Security Access Manager for Enterprise Single Sign-On Web API の概要を説明します。
- 29 ページの『付録 A. Web API のトラブルシューティング』

  Web API のトラブルシューティングに必要な関連エラー・コードおよびログ・ファイルをリストします。
- 31 ページの『付録 B. パスワード・ベースの暗号化』

  Web API の暗号化メカニズムについて説明します。
- 35 ページの『付録 C. セキュリティー・トークン・サービス (STS) 汎用ユーザー文書』

  セキュリティー・トークン・サービス汎用ユーザー文書の要素について説明します。

資料

このセクションには、IBM Security Access Manager for Enterprise Single Sign-On ライブラリーにある資料がリストされています。このセクションでは、Tivoli® 資料へのオンラインでのアクセス方法や Tivoli 資料の注文方法について説明します。

IBM Security Access Manager for Enterprise Single Sign-On ライブラリー

IBM Security Access Manager for Enterprise Single Sign-On ライブラリーでは、以下の資料を入手できます。

- *IBM Security Access Manager for Enterprise Single Sign-On Quick Start Guide*、CF38DML

  IBM Security Access Manager for Enterprise Single Sign-On をデプロイし、使用するための主なインストール・タスクおよび構成タスクについてのクイック・スタート・ガイドとして、この資料をお読みください。

- *IBM Security Access Manager for Enterprise Single Sign-On 計画とデプロイメントのガイド*、SC88-5931-02

  インストール・タスクおよび構成タスクを実行する前に、このガイドをお読みください。このガイドは、デプロイメントの計画と環境の準備に役立ちます。このガイドは、製品のフィーチャーやコンポーネントの概要、必要なインストールと構成、さまざまなデプロイメント・シナリオについて説明しています。また、高可用性と災害復旧を実現する方法について説明しています。

- *IBM Security Access Manager for Enterprise Single Sign-On インストール・ガイド*、GI88-4225-01

  IBM Security Access Manager for Enterprise Single Sign-On のインストール、アップグレード、およびアンインストールの詳細な手順については、このガイドをお読みください。

  このガイドは、各種製品コンポーネントとそれぞれの必須ミドルウェアをインストールするとき、さらには製品のデプロイメントのために必要な初期構成を行うときにも役立ちます。仮想アプライアンス、WebSphere® Application Server Base エディション、および Network Deployment を使用する手順が記載されています。

- *IBM Security Access Manager for Enterprise Single Sign-On 構成ガイド*、GC88-8274-01

  IMS Server の設定、AccessAgent ユーザー・インターフェース、およびその動作を構成する場合に、このガイドをお読みください。

- *IBM Security Access Manager for Enterprise Single Sign-On 管理者ガイド*、SC88-5930-02

  このガイドは、管理者を対象とします。さまざまな管理者タスクについて説明しています。このガイドは、ポリシー・テンプレートの作成と割り当て、ポリシーの値の編集、ログとレポートの生成、および IMS Server とそのデータベースの

バックアップのための手順を記載しています。このガイドは、「IBM Security Access Manager for Enterprise Single Sign-On ポリシー定義ガイド」と併せて使用してください。

- *IBM Security Access Manager for Enterprise Single Sign-On ヘルプ・デスク・ガイド*、SC88-5932-02

  このガイドは、ヘルプ・デスク担当者を対象とします。このガイドは、通常は認証要素に関するユーザーからの照会や要求をヘルプ・デスク担当者が管理する際に役立ちます。このガイドは、「IBM Security Access Manager for Enterprise Single Sign-On ポリシー定義ガイド」と併せて使用してください。

- *IBM Security Access Manager for Enterprise Single Sign-On ポリシー定義ガイド*、SC88-8276-01

  管理者が AccessAdmin で構成できる各種のユーザー・ポリシー、マシン・ポリシー、およびシステム・ポリシーの詳細な説明については、このガイドをお読みください。このガイドは、「IBM Security Access Manager for Enterprise Single Sign-On 管理者ガイド」と併せて使用してください。

- *IBM Security Access Manager for Enterprise Single Sign-On トラブルシューティングとサポート・ガイド*、GC88-8275-01

  インストール、アップグレード、および製品の使用に関して問題が発生した場合は、このガイドをお読みください。このガイドは、製品の既知の問題と制限事項について記載しています。問題の症状を確認し、回避策を見つけ出すのに役立ちます。フィックス、知識ベース、サポートに関する情報も提供します。

- *IBM Security Access Manager for Enterprise Single Sign-On AccessStudio ガイド*、SC88-5934-02

  プロファイルを作成または編集する場合は、このガイドをお読みください。このガイドでは、さまざまなアプリケーション・タイプ用の標準の AccessProfile および拡張 AccessProfile の作成と編集の手順を説明しています。また、認証サービスとアプリケーション・オブジェクトの管理に関する情報、および AccessStudio のその他の機能やフィーチャーについての情報を提供します。

- *IBM Security Access Manager for Enterprise Single Sign-On プロビジョニング・インテグレーション・ガイド*、SC88-5935-02

  プロビジョニングのための各種の Java API および SOAP API の情報については、このガイドをお読みください。また、このガイドでは、プロビジョニング・エージェントのインストールと構成の手順も説明しています。

- *IBM Security Access Manager for Enterprise Single Sign-On 資格情報管理のための Web API*、SA88-4639-00

  このガイドは、資格情報の管理用の Web API をインストールおよび構成する場合にお読みください。

- *IBM Security Access Manager for Enterprise Single Sign-On Lightweight AccessAgent mode on Terminal Server SDK Guide*、SC14-7657-00

  AccessAgent を Terminal Services アプリケーションと統合する仮想チャネル・コネクターを作成する方法の詳細については、このガイドをお読みください。

- *IBM Security Access Manager for Enterprise Single Sign-On Serial ID SPI Guide*、SC14-7626-00

  IBM Security Access Manager for Enterprise Single Sign-On は、RFID などの、シリアル番号を含むデバイスに対するサービス・プロバイダー・インターフェース (SPI) を備えています。シリアル番号を持つ任意のデバイスを統合し、そのデバイスを AccessAgent で第 2 認証要素として使用する方法を知りたい場合に、このガイドを参照してください。

- *IBM Security Access Manager for Enterprise Single Sign-On Context Management Integration Guide*、SC23-9954-03

  このガイドは、Context Management ソリューションをインストールおよび構成する場合にお読みください。

- *IBM Security Access Manager for Enterprise Single Sign-On ユーザーズ・ガイド*、SC88-5929-02

  このガイドは、エンド・ユーザーを対象とします。このガイドでは、AccessAgent および Web Workplace を使用する手順を説明しています。

- *IBM Security Access Manager for Enterprise Single Sign-On エラー・メッセージ リファレンス・ガイド*、GA88-4640-00

  このガイドでは、IBM Security Access Manager for Enterprise Single Sign-On に関連するすべての通知メッセージ、警告メッセージ、エラー・メッセージについて説明しています。

用語へのオンライン・アクセス

IBM Terminology Web サイトは、IBM 製品ライブラリーの用語を 1 箇所にまとめた便利なサイトです。以下に示す Web アドレスで Terminology Web サイトにアクセスできます。

http://www.ibm.com/software/globalization/terminology

マニュアルへのオンライン・アクセス

以下は英語のみの対応となります。IBM は、本製品および他の Tivoli 製品の資料が利用可能になったときおよび更新されたとき、Tivoli ソフトウェア・インフォメーション・センターの Web サイト (http://www.ibm.com/tivoli/documentation) に掲示しています。

注: PDF 文書をレターサイズ以外の用紙に印刷する場合は、Adobe Reader のメニューから「**ファイル**」 > 「**印刷**」を選択して表示されたウィンドウでオプションを設定し、レターサイズのページをご使用の用紙に印刷できるようにしてください。

マニュアルのご注文

日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは http://www.ibm.com/jp/manuals/ の「マニュアル・出版物情報」をご覧ください。(URL は、変更になる場合があります)

アクセシビリティー

アクセシビリティー機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。この製品では、インターフェースを音声出力してナビゲートする支援技術を利用できます。また、マウスではなくキーボードを使用して、グラフィカル・ユーザー・インターフェースのすべての機能を操作することもできます。

詳しくは、「*IBM Security Access Manager for Enterprise Single Sign-On 計画とデプロイメントのガイド*」の『アクセシビリティー機能』を参照してください。

Tivoli 技術研修

以下は英語のみの対応となります。Tivoli 技術研修の情報については、IBM Tivoli 研修 Web サイト (http://www.ibm.com/software/tivoli/education) を参照してください。

Tivoli ユーザー・グループ

Tivoli ユーザー・グループは、Tivoli ユーザーに、Tivoli Software ソリューションの実装時に役立つ情報を提供する、独立した、ユーザーによって実行されるメンバーシップ組織です。これらのグループを介して、メンバーは情報を共有し、他の Tivoli ユーザーの知識および経験から学ぶことができます。Tivoli ユーザー・グループには、以下のメンバーおよびグループが含まれます。

- 23,000 を超えるメンバー
- 144 を超えるグループ

Tivoli Users Group に関して詳しくは、www.tivoli-ug.orgにアクセスしてください。

サポート情報

IBM ソフトウェアで問題が発生した場合、迅速な解決が望まれます。IBM では、以下の方法で必要なサポートを提供しています。

オンライン

IBM ソフトウェア・サポート・サイト (http://www.ibm.com/software/support/probsub.html (英語のみの対応)) にアクセスし、指示に従ってください。

IBM Support Assistant

IBM Support Assistant は、無料のローカル・ソフトウェア保守ワークベンチであり、IBM ソフトウェア製品に関する質問や問題を解決するのに役立ちます。IBM Support Assistant を使用すると、サポートに関連する情報および保守容易性ツールに素早くアクセスして、問題を判別することができます。 IBM Support Assistant ソフトウェアをインストールするには、http://www.ibm.com/software/support/isa (英語のみの対応) にアクセスしてください。

トラブルシューティング・ガイド

問題の解決について詳しくは、『*IBM Security Access Manager for Enterprise Single Sign-On トラブルシューティングとサポート・ガイド*』を参照してください。

本書の規則

本書では、特殊な用語やアクション、オペレーティング・システムに依存するコマンドやパス、およびマージン・グラフィックスに関していくつかの規則を使用しています。

書体の規則

本書では、以下の書体の規則を使用します。

太字

- 周囲のテキストとの区別が難しい小文字コマンドおよび大/小文字混合コマンド
- インターフェース・コントロール (チェック・ボックス、プッシュボタン、ラジオ・ボタン、スピン・ボタン、フィールド、フォルダー、アイコン、リスト・ボックス、リスト・ボックス内の項目、複数列のリスト、コンテナー、メニュー選択、メニュー名、タブ、プロパティー・シート)、ラベル (「**ヒント:**」、および「**オペレーティング・システムの考慮事項:**」など)
- 本文中のキーワードおよびパラメーター

イタリック

- 引用 (例: 資料、ディスケット、および CD のタイトル)
- テキスト内で定義される単語 (例: 非交換回線は *Point-to-Point 回線* と呼ばれる)
- 単語や文字の強調 (単語どおりの単語の例: 「制限節を導くには、単語 *that* を使用します。」文字どおりの文字の例: 「LUN アドレスは先頭に文字 *L* が必要です。」)
- 本文中の新規の用語 (定義リスト内の用語は除く): ビュー とは、ワークスペース内のフレームであり、データが含まれます。
- 指定する必要がある変数および値: ... ここで、*myname* は ... を表します。

モノスペース

- 例およびサンプル・コード
- ファイル名、プログラミング・キーワード、および周囲のテキストとの区別が困難なその他の要素
- ユーザーに対して表示されるメッセージ・テキストおよびプロンプト
- ユーザーが入力する必要があるテキスト
- 引数またはコマンド・オプションの値

オペレーティング・システムに依存する変数およびパス

この資料では、環境変数の指定、およびディレクトリー表記に UNIX 規則が使用されます。

Windows コマンド行を使用する場合、環境変数の **$***variable* を **%** *variable***%** に置き換え、ディレクトリー・パスのスラッシュ (*/*) を円記号 (**¥**) に置き換えてくださ

い。環境変数の名前は、Windows と UNIX 環境で常に同じというわけではありません。例えば、Windows 環境での %TEMP% は、UNIX 環境では $TMPDIR になります。

注: Windows システムで bash シェルを使用している場合、UNIX の規則を使用することができます。

Web API の概要

ユーザーの ISAM ESSO パスワード を知っている場合は、Web アプリケーション・プログラミング・インターフェース (Web API) を使用して、そのユーザーの資格情報の作成、読み取り、更新、または削除を行えます。

Tivoli Federated Identity Manager セキュリティー・トークン・サービス (STS) モジュールを使用して、以下の操作を具体的に説明します。

- ユーザー資格情報の取得。
- 指定の日付以降に更新されたユーザー資格情報の取得。
- 1 つの認証サービスのユーザー資格情報の取得。
- 1 つの認証サービスの、指定の日付以降のユーザー資格情報の取得。
- 1 つの認証サービスのユーザー資格情報の設定。
- ユーザー資格情報の削除。

Web サービスにアクセスするために、セキュリティー・トークン要求メッセージが WS-Trust 経由で Tivoli Federated Identity Manager STS に送信されます。IBM Security Access Manager for Enterprise Single Sign-On Web API は、セキュリティー・トークンの交換に WS-Trust プロトコルを使用します。

Tivoli Federated Identity Manager STS は、SOAP プロトコルおよび SSL セキュリティーを使用して、IBM Security Access Manager for Enterprise Single Sign-On 上の Web サービスと通信します。Tivoli Federated Identity Manager STS は、セキュリティー・トークン要求応答メッセージを使用して応答します。

インストールおよび構成

Web API および Tivoli Federated Identity Manager STS をインストールします。

注: このチュートリアルでは、Web API、および Web API との対話に使用される Tivoli Federated Identity Manager STS を、別々の WebSphere Application Server インスタンスおよびプロファイルにインストールするものとします。

Web API および Tivoli Federated Identity Manager STS をインストールして構成するには、以下のコンポーネントが必要です。

- IMS Server
- Web API EAR ファイル
  - com.ibm.tamesso.webapi.wsEAR.ear
- Tivoli Federated Identity Manager プラグイン
  - com.tivoli.am.fim.sts.modules.tamesso.jar
  - com.ibm.tamesso.webapi.wsClient.jar
  - com.ibm.tamesso.webapi.wsClientWrapper.jar
  - com.ibm.tamesso.webapi.contract.jar

IMS Server と Tivoli Federated Identity Manager は、同じ WebSphere Application Server 上にあっても構いません。Tivoli Federated Identity Manager に挿入されたカスタム STS モジュールによって、IMS Server への Web サービス呼び出しが実行されます。 STS モジュールは、IMS Server の場所を特定できなければなりません。

Web API EAR ファイルのインストール

Web API EAR ファイルを WebSphere Application Server にアップロードして、Web API をインストールします。

このタスクについて

Windows x86 または x64 上の WebSphere Application Server を使用します。

手順

1. 管理コンソールを開始します。「**スタート**」 > 「**すべてのプログラム**」 > 「**IBM WebSphere**」 > 「**Application Server** ***<version>***」 > 「**プロファイル**」 > ***<;profile name>*** > 「**管理コンソール**」と選択します。
2. IBM Integrated Solutions Console にログオンします。
3. Integrated Solutions Console のナビゲーション・ペイン上で、「**アプリケーション**」 > 「**アプリケーション**」 > 「**新規アプリケーション**」をクリックします。
4. 「**新規エンタープライズ・アプリケーション**」をクリックします。
5. アップロードしてインストールする EAR ファイルのパスを指定します。

   ローカル・ファイル・システムまたはリモート・ファイル・システムのいずれかを選択し、「**参照**」を使用して、`com.ibm.tamesso.webapi.wsEAR.ear` ファイルの場所を指定できます。
6. 「**次へ**」をクリックします。「アプリケーション・インストールの準備」ページが表示されます。
7. 以下のいずれかのオプションを選択します。
   - **ファースト・パス - 追加情報が必要な場合のみプロンプトを出す**
   - **詳細 - すべてのインストール・オプションおよびパラメーターを表示する**
8. 「**次へ**」をクリックします。
9. 「**インストール・オプションの選択**」の下にあるデフォルト値はそのままにします。
10. 「**次へ**」をクリックします。
11. クラスターおよびサーバーの選択では、ファースト・パスと詳細パスのいずれの場合も、「**モジュールをサーバーにマップ**」を選択します。

    WebSphere Application Server スタンドアロンを使用する場合は、デフォルト値を使用できます。
12. 「**com.ibm.tamesso.webapi.wsEAR.ear**」を選択します。
13. 「**適用**」をクリックします。
14. 「**次へ**」をクリックします。

15. メッセージ・ボックスで、「**保存**」をクリックします。変更内容がマスター構成に保存されます。

Tivoli Federated Identity Manager のインストール

インストール済みの WebSphere Application Server インスタンスで、Tivoli Federated Identity Manager をインストールし、構成します。IMS Server への Web サービス呼び出しを実行するために、カスタム STS モジュールが Tivoli Federated Identity Manager に挿入されます。

始める前に

1. WebSphere Application Server バージョン 7.0.0.9 (フィックスパック 9) をインストールします。

   WebSphere Application Server バージョン 7.0 フィックスパック 9 は、以下のサイトからダウンロードしてください。

   http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg24025888

2. WebSphere Application Server の `IFIX 7.0.0.3-WS-WAS-IFPM09018.pak` をインストールします。

   この WebSphere Application Server 暫定修正は、以下のサイトからダウンロードしてください。

   http://www-01.ibm.com/support/docview.wss?rs=0&uid=swg24026593

   重要: 正しいフィックスパックおよび暫定修正を適用する必要があります。他のフィックスパックを適用すると、Tivoli Federated Identity Manager に問題が発生する可能性があります。

手順

1. インストール済みの WebSphere Application Server インスタンスに Tivoli Federated Identity Manager 6.2.1.0 をインストールします。
   a. Tivoli Federated Identity Manager 6.2.1.0 インストーラーを実行します。
   b. 機能の選択では、デフォルト設定を受け入れます。 必要なすべてのコンポーネントがインストールされます。

   インストールについて詳しくは、http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc_6.2.1/ic/ic-homepage.htmlを参照してください。

2. Tivoli Federated Identity Manager を構成します。
   a. WebSphere 管理コンソール (`https://<IP address>:<admin_port>/ibm/console/`) にログオンします。デフォルトの *admin port* は `9043` です。
   b. 「**ドメイン**」を選択して、ドメインを作成します。
   c. 新規作成したドメインをアクティブ管理ドメインとして設定します。WebSphere Application Server を再始動するように指示するメッセージが表示されます。
   d. WebSphere Application Server を再始動します。再始動によって、構成変更が Tivoli Federated Identity Manager ランタイムに適用されます。

3. WebSphere Application Server アプリケーション・セキュリティーを無効にします。
   a. WebSphere 管理コンソール (https://<*IP address*>:<*admin port*>/ibm/console/) にログオンします。<*admin port*> のデフォルト値は 9043 です。
   b. 「**セキュリティー**」 > 「**グローバル・セキュリティー**」を選択します。
   c. 「**アプリケーション・セキュリティーを使用可能にする**」チェック・ボックスがクリアされていることを確認します。「**アプリケーション・セキュリティーを使用可能にする**」が選択されている場合は、このチェック・ボックスをクリアして、「**適用**」をクリックします。
   d. 変更内容をマスター構成に保存します。
   e. WebSphere Application Server を再始動します。

2 つの WebSphere インスタンス間での SSL の使用可能化

Tivoli Federated Identity Manager と Web API が別々の WebSphere Application Server にインストールされている場合は、2 つの WebSphere インスタンス間で SSL を有効にすることができます。

手順

1. 「**スタート**」 > 「**すべてのプログラム**」 > 「**IBM WebSphere**」 > 「**Application Server** <*version*>」 > 「**プロファイル**」 > <*;profile name*> > 「**管理コンソール**」と選択します。
2. IBM Integrated Solutions Console にログオンします。
3. Integrated Solutions Console のナビゲーション・ペインで、「**セキュリティー**」 > 「**SSL 証明書および鍵管理**」を選択します。
4. 「**関連項目**」で「**鍵ストアおよび証明書**」をクリックします。
5. 「**鍵ストアの使用法**」リストから「**SSL 鍵ストア**」を選択します。
6. 「**NodeDefaultTrustStore**」をクリックします。
7. 「**追加プロパティー**」の下にある「**署名者証明書**」をクリックします。
8. 「**ポートから検索**」をクリックします。
9. 「**一般プロパティー**」で、以下のフィールドに情報を入力します。

   ホスト

   Web API IMS Server のホスト名を入力します。

   ポート

   SSL ポート番号を入力します。

   アウトバウンド接続の SSL 構成

   「**NodeDefaultSSLSettings**」を選択します。

   別名

   インポートされた証明書の別名を入力します。例えば、webapi_root と入力します。
10. 「**OK**」をクリックします。
11. 「**保存**」をクリックします。

セキュリティー・トークン・サービス・モジュールのデプロイ

Tivoli Federated Identity Manager 管理コンソールを使用して、セキュリティー・トークン・サービスを構成し、デプロイします。

始める前に

- trust チェーンを作成する前に、モジュール・タイプ、モジュール・インスタンス、モジュール・モード、trust サービス・チェーンの概念を理解しておく必要があります。
- Tivoli Federated Identity Manager 管理コンソールをインストールします。この手順については、http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.tivoli.fim.doc_6.2.1%2Ftask%2Finstallingconsole.htmlを参照してください。
- トークン交換シナリオのインストールおよび構成については、http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.tivoli.fim.doc_6.2.1%2Fconcept%2FHowToCfgSTS.htmlを参照してください。

このタスクについて

STS モジュールのデプロイでは、モジュール・タイプのインスタンスを作成し、モジュール・インスタンスを組み合わせてセキュリティー trust サービス・チェーンを作成します。各モジュールは、1 つのトークン処理タスクに対して設定されます。

手順

1. 以下の JAR ファイルを `<TFIM installation root path>/plugins` ディレクトリーにコピーします。
   - com.tivoli.am.fim.sts.modules.tamesso.jar
   - com.ibm.tamesso.webapi.contract.jar
   - com.ibm.tamesso.webapi.wsClient.jar
   - com.ibm.tamesso.webapi.wsClientWrapper.jar
2. Tivoli Federated Identity Manager コンソールを使用して、以下のようにモジュールを公開します。
   a. Tivoli Federated Identity Manager の「**オプション**」ペインを展開します。
   b. 「**ドメイン管理**」 > 「**ランタイム・ノード管理**」を選択します。
   c. 「**プラグインのパブリッシュ**」をクリックします。 構成変更をロードするように指示するメッセージが表示されます。
   d. 「**構成変更を Tivoli Federated Identity Manager ランタイムにロード**」をクリックします。
3. カスタム STS モジュールの新規インスタンスを以下のように作成します。
   a. 管理コンソールで、「**Trust Service の構成**」 > 「**モジュール・インスタンス**」をクリックします。
   b. 「**作成**」をクリックします。
   c. モジュール・タイプを選択します。例えば、「**VerifyUser**」を選択します。
   d. 「**次へ**」をクリックします。

e. 各インスタンスの名前 (例えば、`VerifyUserInstance`) を入力します。

f. ステップ b から d を繰り返して、以下のモジュール・インスタンスをすべて作成します。

モジュール・タイプ	各モジュール・インスタンスに付ける名前
VerifyUser	`VerifyUserInstance`
GetUserCredentials	`GetUserCredentialsInstance`
SetUserCredentials	`SetUserCredentialsInstance`
DeleteUserCredentials	`DeleteUserCredentialsInstance`
EncryptUserCredentials	`EncryptUserCredentialsInstance`
MultiUsernameToken	`MultiUsernameTokenInstance`

g. 「**構成変更を Tivoli Federated Identity Manager ランタイムにロード**」をクリックして、プロセスが完了するのを待ちます。 これらのモジュールをセキュリティー trust サービス・チェーンで使用するには、このステップを実行する必要があります。

4. trust サービス・チェーンを作成します。

trust サービス・チェーンは、さまざまなモード (検証、マップ、発行など) で動作する複数のモジュールのチェーンです。

a. 管理コンソールで、「**Trust Service の構成**」 > 「**Trust Service チェーン**」をクリックします。

b. 「**作成**」をクリックします。

c. 「**チェーン・マッピング識別**」領域の「**チェーン・マッピング名**」で、「`EssoChain`」と入力します。

d. 「**次へ**」をクリックします。

e. 「**要求タイプ**」で、「**検証**」を選択します。

f. 「**ルックアップ・タイプ**」で、「**従来の WS-Trust エレメント (適用先、発行者、およびトークン・タイプ) を使用する**」を選択します。

g. 「**適用先**」領域の「**アドレス**」で、「`esso/*`」と入力します。

h. 「**発行者**」領域の「**アドレス**」で、「`esso/.`」と入力します。

i. 「**トークン・タイプ**」で、「**任意のトークン**」を選択します。

5. 以下のモジュール・インスタンスを下記の順序で trust チェーンに追加し、モードを指定します。

順序	モジュール・インスタンス	モード
1	VerifyUserInstance	検証
2	GetUserCredentialsInstance	マップ
3	SetUserCredentialsInstance	マップ
4	DeleteUserCredentialsInstance	マップ
5	EncryptUserCredentialsInstance	マップ
6	MultiUsernameTokenInstance	発行

VerifyUserInstance

「**検証**」モードでは、このモジュール・インスタンスはセキュリティ

ー・トークン要求 (RST) から **UserName** トークンを抽出します。このモジュールはさらに、IMS Web サービスを呼び出して、IBM Security Access Manager for Enterprise Single Sign-On のユーザー名およびパスワードを使用してユーザーをログオンさせます。

VerifyUserInstance モジュールは、最後にこのセッションと IBM Security Access Manager for Enterprise Single Sign-On のユーザー名およびパスワードを STSUU に設定します。

注: 「**マップ**」モードでは、このモジュールは STSUU からユーザー名とパスワードを抽出します。モジュールはさらに、IMS Web サービスを呼び出し、IBM Security Access Manager for Enterprise Single Sign-On のユーザー名およびパスワードを使用してログオンします。これらの例では、「**マップ**」モードの **VerifyUser** モジュールは使用されません。

GetUserCredentialsInstance
ユーザーのすべてのアプリケーション・ユーザー名およびパスワードを取得し、それらをセキュリティー・トークン・サービス汎用ユーザー (STSUU) 文書に追加します。STSUU について詳しくは、 35 ページの『付録 C. セキュリティー・トークン・サービス (STS) 汎用ユーザー文書』を参照してください。

SetUserCredentialsInstance
指定の認証サービスのユーザー資格情報を設定します。

DeleteUserCredentialsInstance
ユーザー名および認証サービスが指定された場合に、ユーザー資格情報を削除します。

EncryptUserCredentialsInstance
パスワード・ベースの暗号化方式により、Tivoli Access Manager パスワードを使用してすべてのユーザー資格情報を暗号化します。パスワード・ベースの暗号化について詳しくは、 31 ページの『付録 B. パスワード・ベースの暗号化』を参照してください。

MultiUsernameTokenInstance
各ユーザー資格情報のユーザー名トークンを生成し、各トークンをセキュリティー・トークン要求応答 (RSTR) メッセージにラップします。このモジュール・インスタンスはその後、すべての RSTR をセキュリティー・トークン要求応答コレクション (RSTR コレクション) にラップします。

6. モジュール・インスタンス設定を構成します。

   VerifyUserInstance: IBM Security Access Manager for Enterprise Single Sign-On API がインストールされている WebSphere インスタンスの URL を入力します。例えば、http://<*server_name*>:9080 と入力します。

   注: IBM Security Access Manager for Enterprise Single Sign-On API および STS の通信に SSL を使用する場合は、SSL ポートを入力してください。デフォルトの SSL ポートは 9443 です。

7. 「**構成変更を Tivoli Federated Identity Manager ランタイムにロード**」をクリックします。

セキュリティー trust チェーンの使用

単一の trust チェーンを使用して、資格情報管理タスクを実行します。

各操作を実行するには、以下に示す値を RST メッセージ内の各要素で使用してください。

操作	RST メッセージ内の各要素の値
特定ユーザーのすべてのユーザー資格情報を取得する	AppliesTo: `esso/get/`
特定の日付以降に更新された、特定ユーザーのすべてのユーザー資格情報を取得する	AppliesTo: `esso/get/` `Claims:<date>` **注:** `<date>` は、XML DateTime 形式の日付です。
1 つの認証サービスのユーザー資格情報を取得する	AppliesTo: `esso/get/<authentication service>` **注:** *`<authentication service>`* は、ISAM ESSO におけるサービス ID です。例えば、E メール・システムなどを示します。
1 つの認証サービスの、特定の日付以降に更新されたユーザー資格情報を取得する	AppliesTo: `esso/get/<authentication service>` **注:** <authentication service> は、ISAM ESSO におけるサービス ID です。例えば、E メール・システムなどを示します。 Claims:`<date>` **注:** `<date>` は、XML DateTime 形式の日付です。
1 つの認証サービスのユーザー資格情報を設定する	AppliesTo: `esso/set/<authentication service>` **注:** *`<authentication service>`* は、ISAM ESSO におけるサービス ID です。例えば、E メール・システムなどを示します。*`<authentication service>`* には、ユーザーが作成する新規の本人認証サービス ID を指定することもできます。
ユーザー資格情報を削除する	AppliesTo: `esso/delete/<user id>`

セキュリティー trust チェーンのテスト

セキュリティー trust チェーンをテストして、資格情報管理タスクが機能しているかどうかを確認します。

手順

1. 操作ごとに RST メッセージを作成し、保存します。以下で説明されている RST サンプルから `rst.xml` を作成できます。

- 10 ページの『ユーザー資格情報の取得』
- 15 ページの『指定の日付以降に更新されたユーザー資格情報の取得』
- 18 ページの『認証サービスのユーザー資格情報の取得』
- 23 ページの『1 つの認証サービスの、指定の日付以降に更新されたユーザー資格情報の取得』
- 25 ページの『ユーザー資格情報の削除 (Web API)』
- 21 ページの『認証サービスのユーザー資格情報の設定』

注: trust チェーンは、要求を検証するように構成されています。Web サービスへのすべての RST メッセージは、以下の条件を満たしていなければなりません。

- `RequestType` 要素が `Validate` に設定されていること。
- ISAM ESSO ユーザー名およびパスワードが指定された `ValidateTarget` 要素が含まれていること。

rst.xml ファイルの例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
  <soapenv:Header/>
  <soapenv:Body>
  <wst:RequestSecurityToken>
  <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
.
.
.
<wst:ValidateTarget>
        <wss:UsernameTokenwsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>user1</wss:Username>
     <wss:Password>p@ssw0rd</wss:Password>
        </wss:UsernameToken>
      </wst:ValidateTarget>
.
.
.
 </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

注:

- IMS Server でエンタープライズ・ディレクトリーとのパスワード同期が有効になっている場合は、エンタープライズ・ディレクトリー資格情報を使用したログオンが実行されます。
- システムの合言葉も有効になっており、エンタープライズ・パスワードが変更された場合は、ユーザーの IMS Server パスワードおよび Wallet が更新されます。
- エンタープライズ・ディレクトリー・パスワードの有効期限が切れている場合や、このパスワードを変更する必要がある場合は、該当するエラー・コードがユーザーに返されます。 29 ページの『付録 A. Web API のトラブルシューティング』を参照してください。

2. コンソールで、以下のコマンドを実行します。

```
curl --header "soapaction: anything" --data-binary @rst.xml
http://localhost:9080/TrustServer/SecurityTokenService > rstr.xml
```

curl コマンドは、以下の処理を行います。

- セキュリティー・トークン要求 (RST) メッセージを XML ファイル形式で送信します。
- Tivoli Federated Identity Manager を呼び出して、セキュリティー・トークン要求応答 (RSTR) メッセージで応答します。

このメッセージは rstr.xml ファイルに格納されます。

3. rstr.xml ファイルを開きます。

   ファイル rstr.xml には、セキュリティー・トークン要求応答 (RSTR) メッセージが含まれています。

   状況コードおよび説明が RSTR メッセージに格納されています。関連する状況コードと説明のリストについては、29 ページの『付録 A. Web API のトラブルシューティング』を参照してください。

ユーザー資格情報の取得

セキュリティー trust チェーンを使用して、ユーザーのすべての資格情報を取得します。

rst.xml ファイル

RST メッセージ例を使用して、STS モジュールでユーザー資格情報を取得します。

この RST およびチェーン例では、チェーンをマッチングするための TokenType (オプション) は使用しません。RST メッセージを使用する場合は、8 ページの『セキュリティー trust チェーンの使用』を参照してください。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。rst.xml ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明
<wst:ValidateTarget> <wss:UsernameToken ...> <wss:Username>**user1**</wss:Username> <wss:Password>**password**</wss:Password> ... </wss:UsernameToken> </wst:ValidateTarget>	**wss:UserNameToken** 要素には、ISAM ESSO ユーザー名およびパスワードが含まれます。**wss:UserNameToken** は RST の **ValidateTarget** 要素に格納されます。 以下の RST メッセージ例の **ValidateTarget** 要素では、ユーザー名およびパスワードとして user1 および password が指定されています。正しくない値が原因でログオンに失敗した場合、STS チェーンは、ユーザー名またはパスワードが無効であることを示す状況を返します。
<wst:RequestType> **http://schemas.xmlsoap.org/ws/2005/02/trust/Validate** </wst:RequestType>	trust チェーンは要求を検証するように構成されたため、**wst:RequestType** は validate に設定する必要があります。

内容	説明
<wst:Issuer> <wsa:Address>**esso/**</wsa:Address> </wst:Issuer>	発行者のアドレスは、チェーンの構成時に指定した値です。例えば、**esso/** とします。
<wsp:AppliesTo> ... <wsa:Address>**esso/get/**</wsa:Address> ... </wsp:AppliesTo>	**AppliesTo** アドレスは **esso/get/** にする必要があります。

例

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
<soapenv:Header/><soapenv:Body>
 <wst:RequestSecurityToken>
   <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
   <wst:Issuer><wsa:Address>esso/</wsa:Address></wst:Issuer>
   <wsp:AppliesTo><wsa:EndpointReference><wsa:Address>esso/get/</wsa:Address>
    </wsa:EndpointReference></wsp:AppliesTo>
  <wst:ValidateTarget>
<wss:UsernameToken wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <wss:Username>user1</wss:Username>
 <wss:Password>password</wss:Password>
  <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
  </wss:UsernameToken>
 </wst:ValidateTarget>
</wst:RequestSecurityToken>
</soapenv:Body>
</soapenv:Envelope>
```

rstr.xml ファイル

rstr.xml ファイルには、RST メッセージがサービスに送信された後の Tivoli Federated Identity Manager STS テストからの応答が含まれています。

RSTR の内容には、次の特性があります。

- STS からのメッセージ応答は、セキュリティー・トークン要求応答コレクション (RSTRC) の形式になっています。このコレクションは、独立した複数のセキュリティー・トークン要求応答 (RSTR) 要素で構成されます。

  セキュリティー・トークン要求応答コレクションの例:

  ```
  <wst:RequestSecurityTokenResponseCollection ...>
  <wst:RequestSecurityTokenResponse ...> ...</wst:RequestSecurityTokenResponse>
  </wst:RequestSecurityTokenResponseCollection>
  ```

- 各 RSTR には、1 つの認証サービスの **UsernameToken** が含まれます。認証サービスは、各 RSTR の **2 番目** の **AppliesTo** 要素に記述されます。
- パスワード・ベースの暗号化 (PBE) に使用される「ソルト」は、**UsernameToken** の **Nonce** フィールドにあります。 31 ページの『付録 B. パスワード・ベースの暗号化』を参照してください。
- 状況は、RSTR の **wst:status** 要素内にあります。

例

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
<wst:RequestSecurityTokenResponseCollection xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dc72a-012a-14c4-9b65-b007b664caa0"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:Status>  <wst:Code>0x00000000</wst:Code>
  <wst:Reason>Credentials Successfully Fetched</wst:Reason>
</wst:Status>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce6f-012a-1937-b2e6-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
 <wss:UsernameToken wsu:Id="usernamed05dce6e-012a-1824-afcf-b007b664caa0"
 xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user0</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
262FBFGv6RFz/aQLNo+bTGS7yFQ=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">w2bP1aUShGvBw1BmIFI1cg==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes0</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>

<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce71-012a-1d47-b129-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce70-012a-1c34-bf3e-b007b664caa0"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user1</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
FjKLvY4sanhFHgf4jwErvpsostg=</wss:Nonce>
<wss:Password Type="PBEwithSHA-256andAES-128">xeAhpgKD9/gg5mpcIBLBKA==</wss:Password>
</wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes1</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>

<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce73-012a-1b21-9d29-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
```

```
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken><wss:UsernameToken wsu:Id="usernamed05dce72-012a-1a0d-ab81-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user2</wss:Username>
<wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
 exqgk1beWmB4QAHIaoUNCyBvfxc=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">eNpik0bz2ctC31pbUkddRw==</wss:Password>
</wss:UsernameToken></wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes2</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>

<wst:RequestSecurityTokenResponse
wsu:Id="uuidd05dce75-012a-1193-98ee-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
<wss:UsernameToken wsu:Id="usernamed05dce74-012a-1080-8b16-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user3</wss:Username>
<wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
4m4Drdt1SNqjmKt3IDNWHb8rl3Q=</wss:Nonce>

<wss:Password Type="PBEwithSHA-256andAES-128">Yg6BdbItSHy2GT2Y02Z3Fw==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes3</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>

<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce77-012a-1f6d-94c6-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
<wss:UsernameToken wsu:Id="usernamed05dce76-012a-1e5a-8afd-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user4</wss:Username>

<wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
iTq4h9PxL21VpXvLoY8ctFyQ6Ak=</wss:Nonce>

<wss:Password Type="PBEwithSHA-256andAES-128">
 6R8d4wmV9pWwMdDc5MMYmg==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes4</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce79-012a-15e0-bab1-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce78-012a-14cc-a90c-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user5</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01-200401-wss-soap-message-security-1.0#Base64Binary">
 eLShBvgzMlRVrJNONBs7s40w+Zk=</wss:Nonce>

<wss:Password Type="PBEwithSHA-256andAES-128">ZkC1CMcyh0fEiVy5Gs2Dzw==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>

<wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes5</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce7b-012a-13b9-a6af-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce7a-012a-12a6-96a2-b007b664caa0"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user6</wss:Username>
<wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
AYRVH0fb08dfSm0mMh5jSwnfuUg=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">I+6R63rtCLmj41HVHdCUbQ==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes6</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce7d-012a-1a2c-960b-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce7c-012a-1919-afae-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user7</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
qtJwksZRZqSLMb//xR+kKJQHap0=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">GcpnTkwGIf740a0HuWZPlw==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes7</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce7f-012a-1806-96ec-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce7e-012a-16f3-b2e2-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user8</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
Yg/VfWMr5qw2M7/us6GWqrzxp18=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">yvtkFmnpeTtBrWFfsk48Qg==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes8</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd05dce81-012a-1dc2-bb21-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>esso/get/</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
<wst:RequestedSecurityToken>
  <wss:UsernameToken wsu:Id="usernamed05dce80-012a-1caf-9103-b007b664caa0"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

 <wss:Username>app_user9</wss:Username>
 <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
oKW67zrzn7QzF83LO7ERFgHzOmY=</wss:Nonce>
 <wss:Password Type="PBEwithSHA-256andAES-128">kYJReye6L4bMJSDd5U+1QQ==</wss:Password>
  </wss:UsernameToken>
</wst:RequestedSecurityToken>
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsa:EndpointReference>
 <wsa:Address>dir_notes9</wsa:Address>
  </wsa:EndpointReference>
</wsp:AppliesTo>
</wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection></soapenv:Body>
</soapenv:Envelope>
```


指定の日付以降に更新されたユーザー資格情報の取得

セキュリティー trust チェーンを使用して、ユーザーの、更新されたすべての資格情報を取得します。指定の日付以降に更新された資格情報を取得します。

rst.xml

RST メッセージ例を使用して、指定の日付以降に更新されたユーザー資格情報を STS モジュールで取得します。

この RST およびチェーン例では、チェーンをマッチングするための TokenType (オプション) は使用しません。RST メッセージを使用する場合は、8 ページの『セキュリティー trust チェーンの使用』を参照してください。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。`rst.xml` ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明
<wst:ValidateTarget> <wss:UsernameToken...> <wss:Username>**user1** </wss:Username> <wss:Password>**password** </wss:Password> ... </wss:UsernameToken> </wst:ValidateTarget>	wss:UserNameToken 要素には、ISAM ESSO ユーザー名およびパスワードが含まれます。wss:UserNameToken は RST の ValidateTarget 要素に格納されます。 以下の RST メッセージ例では、ユーザー名およびパスワードとして user1 および password が指定されています。正しくない値が原因でログオンに失敗した場合、STS チェーンは、ユーザー名またはパスワードが無効であることを示す状況を返します。
<wst:RequestType> **http://schemas.xmlsoap.org/ws/2005/02/trust/Validate** </wst:RequestType>	trust チェーンは要求を検証するように構成されたため、wst:RequestType は validate に設定する必要があります。
<wst:Issuer> <wsa:Address>**esso/**</wsa:Address> </wst:Issuer>	発行者のアドレスは、チェーンの構成時に指定した値です。例えば、esso/ とします。
<wsp:AppliesTo>... <wsa:Address>**esso/get/**</wsa:Address> ... </wsp:AppliesTo>	AppliesTo アドレスは esso/get/ にする必要があります。
<wst:Claims> **<AccountUpdatedAfter>** **2011-05-22T12:33:00Z** **</AccountUpdatedAfter>** </wst:Claims>	この例では、AccountUpdatedAfter ノードを日付値 2011-05-22T12:33:00Z に設定する必要があります。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
  <soapenv:Header />
  <soapenv:Body>
    <wst:RequestSecurityToken>
      <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
      <wst:Issuer>
        <wsa:Address>esso/</wsa:Address>
      </wst:Issuer>
      <wsp:AppliesTo>
        <wsa:EndpointReference>
          <wsa:Address>esso/get/</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:ValidateTarget>
        <wss:UsernameToken wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
  xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401- wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>user1</wss:Username>
          <wss:Password>p@ssw0rd</wss:Password>
        </wss:UsernameToken>
      </wst:ValidateTarget>
      <wst:Claims>
        <AccountUpdatedAfter>2011-05-22T12:33:00Z</AccountUpdatedAfter>      </wst:Claims>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

rstr.xml

rstr.xml ファイルには、指定の日付以降の資格情報を取得するために RST メッセージが送信された後の STS テストからの応答が含まれています。

RSTR の内容には、次の特性があります。

- STS からのメッセージ応答は、セキュリティー・トークン要求応答コレクション (RSTRC) の形式になっています。このコレクションは、独立した複数のセキュリティー・トークン要求応答 (RSTR) 要素で構成されます。

  セキュリティー・トークン要求応答コレクションの例:

  ```
  <wst:RequestSecurityTokenResponseCollection ...>
  <wst:RequestSecurityTokenResponse ...> ...</wst:RequestSecurityTokenResponse>
  </wst:RequestSecurityTokenResponseCollection>
  ```

- 各 RSTR には、1 つの認証サービスの **UsernameToken** が含まれます。認証サービスは、各 RSTR の **2 番目** の **AppliesTo** 要素に記述されます。
- パスワード・ベースの暗号化 (PBE) に使用される「ソルト」は、**UsernameToken** の **Nonce** フィールドにあります。 31 ページの『付録 B. パスワード・ベースの暗号化』を参照してください。
- 状況は、RSTR の **wst:status** 要素内にあります。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header />
  <soapenv:Body>
    <wst:RequestSecurityTokenResponseCollection xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
    <wst:RequestSecurityTokenResponse wsu:Id="uuid2f8e7b3c-0130-14ff-87b9-f88a0599ea80"
    xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

      <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference>
          <wsa:Address>esso/get/</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:Status>
        <wst:Code>0x0</wst:Code>
        <wst:Reason>Account Credentials retrieved successfully</wst:Reason>
      </wst:Status>
    </wst:RequestSecurityTokenResponse>
    <wst:RequestSecurityTokenResponse wsu:Id="uuid2f8e7dce-0130-1314-85e4-f88a0599ea80"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

      <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference>
          <wsa:Address>esso/get/</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:RequestedSecurityToken>
      <wss:UsernameToken wsu:Id="username2f8e7dcd-0130-164e-a8fc-f88a0599ea80"
      xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

        <wss:Username>user2</wss:Username>
        <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
Nl1h9u+vT1u06/OfbiunwrAfEy0=</wss:Nonce>
        <wss:Password Type="PBEwithSHA-256andAES-128">Hl12u2zsoVPvD4kcpj6s0A==</wss:Password>
      </wss:UsernameToken></wst:RequestedSecurityToken>
      <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference>
          <wsa:Address>mail</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
    </wst:RequestSecurityTokenResponse>
    <wst:RequestSecurityTokenResponse wsu:Id="uuid2f8e7dd0-0130-1724-93c5-f88a0599ea80"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
    <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <wsa:EndpointReference>
        <wsa:Address>esso/get/</wsa:Address>
      </wsa:EndpointReference>
    </wsp:AppliesTo>
    <wst:RequestedSecurityToken>
    <wss:UsernameToken wsu:Id="username2f8e7dcf-0130-1427-917a-f88a0599ea80"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

      <wss:Username>user2</wss:Username>
      <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
      W3PBe5qjpb+XiS4Md1REIP9mBpw=</wss:Nonce>
      <wss:Password Type="PBEwithSHA-256andAES-128">3aZu1VwBRY4MeByUpRm3rQ==</wss:Password>
    </wss:UsernameToken></wst:RequestedSecurityToken>
    <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <wsa:EndpointReference>
        <wsa:Address>notes</wsa:Address>
      </wsa:EndpointReference>
    </wsp:AppliesTo></wst:RequestSecurityTokenResponse>
    <wst:RequestSecurityTokenResponse wsu:Id="uuid2f8e7dd2-0130-14fd-b898-f88a0599ea80"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

      <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference>
          <wsa:Address>esso/get/</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:RequestedSecurityToken>
        <wss:UsernameToken wsu:Id="username2f8e7dd1-0130-1837-b349-f88a0599ea80"
        xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

          <wss:Username>user2</wss:Username>
  <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
UrZoQGodbzS/EPlFSkaXiZPkFf4=</wss:Nonce>
          <wss:Password Type="PBEwithSHA-256andAES-128">bA9gvoiAPfT46tFh/lxxgg==</wss:Password>
        </wss:UsernameToken>
      </wst:RequestedSecurityToken>
      <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">-
      <wsa:EndpointReference>
        <wsa:Address>dir_skype</wsa:Address>
      </wsa:EndpointReference></wsp:AppliesTo>
    </wst:RequestSecurityTokenResponse>
    <wst:RequestSecurityTokenResponse wsu:Id="uuid2f8e7dd4-0130-1b70-af1a-f88a0599ea80"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <wsa:EndpointReference>
        <wsa:Address>esso/get/</wsa:Address>
      </wsa:EndpointReference>
    </wsp:AppliesTo>
    <wst:RequestedSecurityToken>
    <wss:UsernameToken wsu:Id="username2f8e7dd3-0130-1611-96d0-f88a0599ea80"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

      <wss:Username>user1</wss:Username>
      <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
      tz5f6KP56AZ3jXEnnRFejSGcodc=</wss:Nonce>
      <wss:Password Type="PBEwithSHA-256andAES-128">ki40fjbr8H9RRtr0hPbu7Q==</wss:Password>
    </wss:UsernameToken></wst:RequestedSecurityToken>
    <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <wsa:EndpointReference>
        <wsa:Address>dir_ibm.example.com</wsa:Address>
      </wsa:EndpointReference>
    </wsp:AppliesTo></wst:RequestSecurityTokenResponse></wst:RequestSecurityTokenResponseCollection>
  </soapenv:Body>
</soapenv:Envelope>
```


認証サービスのユーザー資格情報の取得

Web API を使用して、認証サービスのすべてのユーザー資格情報を取得します。

rst.xml ファイル

RST メッセージ例を使用して、認証サービスの資格情報を取得します。

この RST およびチェーン例では、チェーンをマッチングするための TokenType (オプション) は使用しません。RST メッセージのサンプルを使用してサービス要求を送信する場合は、 8 ページの『セキュリティー trust チェーンの使用』を参照してください。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。`rst.xml` ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明

`<wst:ValidateTarget>`
`<wss:UsernameToken ...>...`
`  </wss:UsernameToken>`
`</wst:ValidateTarget>`
	wss:UserNameToken 要素には、ISAM ESSO ユーザー名およびパスワードが含まれます。
	wss:UserNameToken は RST の **ValidateTarget** 要素に格納されます。

`<wst:RequestType>`
http://schemas.xmlsoap.org/ws/2005/02/trust/Validate
`</wst:RequestType>`
	trust チェーンは要求を検証するように構成されているため、**RequestType** 要素は validate に設定されています。

`<wst:Issuer>`
`<wsa:Address>`**esso/**`</wsa:Address>`
`</wst:Issuer>`
	発行者のアドレスは、チェーンの構成時に指定した値に一致するアドレスでなければなりません。例えば、`esso/` とします。

`<wsp:AppliesTo>...`
`<wsa:Address>`
esso/get/*<authentication service>*
`... </wsp:AppliesTo>`
	wsp:AppliesTo 要素のアドレスは、**esso/get/*<authentication service>*** にする必要があります。***<authentication service>*** は、ISAM ESSO におけるサービス ID です。例えば、E メール・システムの場合は、**mail** です。

`<wss:Username>user1</wss:Username>` `<wss:Password>password</wss:Password>`	**wss:Username** 要素には、ISAM ESSO ユーザー名が含まれます。 **wss:Password** 要素には、ISAM ESSO パスワードが含まれます。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <soapenv:Header/>
  <soapenv:Body>
  <wst:RequestSecurityToken>
   <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
   <wst:Issuer>
    <wsa:Address>esso/</wsa:Address>
   </wst:Issuer>
  <wsp:AppliesTo>
<wsa:EndpointReference>
<wsa:Address>esso/get/mail</wsa:Address>
  </wsa:EndpointReference>
  </wsp:AppliesTo>
  <wst:ValidateTarget>
  <wss:UsernameToken
wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wss:Username>user1</wss:Username>
     <wss:Password>password</wss:Password>
```

```
     <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
        </wss:UsernameToken>      </wst:ValidateTarget>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

rstr.xml ファイル

rstr.xml ファイルには、認証サービスからユーザー資格情報を取得するための STS テストからの応答が含まれています。

メッセージ応答には、次の特性があります。

- STS からの応答は、セキュリティー・トークン要求応答コレクション (RSTRC) の形式になっています。このコレクションは、独立した複数のセキュリティー・トークン要求応答 (RSTR) 要素で構成されます。

  セキュリティー・トークン要求応答コレクションの例:

  ```
  <wst:RequestSecurityTokenResponseCollection...>
   <wst:RequestSecurityTokenResponse...>...</wst:RequestSecurityTokenResponse>
  </wst:RequestSecurityTokenResponseCollection>
  ```

- 各 RSTR には、1 つの認証サービスの UsernameToken が含まれます。
- 認証サービスは、各 RSTR の 2 番目の **AppliesTo** 要素に記述されます。
- 状況は、RST の **wst:status** 要素内にあります。

例

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <wst:RequestSecurityTokenResponseCollection
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
      <wst:RequestSecurityTokenResponse wsu:Id="uuidd058f7b8-012a-110c-afe2-b007b664caa0"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference>
            <wsa:Address>esso/get/mail</wsa:Address>
          </wsa:EndpointReference>
        </wsp:AppliesTo>
    <wst:Status>
       <wst:Code>0x00000000</wst:Code>
       <wst:Reason>Credentials Successfully Fetched</wst:Reason>
    </wst:Status>      </wst:RequestSecurityTokenResponse>
      <wst:RequestSecurityTokenResponse wsu:Id="uuidd058fad6-012a-1ae4-9fe0-b007b664caa0"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference>
            <wsa:Address>esso/get/mail</wsa:Address>
          </wsa:EndpointReference>
        </wsp:AppliesTo>
        <wst:RequestedSecurityToken>
<wss:UsernameToken wsu:Id="usernamed058fad5-012a-1e1d-b48c-b007b664caa0"
 xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <wss:Username>app_user0</wss:Username>
  <wss:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
0+Nx140dzSSQHPqFfa8xxlAUdN8=
   </wss:Nonce> <wss:Password Type="PBEwithSHA-256andAES-128">dmISqch7uCsCw7xr/URYjw==
 </wss:Password>
</wss:UsernameToken>   </wst:RequestedSecurityToken>
   <wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference>
            <wsa:Address>mail</wsa:Address>
          </wsa:EndpointReference>
```

```
      </wsp:AppliesTo>
     </wst:RequestSecurityTokenResponse>
    </wst:RequestSecurityTokenResponseCollection>
  </soapenv:Body>
</soapenv:Envelope>
```

認証サービスのユーザー資格情報の設定

セキュリティー trust チェーンを使用して、認証サービスのユーザー資格情報を設定します。認証サービスが存在しない場合は、ユーザー用の本人認証サービスを作成し、資格情報を保存します。

rst.xml ファイル

RST メッセージ内容の例を使用して、認証サービスのユーザー資格情報を設定します。

この RST およびチェーン例では、チェーンをマッチングするための TokenType (オプション) は使用しません。RST メッセージをテストする場合は、8 ページの『セキュリティー trust チェーンの使用』を参照してください。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。`rst.xml` ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明

`<wst:ValidateTarget>`
`<wss:UsernameToken ...>`
`<wss:Username>`**`user1`**`</wss:Username>`
`<wss:Password>`**`password`**`</wss:Password>`
`...</wss:UsernameToken>`
`</wst:ValidateTarget>`
	`wss:UserNameToken` 要素には、ISAM ESSO ユーザー名およびパスワードが含まれます。**`wss:UserNameToken`** は RST の **`ValidateTarget`** 要素内にあります。
	
	以下の RST メッセージ例の **`ValidateTarget`** 要素では、ユーザー名およびパスワードとして **`user1`** および **`password`** が指定されています。正しくない値が原因でログオンに失敗した場合、STS チェーンは、ユーザー名またはパスワードが無効であることを示す状況を返します。

`<wst:RequestType>`
`http://schemas.xmlsoap.org/ws/2005/02/trust/Validate`
`</wst:RequestType>`
	trust チェーンは要求を検証するように構成されているため、**`RequestType`** を validate に設定します。

`<wst:Issuer>`
`<wsa:Address>`**`esso/`**
`</wsa:Address>`
`</wst:Issuer>`
	発行者のアドレスは、チェーンの構成時に指定した値でなければなりません。例えば、**`esso/`** とします。

`<wsp:AppliesTo>`
`<wsa:EndpointReference>`
`<wsa:Address>`**`esso/set/mail`**
`</wsa:Address>`
`</wsa:EndpointReference>`
`</wsp:AppliesTo>`
	`AppliesTo` アドレスは **`esso/set/`*`<authentication service>`*** です。***`<authentication service>`*** は、ISAM ESSO におけるサービス ID です。例えば、**`mail`** とします。ユーザーが自分用に作成する新規サービス ID を指定することもできます。

`<wst:Base>`
`<wss:UsernameToken ...>...`
`</wss:UsernameToken>`
`</wst:Base>`
	設定する資格情報が含まれている **`UsernameToken`** を RST の **`wst:Base`** 要素内に置きます。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<soapenv:Header/>
  <soapenv:Body>
    <wst:RequestSecurityToken>
      <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
      <wst:Issuer>
        <wsa:Address>esso/</wsa:Address>
      </wst:Issuer>
      <wsp:AppliesTo>
        <wsa:EndpointReference>
          <wsa:Address>esso/set/mail</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:ValidateTarget><wss:UsernameToken
wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>user1</wss:Username>
    <wss:Password>password</wss:Password>
          <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
        </wss:UsernameToken>
      </wst:ValidateTarget><wst:Base>
<wss:UsernameToken
wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>user1</wss:Username>
    <wss:Password>password</wss:Password>
          <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
        </wss:UsernameToken>
      </wst:Base>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

rstr.xml ファイル

rstr.xml ファイルには、認証サービスのユーザー資格情報を設定するための STS テストからの応答が含まれています。

- 状況 (資格情報の設定が正常に行われたか失敗したか) は、**wst:Status** 要素に反映されます。

例

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header/>
<soapenv:Body>
<wst:RequestSecurityTokenResponse wsu:Id="uuidd06e666a-012a-150a-8eb8-b007b664caa0"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsp:AppliesTo
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
 <wsa:EndpointReference>
  <wsa:Address>esso/set/mail</wsa:Address>
  </wsa:EndpointReference>
 </wsp:AppliesTo>
<wst:Status>
  <wst:Code>0x00000000</wst:Code>
```

```
  <wst:Reason>Setting of Credential Successful</wst:Reason>
</wst:Status> </wst:RequestSecurityTokenResponse>
</soapenv:Body>
</soapenv:Envelope>
```

1 つの認証サービスの、指定の日付以降に更新されたユーザー資格情報の取得

セキュリティー trust チェーンを使用して、ユーザーの、更新されたすべての資格情報を取得します。指定の日付以降に更新された資格情報を取得します。

rst.xml ファイル

RST メッセージ例を使用して、STS モジュールでユーザー資格情報を取得します。

この RST およびチェーン例では、チェーンをマッチングするための TokenType (オプション) は使用しません。RST メッセージを使用する場合は、8 ページの『セキュリティー trust チェーンの使用』を参照してください。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。`rst.xml` ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明

`<wst:ValidateTarget>`
`  <wss:UsernameToken...>`
`  <wss:Username>user1</wss:Username>`
`  <wss:Password>password</wss:Password>`
`  ...`
`  </wss:UsernameToken>`
`</wst:ValidateTarget>`
	`wss:UserNameToken` 要素には、ISAM ESSO ユーザー名およびパスワードが含まれます。`wss:UserNameToken` は RST の `ValidateTarget` 要素に格納されます。
	
	以下の RST メッセージ例の **`ValidateTarget`** 要素では、ユーザー名およびパスワードとして `user1` および `password` が指定されています。正しくない値が原因でログオンに失敗した場合、STS チェーンは、ユーザー名またはパスワードが無効であることを示す状況を返します。

`<wst:RequestType>`
`http://schemas.xmlsoap.org/ws/2005/02/trust/Validate`
`</wst:RequestType>`
	trust チェーンは要求を検証するように構成されたため、`wst:RequestType` は `validate` に設定する必要があります。

`<wst:Issuer>`
`  <wsa:Address>`**`esso/`**`</wsa:Address>`
`</wst:Issuer>`
	発行者のアドレスは、チェーンの構成時に指定した値です。例えば、**`esso/`** とします。

`<wsp:AppliesTo>...`
`  <wsa:Address>`
`esso/get/`
`</wsa:Address>...`
`</wsp:AppliesTo>`
	AppliesTo アドレスは **`esso/get/`**
	`<authentication service>` にする必要があります。***`<authentication service>`*** は、ISAM ESSO におけるサービス ID です。例えば、E メール・システムの場合は、**`mail`** です。

`<wss:Username>user1</ wss:Username>` `<wss:Password>password</ wss:Password>`	`wss:Username` 要素には、ISAM ESSO ユーザー名が含まれます。 `wss:Password` 要素には、ISAM ESSO パスワードが含まれます。

内容	説明
<wst:Claims> <AccountUpdatedAfter> **2011-05-22T12:33:00Z** </AccountUpdatedAfter> </wst:Claims>	例えば、Claims の AccountUpdatedAfter ノードを日付値 2011-05-22T12:33:00Z に設定する必要があります。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
<soapenv:Header />
<soapenv:Body>
<wst:RequestSecurityToken>
  <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
  <wst:Issuer>
 <wsa:Address>esso/</wsa:Address>
  </wst:Issuer>
  <wsp:AppliesTo>
 <wsa:EndpointReference>
   <wsa:Address>esso/get/</wsa:Address>
 </wsa:EndpointReference>
  </wsp:AppliesTo>
  <wst:ValidateTarget>
 <wss:UsernameToken wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
 xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <wss:Username>test22</wss:Username>
   <wss:Password>p@ssw0rd</wss:Password>
 </wss:UsernameToken>
  </wst:ValidateTarget>
  <wst:Claims>
 <AccountUpdatedAfter>2011-05-22T12:33:00Z</AccountUpdatedAfter>
  </wst:Claims>
</wst:RequestSecurityToken>
</soapenv:Body>
</soapenv:Envelope>
```

rstr.xml ファイル

rstr.xml ファイルには、指定の日付以降に更新された資格情報を取得するための STS テストからの応答が含まれています。

RSTR の内容には、次の特性があります。

- STS からのメッセージ応答は、セキュリティー・トークン要求応答コレクション (RSTRC) の形式になっています。このコレクションは、独立した複数のセキュリティー・トークン要求応答 (RSTR) 要素で構成されます。

  セキュリティー・トークン要求応答コレクションの例:

  ```
  <wst:RequestSecurityTokenResponseCollection ...>
  <wst:RequestSecurityTokenResponse ...> ...</wst:RequestSecurityTokenResponse>
  </wst:RequestSecurityTokenResponseCollection>
  ```

- 各 RSTR には、1 つの認証サービスの **UsernameToken** が含まれます。認証サービスは、各 RSTR の **2 番目** の **AppliesTo** 要素に記述されます。
- パスワード・ベースの暗号化 (PBE) に使用される「ソルト」は、**UsernameToken** の **Nonce** フィールドにあります。 31 ページの『付録 B. パスワード・ベースの暗号化』を参照してください。
- 状況は、RSTR の **wst:status** 要素内にあります。

例

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
<wst:RequestSecurityTokenResponseCollection xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
  <wst:RequestSecurityTokenResponse wsu:Id="uuid2fb0cee5-0130-1e07-86b2-dcb8f1d2f433"
  xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
 xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
   <wsa:EndpointReference>
  <wsa:Address>esso/get/mail</wsa:Address>
   </wsa:EndpointReference>
 </wsp:AppliesTo>
 <wst:Status>
   <wst:Code>0x0</wst:Code>
   <wst:Reason>Account Credentials retrieved successfully</wst:Reason>
 </wst:Status>
  </wst:RequestSecurityTokenResponse>
  <wst:RequestSecurityTokenResponse wsu:Id="uuid2fb0d03e-0130-1b95-bfdd-dcb8f1d2f433"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
 xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
   <wsa:EndpointReference>
  <wsa:Address>esso/get/mail</wsa:Address>
   </wsa:EndpointReference>
 </wsp:AppliesTo>
 <wst:RequestedSecurityToken>
   <wss:UsernameToken wsu:Id="username2fb0d03d-0130-1ecf-9049-dcb8f1d2f433"
   xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wss:Username>user2</wss:Username>
  <wss:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-soap-message-security-1.0#Base64Binary">
veFaIFIzRoD6PqG7/F67a2M9Xdc=</wss:Nonce>
<wss:Password Type="PBEwithSHA-256andAES-128">j03hcXeHYZmRjdWpXvu8+Q==</wss:Password>
   </wss:UsernameToken>
 </wst:RequestedSecurityToken>
 <wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
 xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
   <wsa:EndpointReference>
  <wsa:Address>mail</wsa:Address>
   </wsa:EndpointReference>
 </wsp:AppliesTo>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
</soapenv:Body>
</soapenv:Envelope>
```

ユーザー資格情報の削除 (Web API)

特定のユーザー名に関するユーザー資格情報を削除します。

rst.xml ファイル

サンプル RST を使用して、ユーザーを削除します。

注: アンパーサンド文字 (&) は、有効な XML 文字ではありません。`rst.xml` ファイルにこの文字が含まれないようにしてください。そうしないと、エラーが発生します。

内容	説明

`<wst:ValidateTarget>`
`<wss:UsernameToken ...>`
`...</wss:UsernameToken>`
`</wst:ValidateTarget>`
	ESSO ユーザー名およびパスワードが含まれている `UsernameToken` を RST の `ValidateTarget` 要素内に置きます。

`<wst:RequestType>`
`http://schemas.xmlsoap.org/ws/2005/02/`
`trust/Validate`
`</wst:RequestType>`
	trust チェーンは要求を検証するように構成されたため、`wst:RequestType` は `Validate` に設定する必要があります。

内容	説明
<wst:Issuer><wsa:Address>**esso/** </wsa:Address> </wst:Issuer>	発行者のアドレスは、チェーンの構成時に指定した値に一致するアドレス (esso/ など) でなければなりません。
<wsp:AppliesTo><wsa:EndpointReference> <wsa:Address>**esso/delete/mail** </wsa:Address></wsa:EndpointReference> </wsp:AppliesTo>	AppliesTo アドレスは、esso/delete/*<authentication service>* にする必要があります。ここで、*<authentication service>* は資格情報の削除対象となる認証サービスです。例えば、esso/delete/mail にします。
<wss:UsernameToken...>... </wss:UsernameToken>	削除する資格情報が含まれている UsernameToken を RST の Base 要素内に置きます。このトークンの password 要素および created 要素は重要ではありません。これらは省略可能です。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <soapenv:Header/>
  <soapenv:Body>
    <wst:RequestSecurityToken>
      <wst:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Validate</wst:RequestType>
      <wst:Issuer>
        <wsa:Address>esso/</wsa:Address>
      </wst:Issuer>
      <wsp:AppliesTo>
        <wsa:EndpointReference>
          <wsa:Address>esso/delete/mail</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:Base>
  <wss:UsernameToken
      wsu:Id="username8a2fcf8c-0128-124a-b5d0-adafae3d9ad4"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>userX2</wss:Username>
   <wss:Password>p@ssw0rd1</wss:Password>
          <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
        </wss:UsernameToken>
      </wst:Base>
      <wst:ValidateTarget>
        <wss:UsernameToken
     wsu:Id="username8a2fcf7b-0128-124a-b5d0-adafae3d9ad1"
   xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wss:Username>imsTest3</wss:Username>
     <wss:Password>p@ssw0rd1</wss:Password>
          <wsu:Created>2010-05-25T01:45:08Z</wsu:Created>
        </wss:UsernameToken>
      </wst:ValidateTarget>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

rstr.xml ファイル

rstr.xml ファイルには、ユーザーを削除するための STS テストからの応答が含まれています。

項目	説明
<wst:Status> <wst:Code>**0x0**</wst:Code> <wst:Reason>Account Credentials retrieved successfully</wst:Reason> </wst:Status>	資格情報の設定が正常に行われたかどうかを示す状況は、RST の wst:Status 要素に反映されます。

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header/>
<soapenv:Body>
<wst:RequestSecurityTokenResponse wsu:Id="uuid145c7e1f-012d-1b3f-b991-fccbe5dc5e42"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsp:AppliesTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
<wsa:EndpointReference><wsa:Address>esso/get/dir_skype</wsa:Address>
                              </wsa:EndpointReference></wsp:AppliesTo>
   <wst:Status>
     <wst:Code>0x0</wst:Code>
       <wst:Reason>Account Credentials retrieved successfully</wst:Reason>
   </wst:Status>
</wst:RequestSecurityTokenResponse></soapenv:Body></soapenv:Envelope>
```

付録 A. Web API のトラブルシューティング

システム・ログおよびトレース・ログを使用して、発生している可能性のある問題のトラブルシューティングを行い、解決策を適用します。

各種タスクの関連エラー・コードは、以下のとおりです。

- 成功: 0x00000000
- 無効なログイン: 0x53000250
- エンタープライズ・ディレクトリー・パスワードを変更する必要がある: 0x53000306[1]
- エンタープライズ・ディレクトリー・パスワードの有効期限が切れている: 0x53000315[2]
- 無効な認証サービス: 0x53008150
- 認証サービスが不明: 0x53008101
- ユーザーがログインしていない: 0x53009301
- 予期しないエラー: 0x53009300
- 無効な引数: 0x53008405

  注: このエラー・コードは、通常、1 つ以上の引数が NULL である場合に発生します。
- データ・ストア例外: 0x23005000
- アカウントが見つからない (削除操作の場合): 0x53008403

  注: このエラー・コードは、通常、アカウントの削除中に発生します。

SystemOut.log および SystemErr.log の各ログは、<was_home>¥profiles¥AppSrv14¥logs¥server1 にあります。例えば、C:¥Program Files¥IBM¥WebSphere¥AppServer7.0¥profiles¥AppSrv14¥logs¥server1 にあります。

メモリー不足エラー

Web API のデプロイ中にメモリー不足の問題が発生した場合は、JVM 最大ヒープ・サイズを少なくとも 1024 MB に増やしてください。

1. Active Directory のパスワード同期が有効である場合に適用されます。
2. Active Directory のパスワード同期が有効である場合に適用されます。

付録 B. パスワード・ベースの暗号化

Web API は、パスワード・ベースの暗号化プロセスを使用してすべてのアプリケーション・パスワードを暗号化し、それらを **RequestSecurityTokenResponseCollection** に含めて返します。

暗号化プロセスでは、ISAM ESSO パスワードから秘密鍵を生成します。秘密鍵を生成するときに、暗号化プロセスは *ソルト* と呼ばれるランダム・バイトのコレクションを使用します。秘密鍵は、AES-128 ビット暗号化アルゴリズムを使用してアプリケーション・パスワードを暗号化します。このアルゴリズムでは、以下のプロセスに従って鍵が生成されます。

1. パスワードにソルトを付加し、SHA-256 アルゴリズムを使用してハッシュを生成します。
2. ダイジェストにソルトを付加し、SHA-256 アルゴリズムを使用してハッシュを生成します。このプロセスを 1000 回繰り返します。

最終ダイジェストの最初の 16 バイトが、鍵となります。

CryptoUtil.java ファイル

CryptoUtils クラスを使用して、アプリケーション・パスワードを暗号化解除し、パスワードをプレーン・テキストで返します。`encryptedAppPassword`、*ソルト*、および ISAM ESSO パスワードを指定して、`getDecryptedAppPassword` メソッドを呼び出します。

```
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;
import java.security.*;
import java.util.logging.*;

/*
 * Password-Based Encryption(PBE)

* In this class we are using the ISAMESSO password to encrypt the application
* passwords which are returned in the RSTR. In this process we generate a secret key
* using the ISAMESSO password and a collection of random bytes called the 'Salt'. This
* key is then used to encrypt the application password using the AES-128 encryption
* algorithm.
* The algorithm to generate the key is as follows:
*  1. Append the Salt(a collection of random bytes) to the password and generate a hash
*     using the SHA-256 hashing algorithm.
*  2. Append the Salt to the digest and generate the hash using the SHA-256 algorithm.
*     Repeat this process 1000 times.
*  3. The first 16 bytes of the final digest is our key.
*
*
* This class provides functions to implement PBE
*/
public class CryptoUtils {

 /**This decrypts the encryptedAppPassword and returns it as plain-text. It converts
   * the byte array returned by decrypt into UTF-8
  * @param Encrypted Application Password
  * @param Salt
  * @param Esso Password
  * @return Decrypted Application Password
  */

 private static final String CLASS = CryptoUtils.class.getName();
 private static final Logger log = Logger.getLogger(CLASS);
```

```
private static final String ENCRYPTION_ALGO = "AES";
private static final String HASHING_ALGO = "SHA-256";
private static final String PROVIDER = "com.ibm.crypto.fips.provider.IBMJCEFIPS";
 //JCE provider
private static final String ENCODING = "UTF-8";
private static final int ITERATIONS = 1000;
 //number of ITERATIONS for generating the Secret Key

public static String getDecryptedAppPassword(String encryptedAppPassword,
            String salt,String essoPassword)
throws IllegalArgumentException
{
 String methodName_ = "getDecryptedAppPassword()";
 log.entering(CLASS, methodName_);
 if(encryptedAppPassword==null||salt==null||essoPassword==null) throw
            new IllegalArgumentException
 ("One or more of the parameters are Null");
 String result = null;
 try{
  Security.addProvider((Provider) ((Class.forName(PROVIDER))
          .newInstance()));
  byte[] encryptedAppPassword1 = Base64.decode(encryptedAppPassword);
  byte[] salt1 = Base64.decode(salt);
  byte[] arr = decrypt(essoPassword.getBytes(ENCODING),encryptedAppPassword1,salt1);
  result = new String(arr,ENCODING);
  log.exiting(CLASS,methodName_);
 }
 catch(Exception e){
  log.log(Level.WARNING,e.getMessage(),e);
 }
 return result;
}

/**This decrypts the encryptedAppPassword and returns it as a byte array
  * given the Salt and the essoPassword using PBE
 * @param essoPassword
 * @param appPassword
 * @param salt
 * @return De-crypted Application Password as an array of bytes
 */
public static byte[] decrypt(byte[] essoPassword,byte[] appPassword,byte[] salt)
throws IllegalArgumentException
{
 String methodName_ = "decrypt";
 log.entering(CLASS, methodName_);
 if(essoPassword==null||appPassword==null||salt==null) throw
                new IllegalArgumentException
 ("One or more of the parameters are Null");
 byte[] result = null;
 try{
  if(essoPassword==null||appPassword==null||salt==null) throw
                  new NullPointerException();
  SecretKey key = generateKey(essoPassword,salt);
  Cipher aesCipher;
  aesCipher = Cipher.getInstance(ENCRYPTION_ALGO);
  aesCipher.init(Cipher.DECRYPT_MODE,key);
  log.exiting(CLASS,methodName_);
  result = aesCipher.doFinal(appPassword);
 }
 catch(Exception e){
  log.log(Level.WARNING,e.getMessage(),e);
 }
 return result;
}

/**This encrypts the AppPassword and returns it as a byte array given the
  * Salt and the essoPassword using PBE
 * @param essoPassword
 * @param appPassword
 * @param salt
 * @return De-crypted Application Password as an array of bytes
 * @throws IllegalArgumentException
 */
public static byte[] encrypt(byte[] essoPassword,byte[] appPassword,byte[] salt)
throws IllegalArgumentException
{
 String methodName_ = "encrypt";
 log.entering(CLASS,methodName_);
 if(essoPassword==null||appPassword==null||salt==null) throw
```

```
                        new IllegalArgumentException
 ("One or more of the parameters are Null");
 byte[] result = null;
 try{
  SecretKey key = generateKey(essoPassword,salt);
  Cipher aesCipher;
  aesCipher = Cipher.getInstance("AES");
  aesCipher.init(Cipher.ENCRYPT_MODE,key);
  log.entering(CLASS,methodName_);
  result = aesCipher.doFinal(appPassword);
 }
 catch(Exception e){
  log.log(Level.WARNING,e.getMessage(),e);
 }
 return result;
}

/**This generates the secret key given the essoPassword and Salt using PBE
 * @param essoPassword
 * @param salt
 * @return SecretKey obtained from the ESSO password using PBE
 * @throws IllegalArgumentException
 */
public static SecretKey generateKey(byte[] essoPassword,byte[] salt)
throws IllegalArgumentException
{
 String methodName_ = "generateSecretKey";
 log.entering(CLASS, methodName_);
 if(essoPassword==null||salt==null) throw new IllegalArgumentException
 ("One or more of the parameters are Null");
 SecretKey aesKey = null;
 try{
  essoPassword = append(essoPassword,salt);
  SecretKeyFactory factory = SecretKeyFactory.getInstance(ENCRYPTION_ALGO);
  MessageDigest md = MessageDigest.getInstance(HASHING_ALGO);
  byte[] hashedKey = md.digest(essoPassword);
  for(int i=0;i<ITERATIONS-1;i++){
      hashedKey = append(hashedKey,salt);
      hashedKey = md.digest(hashedKey);
  }
  SecretKeySpec keySpec = new SecretKeySpec(hashedKey,0,16,HASHING_ALGO);
  aesKey = factory.generateSecret(keySpec);
 }
 catch(Exception e){
  log.log(Level.WARNING,e.getMessage(),e);
 }
 return aesKey;
}

/**This appends the second byte array to the first one.
 * @param array1
 * @param array2
 * @return array2 appended to array1
 */
private static byte[] append(byte[] array1,byte[] array2){
 byte[] array3 = new byte[array1.length+array2.length];
 int index = 0;
 for(int i=0;i<array1.length;i++){
  array3[index++] = array1[i];
 }
 for(int i=0;i<array2.length;i++){
  array3[index++] = array2[i];
 }
 return array3;
}
}
```

Base64.java ファイル

Base64 クラスを使用して、Base64 エンコード・スキームおよびデコード・スキームのメソッドを提供します。

```
package com.ibm.tamesso.utils;

import java.util.prefs.*;
import java.util.logging.*;
```

```
/**This class provides methods for Base-64 encoding and decoding
 */
public class Base64 {

 private static final String CLASS = Base64.class.getName();
 private static final Logger log=Logger.getLogger(CLASS);

 /**This encodes a byte array into a Base-64 string
  * @param array
  * @return Base-64 encoded string
  */
 public static String encode(byte[] array){
  String methodName_ = "encode";
  log.entering(CLASS, methodName_);
  Preferences prefs = Preferences.userNodeForPackage(Base64.class);
  prefs.putByteArray("key1",array);
  log.exiting(CLASS,methodName_);
  return prefs.get("key1", null);
 }

 /**This decodes a base-64 string into a byte array
  * @param str
  * @return byte array
  */
 public static byte[] decode(String str){
  String methodName_ = "decode";
  log.entering(CLASS, methodName_);
  Preferences prefs = Preferences.userNodeForPackage(Base64.class);
  prefs.put("key2", str);
  log.exiting(CLASS, methodName_);
  return prefs.getByteArray("key2", null);
 }

}
```

付録 C. セキュリティー・トークン・サービス (STS) 汎用ユーザー文書

セキュリティー・トークン・サービス汎用ユーザー (STSUU) 文書は、STS 内の trust モジュール・チェーンを通過する要求の XML 表現です。

STS 汎用ユーザー文書内には、次の 3 つの要素があります。

- Principal
- AttributeList
- RequestSecurityToken

セキュリティー・トークン・サービス (STS) 汎用ユーザー文書について詳しくは、Tivoli Federated Identity Manager のインフォメーション・センター (http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc_6.2.1/ic/ic-homepage.html) を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラット

フォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com® は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open、LTO、LTO ロゴ、Ultrium、および Ultrium ロゴは、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。

用語集

合言葉 (secret). ユーザーしか知らない情報。

アカウント・データ (account data). 認証サービスを検証するために必要なログオン情報。ユーザー名、パスワード、およびログオン情報が格納されている認証サービスが該当する。

アカウント・データ・テンプレート (account data template). 特定の AccessProfile を使用して収集される資格情報用に、保管されるアカウント・データのフォーマットを定義する。

アカウント・データ・バッグ (account data bag). アプリケーションでシングル・サインオンが実行されている間、ユーザー資格情報をメモリーに保持するデータ構造体。

アカウント・データ項目 (account data item). ログオンに必要なユーザー資格情報。

アカウント・データ項目テンプレート (account data item template). アカウント・データ項目のプロパティーを定義する。

アクション (action). プロファイルにおいて、アクションとは、トリガーへの応答として実行できる操作のことである。例えば、サインオン・ウィンドウが表示されるとすぐに、ユーザー名とパスワードの詳細を自動入力するなど。

アクティブ RFID (ARFID) (Active RFID (ARFID)). ARFID は、第 2 要素とプレゼンス検出機能の両方である。これはユーザーのプレゼンスを検出でき、特定のアクションを実行するように AccessAgent を構成することができる。

アクティブな近接バッジ (Active Proximity Badge). ARFID カードとも呼ばれる。RFID カードと似ているが、より遠方の近接リーダーで検出される。

アプリケーション (application). AccessStudio では、認証資格情報を読み取ったり入力したりするためのユーザー・インターフェースを提供するシステムを指す。

アプリケーション・プログラミング・インターフェース (API) (Application Programming Interface (API)). 高水準言語で書かれたアプリケーション・プログラムが、オペレーティング・システムまたは他のプログラムの特定データまたは機能を使用できるようにするためのインターフェース。

アプリケーション・ポリシー (application policies). アプリケーションに割り当てられたポリシー。例えば、パスワード・ポリシー、再認証ポリシー、ログオフ・ポリシーなどがある。

暗号化アプリケーション・プログラミング・インターフェース (CAPI) (Cryptographic Application Programming Interface (CAPI)). Windows オペレーティング・システムのアプリケーション・プログラミング・インターフェースの一つ。暗号化を使用して Windows ベースのアプリケーションを保護するためのツールを開発者に提供する。

暗号化サービス・プロバイダー (CSP) (Cryptographic Service Provider (CSP)). このモジュールは、暗号化機能とスマート・カードに対する CAPI インターフェースを提供する。

暗証番号 (PIN) (Personal Identification Number (PIN)). スマート・カードへのアクセスを保護するパスワードを表す。 PIN は、スマート・カードで使用される。

イベント・コード (event code). 追跡されて ESSO 監査ログ・テーブルに記録される特定の ESSO イベントを表す。

エンタープライズ・ディレクトリー (enterprise directory). IBM Security Access Manager for Enterprise Single Sign-On ユーザーを定義するユーザー・アカウントのディレクトリー。これは、パスワードがエンタープライズ・ディレクトリー・パスワードと同期している場合、ユーザーの資格情報をサインアップおよびログオンのときに検証する。エンタープライズ・ディレクトリーの例としては、Active Directory がある。

エンタープライズ・ユーザー名 (enterprise user name). エンタープライズ・ディレクトリー内のユーザー・アカウントのユーザー名。

カード・シリアル番号 (CSN) (Card Serial Number (CSN)). ハイブリッド・スマート・カードを識別する固有のデータ。スマート・カードにインストールされている証明書とは無関係である。

鍵ストア (keystore). セキュリティーにおいて、ID および秘密鍵が保管されるファイルまたはハードウェア暗号カードで、認証および暗号化に使用する。一部の鍵ストアは、トラステッド鍵つまり公開鍵も含む。

仮想アプライアンス (virtual appliance). 特定用途向けの仮想マシン・イメージであり、仮想化プラットフォームにデプロイされる。

仮想チャネル・コネクター (virtual channel connector). 仮想チャネル・コネクターは、端末サービス環境で使用される。仮想チャネル・コネクターは、クライアント AccessAgent コンポーネントとサーバー AccessAgent コンポーネント間のリモート・セッションを管理するための仮想通信チャネルを確立する。

仮想プライベート・ネットワーク (VPN) (Virtual Private Network (VPN)). 公衆または私設ネットワークの既存のフレームワーク上に拡張した企業イントラネット。VPN は、接続の 2 つの終点間で送信されたデータを確実にセキュアに保つ。

仮想メンバー・マネージャー (VMM) (Virtual Member Manager (VMM)). 基本的な組織エンティティー・データ (個人、ログオン・アカウント、セキュリティー役割など) に安全にアクセスする機能をアプリケーションに提供する WebSphere Application Server コンポーネント。

監査 (audit). ユーザー、管理者、およびヘルプ・デスクのアクティビティーをログに記録するプロセス。

間接認証情報 (indirect auth-info). プロファイルにおいて、間接認証情報とは、既存の認証サービスに対する間接的な参照である。

完全修飾ドメイン名 (FQDN) (fully qualified domain name (FQDN)). ドメイン・ネームのすべてのサブネームを含むホスト・システムの名前。例: ims.example.com。

管理者 (Administrator). この役割を所有すると、各ユーザー、役割、ポリシー、およびレポートを管理できるようになる。管理者は、IMS Server から AccessProfile の作成、アップロードおよびダウンロードを行える。

管理対象ノード (managed node). Websphere Application Server では、デプロイメント・マネージャーに統合されており、ノード・エージェントが組み込まれているノードを指す。このノードには、管理対象サーバーを含めることができる。

基本識別名 (base distinguished name). ディレクトリー・サーバー内の検索開始点を示す。

共有デスクトップ (shared desktop). 複数のユーザーが汎用の Windows デスクトップを共有するデスクトップ・スキーム。

共有ワークステーション (shared workstation). 複数のユーザー間で共有されるワークステーション。

強力なデジタル ID (strong digital identity). 偽名を使用するのが困難な、オンライン上の人物。スマート・カードの秘密鍵で保護されている場合がある。

強力な認証 (strong authentication). 企業の境界の内部および外部の両方で、多要素認証デバイスを使用して機密の企業情報と IT ネットワークへの無許可アクセスを防止するソリューション。

許可コード (authorization code). IBM ヘルプ・デスク・ユーザーが生成する英数字コード。AccessAgent、AccessAssistant、および Web Workplace でパスワード・リセットまたは 2 要素認証バイパスを完了するためにユーザーに提供される。

クライアント AccessAgent (Client AccessAgent). クライアント・マシンにインストールされ、稼働している AccessAgent。

クライアント・ワークステーション、クライアント・マシン、クライアント・コンピューター (client workstation, client machine, client computers). AccessAgent がインストールされているコンピューター。

クラスター (clusters). WebSphere Application Server において、クラスターとはワークロード・バランシングとフェイルオーバーのためにコラボレーションするアプリケーション・サーバーのグループである。

クラスター化 (clustering). WebSphere Application Server では、クラスター化とはアプリケーション・サーバーをグループ化する機能である。

グラフィカル識別および認証 (GINA) (Graphical Identification and Authentication (GINA)). 認証要素に強固に統合されたユーザー・インターフェース、およびパスワードのリセット・オプションと第 2 要素のバイパス・オプションを提供する Windows のダイナミック・リンク・ライブラリー。

グループ・ポリシー・オブジェクト (GPO) (Group Policy Object (GPO)). Active Directory 内のグループ・ポリシー設定のコレクション。グループ・ポリシー・オブジェクトは、グループ・ポリシー・スナップインによって作成される文書である。グループ・ポリシー・オブジェクトはドメイン・レベルで保管され、サイト、ドメイン、および組織単位に含まれているユーザーとコンピューターに影響を及ぼす。

軽量モード (lightweight mode). サーバー AccessAgent のモード。軽量モードで実行すると、Citrix/Terminal

Server 上の AccessAgent のメモリー・フットプリントが削減され、シングル・サインオンの起動の所要時間が短縮される。

公開済みアプリケーション (published application). Citrix XenApp サーバーにインストールされており、Citrix ICA クライアントからアクセス可能なアプリケーション。

公開デスクトップ (published desktop). ユーザーがいつでも、どこでも、どのデバイスからでも、Windows デスクトップ全体に対してリモート・アクセスできる Citrix XenApp の機能。

高可用性 (HA) (high availability (HA)). 事前定義されたサービス・レベルにしたがってすべての停止に耐え、処理機能を提供し続ける、IT サービスの能力。対象となる停止には、保守やバックアップなどの計画されたイベントと、ソフトウェア障害、ハードウェア障害、電源障害、および災害などの計画外のイベントの両方が含まれる。

個人用アプリケーション (personal applications). AccessAgent が資格情報を保管および入力できる Windows および Web ベースのアプリケーション。

個人用アプリケーションの例としては、Web ベースのメール・サイト (企業のメールなど)、インターネット・バンキング・サイト、オンライン・ショッピング・サイト、チャット、インスタント・メッセージング・プログラムなどがある。

個人用デスクトップ (personal desktop). このデスクトップは、他のどのユーザーとも共有されない。

コマンド行ツール (CLT) (Command Line tool (CLT)). 特定のタスクを実行するコマンドを入力することにより、コンピューターのオペレーティング・システムまたはソフトウェアと対話するメカニズム。

サーバー AccessAgent (Server AccessAgent). Microsoft Windows Terminal Server または Citrix サーバーにデプロイされた AccessAgent。

サーバー・ロケーター (server locator). 同じ認証サービスでの認証が必要な、関連する一連の Web アプリケーションをグループ化する。AccessStudio では、サーバー・ロケーターは、アプリケーション画面が関連付けられている認証サービスを識別する。

サービス・プロバイダー・インターフェース (SPI) (Service Provider Interface (SPI)). このインターフェースにより、ベンダーは、シリアル番号を備えた任意のデバイスを IBM Security Access Manager for Enterprise Single Sign-On に統合し、そのデバイスを AccessAgent の第 2 要素として使用できる。

災害復旧 (disaster recovery). サービスとデータを復元して、災害から復旧するプロセス。

災害復旧サイト (disaster recovery site). 2 次的な実稼働環境の場所。

サイレント・モード (silent mode). ユーザーがプログラムと対話しないインストール・モード。ソフトウェアは、スクリプトを通じてインストールされる。

サインアップ (sign-up). IMS Server を使用したアカウントの要求。プロセスの一環として、ユーザーに Wallet が発行される。その後ユーザーは、1 つ以上の第 2 要素を IMS Server に登録できる。

サインオン情報 (sign-on information). セキュア・アプリケーションへのアクセス権をユーザーに付与するために必要な情報。この情報には、ユーザー名、パスワード、ドメイン情報、および証明書が含まれる。

サインオンの自動化 (sign-on automation). アプリケーション・ユーザー・インターフェースと連動してユーザーのサインオン・プロセスを自動化するテクノロジー。

参照ユーザー (lookup user). エンタープライズ・ディレクトリーで自分自身を認証して、他のユーザーを検索するユーザー。IBM Security Access Manager for Enterprise Single Sign-On は、参照ユーザーを使用して、Active Directory または LDAP エンタープライズ・リポジトリーからユーザー属性を取得する。

資格情報 (credentials). ユーザー名、パスワード、証明書、および認証に必要なその他の情報。認証要素は、資格情報としての役割を果たす。IBM Security Access Manager for Enterprise Single Sign-On では、資格情報は Wallet で保管および保護される。

識別名 (distinguished name). ディレクトリー内の項目を一意的に識別する名前。識別名は、属性と値のペアをコンマで区切ったものから構成される。例えば、CN は個人名であり、C は国または地域である。

シグニチャー (signature). プロファイルにおいて、シグニチャーとは、任意のアプリケーション、ウィンドウ、またはフィールドの固有識別情報である。

システム・モーダル・メッセージ (system modal message). 通常、重要なメッセージを表示するために使用されるシステム・ダイアログ・ボックス。システム・モーダル・メッセージが表示されたときは、そのメッセージを閉じるまで、画面上で他に何も選択できない。

質問と合言葉 (secret question and answer). ユーザーしか答えを知らない質問。IBM Security Access Manager for Enterprise Single Sign-On の知識ベース認証の一環として、ユーザーは、いくつかの合言葉について質問される。

自動サインオン (automatic sign-on). ユーザーがサインオン自動化システムにログオンすると、システムがそのユーザーを他のすべてのアプリケーションにログオンさせる機能。

自動取得 (auto-capture). システムが、各種アプリケーションのユーザー資格情報を記憶することを可能にする機能。これらの資格情報は、最初に使用されるときに取り込まれ、今後の使用のために Wallet で保管および保護される。

自動注入 (auto-inject). システムが、ログオン自動化により、各種アプリケーションのユーザー資格情報を自動的に入力することを可能にする機能。

従来のシングル・サインオン (conventional single sign-on). Web ベースのシングル・サインオン・システム。通常は、集中型アーキテクチャーによるサーバー・サイドの統合が必要である。

シリアル ID サービス・プロバイダー・インターフェース (SPI) (Serial ID Service Provider Interface (SPI)). 2 要素認証に使用されるサード・パーティー製のシリアル ID デバイスに AccessAgent を統合するためのプログラマチック・インターフェース。

シリアル番号 (serial number). IBM Security Access Manager for Enterprise Single Sign-On キーに埋め込まれる固有の番号。キーごとに固有であり、変更できない。

シン・クライアント (thin client). ソフトウェアがほとんど、または全くインストールされていないクライアント・マシン。接続先のネットワーク・サーバー上で稼働しているアプリケーションとデスクトップ・セッションにアクセスできる。シン・クライアント・マシンは、ワークステーションのような完全な機能を持つクライアントの代替である。

シングル・サインオン (single sign-on). ユーザー ID とパスワードを 1 回指定するだけで、複数のアプリケーションにアクセスできる機能。

スタンドアロン・サーバー (stand-alone server). 他のすべてのサーバーから独立して管理される、完全に作動可能なサーバー。独自の管理コンソールを使用する。

スタンドアロン・デプロイメント (stand-alone deployment). 独立した WebSphere Application Server プロファイル上に IMS Server をデプロイするデプロイメント・タイプ。

スマート・カード (smart card). スマート・カードは、内蔵回路のネットワークを使用してデータ処理を行うよう作成された、ポケット・サイズのカード。スマート・カードは、アプリケーションから入力を受信することができ、情報を送信することもできる。

スマート・カード・ミドルウェア (smart card middleware). スマート・カード・アプリケーションとスマート・カード・ハードウェア間のインターフェースとして機能するソフトウェア。通常、このソフトウェアは、PKCS#11 を実装したライブラリーと、スマート・カードへの CAPI インターフェースから構成される。

制御 (control). 画面上の任意のフィールド。例えば、Web ページ上のユーザー名のテキスト・ボックスまたは「OK」ボタン。

セキュア・リモート・アクセス (Secure Remote Access). ファイアウォールの外部から、すべてのアプリケーションへの Web ブラウザー・ベースのシングル・サインオンを提供するソリューション。

セキュリティー trust サービス・チェーン（security trust service chain). trust サービス・チェーンは、一緒に使用するように構成されたモジュール・インスタンスのグループである。チェーン内の各モジュール・インスタンスは、順次呼び出され、要求に対する全体の処理の一部として特定の機能を実行する。

セキュリティー・トークン・サービス (STS) (Security Token Service (STS)). セキュリティー・トークンの発行および交換に使用される Web サービス。

セキュリティー担当者 (security officer). ID Wallet セキュリティー・ポリシーおよびその他のアプリケーション・ポリシーを定義する担当者。

セッション管理 (session management). 専用デスクトップおよび共有デスクトップ上でのユーザー・セッションの管理。

セル (cell). WebSphere Application Server において、セルとはデプロイメント・マネージャーと 1 つ以上のノードから構成される仮想的な単位である。

セルフサービス機能 (self-service features). ユーザーがヘルプ・デスクや管理者からの支援をできるだけ受けずに、基本的なタスク (パスワードや合言葉のリセットなど) を行えるようにする IBM Security Access Manager for Enterprise Single Sign-On の機能。

双方向言語サポート (bidirectional language support). テキストが右から左へと表示されるヘブライ語とアラビア語に対するサポート。

ダイナミック・リンク・ライブラリー (DLL) (dynamic link library (DLL)). リンク中ではなく、ロード時または実行時にプログラムにバインドされる実行可能コードおよびデータを含むファイル。DLL 内のコードおよびデータは、複数のアプリケーションで同時に共有できる。このファイルは、Windows プラットフォームにのみ適用される。

対話式グラフィック・モード (interactive graphical mode). 一連のパネルが順次表示され、指示に従って情報を入力していくとインストールが完了する。

直接認証情報 (direct auth-info). プロファイルにおいて、直接認証情報とは既存の認証サービスに対する直接的な参照である。

データ・ソース (data source). アプリケーションがデータベースからデータにアクセスする方法。

データベース (DB) サーバー (Database (DB) server). データベース・マネージャーを使用してソフトウェア・プログラムまたはコンピューターにデータベース・サービスを提供するソフトウェア・プログラム。

データベース・レプリケーション (database replication). 同じデータベースのコピーを複数作成して保持すること。

ディレクトリー (directory). 組織内の個人とリソースに関する情報の構造化リポジトリー。これにより、管理と通信が容易になる。

ディレクトリー・サーバー (directory server). ディレクトリー・サービスをホストするサーバー。

ディレクトリー・サービス (directory service). ネットワーク上のすべてのユーザーおよびリソースの名前、プロファイル情報、マシン・アドレスのディレクトリー。ユーザー・アカウントおよびネットワーク・アクセス権を管理する。ユーザー名を送信すると、そのユーザーの属性が返される。これには、電話番号および E メール・アドレスが含まれる場合がある。ディレクトリー・サービスでは、通常は階層化された設計で素早い検索を可能にする、高度に分化したデータベースを使用する。

デスクトップ・アプリケーション (desktop application). デスクトップで実行されるアプリケーション。

デスクトップ・マネージャー (Desktop Manager). 1 つのワークステーション上の同時ユーザー・デスクトップを管理する。

デプロイメント・マネージャー (deployment manager). 他のサーバーの論理グループ、すなわちセルの操作を管理および構成するサーバー。

デプロイメント・マネージャー・プロファイル (deployment manager profiles). 他のサーバーの論理的なグループまたはセルの運用を管理する WebSphere Application Server ランタイム環境。

透過的画面ロック (transparent screen lock). IBM Security Access Manager for Enterprise Single Sign-On の機能。有効にすると、ユーザーはデスクトップ画面をロックできるが、デスクトップの内容は引き続き参照できる。

登録 (register). IBM Security Access Manager for Enterprise Single Sign-On アカウントにサインアップして、IMS Server に第 2 要素を登録すること。

ドメイン・ネーム・サーバー (DNS) (domain name server (DNS)). ドメイン・ネームを IP アドレスにマップすることにより、名前とアドレス間の変換を提供するサーバー・プログラム。

トラストストア (truststore). セキュリティーにおける記憶オブジェクト (ファイルまたはハードウェア暗号カード)。このオブジェクトには、Web トランザクションの認証に使用するために、信頼できる証明書の形式で公開鍵が保管される。アプリケーションによっては、これらの信頼できる証明書がアプリケーション鍵ストアに移動され、秘密鍵と一緒に保管される。

トリガー (trigger). プロファイルにおいて、トリガーとは、状態エンジン内で状態の遷移を引き起こすイベントである (Web ページのロードや、デスクトップ上のウィンドウの表示など)。

取り消し (revoke). IBM Security Access Manager for Enterprise Single Sign-On において、取り消しとは、ユーザーの認証要素を取り消すこと、またはユーザー自体を取り消すことを指す。

認証局 (CA) (Certificate authority (CA)). デジタル証明書を発行する信頼できる組織または企業。認証局は、通常、固有の証明書を付与された個人の身元を証明する。

認証サービス (authentication service). IBM Security Access Manager for Enterprise Single Sign-On において、アカウントの妥当性を当該アカウントのユーザー・ストアまたは企業のディレクトリーと照合して検証する

サービス。画面に関連付けられた認証サービスを識別する。特定の認証サービスで保存されたアカウント・データが、定義中のログオン画面に取得され、自動入力される。定義されたログオン画面から収集されたアカウント・データは、この認証サービスで保存される。

認証要素 (authentication factor). デジタル ID の検証のために資格情報として必要な、各種デバイス、バイオメトリクス、または合言葉。認証要素の例としては、パスワード、スマート・カード、RFID、バイオメトリクス、およびワンタイム・パスワード・トークンがある。

ネットワーク・デプロイメント (network deployment). クラスター・デプロイメントとも呼ばれる。WebSphere Application Server クラスターに IMS Server をデプロイするデプロイメント・タイプ。

ノード (nodes). Websphere Application Server において、ノード・エージェントと 1 つ以上のサーバー・インスタンスから構成される仮想的な単位を指す。

ノード・エージェント (node agent). Websphere Application Server において、ノード・エージェントは、サーバー・プロセスを作成および終了するプロセスである。ノード・エージェントは、デプロイメント・マネージャーとノード間で構成の同期も行う。

バイオメトリクス (biometrics). ユーザーの身体的特性 (指紋、虹彩、顔、声、または筆跡など) に基づいたユーザーの識別。

ハイブリッド・スマート・カード (hybrid smart card). 公開鍵暗号チップと RFID チップを搭載した、ISO-7816 準拠のスマート・カード。暗号チップは、接触インターフェースを通じてアクセスできる。RFID チップは、非接触 (RF) インターフェースを通じてアクセスできる。

バインド識別名 (bind distinguished name). アプリケーション・サーバーがディレクトリー・サービスに接続するために使用する資格情報を指定する。ディレクトリー内の項目を一意的に識別する識別名。*「識別名 (distinguished name)」*も参照。

パスワード・エージング (password aging). ユーザーが自分のパスワードを変更できる頻度を示す。

パスワード入力オプション (password entry option). AccessAgent のパスワード入力方法に関するオプション。「自動ログオン」、「常時」、「尋ねる」、「何もしない)」を選択できる。

パスワードの複雑さ (password complexity). パスワードの最小長と最大長、英数字の最小数、および大/小文字の混在の可否を示す。

秘密鍵 (private key). 所有者によって秘密のままにされる、暗号化または暗号化解除の鍵。公開鍵暗号方式で暗号化および暗号化解除に使用される 2 つの鍵のペアのうちの 1 つである。

フィックスパック (fix pack). スケジュールされたリフレッシュ・パック、マニュファクチャリング・リフレッシュ、またはリリースの間で使用可能になったフィックスの累積の集合。これは、お客様が特定の保守レベルに移行できることを目的としている。

フェイルオーバー (failover). ソフトウェア、ハードウェア、またはネットワークの中断が発生した場合に冗長システムまたはスタンバイ・システムにシステムを切り替える自動操作。

プライベート・デスクトップ (private desktop). このデスクトップ・スキームでは、ユーザーはワークステーション内に各自の Windows デスクトップを持つ。前のユーザーがワークステーションに戻ってアンロックすると、AccessAgent は前のユーザーのデスクトップ・セッションに切り替え、最後に行われていたタスクを再開する。

プレゼンス検出機能 (presence detector). このデバイスをコンピューターに設置すると、ユーザーがコンピューターを離れたことが検出されるようになる。このデバイスにより、短時間コンピューターから離れるときに、コンピューターを手動でロックする必要がなくなる。

プロビジョニング API (Provisioning API). IBM Security Access Manager for Enterprise Single Sign-On がユーザーのプロビジョニング・システムと統合できるようにするインターフェース。

プロビジョニング・システム (provisioning system). エンタープライズ内のアプリケーション・ユーザーの ID ライフ・サイクル管理を提供し、それらのユーザーの資格情報を管理するシステム。

プロビジョニング・ブリッジ (provisioning bridge). SOAP 接続を使用する API ライブラリーを使用して、サード・パーティー・プロビジョニング・システムによる IMS Server 資格情報の配布プロセスを自動化する。

プロビジョン (provision). サービス、コンポーネント、アプリケーション、またはリソースを提供、デプロイ、および追跡すること。

プロビジョン解除 (deprovision). サービスまたはコンポーネントを削除すること。例えば、アカウントのプロビジョン解除はリソースからのアカウントの削除を意味する。

分散 IMS Server (distributed IMS Server). IMS Server は、複数の地域にデプロイされる。

ヘルプ・デスク役割 (Help desk role). IBM Security Access Manager for Enterprise Single Sign-On ユーザーの特定のグループを管理する権限を所有者に付与する役割。ヘルプ・デスクのタスクとしては、パスワードのリセット、許可コードの発行、ユーザーのアクセス権の取り消しなどがある。

ポータル (portal). カスタマイズおよび個別設定できる、さまざまな情報、アプリケーション、および個人にアクセスするための単一の安全なポイント。

ホスト名 (host name). インターネット通信において、コンピューターに与えられる名前。ホスト名は、完全修飾ドメイン・ネーム (例: `mycomputer.city.company.com`) あるいは特定のサブネーム (例: `mycomputer`) を指定できる。

ホット・キー (hot key). 異なるアプリケーション間、またはアプリケーションの異なる機能間で操作をシフトするために使用されるキー・シーケンス。

ポリシー (policy). IBM Security Access Manager for Enterprise Single Sign-On Enterprise の運用を管理するためのもの。

ポリシー・テンプレート (policy template). ユーザーが固定ポリシー要素と可変ポリシー要素を指定することによりポリシーを定義できる、事前定義のポリシー・フォーム。これには、マシン・ポリシー・テンプレート、ユーザー・ポリシー・テンプレート、およびシステム・ポリシー・テンプレートがある。

マシン登録/サインアップ (machine registration / sign up). サービスを使用するためにマシンを ISAM ESSO に登録するプロセス。

無線による個体識別 (RFID) (Radio Frequency Identification (RFID)). 製品のシリアル番号を、タグからスキャナーへ、人が介入することなく伝送する無線テクノロジー。

モバイル認証 (mobile authentication). モバイル・ユーザーが、ネットワークのどこからでも企業のリソースに安全にサインオンできるようにする認証要素。IBM Security Access Manager for Enterprise Single Sign-On では、Wallet およびその他のエンタープライズ・アプリケーション用のオプションの認証要素が提供される。例えば、SMS やショート・メッセージング・サービスを通じて、IBM Security Access Manager for Enterprise Single Sign-On ActiveCode がモバイル・デバイスに送信される。

ユーザー・プロビジョニング (user provisioning). IBM Security Access Manager for Enterprise Single Sign-On を使用するためにユーザーをサインアップすること。

ユーザー・プロビジョン解除 (user deprovisioning). IBM Security Access Manager for Enterprise Single Sign-On からユーザー・アカウントを削除すること。

ユーザー資格情報 (user credential). ユーザー、グループ関連付け、またはその他のセキュリティー関連の識別属性を記述する情報。認証中に取得され、権限付与、監査、委任などのサービスを実行するために使用される。例えば、ユーザー ID とパスワードは、ネットワークおよびシステム・リソースへのアクセスを可能にする資格情報である。

ユーザー登録/サインアップ (user registration / sign up). サービスを使用するためにユーザーをシステムに登録するプロセス。

ユーザーの簡易切り替え (fast user switching). アプリケーションを終了してログアウトしなくても、単一のワークステーション上で複数のユーザー・アカウントを切り替えることのできる機能。

ユーザー役割 (user role). サインオン自動化のため AccessAgent を使用するのに必要な役割。この役割は、IBM Security Access Manager for Enterprise Single Sign-On システムにおいて、3 つある事前定義済みの IBM Security Access Manager for Enterprise Single Sign-On 役割の一つである。

有効範囲 (scope). IBM Security Access Manager for Enterprise Single Sign-On では、ポリシーが適用される範囲を指す。システム、ユーザー、またはマシン・レベルに設定できる。

ランダム・パスワード (random passwords). 生成されるパスワードで、クライアントとサーバー間の認証セキュリティーを強化する。ランダム・パスワードの変更とは、クライアントとサーバー間のアクセス・コードをランダムな文字列を使用して変更するプロセスである。

この変更は、クライアントとサーバーがセキュアなセッションを共有している場合にのみ実行できる。次回クライアントがサーバーにアクセスする必要があるときは、新しいランダム・パスワードを使用して、セキュアなセッションを再確立できる。

リモート・デスクトップ・プロトコル (RDP) (Remote Desktop Protocol (RDP)). Windows ベースのサーバー・アプリケーションのリモート表示および入力をネットワーク接続を介して容易に行うことができるようにするプロトコル。RDP は、さまざまなネットワーク・トポロジーと、複数の接続をサポートしている。

ルート認証局 (CA) (root certificate authority (CA)). 認証局階層の最上部の認証局。証明書所有者の ID の正真性を証明する。

レジストリー (registry). マシン・ポリシーは、通常 AccessAdmin で構成されるが、必要であれば Windows レジストリーで構成することもできる。特に pid_machine_policy_override_enabled ポリシーが「はい」に設定されている場合はこの構成を使用する。「はい」は、管理者が Windows レジストリーを使用してマシン・ポリシーを変更しなければならないことを意味する。

レジストリー・ハイブ (registry hive). Windows システムで、レジストリーに保管されているデータの構造。

連邦情報処理標準 (FIPS) (Federal Information Processing Standard (FIPS)). 米国連邦情報・技術局が作成した標準規格。

ロード・バランサー (load balancer). ネットワークまたはアプリケーションのトラフィックを多数のサーバーに分散するハードウェアまたはソフトウェア。

ロード・バランシング (load balancing). アプリケーション・サーバーのモニター、およびサーバー上のワークロード管理。1 つのサーバーがそのワークロードを超えると、より能力の高い別のサーバーに要求が転送される。

ワンタイム・パスワード (OTP) (One-Time Password (OTP)). 認証イベントのために生成される、1 回限りのパスワード。クライアントとサーバーの間で、セキュアなチャネルを介して伝達されることがある。

1 次認証要素 (primary authentication factor). IBM Security Access Manager for Enterprise Single Sign-On のパスワード、またはディレクトリー・サーバーの資格情報。

2 要素認証 (two-factor authentication). ユーザー認証時に 2 要素を使用すること。例えば、AccessAgent へのログオン時にパスワードと RFID カードを使用すること。

AccessAdmin. 管理者とヘルプ・デスク担当者が IMS Server の管理、およびユーザーとポリシーの管理のために使用する、Web ベースの管理コンソール。

AccessAgent. ユーザー ID の管理、ユーザーの認証、およびシングル・サインオン/サインオフの自動化を行うクライアント・ソフトウェア。

AccessAgent プラグイン (AccessAgent plug-in). 条件のカスタム検査またはカスタム・アクションの実行のために AccessProfile 内に埋め込まれる、VBscript または Javascript で作成されたスクリプトの断片。AccessProfile の機能を組み込みトリガーおよびアクションを超えて拡張するために使用される。

AccessAssistant. ユーザーが自分のパスワードをリセットしたりアプリケーションの資格情報を取得したりするために役立つ、Web ベースのインターフェース。

AccessProfiles. AccessAgent は、この XML 仕様を使用して、シングル・サインオンと自動化を実行できるアプリケーション画面を識別する。

AccessStudio. 管理者が AccessProfile の作成と保守のために使用するアプリケーション。

Active Directory (AD). ネットワーク全体の安全な集中管理を可能にする階層ディレクトリー・サービス。Microsoft Windows プラットフォームの中心構成要素である。

Active Directory 資格情報 (Active Directory credentials). Active Directory ユーザー名およびパスワード。

Active Directory パスワードの同期 (Active Directory password synchronization). ISAM ESSO パスワードと Active Directory パスワードを同期させる IBM Security Access Manager for Enterprise Single Sign-On の機能。

ActiveCode. IBM Security Access Manager for Enterprise Single Sign-On で生成および検証される、一時的な認証コード。 ActiveCode には、Mobile ActiveCode と Predictive ActiveCode の 2 種類がある。

Mobile ActiveCode は、IBM Security Access Manager for Enterprise Single Sign-On で生成され、ユーザーの携帯電話または E メール・アカウントにディスパッチされる。Predictive ActiveCode (ワンタイム・パスワード) は、ユーザーがボタンを押したときに OTP トークンから生成される。

代替チャネルまたはデバイスと結合すると、ActiveCodes は有効な第 2 認証要素を提供する。

Clinical Context Object Workgroup (CCOW) (Clinical Context Object Workgroup (CCOW)). 健康管理業界において臨床アプリケーション間で情報を交換するための、ベンダーに依存しない標準。

DB2®. リレーショナル・データベース管理用の IBM ライセンス・プログラム・ファミリー。

Enterprise Single Sign-On (ESSO). ユーザー名および関連する資格情報 (パスワードなど) を 1 回指定するだ

けで、エンタープライズにデプロイされたすべてのアプリケーションにログオンできるメカニズム。

ESSO GINA. 以前は、Encentuate GINA (EnGINA) と呼ばれていた。IBM Security Access Manager for Enterprise Single Sign-On GINA には、認証要素に統合されたユーザー・インターフェースと、パスワードのリセット・オプションおよび第 2 要素のバイパス・オプションがある。

ESSO 監査ログ (ESSO audit logs). システム・イベントおよび応答のレコードが含まれるログ・ファイル。ESSO 監査ログは IMS データベースに保管される。

ESSO 資格情報 (ESSO credentials). ISAM ESSO ユーザー名およびパスワード。

ESSO 資格情報プロバイダー (ESSO Credential Provider). これは、Windows Vista と Windows 7 用の IBM Security Access Manager for Enterprise Single Sign-On GINA である。以前は、Encentuate 資格情報プロバイダー (EnCredentialProvider) と呼ばれていた。

ESSO ネットワーク・プロバイダー (ESSO Network Provider). 以前は、Encentuate ネットワーク・プロバイダー (EnNetworkProvider) と呼ばれていた。 Active Directory サーバーの資格情報を収集し、それらの資格情報を使用してユーザーを自動的に Wallet にログオンさせる AccessAgent モジュール。

ESSO パスワード (ESSO password). ユーザー Wallet へのアクセスを保護するパスワード。

IBM HTTP Server. Web サーバー。IBM は、IBM HTTP Server という名前の Web サーバーを提供している。この Webサーバーは、クライアントからの要求を受け入れて、アプリケーション・サーバーに転送する。

ID Wallet (identity wallet). ユーザーのアクセス資格情報および関連する情報 (ユーザー ID、パスワード、証明書、暗号鍵など) を保管する、保護されたデータ・ストア。Wallet とは、ID Wallet である。

IMS Server. エンタープライズのセキュア・アクセス管理の中心点を提供する ISAM ESSO の統合管理システム。これにより、ユーザー ID、AccessProfiles、認証ポリシーの中央管理が可能になり、エンタープライズの損失管理、証明書管理、および監査管理を行える。

IMS Server 証明書 (IMS Server Certificate). IBM Security Access Manager for Enterprise Single Sign-On で使用される。IMS Server 証明書を使用すると、クライアントは、IMS Server を識別して認証することができる。

IMS 構成ウィザード (IMS Configuration Wizard). インストール時、管理者はこのウィザードを使用して IMS Server を構成する。

IMS 構成ユーティリティー (IMS Configuration Utility). 管理者が IMS Server の下位レベル構成設定を管理できるようにする、IMS Server のユーティリティー。

IMS コネクター (IMS Connector). IMS™ を外部システムに接続して、モバイル・アクティブ・コードをメッセージング・ゲートウェイにディスパッチするためのモジュール。

IMS データ・ソース (IMS data source). IMS データベースにアクセスするための場所とパラメーターを定義した Websphere Application Server の構成オブジェクト。

IMS データベース (IMS Database). IMS Server が ESSO システム、マシン、およびユーザーに関するデータと監査ログをすべて保管するリレーショナル・データベース。

IMS ブリッジ (IMS Bridge). プロビジョニングなどを目的として IMS API を呼び出すために、サード・パーティー製のアプリケーションやシステムに埋め込まれるモジュール。

IMS ルート CA (IMS Root CA). AccessAgent と IMS Server 間のトラフィックを保護するための証明書に署名するルート認証局。

IP アドレス (IP address). インターネット・プロトコルの標準規格を使用する、ネットワーク上のデバイスまたは論理装置の固有のアドレス。

iTag. すべての写真入りバッジまたは個人用オブジェクトを、近接型デバイス (強力な認証のために使用可能) に変換できる、特許出願中のテクノロジー。

Java Management Extensions (JMX). Java テクノロジーを介して Java テクノロジーの管理を行う手段のこと。JMX は、管理用の Java プログラミング言語のユニバーサルかつオープンな拡張機能であり、管理が必要とされるすべての業界でデプロイできる。

Java 仮想マシン (JVM) (Java Virtual Machine (JVM)). コンパイルされた Java コード (アプレットおよびアプリケーション) を実行するプロセッサーのソフトウェア実装。

Java ランタイム環境 (JRE) (Java Runtime Environment (JRE)). 標準的な Java プラットフォームを構成する中核の実行可能プログラムおよびファイルを含む Java Developer Kit のサブセット。 JRE には、

Java 仮想マシン (JVM)、コア・クラス、およびサポート・ファイルが組み込まれている。

Lightweight Directory Access Protocol (LDAP). TCP/IP を使用して X.500 モデルをサポートするディレクトリーにアクセスできるようにするオープン・プロトコル。LDAP を使用して、インターネットまたはイントラネット・ディレクトリー内の個人、組織、その他のリソースを見つけることができる。

Microsoft Cryptographic Application Programming Interface (CAPI). スマート・カードへのアクセスが可能であり、暗号機能を備えたモジュール用の、Microsoft によるインターフェース仕様。

Mobile ActiveCode (MAC). Web Workplace や AccessAssistant などのアプリケーションで 2 要素認証のためにユーザーが使用するワンタイム・パスワード。この OTP は、ランダムに生成され、SMS または E メールを通じてユーザーにディスパッチされる。

OTP トークン. デジタル・システムまたは物理資産 (あるいはその両方) へのアクセスを許可するために所有者が持っている、小型で携帯性に優れたハードウェア・デバイス。

PKCS#11. RSA 研究所が定義した Public Key Cryptography Standard 11 は、暗号機能を実行するデバイス (スマート・カードなど) に対するインターフェースである。

RADIUS. Remote Authentication Dial-In User Service。

Secure Sockets Layer (SSL). 通信プライバシーを提供するセキュリティー・プロトコル。 SSL を使用すると、クライアント/サーバー・アプリケーションは、盗聴、改ざん、およびメッセージ偽造を防ぐように設計された方法で通信することができる。

Simple Mail Transfer Protocol (SMTP). インターネット・ユーザー間でメールの転送を行うための インターネット・アプリケーション・プロトコル。

Simple Object Access Protocol (SOAP). XML ベースのメッセージをコンピューター・ネットワーク上で交換するためのプロトコルで、通常は HTTP を使用する。SOAP は、Web サービス・スタックのファウンデーション層を形成し、より抽象的な層を構築できる基本的なメッセージング・フレームワークを提供する。

SSL 仮想プライベート・ネットワーク (SSL VPN) (Secure Sockets Layer virtual private network (SSL VPN)). 標準の Web ブラウザーで使用できる形式の VPN。

SSO 項目 (SSO-items). AccessAgent が資格情報を収集または自動入力する際の、情報の取得元の画面上のフィールド。「情報項目 (info-items)」も参照。

Tivoli Common Reporting ツール (Tivoli Common Reporting Tool). レポートの作成、カスタマイズ、および管理が可能なレポート作成コンポーネント。

Tivoli Identity Manager アダプター (Tivoli Identity Manager Adapter). IBM Security Access Manager for Enterprise Single Sign-On が Tivoli Identity Manager と通信できるようにする仲介ソフトウェア・コンポーネント。

trust サービス・チェーン (trust service chain). さまざまなモード (検証、マップ、発行など) で動作するモジュールのチェーン。

TTY. 端末エミュレーター、端末アプリケーション。他のディスプレイ・アーキテクチャー内でビデオ端末をエミュレートするプログラム。「端末」という用語は通常、コマンド行シェルまたはテキスト端末の同義語として使われるが、この用語はグラフィカル・インターフェースを含むすべてのリモート端末を表すこともある。通常、グラフィカル・ユーザー・インターフェース内の端末エミュレーターは、端末ウィンドウと呼ばれる。

Uniform Resource Identifier. 抽象的または物理的なリソースを識別するための簡潔な文字ストリング。

Visual Basic (VB). Microsoft が提供するイベント・ドリブン・プログラミング言語および統合開発環境 (IDE)。

Wallet. ユーザーのアクセス資格情報および関連情報 (ユーザー ID、パスワード、証明書、暗号鍵など) を保管する ID Wallet。それぞれが、ユーザーの個人用メタディレクトリーとして機能する。資料には、ID Wallet、ユーザー Wallet、マシン Wallet、資格情報 Wallet などが記載されている。

Wallet キャッシング (Wallet caching). アプリケーションに対してシングル・サインオンを実行する場合、AccessAgent は、ユーザー資格情報 Wallet からログオン資格情報を取得する。ユーザー資格情報 Wallet は、ユーザー・マシンのキャッシュに格納されると共に、IMS Server にも安全に保管される。そのため、ユーザーは、後で別のマシンから IBM Security Access Manager for Enterprise Single Sign-On にログオンしたときでも、自分の Wallet にアクセスできる。

Wallet パスワード (Wallet Password). Wallet へのアクセスを保護するパスワード。

Wallet マネージャー (Wallet manager). ユーザーが個人用 ID Wallet でアプリケーション資格情報を管理できるようにする IBM Security Access Manager for Enterprise Single Sign-On GUI コンポーネント。

Web Workplace. アプリケーションごとにパスワードを入力しなくても、リンクをクリックするだけでエンタープライズ Web アプリケーションにログオンできる Web ベースのインターフェース。このインターフェースは、ユーザーの既存のポータルまたは SSL VPN と統合できる。

Web サーバー (web server). Hypertext Transfer Protocol (HTTP) 要求を管理できるソフトウェア・プログラム。IBM は、Apache をベースとした IBM HTTP Server と呼ばれる Web サーバーを提供している。

Web サービス (web service). 内蔵タイプの自己記述型モジュラー・アプリケーションであり、標準のネットワーク・プロトコルを使用することによりネットワークを介して公開、検出、および起動できる。通常、データは、XML を使用してタグ付けされる。データを転送するときは、SOAP が使用される。使用可能なサービスについて記述するときは WSDL が使用され、使用可能なサービスをリストするときは UDDI が使用される。

WebSphere Application Server. e-ビジネス・アプリケーションのデプロイ、統合、実行、および管理が可能な、Web サーバー上で稼働するソフトウェア。

WebSphere Application Server プロファイル (WebSphere Application Server profile). WebSphere Application Server の管理者のユーザー名とプロファイル。ランタイム環境を定義する。

WebSphere 管理コンソール (WebSphere Administrative console). 管理サーバー内のリソース Bean に対するメソッド呼び出しを作成してドメイン内のリソースへのアクセスまたはリソースの変更を行う、グラフィカルな管理用 Java アプリケーション・クライアント。

Windows Terminal Services. リモート・コンピューター上のアプリケーションとデータにネットワーク経由でアクセスするためにユーザーが使用する Microsoft Windows コンポーネント。

Windows ネイティブのユーザーの簡易切り替え (Windows native fast user switching). 複数のユーザー・アカウントを迅速に切り替えることができる Windows XP の機能。

Windows ログオン画面、Windows ログオン UI モード (Windows logon screen, Windows logon UI mode). Windows デスクトップにログオンするために、ユーザーが自分のユーザー名とパスワードを入力する画面。

WS-Trust. トラスト・モデルのフレームワークを定義して Webサービス間のトラストを確立する、Web サービス・セキュリティー仕様。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ハ行]

[マ行]

[ヤ行]

[ラ行]

A

B

C

E

J

P

R

S

T

W

Printed in Japan

SA88-4639-00

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21