HP Universal CMDB

Windows および Red Hat Enterprise Linux オペレーティング・システム向け

ソフトウェア・アバージョン: 10.00

Hardening the HP Universal CMDB and Configuration Manager

ドキュメント・リリース日: 2012 年 6 月 (英語版)

ソフトウェア・リリース日: 2012 年 6 月 (英語版)

ご注意

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピュータソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピュータソフトウェア、コンピュータソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

商標について

Adobe™は、Adobe Systems Incorporated (アドビシステムズ社) の商標です。

Microsoft®およびWindows®は、米国におけるMicrosoft Corporationの登録商標です。

UNIX®は、The Open Groupの登録商標です。

本製品には 'zlib' 汎用圧縮ライブラリのインタフェースが使用されています。'zlib': Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアのバージョン番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

最新の更新のチェック、またはご使用のドキュメントが最新版かどうかのご確認には、次のサイトをご利用ください。

http://support.openview.hp.com/selfsolve/manuals

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの取得登録は、次のWebサイトから行なうことができます。

http://h20229.www2.hp.com/passport-registration.html(英語サイト)

または、HP Passport のログインページの [**New users - please register**] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。

http://www.hp.com/go/hpsoftwaresupport

HPソフトウェアが提供する製品、サービス、サポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様の業務の管理に必要な対話型の技術支援ツールに素早く効率的にアクセスいただけます。HPソフトウェアサポートWebサイトのサポート範囲は次のとおりです。

- 関心のある技術情報の検索
- サポートケースとエンハンスメント要求の登録とトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HP サポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部を除き、サポートのご利用には、HP Passportユーザとしてご登録の上、ログインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDの登録は、次の場所で行います。

http://h20229.www2.hp.com/passport-registration.html(英語サイト)

アクセスレベルに関する詳細は、以下のWebサイトにアクセスしてください。

http://h20230.www2.hp.com/new_access_levels.jsp

目次

第1章

強化の紹介

本章の内容

強化の概要

本項では，セキュリティで保護されたHP Universal CMDBアプリケーションの概念について紹介し，セキュリティを実装するために必要な計画とアーキテクチャについて説明します。セキュリティ強化について説明する次のセクションに進む前に，このセクションを読むことを強くお勧めします。

HP Universal CMDBは，セキュリティ保護アーキテクチャの一部となるように設計されており，さらされる可能性のあるセキュリティ上の脅威に対処するという課題に対応しています。

セキュリティ強化のガイドラインでは，より安全な(セキュリティ強化された)HP Universal CMDBの実装に必要な設定作業について取り上げます。

提供されるセキュリティ強化の情報は主に，セキュリティ強化の手順を開始する前にセキュリティ強化の設定と推奨事項について精通しているHP Universal CMDB管理者を対象としています。

HP Universal CMDBにリバース・プロキシを組み合わせて使用して，セキュリティ保護アーキテクチャを実現することを強くお勧めします。HP Universal CMDBでリバース・プロキシを使うための設定方法の詳細については，32ページ「リバース・プロキシの使用」を参照してください。

HP Universal CMDBで，このドキュメントで説明する以外の別の安全なアーキテクチャを使用する必要がある場合は，どのアーキテクチャが最良かを決定するためにHPソフトウェア・サポートにお問い合わせください。

Data Flow Probeのセキュリティ強化の詳細については，59ページ「データ・フロー・プローブの強化」を参照してください。

注：

- このセキュリティ強化の手順は，これらの章で提供された指示だけを実装しており，ほかで記述されているセキュリティ強化の手順は実行していないことを前提としています。
- セキュリティ強化手順が，特定の分散アーキテクチャを前提としている場合であっても，そのアーキテクチャがユーザの組織のニーズに合う最適なアーキテクチャであるとは限りません。
- 以降の章に含まれる手順は，HP Universal CMDB 専用のマシンで実行されることを想定しています。コンピュータを HP Universal CMDB 以外の用途に使用した場合，問題が生じることがあります。
- 本項に示すセキュリティ強化に関する情報は，ご利用のコンピュータ・システムのセキュリティ・リスク評価を行うためのガイドラインを意図したものではありません。

強化の準備

- 利用するネットワーク全体のセキュリティ上の危険やセキュリティの状態の評価を行い，その評価結果に基づいて，HP Universal CMDB をネットワークに最適な形で統合する方法を判断します。
- HP Universal CMDB の技術フレームワークと HP Universal CMDB のセキュリティ機能についてよく理解してください。
- セキュリティ強化ガイドラインのすべての内容に目を通します。
- HP Universal CMDB が完全に機能していることを確認してから，セキュリティ強化手順を開始します。
- セキュリティ強化手順は，各章に記載されている順序どおりに実行します。たとえば，HP Universal CMDB サーバで SSL をサポートするには，17ページ「Secure Sockets Layer(SSL) 通信の有効化」をまず読んでから，記載されている順序のとおりに手順を実行します。
- HP Universal CMDB では，空のパスワードでの基本認証をサポートしていません。基本認証の接続パラメータを設定するときは，パスワードを省略しないでください。

ヒント： セキュリティ強化手順を印刷して，作業時に確認しながら作業してください。

安全なアーキテクチャへの UCMDB のデプロイ

HP Universal CMDB サーバを安全にデプロイするのに，いくつかの方法を推奨します。

- **ファイアウォールを使用する DMZ アーキテクチャ**

  ここでのセキュリティ保護アーキテクチャとは，デバイスをファイアウォールとして使用する典型的な DMZ アーキテクチャのことです。アーキテクチャの基本的な概念は，HP Universal CMDB クライアントと HP Universal CMDB サーバを完全に切り離し，これらの間の直接アクセスをなくすことです。

- **安全なブラウザ**

  Windows 環境での Internet Explorer および Firefox は，Java スクリプト，アプレット，cookie を安全に処理するよう設定されている必要があります。

- **SSL 通信プロトコル**

  Secure Sockets Layer プロトコルは，クライアントとサーバ間の通信を保護します。SSL 通信を必要とする URL は，ハイパーテキスト転送プロトコルのセキュリティ保護されたバージョン (HTTPS) を使用します。詳細については，17ページ「Secure Sockets Layer(SSL) 通信の有効化」を参照してください。

- **リバース・プロキシ・アーキテクチャ**

  より安全で推奨されるソリューションの 1 つは，リバース・プロキシを使用して HP Universal CMDB をデプロイすることです。HP Universal CMDB は安全なリバース・プロキシ・アーキテクチャを完全サポートしています。詳細については，32ページ「リバース・プロキシの使用」を参照してください。

システムのアクセス権

Java JMX Access の強化

> **注：** ここに記載の手順は Data Flow Probe JMX にも使用できます。

ユーザ資格情報を提供する場合のみに JMX RMI ポートへのアクセスを可能にするには，次の手順を実行してください。

1. **C:\hp\UCMDB\UCMDBServer\bin** にあるサーバ上の **wrapper.conf** ファイルで次を設定します。

   wrapper.java.additional.16=-Dcom.sun.management.jmxremote.authenticate=true

   この設定では，JMX が認証を要求する必要があります。

   - **Data Flow Probe JMX** の場合は，次を実行します。

     C:\hp\UCMDB\DataFlowProbe\bin にある **WrapperGateway.conf** および **WrapperManager.conf** のファイルで次の設定を行います。

     wrapper.java.additional.17=-Dcom.sun.management.jmxremote.authenticate=true

2. **jmxremote.password.template**(場所：**C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**) の名前を **jmxremote.password** に変更します。

   > **注：** Data Flow Probe JMX の場合，このファイルは **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management** にあります。

3. **jmxremote.password** で **monitorRole** および **controlRole** の各ルールのパスワードを追加します。

   例：

   monitorRole QED

   controlRole R&D

ではパスワード **QED** が **monitorRole** に，パスワード **R&D** が **controlRole** に割り当てられます。

> **注：** パスワードがクリア・テキストで保存されますので，必ず所有者のみが **jmxremote.password** を読み書きするようにしてください。ファイル所有者は，必ず UCMDB を実行しているユーザと同じでなければなりません。

4. **jmxremote.access**(場所：**C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**) で **monitorRole** および **controlRole** にアクセス権を割り当てます。

   例：

   monitorRole 読み取り専用

   controlRole 読み書き

   では読み取り専用アクセス権が **monitorRole** に，読み書きアクセス権が **controlRole** に割り当てられます。

   > **注：** Data Flow Probe JMX の場合，このファイルは **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management** にあります。

5. ファイルは次のようにしてセキュアにします。

   - **Windows のみの場合：**コマンド・ラインから次のコマンドを実行してファイルをセキュアにします。

     cacls jmxremote.password /P <username>:F

     cacls jmxremote.access /P <username>:R

     この場合，**<username>** は両方のファイルのプロパティで確認できるファイル所有者です。これらのファイルのプロパティを開いて，内容が正しいことと，所有者が 1 人しかいことを確認してください。

   - **Solaris および Linux オペレーティング・システムの場合：**次を実行してパスワードのファイル権限を設定します。

     chmod 600 jmxremote.password

6. **サービス・パックのアップグレード，サーバ移行および障害回復の場合 :jmxremote.access**(場所 **C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**) ファイルの所有権を，アップグレードまたは移行インストールを実行するオペレーティング・システムのユーザに変更します。

   > **注：** Data Flow Probe JMX の場合，このファイルは **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management** にあります。

JMX コンソールのシステム・ユーザ名またはパスワードの変更

JMX コンソールでは，マルチ顧客環境での顧客間ユーザであるシステム・ユーザが使用されます。JMX コンソールには，任意のシステム・ユーザ名でログインできます。標準設定の名前およびパスワードは **sysadmin/sysadmin** です。

パスワードは，JMX コンソールまたはサーバ管理ツールで変更できます。

JMX コンソールで標準設定のシステム・ユーザ名またはパスワードを変更するには，次の手順を実行します。

1. Web ブラウザを起動し，アドレスに**http://localhost.<domain_name>:8080/jmx-console** を入力します。
2. JMX コンソールの認証資格情報を入力します。標準設定値は次のとおりです。
   - ログイン名 = **sysadmin**
   - パスワード = **sysadmin**
3. **UCMDB:service=Authorization Services** を見つけ，リンクをクリックして［操作］ページを開きます。
4. **resetPassword** 操作を見つけます。
   - ［**ユーザ名**］フィールドに **sysadmin** を入力します。
   - ［**パスワード**］フィールドに新しいパスワードを入力します。
5. ［**Invoke**］をクリックして変更内容を保存します。

サーバ管理ツールで標準設定のシステム・ユーザ名またはパスワードを変更するには，次の手順を実行します。

1. **Windows の場合，C:\hp\UCMDB\UCMDBServer\tools\server_management.bat** というファイルを実行します。

   Linux の場合，**/opt/hp/UCMDB/UCMDBServer/tools/** というフォルダにある **server_management.sh** を実行します。
2. 認証資格情報 **sysadmin/sysadmin** を使用してツールにログインします。
3. ［ユーザ］リンクをクリックします。
4. システム・ユーザを選択し，［**ログオン ユーザのパスワードを変更**］をクリックします。
5. 古いパスワードと新しいパスワードを入力し，［**OK**］をクリックします。

HP Universal CMDB Server サービス・ユーザの変更

Windows プラットフォームでは，HP Universal CMDB サービス(すべての HP Universal CMDB サービスとプロセスを実行します) は，Server and Database Configuration ユーティリティの実行時にインストールされます。標準設定では，このサービスは local system ユーザのもとで実行されます。しかし，別のユーザがサービスを実行するように割り当てる必要がある場合があります (NTLM 認証を使用している場合など) 。

サービスを実行するように割り当てるユーザは，次の権限を持っている必要があります。

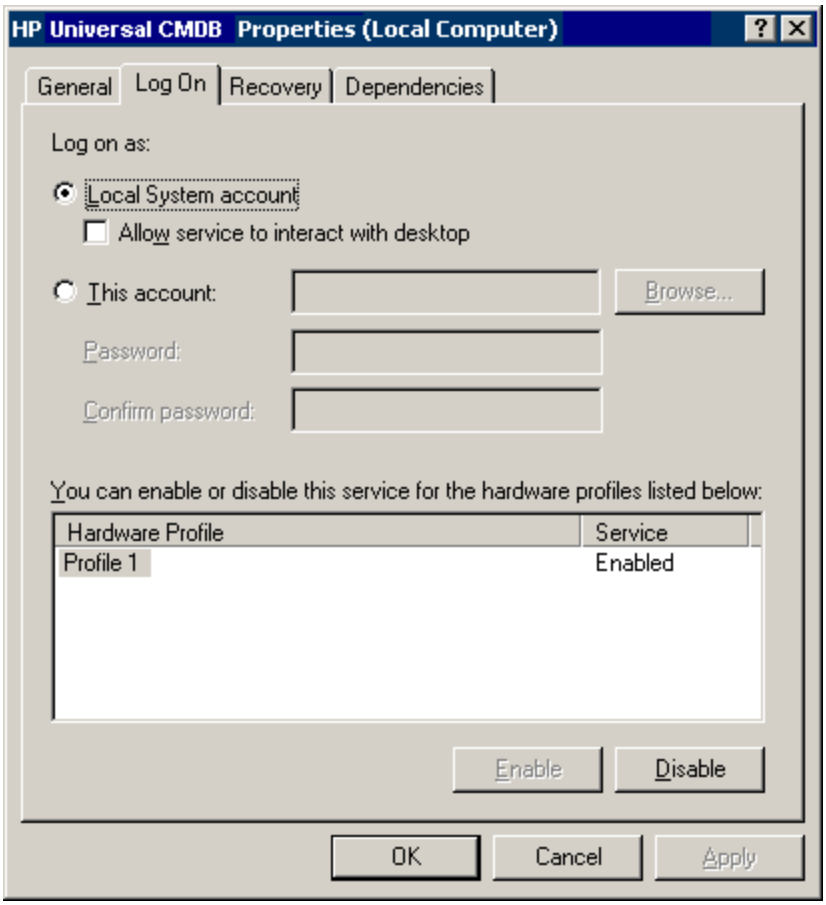
- 十分なデータベース権限 (データベース管理者によって定義されます)
- 十分なネットワーク権限
- ローカル・サーバでの管理者権限

サービス・ユーザを変更するには，次の手順を実行します。

1. ［スタート］メニュー(**［スタート］>［すべてのプログラム］>［HP UCMDB］>［HP Universal CMDB サーバの停止］**) を使用するか，Stop HP Universal CMDB サーバ・サービスを停止して HP

Universal CMDB を無効にします。詳細については,「HP Universal CMDB Server サービスの開始と停止」を参照してください。

2. Windows の[**サービス**]ウィンドウで,[**UCMDB_Server**]をダブルクリックします。[**UCMDB Server のプロパティ**] ダイアログ・ボックスが開きます。
3. [ログオン]タブをクリックします。



4. [**アカウント**]を選択し,マシンで有効なユーザのリストから別のユーザを参照して選択します。
5. 選択したユーザの Windows パスワードを入力し,このパスワードを確認します。
6. [**適用**]をクリックして設定を保存し,[**OK**]をクリックしてダイアログ・ボックスを閉じます。
7. [スタート]メニュー(**[スタート]>[すべてのプログラム]>[HP UCMDB]>[HP Universal CMDB サーバの開始]**) を使用するか, Stop HP Universal CMDB サーバ・サービスを開始して HP Universal CMDB を有効にします。詳細については,「HP Universal CMDB Server サービスの開始と停止」を参照してください。

構成マネージャのデータベース・パスワードの暗号化

CM データベースのパスワードは**<構成マネージャのインストール・ディレクトリの >\conf\database.properties** ファイルに保存されます。パスワードを暗号化する場合,デフォルトの暗号化アルゴリズムは FIPS 140-2 の基準に準拠します。

暗号化はキーを用いて行われ，このキーを通してパスワードが暗号化されます。そして，キー自体もマスタ・キーと呼ばれる別のキーを使用して暗号化されます。いずれのキーも，同じアルゴリズムによって暗号化されます。暗号化プロセスで使用されるパラメータの詳細については，15ページ「構成マネージャ・データベースのパスワード暗号化のためのパラメータ」を参照してください。

注意： 暗号化アルゴリズムを変更すると，これまでに暗号化されたパスワードはすべて使用できなくなります。

データベース・パスワードの暗号化を変更するには，次の手順を実行します。

1. **<構成マネージャのインストール・ディレクトリの >\conf \encryption.properties** のファイルを開き，次のフィールドを編集します。
   - **engineName :** 暗号化アルゴリズムの名前を入力します。
   - **keySize :** 選択したアルゴリズムのマスタ・キーのサイズを入力します。
2. **generate-keys.bat** スクリプトを実行すると，次のファイルが作成されます。 **<構成マネージャのインストール・ディレクトリ >\security\encrypt_repository**。また，暗号化鍵も生成されます。
3. **bin\encrypt-password.bat** ユーティリティを実行してパスワードを暗号化します。**-h** フラグを設定して利用可能なオプションを確認します。
4. パスワード暗号化ユーティリティの結果をコピーして **conf\database.properties** ファイルに結果の暗号化を貼り付けます。

構成マネージャ・データベースのパスワード暗号化のためのパラメータ

次のテーブルには，CM データベースのパスワード暗号化に使用する **encryption.properties** ファイルに含まれるパラメータが一覧表示されます。データベースのパスワード暗号化の詳細については，14ページ「構成マネージャのデータベース・パスワードの暗号化」を参照してください。

パラメータ	詳細
cryptoSource	暗号化アルゴリズムを実装するインフラストラクチャを示します。次のオプションを使用できます。 • **lw。**［Uses Bouncy Castle lightweight implementation］(標準設定) • **jce。**［Java Cryptography Enhancement］(標準 Java 暗号化方式インフラストラクチャ)
storageType	キー保存のタイプを示します。 現在は，**バイナリ・ファイル**のみがサポートされています。
binaryFileStorageName	マスタ・キーが保存されているファイル内の場所を示します。
cipherType	暗号化のタイプです。現在は，**symmetricBlockCipher** のみがサポートされています。

パラメータ	詳細
engineName	暗号化アルゴリズムの名前です。 次のオプションを利用できます。 • **AES。** American Encryption Standard。この暗号化は FIPS 140-2 に準拠しています。(標準設定) • **Blowfish** • **DES** • **3DES。** (FIPS 140-2 準拠) • **ヌル。** 暗号化なし
keySize	マスタ・キーのサイズです。サイズは次のアルゴリズムによって決定されます。 • **AES。** 128, 192 または 256(標準設定は 256) • **Blowfish。** 0-400 • **DES。** 56 • **3DES。** 156
encodingMode	バイナリ暗号化結果の ASCII エンコーディング。 次のオプションを利用できます。 • **Base64**(標準設定) • **Base64Url** • **Hex**
algorithmModeName	アルゴリズムのモードです。現在は, **CBC** のみがサポートされています。
algorithmPaddingName	使用するパディング・アルゴリズムです。 次のオプションを利用できます。 • **PKCS7Padding**(標準設定) • **PKCS5Padding**
jceProviderName	JOE 暗号化アルゴリズムの名前です。 **注 :** crytpSource が jce の場合のみ該当します。lw の場合は engineName が使用されいます。

第2章

Secure Sockets Layer(SSL) 通信の有効化

本章の内容

自己署名証明書を使ったサーバ・マシンの SSL 有効化 - UCMDB

本項では，Secure Sockets Layer(SSL) チャネルを使用した通信をサポートするようHP Universal CMDB を設定する方法について説明します。

HP Universal CMDB は，標準の Web サーバとして Jetty 6.1 を使用します。

1. **前提条件**

   a. 次の手順を開始する前に，**C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore** にある古い **server.keystore** を削除してください。

   b. HP Universal CMDB キーストア(JKS タイプ) を **C:\hp\UCMDB\UCMDBServer\conf\security** フォルダに置きます。

2. **サーバ・キーストアの生成**

a. 自己署名証明書と秘密鍵を使用してキーストア(JKS タイプ) を作成します。

- **C:\hp\UCMDB\UCMDBServer\bin\jre\bin** から次のコマンドを実行します。

  ```
  keytool -genkey -alias hpcert -keystore
  C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore
  ```

  コンソール・ダイアログ・ボックスが開きます。

- キーストアのパスワードを入力します。パスワードが変更された場合は，**UCMDB:service=Security Services** で JMX 操作 **changeKeystorePassword** を実行します。パスワードが変更されていない場合，標準設定のパスワード **hppass** を使用します。
- **「What is your first and last name?」**という質問に回答します。HP Universal CMDB の Web サーバ名を入力します。所属する組織に応じて，ほかのパラメータを入力します。
- キーのパスワードを入力します。キーのパスワードは，キーストアのパスワードと一致する必要があります。

  JKS キーストアが **server.keystore** という名前で，**hpcert** という名前のサーバ証明書とともに作成されます。

b. 自己署名証明書をファイルにエクスポートします。

C:\hp\UCMDB\UCMDBServer\bin\jre\bin から次のコマンドを実行します。

```
keytool -export -alias hpcert -keystore
C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore -storepass
<your password> -file hpcert
```

3. **クライアントのトラスト・ストアへの証明書の配置**

   server.keystore を生成してサーバ証明書をエクスポートした後，自己署名証明書を使用して HP Universal CMDB とSSL 通信をする必要のあるすべてのクライアントで，この証明書をクライアントのトラスト・ストアに配置します。

   > **注：** **server.keystore** でのみ，サーバ証明書を 1 つ持つことができます。

4. **HTTP ポート 8080 の無効化**

   詳細については，27ページ「HTTP/HTTPS ポートの有効化と無効化」を参照してください。

   > **注：** HTTP ポートを閉じる前に，HTTPS 通信が動作していることを確認します。

5. **サーバの再起動**
6. **HP Universal CMDB の表示**

   UCMDB サーバがセキュアであることを確認するには，Web ブラウザに次の URL を入力します :**https://<UCMDB サーバ名または IP アドレス>:8443/ucmdb-ui**

自己署名証明書を使ったサーバ・マシンの SSL 有効化 - 構成マネージャ

本項では，Secure Sockets Layer(SSL) チャネルを使用した認証および暗号化をサポートするよう構成マネージャを設定する方法について説明します。

構成マネージャでは Tomcat 7.0.19 がアプリケーション・サーバとして使用されます。

注： すべてのディレクトリおよびファイル・ロケーションは特定のプラットフォーム，OS，インストール設定によって異なります。

1. **前提条件**

   次の手順を開始する前に，**<<構成マネージャのインストール・ディレクトリ>>\java\windows\x86_64\lib\security** フォルダまたは**<<構成マネージャのインストール・ディレクトリ>>\java\linux\x86_64\lib\security** フォルダ(いずれか該当するほう) に古い **tomcat.keystore** ファイルがあれば削除します。

2. **サーバ・キーストアの生成**

   自己署名証明書と秘密鍵を使用してキーストア (JKS タイプ) を作成します。

   - 構成マネージャのインストール・ディレクトリの Java インストールの bin ディレクトリから次のコマンドを実行します。

     ```
     keytool -genkey -alias tomcat -keyalg RSA -keystore ..\lib\
     security\tomcat.keystore
     ```

     コンソール・ダイアログ・ボックスが開きます。

   - キーストアのパスワードを入力します。パスワードが変更されている場合はファイル内で手動で変更します。
   - **「What is your first and last name?」**という質問に回答します。構成マネージャの Web サーバ名を入力します。所属する組織に応じて，ほかのパラメータを入力します。
   - キーのパスワードを入力します。キーのパスワードは，キーストアのパスワードと一致する必要があります。

     JKS キーストアが **tomcat.keystore** という名前で，**hpcert** という名前のサーバ証明書とともに作成されます。

3. **クライアントのトラスト・ストアへの証明書の配置**

   使用しているコンピュータの Internet Explorer 内のクライアントの信頼済みストアに証明書を追加します(**[ツール]>[インターネット オプション]>[コンテンツ]>[証明書]**) 。これを行わないと，最初に構成マネージャを使用しようとしたときに証明書を追加するよう要求されます。

   制限事項 :**tomcat.keystore** でのみ，サーバ証明書を 1 つ持つことができます。

4. **server.xml ファイルの修正**

<<構成マネージャのインストール・ディレクトリ>>\servers\server-0\conf にある **server.xml** ファイルを開きます。次で始まるセクションを見つけます。

```
Connector port="8143"
```

これはコメントに表示されます。コメント文字を削除してスクリプトをアクティブ化し，次の属性を HTTPS 接続に追加します。

```
keystoreFile="<tomcat.keystore file location>"( ステップ 2参照)
keystorePass="<password>"
```

次のラインをコメント・アウトします。

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
```

注： HTTP 接続ポートをブロックしないでください。HTTP 通信をブロックしたい場合は，ファイアウォールが使用できます。

5. **サーバの再起動**

   構成マネージャ・サーバを再起動します。

6. **サーバ・セキュリティの確認**

   構成マネージャ・サーバがセキュアであることを確認するには，Web ブラウザに次の URL を入力します :**https://<<構成マネージャ・サーバの名前または IP アドレス>>:8143/cnc**.

ヒント： 接続を確立できない場合は，別のブラウザを使用してみるかブラウザのバージョンを更新してみてください。

認証局から取得した証明書を使ったサーバ・マシンの SSL 有効化 - UCMDB

認証局(CA) が発行した証明書を使用するには，キーストアが Java 形式である必要があります。次の例を使って，Windows マシンでキーストアをフォーマットする方法を説明します。

1. **前提条件**

   次の手順を開始する前に，**C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore** にある古い **server.keystore** を削除してください。

2. **サーバ・キーストアの生成**

   a. 認証局署名証明書を生成して，Windows にインストールします。

   b. Microsoft 管理コンソール(**mmc.exe**) を使って，証明書を ***.pfx** ファイルに(秘密鍵を含めて) エクスポートします。

      pfx ファイルのパスワードとして任意の文字列を入力します(キーストアのタイプを JAVA キーストアに変換するとき，このパスワードを入力する必要があります) 。これで **.pfx** ファイルには公開証明書と秘密鍵が含まれ，パスワードで保護されます。

   c. 作成した **.pfx** ファイルを次のフォルダにコピーします

:**C:\hp\UCMDB\UCMDBServer\conf\security**.

d. コマンド・プロンプトを開いて，ディレクトリを **C:\hp\UCMDB\UCMDBServer\bin\jre\bin** に変更します。

次のコマンドを実行して，キーストアのタイプを **PKCS12** から **JAVA** キーストアに変更します。

```
keytool -importkeystore -srckeystore
c:\hp\UCMDB\UCMDBServer\conf\security\<pfx ファイル名> -srcstoretype
PKCS12 -destkeystore server.keystore
```

ソース(**.pfx**) キーストアのパスワードを入力するメッセージが表示されます。このパスワードは，手順 b で **pfx** ファイルを作成したときに指定したものです。

e. 変換先キーストアのパスワードを入力します。ここでは，セキュリティ・サービスで JMX メソッド **changeKeystorePassword** に先ほど定義したものと同じパスワードを使う必要があります。パスワードが変更されていない場合，標準設定のパスワード **hppass** を使用します。

f. 証明書を生成した後，HTTP ポート 8080 を無効にします。詳細については，27ページ「HTTP/HTTPS ポートの有効化と無効化」を参照してください。

g. **hppass** または **.pfx** ファイルで使ったパスワード以外のパスワードを使用した場合，JMX メソッド **changeKeystorePassword** を実行して，キーが同じパスワードを持つようにします。

> **注：** HTTP ポートを閉じる前に，HTTPS 通信が動作していることを確認します。

3. **サーバの再起動**
4. **サーバ・セキュリティの確認**

UCMDB サーバがセキュアであることを確認するには，Web ブラウザに次の URL を入力します :**https://<UCMDB サーバ名 または IP アドレス>:8443/ucmdb-ui**

> **注意：** **server.keystore** でのみ，サーバ証明書を 1 つ持つことができます。

認証局から取得した証明書を使ったサーバ・マシンの SSL 有効化 - 構成マネージャ

構成マネージャの場合，認証局(CA) が発行した証明書を使用するには，キーストアが Java 形式である必要があります。次の例を使って，Windows マシンでキーストアをフォーマットする方法を説明します。

1. **前提条件**

次の手順を開始する前に，**<<構成マネージャのインストール・ディレクトリ>>\java\windows\x86_64\lib\security** フォルダまたは**<<構成マネージャのインストール・ディレクトリ>>\java\linux\x86_64\lib\security** フォルダ(いずれか該当するほう) に古い **tomcat.keystore** ファイルがあれば削除します。

2. **サーバ・キーストアの生成**

a. 認証局署名証明書を生成して，Windows にインストールします。

b. Microsoft 管理コンソール(**mmc.exe**) を使って，証明書を ***.pfx** ファイルに(秘密鍵を含めて) エクスポートします。

pfx ファイルのパスワードとして任意の文字列を入力します(キーストアのタイプを JAVA キーストアに変換するとき，このパスワードを入力する必要があります) 。
これで **.pfx** ファイルには公開証明書と秘密鍵が含まれ，パスワードで保護されます。

作成した **.pfx** ファイルを次のフォルダにコピーします :**<＜構成マネージャのインストール・ディレクトリ＞ >\java\lib\security**

c. コマンド・プロンプトを開いて，ディレクトリを **<＜構成マネージャのインストール・ディレクトリ＞ >\java\bin** に変更します。

次のコマンドを実行して，キーストアのタイプを **PKCS12** から **JAVA** キーストアに変更します。

```
keytool -importkeystore -srckeystore <＜構成マネージャのインストール・ディレクトリ＞ >\conf\security\<＜pfx ファイル名＞ > -srcstoretype PKCS12 -destkeystore tomcat.keystore
```

ソース(**.pfx**) キーストアのパスワードを入力するメッセージが表示されます。このパスワードは，手順 b で pfx ファイルを作成したときに指定したものです。

3. **server.xml ファイルの修正**

<＜構成マネージャのインストール・ディレクトリ＞ >\servers\server-0\conf にある **server.xml** ファイルを開きます。次で始まるセクションを見つけます。

```
Connector port="8143"
```

これはコメントに表示されます。コメント文字を削除してスクリプトをアクティブ化し，次のラインを追加します。

```
keystoreFile="../../java/lib/security/tomcat.keystore"
keystorePass="password" />
```

次のラインをコメント・アウトします。

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
```

注 : HTTP 接続ポートをブロックしないでください。HTTP 通信をブロックしたい場合は，ファイアウォールが使用できます。

4. **サーバの再起動**

構成マネージャ・サーバを再起動します。

5. **サーバ・セキュリティの確認**

構成マネージャ・サーバがセキュアであることを確認するには，Web ブラウザに次の URL を入力します :**https://<＜構成マネージャ・サーバの名前または IP アドレス＞ >:8143/cnc**.

制限事項 :**tomcat.keystore** でのみ，サーバ証明書を 1 つ持つことができます。

注： すべてのディレクトリおよびファイル・ロケーションは特定のプラットフォーム，オペレーティング・システム，インストール設定によって異なります。

例 :`java/{os name}/lib`。

クライアント・マシンでの SSL の有効化 - UCMDB

HP Universal CMDB Web サーバによって使用されている証明書が良く知られている認証局 (CA) で発行されたものである場合，使用している Web ブラウザは特別なアクションを行わずに検証することができます。

認証局が Web ブラウザに信頼されていない場合は，証明書のトラスト・パス全体または HP Universal CMDB によって使用されている証明書を，ブラウザのトラスト・ストアに明示的にインポートする必要があります。

次の例では，自己署名 **hpcert** 証明書を Windows のトラスト・ストアにインポートして，Internet Explorer で使用できるようにする方法について説明します。

証明書を Windows のトラスト・ストアにインポートするには，次の手順を実行します。

1. **hpcert** 証明書を **hpcert.cer** という名前に変更します。

   Windows Explorer に，ファイルがセキュリティ証明書であることを示すアイコンが表示されます。

2. **hpcert.cer** をダブルクリックして，Internet Explorer の[証明書]ダイアログ・ボックスを開きます。
3. Certificate Import Wizard で証明書をインストールすることによってトラストを有効にするための指示に従います。

注： UCMDB サーバが発行した証明書を Web ブラウザにインポートするには，別の方法もあります。これには，まず UCMDB にログインし，信頼されていない証明書であるという警告が表示されたときに証明書をインストールします。

クライアント証明書で SSL を有効化 - 構成マネージャ

構成マネージャ Web サーバによって使用されている証明書が良く知られている認証局 (CA) で発行されたものである場合は，使用している Web ブラウザはこれ以上のアクションなしでもおそらく検証できます。

CA がサーバ・トラスト・ストアによって信頼されていない場合は，CA 証明書をサーバ・トラスト・ストアにインポートします。

次の例では，自己署名 **hpcert** 証明書をサーバ・トラスト・ストア(cacerts) にインポートする方法について説明します。

証明書をサーバ・トラスト・ストアにインポートするには，次の手順を実行します。

1. クライアント・マシン上で **hpcert** 証明書を見つけて名前を **hpcert.cer** に変更します。
2. **hpcert.cer** を**<＜構成マネージャのインストール・ディレクトリ＞ >\java\bin** フォルダのサーバ・マシ

ンにコピーします。

3. 次のコマンドによりキーツール・ユーティリティを使用して，サーバ・マシン上で CA 証明書をトラスト・ストア(cacerts) にインポートします。

   <＜構成マネージャのインストール・ディレクトリ＞>\java\bin\keytool.exe -import -alias hp -file hpcert.cer -keystore ..\lib\security\cacerts

4. **server.xml** ファイル(場所 : **<＜構成マネージャのインストール・ディレクトリ＞>\servers\server-0\conf**) を修正します。

   a. 22ページ「server.xml ファイルの修正」で説明している変更を行います。

   b. これらの変更後すぐに，次の属性を HTTPS コネクタに追加します。

      ```
      truststoreFile="../../java/lib/security/cacerts"
      truststorePass="changeit" />
      ```

   c. clientAuth="true" を設定します。

5. 22ページ「サーバ・セキュリティの確認」の説明に従ってサーバ・セキュリティを確認します。

クライアント SDK での SSL の有効化

クライアント SDK とサーバ SDK の間では，HTTPS 通信を利用することができます。

1. クライアント・マシンで，クライアント SDK を埋め込んだ製品から通信設定を開き，HTTP ではなく HTTPS を使うよう設定します。
2. 認証局署名証明書または自己署名証明書をクライアント・マシンにダウンロードし，サーバに接続する JRE の **cacerts** トラスト・ストアにインポートします。

   次のコマンドを使います。

   Keytool -import -alias <CA name> -trustcacerts -file <サーバの公開証明書のパス> -keystore <クライアント JRE の信頼済み cacerts ストアのパス (x:\program files\java\jre\lib\security\cacerts など)>

SDK の相互証明書認証の有効化

このモードでは SSL を使用し，UCMDB によるサーバ認証と UCMDB-API クライアントによるクライアント認証の両方を有効化します。サーバおよび UCMDB-API クライアントはどちらも，認証のためにほかのエンティティに証明書を送信します。

注： SDK で SSL を使用して相互認証を有効化する次の方法は，最も安全で，推奨される通信モードです。

1. UCMDB の UCMDB-API クライアント・コネクタのセキュリティを強化します。

   a. UCMDB JMX コンソールにアクセスします。Web ブラウザを起動し，アドレスに「**http://<UCMDB machine name or IP address>:8080/jmx-console**」と入力します。ユーザ名とパスワードでのログインが必要な場合もあります(標準設定は sysadmin/sysadmin) 。

   b. **UCMDB:service=Ports Management Services** を見つけ，リンクをクリックして［操作］ペー

ジを開きます。

c. **PortsDetails** 操作を見つけ，[**Invoke**]をクリックします。クライアント認証に使用する HTTPS のポート番号を記録します。標準設定は 8444 で，有効になっている必要があります。

d. [操作]ページに戻ります。

e. ucmdb-api コネクタを相互認証モードにマップするには，次のパラメータを指定して **mapComponentToConnectors** メソッドを呼び出します。

- **componentName**:ucmdb-api
- **isHTTPSWithClientAuth**: true
- ほかのすべてのフラグ: false

次のメッセージが表示されます。

```
Operation succeeded.Component ucmdb-api is now mapped to: HTTPS_
CLIENT_AUTH ports.
```

f. [操作]ページに戻ります。

2. UCMDB-API クライアントを実行する JRE に，クライアント証明書を含むキー・ストアがあることを確認します。

3. キー・ストアから UCMDB-API クライアント証明書をエクスポートします。

4. エクスポートした UCMDB-API クライアント証明書を UCMDB サーバのトラスト・ストアにインポートします。

a. UCMDB マシンで，作成した UCMDB-API クライアント証明書ファイルを UCMDB の次のディレクトリにコピーします。

C:\HP\UCMDB\UCMDBServer\conf\security

b. 次のコマンドを実行します。

C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -import -v -keystore C:\HP\UCMDB\UCMDBServer\conf\security\server.truststore -file <<エクスポートされた UCMDB API クライアント証明書>> - alias ucmdb-api

c. UCMDB サーバのトラスト・ストア・パスワード(標準設定は **hppass**) を入力します。

d. [**Trust this certificate?**]と表示された場合は，[**y**]を押して **Enter** キーを押します。

e. 出力が[**証明書**がキー・ストアに追加されました]であることを確認します。

5. サーバ・キー・ストアから UCMDB サーバ証明書をエクスポートします。

a. UCMDB マシンで，次のコマンドを実行します。

C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias hpcert -keystore C:\HP\UCMDB\UCMDBServer\conf\security\server.keystore -file C:\HP\UCMDB\conf\security\server.cert

b. UCMDB サーバのトラスト・ストア・パスワード(標準設定は **hppass**) を入力します。

c. 証明書が次のディレクトリに作成されていることを確認します。

C:\HP\UCMDB\UCMDBServer\conf\security\server.cert

6. エクスポートしたUCMDB証明書をUCMDB-APIクライアントのトラスト・ストアのJREにインポートします。
7. UCMDBサーバおよびUCMDB-APIクライアントを再起動します。
8. UCMDB-APIクライアントからUCMDB-APIサーバに接続するには，次のコードを使用します。

```
UcmdbServiceProvider provider =
UcmdbServiceFactory.getServiceProvider("https", <SOME_HOST_NAME>,
<HTTPS_WITH_CLIENT_AUTH_PORT_NUMBER (default:8444>));
UcmdbService ucmdbService = provider.connect
(provider.createCertificateCredentials(<TheClientKeystore.
e.g:"c:\\client.keystore">, <KeystorePassword>),
provider.createClientContext(<ClientIdentification>));
```

サーバ・キーストアのパスワードの変更

サーバをインストールすると，HTTPSポートが開き，弱いパスワード(標準設定の**hppass**) でストアが保護されます。SSLのみ使用する場合は，パスワードを変更する必要があります。

次に，**server.keystore**パスワードだけを変更する方法について説明します。ただし，**server.truststore**パスワードの変更でも同じ手順を実行します。

注： この操作ではすべての手順を実行する必要があります。

1. UCMDBサーバを開始します。
2. JMXコンソールでパスワード変更を実行します。
   a. Webブラウザを起動して，次のサーバ・アドレスを入力します:**http://<UCMDB サーバのホスト名 または IP>:8080/jmx-console**。

      ユーザ名とパスワードを使用してログインする必要がある場合もあります。

   b. UCMDBで，**UCMDB:service=Security Services**をクリックして[操作]ページを開きます。
   c. **changeKeystorePassword**操作を実行します。

      このフィールドは空にせず，6文字以上を入力する必要があります。パスワードの変更は，データベースのみが対象です。
3. UCMDBサーバを停止します。
4. コマンドを実行します。

   C:\hp\UCMDB\UCMBServer\bin\jre\binから次のコマンドを実行します。

   a. ストアのパスワードを変更します。

      keytool -storepasswd -new <新しいキーストアのパス> -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore -storepass <現在のキーストアのパス>

   b. 次のコマンドで，キーストアの内部キーを表示します。最初のパラメータはエイリアスです。次のコマンドで必要なので，このパラメータを保存しておいてください。

keytool -list -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore

c. (ストアが空ではない場合) キーのパスワードを変更します。

keytool -keypasswd -alias <別名> -keypass <現在のパス> -new <新しいパス> -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore

d. 新しいパスワードを入力します。

5. UCMDB サーバを開始します。
6. サーバのトラスト・ストアでこの手順を繰り返します。

HTTP/HTTPS ポートの有効化と無効化

ユーザ・インタフェースまたは JMX コンソールから HTTP ポートおよび HTTPS ポートを有効化 / 無効化できます。

ユーザ・インタフェースから HTTP ポートまたは HTTPS ポートを有効化 / 無効化するには, 次の手順を実行します。

1. HP Universal CMDB にログオンします。
2. **[管理]>[インフラストラクチャ設定]**を選択します。
3. [**フィルタ**](名前を使用) ボックスで **http** または **https** を入力して, HTTP 設定を表示します。
   - **HTTP 接続を有効化** :**true** の場合はポートは有効になっています。**False** の場合はポートは無効になっています。
4. サーバを再起動して, 変更を適用します。

> **注意 :** HTTPS ポートは標準設定で開いています。このポートを閉じると, **Server_Management.bat** が動作しなくなります。

JMX コンソールから HTTP ポートまたは HTTPS ポートを有効化 / 無効化するには, 次の手順を実行します。

1. Web ブラウザを起動し, アドレスにhttp://localhost.<domain_name>:8080/jmx-console を入力します。
2. JMX コンソールの認証資格情報を入力します。標準設定値は次のとおりです。
   - ログイン名 = **sysadmin**
   - パスワード = **sysadmin**
3. **UCMDB:service=Ports Management Services** を見つけ, リンクをクリックして[操作]ページを開きます。
4. HTTP ポートを有効化 / 無効化するには, **HTTPSetEnable** 操作を見つけ, 値を設定します。
   - **True にすると**, ポートは有効になっています。
   - **False にすると**, ポートは無効になっています。
5. HTTPS ポートを有効化 / 無効化するには, **HTTPSSetEnable** 操作を見つけ, 値を設定します。

- **true の場合 :** ポートは有効になっています。
- **false の場合 :** ポートは無効になっています。

6. クライアント認証を使用して HTTPS ポートを有効化 / 無効化するには, **HTTPSClientAuthSetEnable** 操作を見つけ, 値を設定します。
   - **true の場合 :** ポートは有効になっています。
   - **false の場合 :** ポートは無効になっています。

UCMDB Web コンポーネントのポートへのマップ

各 UCMDB コンポーネントで, JMX コンソールから利用可能なポートへのマップを設定できます。

現在のコンポーネント設定を表示するには, 次の手順を実行します。

1. Web ブラウザを起動し, アドレスに**http://localhost.<domain_name>:8080/jmx-console** を入力します。
2. JMX コンソールの認証資格情報を入力します。標準設定値は次のとおりです。

   ログイン名 = **sysadmin**

   パスワード = **sysadmin**
3. **UCMDB:service=Ports Management Services** を見つけ, リンクをクリックして[操作]ページを開きます。
4. **ComponentsConfigurations** メソッドを見つけ, [**Invoke**]をクリックします。
5. コンポーネントごとに, 有効なポートおよび現在マップされているポートが表示されます。

コンポーネントをマップするには, 次の手順を実行します。

1. **UCMDB:service=Ports Management Services** を見つけ, リンクをクリックして[操作]ページを開きます。
2. **mapComponentToConnectors** メソッドを見つけます。
3. [値]ボックスにコンポーネントの名前を入力します。選択項目に対応するポートごとに[**True**]または[**False**]を選択します。[**Invoke**]をクリックします。選択したコンポーネントが選択したポートにマップされます。**serverComponentsNames** メソッドを呼び出して, コンポーネント名を検索できます。
4. 関連するコンポーネントごとにこのプロセスを繰り返します。

注 :

- すべてのコンポーネントを少なくとも 1 つのポートにマップする必要があります。コンポーネントをどのポートにもマップしない場合は, 標準設定で HTTP ポートにマップされます。
- クライアント認証を使用して HTTP ポートと HTTPS ポートの両方にコンポーネントをマップすると, クライアント認証オプションのみがマップされます(この場合, ほかのオプションは余剰です) 。

また, 各ポートに割り当てる値を変更することもできます。

ポートの値を設定するには, 次の手順を実行します。

1. **UCMDB:service=Ports Management Services** を見つけ，リンクをクリックして［操作］ページを開きます。
2. HTTP ポートの値を設定するには，**HTTPSetPort** メソッドを見つけ，［**値**］ボックスに値を入力します。［**Invoke**］をクリックします。
3. HTTPS ポートの値を設定するには，**HTTPSSetPort** メソッドを見つけ，［**値**］ボックスに値を入力します。［**Invoke**］をクリックします。
4. クライアント認証を使用して HTTPS ポートの値を設定するには，**HTTPSClientAuthSetPort** メソッドを見つけ，［**値**］ボックスに値を入力します。［**Invoke**］をクリックします。

SSL を使用して UCMDB の作業を行うための構成マネージャの設定

SSL(Secure Sockets Layer) を使用して UCMDB の作業を行うために構成マネージャの設定を行うことができます。ポート 8443 の SSL コネクタは UCMDB の標準設定で有効化されています。

1. **<＜UCMDB インストール・ディレクトリ＞>\bin\jre\bin** に移動して次のコマンドを実行します。

   ```
   keytool -export -alias hpcert -keystore <UCMDB server dir>
   \conf\security\server.keystore -storepass hppass -file
   <certificatefile>
   ```

2. 証明書ファイルをローカル構成マネージャ・マシンの一時ロケーションにコピーします。
3. 新しいインストールを実施するか，既存の構成マネージャのインストールを再設定します。その方法については，対話型の『HP Universal CMDB デプロイメント・ガイド』の関連セクションを参照してください。

   UCMDB 設定画面で，プロトコルを HTTPS に設定し，ステップ 2 でコピーした証明書ファイルを選択します。

SSL を使用してその他の製品 (ロード・バランサなど) の作業を行うよう構成マネージャを設定するには，次のコマンドを実行して製品のセキュリティ証明書を構成マネージャのトラスト・ストア(標準設定の jre トラスト・ストア) にインポートします。

```
<CM_JAVA_HOME>\bin\keytool -import -trustcacerts -alias <alias>
-keystore <CM_JAVA_HOME>\lib\security\cacerts -storepass changeit
-file <certificatefile>
```

SSL で使用する UCMDB KPI アダプタの有効化

SSL(Secure Sockets Layer) を使用して送信されるよう UCMDB KPI アダプタ情報を設定することができます。

1. 構成マネージャ証明書のエクスポート：

   ```
   <CM_JAVA_HOME>\bin\keytool -export -alias tomcat -keystore
   <CM_JAVA_HOME>\lib\security\tomcat.keystore -storepass
   <keystore pass> -file <＜証明書のファイル名＞>
   ```

2. 構成マネージャからエクスポートした証明書を次のように UCMDB トラスト・ストアにインポートしま

す。

```
<UCMDB server dir>\bin\jre\bin keytool -import -trustcacerts
-alias tomcat -keystore <UCMDB server dir>\bin\jre\lib
\security\cacerts -storepass changeit -file <certificatefile>
```

3. 構成マネージャからエクスポートした証明書を次のようにプローブのトラスト・ストアにインポートします。

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   <DataFlowProbe dir>\bin\jre\bin\keytool.exe -import -v -keystore
   <DataFlowProbe dir>\conf\security\MAMTrustStoreExp.jks -file
   <certificatefile> -alias tomcat
   ```

   b. キーストアのパスワード，logomaniaを入力します。

   c. [**Trust this certificate?**]と表示された場合は，[**y**]を押して **Enter** キーを押します。

   次のメッセージが表示されます。

   > **証明書がキーストアに追加されました。**

   Data Flow Probe のセキュリティ強化の詳細については，59ページ「データ・フロー・プローブの強化」を参照してください。

4. UCMDB，Data Flow Probe，構成マネージャを再起動します。

UCMDB ブラウザの SSL サポートの構成

> **注：** ここに記載の指示は UCMDB ブラウザのバージョン 1.7 に対応しています。他の UCMDB 製品スイートとは別にアップグレードされた，新しいバージョンの UCMDB ブラウザを使用している場合は，当該バージョンの『UCMDB ブラウザのインストールおよび構成ガイド』に記載されている SSL サポートの構成についてのセクションを参照してください。

Tomcat で SSL サポートをインストール，構成するには

1. 次のコマンドの 1 つを実行し，サーバのプライベート・キーと自己署名証明書を保存するキーストア・ファイルを作成します。

   - Windows の場合 :**%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA**
   - UNIX の場合 :**$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA**

   両方のコマンドで，パスワード値 **changeit** を使用します(開いているコンソール・ダイアログ・ボックスの他のすべてのフィールドについても，同じ値を使用できます) 。

2. **$CATALINA_BASE/conf/server.xml** の **SSL HTTP/1.1 Connector** エントリからコメント行指定を解除します(**$CATALINA_BASE** は Tomcat をインストールしたディレクトリ) 。

   > **注：** **server.xml** を構成して SSL を使用する方法の詳細については，Apache Tomcat オフィシャル・サイトを参照してください。http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html(英語サイト)

3. Tomcat サーバを再起動します。

UCMDB サーバへの接続にHTTPS プロトコルを使用するには:

1. **ucmdb_browser_config.xml** で，タグ **<protocol>** に値 **https**を割り当て，タグ **<port> に UCMDB サーバ HTTPS ポート値 (デフォルトでは 8443) を割り当てます。**
2. UCMDB ブラウザ・マシンにUCMDB サーバの公開証明書をダウンロードします(UCMDB-Server で SSL を使用している場合，UCMDB 管理者がこの証明書を提供できます) 。そして次のコマンドを実行して，サーバに接続しようとしている JRE の **cacerts** トラスト・ストアに公開証明書をインポートします。

   "%JAVA_HOME%\bin\keytool" -import -alias ucmdb -trustcacerts -file <UCMDB-Server-certificate-file> -keystore "%JAVA_HOME%\jre\lib\security\cacerts"

   <UCMDB-Server-certificate-file> は，UCMDB サーバ公開証明書ファイルへのフルパスです。
3. Tomcat サーバを再起動します。

第3章

リバース・プロキシの使用

本項では，リバース・プロキシのセキュリティとの関係について概説し，HP Universal CMDB および構成マネージャでリバース・プロキシを使用する手順について説明します。リバース・プロキシのセキュリティの側面については説明しますが，その他の側面については取り上げません。

本章の内容

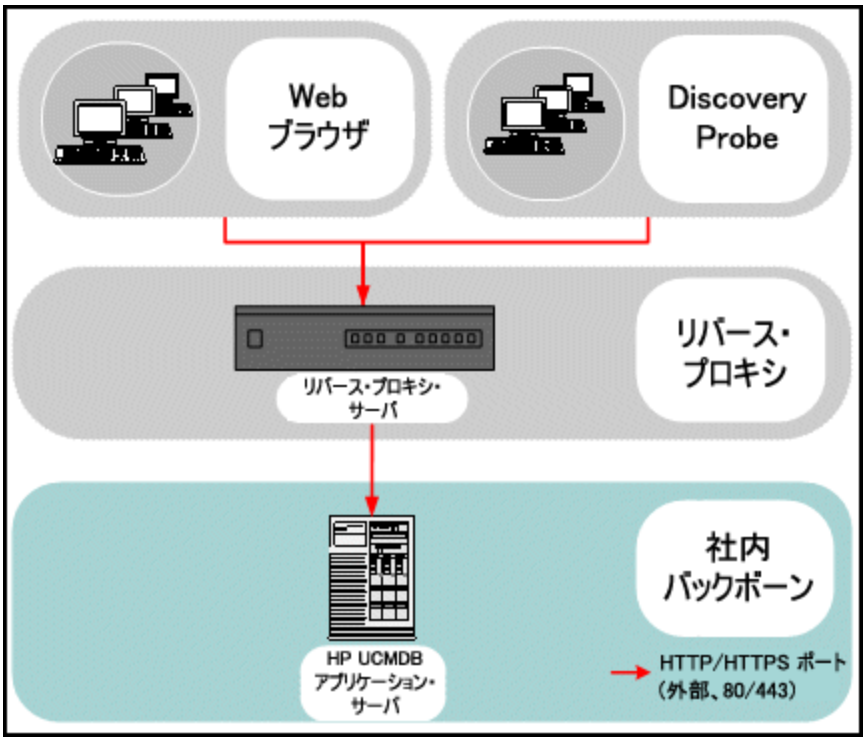
リバース・プロキシの概要

リバース・プロキシは，クライアント・マシンとWeb サーバ間に位置する中間サーバです。リバース・プロキシは，クライアント・マシンからはクライアント・マシンのHTTP プロトコル要求を提供する通常のWeb サーバのように見えます。

クライアント・マシンは，Web サーバ名の代わりにリバース・プロキシ名を使用して，Web コンテンツを求める通常の要求を送信します。リバース・プロキシはその要求をWeb サーバの1つに送信します。応答はリバース・プロキシによってクライアント・マシンに戻されますが，クライアント・マシンには応答がWeb サーバから送信されたように見えます。

異なるURL で同じUCMDB/CM インスタンスを表す複数のリバース・プロキシを持つことが可能です。または，単一のリバース・プロキシを使用して各 UCMDB/CM サーバに異なるルート・コンテキストを設定することで複数のUCMDB/CM サーバにアクセスすることも可能です。

HP Universal CMDB および構成マネージャは，DMZ アーキテクチャのリバース・プロキシをサポートしています。リバース・プロキシは，Data Flow Probe とWeb クライアントおよびHP Universal CMDB/CM サーバ間のHTTP メディエータです。



注：

- リバース・プロキシの種類によって，必要な構成の構文が異なります。Apache 2.0.x リバース・プロキシ設定の例については，35ページ「 例：Apache 2.0.x 設定 」を参照してください。
- フロント・エンド URL の設定は，スケジューラを使用したレポートへの直接リンク作成時にのみ必要となります。

リバース・プロキシ・サーバの使用のセキュリティ面

リバース・プロキシ・サーバは，要塞ホストとして機能します。リバース・プロキシは外部クライアントから直接宣言される唯一のマシンとして設定されるため，残りの内部ネットワークは外部から見えなくなります。リバース・プロキシを使うことで，アプリケーション・サーバを内部ネットワークの別のマシンへ置くことが可能になります。

本項では，バック・ツー・バック・トポロジ環境でのリバース・プロキシとDMZの使用について解説します。

このような環境でのリバース・プロキシの使用には，主に次のような利点があります。

- DMZ でのプロトコル変換が発生しない。受信プロトコルと送信プロトコルが同一（ヘッダの変更のみ発生）
- リバース・プロキシに対するHTTP アクセスのみを許可することにより，ファイアウォールのステートフル・パケットインスペクションによる通信の高度な保護が可能
- 静的な制限付きのリダイレクト要求の設定をリバース・プロキシにおいて定義が可能

- Web サーバのセキュリティ機能のほとんどが，リバース・プロキシで利用可能（認証方式や暗号化など）
- リバース・プロキシにより，実際のサーバの IP アドレスと内部ネットワークのアーキテクチャが見えなくなる
- Web サーバにアクセスが可能なクライアントはリバース・プロキシのみ
- この構成はほかのソリューションとは異なり，NAT 型ファイアウォールをサポートする
- リバース・プロキシでは，ファイアウォールに開いておく必要のあるポートの数は最小限で済む
- リバース・プロキシは，ほかの要塞ホスト・ソリューションと比較して高度なパフォーマンスを提供

リバース・プロキシの設定

本項では，リバース・プロキシを設定する方法について説明します。

インフラストラクチャ設定を使用したリバース・プロキシの設定

次に，インフラストラクチャ設定にアクセスしてリバース・プロキシを設定にする方法について説明します。この設定は，スケジューラを使用したレポートへの直接リンク作成時にのみ必要となります。

リバース・プロキシを設定するには，次の手順を実行します。

1. [**管理**]>[**インフラストラクチャ設定**]>[**全般設定**]カテゴリを選択します。
2. [フロントエンド URL]設定を変更します。**https://my_proxy_server:443/** などのアドレスを入力します。

> **注：** この変更を行うと，クライアントから直接 HP Universal CMDB サーバにアクセスできなくなります。リバース・プロキシの設定を変更するには，サーバ・マシンで JMX コントロールを使用します。詳細については，次の「JMX コントロールを使用したリバース・プロキシの設定」を参照してください。

JMX コントロールを使用したリバース・プロキシの設定

HP Universal CMDB サーバ・マシンで JMX コントロールを使用してリバース・プロキシの設定を変更できます。この設定は，スケジューラを使用したレポートへの直接リンク作成時にのみ必要となります。

リバース・プロキシの設定を変更するには，次の手順を実行します。

1. HP Universal CMDB サーバ・マシンで Web ブラウザを起動し，次のアドレスを入力します。

   http://<<マシン名または IP アドレス>>.<<ドメイン名>>:8080/jmx-console

   <マシン名または IP アドレス>には，HP Universal CMDB がインストールされているマシンを指定します。ユーザ名とパスワードでログインする必要がある場合もあります。

2. [**UCMDB-UI**]>[**UCMDB-UI:name=UI Server frontend settings**]リンクをクリックします。

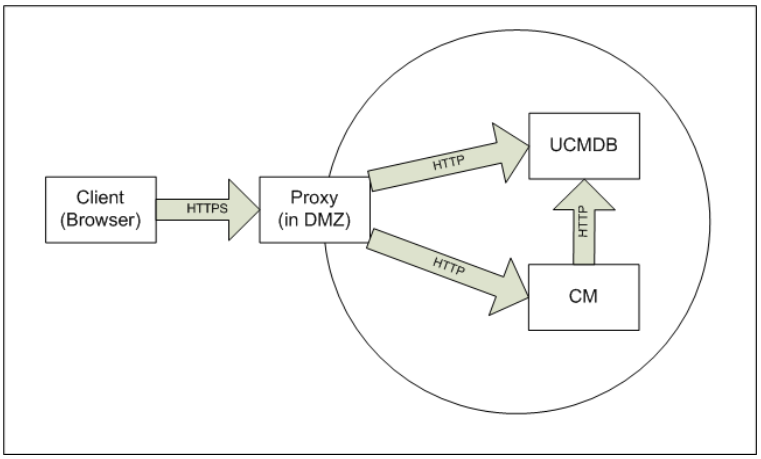
   [**setUseFrontendURLBySettings**]フィールドに，`https://my_proxy_server:443/` などのサーバ・プロキシ URL を入力します。

3. [**Invoke**]をクリックします。
4. この設定の値を確認するには，**showFrontendURLInSettings** メソッドを使用します。

例 : Apache 2.0.x 設 定

本 項 では, Data Flow Probe プローブとアプリケーション・ユーザの両 方 が, HP Universal CMDB に接 続 する場 合 の, Apache 2.0.x リバース・プロキシの使 用 をサポートする設 定 ファイルの例 を説 明 します。

次 の図 には, 構 成 マネージャおよび UCMDB のリバース・プロキシの設 定 プロセスを示 しています。



注 :

- この例 では, HP Universal CMDB マシンの DNS 名 およびポートは UCMDB_server です。
- この例 では, HP 構 成 マネージャの DNS 名 およびポートは UCMDB_CM_server です。
- この変 更 は, Apache 管 理 の知 識 を持 つユーザのみが行 なえます。

1. **<Apache マシンのルート・ディレクトリ>\Webserver\conf\httpd.conf** ファイルを開きます。
2. 次 のモジュールを有 効 にします。
   - **LoadModule proxy_module modules/mod_proxy.so**
   - **LoadModule proxy_http_module modules/mod_proxy_http.so**
   - **LoadModule headers_module modules/mod_headers.so**
3. 次 の行 を httpd.conf ファイルに追 加 します。

```
ProxyRequests off

<Proxy *>

Order deny,allow

Deny from all

Allow from all

</Proxy>

ProxyPass /mam http://UCMDB_server/mam

ProxyPassReverse /mam http://UCMDB_server/mam

ProxyPass /mam_images http://UCMDB_server/mam_images
```

```
ProxyPassReverse /mam_images http://UCMDB_server/mam_images
ProxyPass /mam-collectors http://UCMDB_server/mam-collectors
ProxyPassReverse /mam-collectors http://UCMDB_server/mam-collectors
ProxyPass /ucmdb http://UCMDB_server/ucmdb
ProxyPassReverse /ucmdb http://UCMDB_server/ucmdb
ProxyPass /site http://UCMDB_server/site
ProxyPassReverse /site http://UCMDB_server/site
ProxyPass /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPassReverse /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPass /status http://UCMDB_server/status
ProxyPassReverse /status http://UCMDB_server/status
ProxyPass /jmx-console http://UCMDB_server/jmx-console
ProxyPassReverse /jmx-console http://UCMDB_server/jmx-console
ProxyPass /axis2 http://UCMDB_server/axis2
ProxyPassReverse /axis2 http://UCMDB_server/axis2
ProxyPass /icons http://UCMDB_server/icons
ProxyPassReverse /icons http://UCMDB_server/icons
ProxyPass /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPassReverse /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPass /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPassReverse /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPass /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPassReverse /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPass /cm http://UCMDB_Server/cm
ProxyPassReverse /cm http://UCMDB_Server /cm
ProxyPass /cnc http://UCMDB_CM_server/cnc
ProxyPassReverse /cnc http://UCMDB_CM_server/cnc
ProxyPass /docs http://UCMDB_CM_server/docs
ProxyPassReverse /docs http://UCMDB_CM_server/docs
ProxyPass /ucmdb-browser http://UCMDB_CM_server/ucmdb-browser
ProxyPassReverse /ucmdb-browser http://UCMDB_CM_server/ucmdb-
browser
ProxyPreserveHost On
RequestHeader set X-Reverse-Proxy "https://<SRP host>:<SRP port>"
```

> **注：** `ProxyPreserveHost On` のラインは，仮想ホストがある場合にのみ必要です。

> **注意：** `RequestHeader set X-Reverse-Proxy "https://<SRP host>:<SRP port>"` のラインの追加が不可欠です。これがなければ構成は機能しません。

4. 変更を保存します。

相互認証を使用するリバース・プロキシまたはロード・バランサによる，データ・フロー・プローブの接続

相互認証によるリバース・プロキシまたはロード・バランサを使用して Data Flow Probe に接続するには，次の手順を実行します。この手順は次の構成に適用されます。

- プローブによって提供され，リバース・プロキシまたはロード・バランサによって要求されるクライアント証明書に基づいた，プローブとリバース・プロキシまたはロードバランサの間の相互 SSL 認証。
- リバース・プロキシまたはロード・バランサおよび UCMDB サーバとの間の通常の SSL 接続。

> **注：** 次の手順では，**cKeyStoreFile** キー・ストアをプローブ・キー・ストアとして使用します。これは Data Flow Probe インストールの一部である，事前定義されたクライアント・キー・ストアで，セルフ署名証明書が含まれています。詳細については，73ページ「サーバとデータ・フロー・プローブのデフォルトのキー・ストアとトラスト・ストア」を参照してください。
>
> 新たに生成された秘密鍵を含む新しい一意のキー・ストアを作成することをお勧めします。詳細については，72ページ「データ・フロー・プローブのキー・ストアの作成」を参照してください。

証明権限から証明書を取得

CA ルート証明書を取得し次の場所にインポートします。

- Data Flow Probe のトラスト・ストア
- Data Flow Probe JVM cacerts
- UCMDB サーバのトラスト・ストア
- リバース・プロキシのトラスト・ストア

1. CA ルート証明書を Data Flow Probe トラスト・ストアにインポートします。
   a. CA ルート証明書を<＜Data Flow Probe のインストール・ディレクトリ＞>\conf\security\<＜証明書ファイル名＞> のディレクトリに置きます。
   b. 次のスクリプトを実行して，CA ルート証明書を Data Flow トラスト・ストアにインポートします。

      ```
      <＜Data Flow Probe のインストール・ディレクトリ＞>\bin\jre\bin\keytool.exe -import -trustcacerts -alias <YourAlias> -file C:\hp\UCMDB\DataFlowProbe\conf\security\<＜証明書ファイル名＞> -keystore <＜Data Flow のインストール・ディレクトリ＞>\conf\security\MAMTrustStoreExp.jks
      ```

      標準設定ではパスワードは **logomania** です。

2. 次のスクリプトを実行して，CA ルート証明書を Data Flow Probe JVM cacerts にインポートします。

   ```
   <<Data Flow Probe のインストール・ディレクトリ>>\bin\jre\bin\keytool.exe -import -trustcacerts -alias <YourAlias> -file <<Data Flow Probe のインストール・ディレクトリ>>\conf\security\<<証明書ファイル名>> -keystore <<Data Flow Probe のインストール・ディレクトリ>>\bin\jre\lib\security\cacerts
   ```

   標準設定ではパスワードは **changeit** です。

3. CA ルート証明書を UCMDB トラスト・ストアにインポートします。

   a. CA ルート証明書を <<UCMDB のインストール・ディレクトリ>>\conf\security\<<証明書ファイル名>> のディレクトリに置きます。

   b. 次のスクリプトを実行して，CA ルート証明書を UCMDB トラスト・ストアにインポートします。

      ```
      <<UCMDB のインストール・ディレクトリ>>\bin\jre\bin\keytool.exe -import -trustcacerts -alias <YourAlias> -file <<UCMDB のインストール・ディレクトリ>>\conf\security\<<証明書ファイル名>> -keystore <<UCMDB のインストール・ディレクトリ>>\conf\security\sever.truststore
      ```

      標準設定ではパスワードは **hppass** です。

4. CA ルート証明書をリバース・プロキシのトラスト・ストアにインポートします。このステップはベンダーによって異なります。

証明書の Java キーストアへの変換

Data Flow Probe のクライアント証明書(および秘密鍵)を使用している証明書権限(CA)から PFX/PKCS12 形式で取得し，次のスクリプトを実行して Java キーストアに変換します。

```
<<Data Flow Probe のインストール・ディレクトリ>>\bin\jre\bin\keytool.exe -importkeystore -srckeystore <<PFX キーストアのフル・パス>> -destkeystore <<新しい対象キーストアのフル・パス>> -srcstoretype PKCS12
```

ソースおよび対象キーストアのパスワードの入力を求められます。

ソース・キーストアのパスワードの場合は，PFX キーストアをエクスポートした際と同じパスワードを使用します。

Data Flow Probe キーストアの標準の対象キーストア・パスワードは **logomania** です。

> **注：** 標準の Data Flow Probe キーストア・パスワード(logomania)から異なる対象キーストア・パスワードを入力した場合，**<<Data Flow Probe のインストール・ディレクトリ>>\conf\ssl.properties** ファイル(javax.net.ssl.keyStorePassword)に暗号化された形式で新しいパスワードを入力しなければならなくなります。詳細については，72ページ「プローブのキー・ストアとトラスト・ストアのパスワードを暗号化」を参照してください。

新しいキーストアを**<<Data Flow Probe のインストール・ディレクトリ>>\conf\security** のディレクトリに置きます。

> **注意：** **MAMKeyStoreExp.jks** ファイルは上書きしないでください。

新規作成したキーストアを使用するためのSSLプロパティ・ファイルの変更

<<Data Flow Probeのインストール・ディレクトリ>>\conf\ssl.propertiesのファイルにクライアント証明書を含んでいるキーストアを**javax.net.ssl.keyStore**に設定します。

キーストアのパスワードが標準のData Flow Probeキーストア・パスワード(logomania)ではない場合，暗号化後に**javax.net.ssl.keyStorePassword**を更新します。パスワード暗号化の詳細については，72ページ「プローブのキー・ストアとトラスト・ストアのパスワードを暗号化」を参照してください。

Data Flow Probe設定の確認

<<Data Flow Probeのインストール・ディレクトリ>>\conf\DataFlowProbe.propertiesファイルを次のように編集します。

```
appilog.agent.probe.protocol = HTTPS

serverName = <リバース・プロキシ・サーバ・アドレス>

serverPortHttps = <UCMDB に要求をリダイレクトするためにリバース・プロキシがリッスンするHTTPS ポート>
```

SSLを使用して作業するためのUCMDBの設定

詳細については，17ページ「Secure Sockets Layer(SSL)通信の有効化」を参照してください。

この手順で残りの証明書を作成したCAと同じCAによってUCMDBサーバ証明書が作成された場合，リバース・プロキシまたはロード・バランサはUCMDB証明書を信頼します。

第4章

データ・フロー資格情報管理

本章の内容

データ・フロー資格情報管理の概要

ディスカバリを操作またはインテグレーションを実行するには，リモート・システムにアクセスするための資格情報を設定する必要があります。資格情報は[Data Flow Probe設定]ウィンドウで設定され，UCMDBサーバに保存されます。詳細については，『HP Universal CMDB データ・フロー管理ガイド』で，Deta Flow Probeのセットアップについて説明した項を参照してください。

資格情報のストレージは，資格情報マネージャ・コンポーネントによって管理されます。詳細については，86ページ「機密マネージャ」を参照してください。

Data Flow Probeは機密マネージャ・クライアントを使用して資格情報にアクセスできます。機密マネージャ・クライアントはData Flow Probe上にあり，UCMDBサーバ上にある機密マネージャ・サーバと通信します。機密マネージャ・クライアントと機密マネージャ・サーバ間の通信は暗号化され，機密マネージャ・サーバに接続するときには機密マネージャ・クライアントの認証が必要になります。

機密マネージャ・サーバ上での機密マネージャ・クライアントの認証は，LW-SSOコンポーネントに基づいています。機密マネージャ・サーバに接続する前に，機密マネージャ・クライアントはまずLW-SSOクッキーを送信します。機密マネージャ・サーバはクッキーを検証し，検証にパスしたら，機密マネージャ・クライアントとの通信が開始されます。LW-SSOの詳細については，45ページ「LW-SSO設定の構成」を参照してください。

機密マネージャ・クライアントと機密マネージャ・サーバ間の通信は暗号化されます。暗号化設定の更新の詳細については，45ページ「機密マネージャ・コミュニケーション暗号化の設定」を参照してください。

注意： 機密マネージャの認証では，コンピュータ上に定義された世界時(UTC)が使用される認証を正常に行うには，Data Flow ProbeとUCMDBサーバ上の世界時を同じにしてください。UTCはタイムゾーンやサマータイムとは独立しているため，サーバとプローブは異なるタイムゾーンでも可能です。

機密マネージャ・クライアントは資格情報のローカル・キャッシュを保持します。機密マネージャ・クライアントは，機密マネージャ・サーバからすべての資格情報をダウンロードしてキャッシュに保存するように設定されます。資格情報の変更は，機密マネージャ・サーバから継続的に自動同期されます。キャッシュは，事前設定によってファイルシステム・キャッシュまたはメモリ内キャッシュを使用できます。また，キャッシュは暗号化され，外部からはアクセスできません。キャッシュ設定の更新の詳細については，48ページ「プローブでの機密マネージャ・クライアントのキャッシュ・モードの設定」を参照してください。キャッシュの暗号化の更新の詳細については，49ページ「プローブでの機密マネージャ・クライアントのキャッシュ暗号化文字列の設定」を参照してください。

トラブルシューティングの詳細については，52ページ「資格情報マネージャ・クライアントのログ・ファイル・メッセージ・レベルの変更」を参照してください。

資格情報をUCMDBサーバ間でコピーできます。詳細については，50ページ「暗号化された形式による，資格情報および範囲情報のエクスポートとインポート」を参照してください。

注： プローブで(UCMDBバージョン9.01以前で)資格情報のストレージに使用されていた**DomainScopeDocument**(DSD)に，資格情報に関する機密情報が含まれなくなりました。現在，このファイルにはプローブのリストとネットワーク範囲情報が含まれます。各ドメインの資格情報エントリのリストも含まれます。各エントリには資格情報IDとネットワーク範囲(この資格情報エントリに定義されている)のみが含まれます。

本項の内容

セキュリティ上の基本的な前提条件

セキュリティ上の前提条件は次のとおりです。

UCMDBサーバおよびプローブJMXコンソールのセキュリティが確保されていて，UCMDBシステム管理者だけがアクセスを許可されるようになっている。localhostからのアクセスに限定されているのが望ましい。

別々のモードで実行されているデータ・フロー・プローブ

プローブ・ゲートウェイとマネージャを個別のプロセスとして実行する場合，資格情報マネージャ・クライアント・コンポーネントはマネージャ・プロセスの一部となります。資格情報はキャッシュされ，プローブ・マネージャによってのみ使用されます。UCMDBシステムで機密マネージャ・サーバにアクセスする場合，ゲートウェイ・プロセスで機密マネージャ・クライアント要求が処理され，そこからUCMDBに転送されます。

プローブが個別のモードで設定されている場合，この設定は自動です。

資格情報キャッシュを最新に保つ

機密マネージャ・サーバへの初回接続時，機密マネージャ・クライアントは関連するすべての資格情報をダウンロードします(プローブのドメインで設定されているすべての資格情報)。初回通信に成功した後，機密マネージャ・クライアントは機密マネージャ・サーバとの同期状態を維持します。1分間隔で差分同期が実行され，機密マネージャ・サーバと機密マネージャ・クライアント間の差異のみが同期されます。UCMDBサーバ側で資格情報が変更されると(新しい資格情報の追加，または既存の資格情報の更新/削除など)，機密マネージャ・クライアントはUCMDBサーバから直ちに通知を受信し，追加の同期を実行します。

すべてのプローブでの設定変更の同期

正常に通信するためには，機密マネージャ・クライアントを機密マネージャ・サーバの認証設定(LW-SSO init 文字列)および暗号化設定(機密マネージャ通信の暗号化)で更新する必要があります。たとえば，サーバで init 文字列が変更された場合，プローブは認証するために新しい init 文字列を認識している必要があります。

UCMDB サーバは機密マネージャ通信の暗号化設定および機密マネージャ認証設定の変更を常に監視します。このモニタリングは 15 秒毎に実施され，変更が発生すると，更新された設定がプローブに送信されます。設定は暗号化された形式でプローブに渡され，プローブ側の安全なストレージに保存されます。送信される設定は，対称暗号鍵を使用して暗号化されます。標準設定では，UCMDB サーバとData Flow Probe は同じ標準設定の対称暗号鍵を使用してインストールされます。最適なセキュリティを実現するために，システムに資格情報を追加する前に，この鍵を変更することをお勧めします。詳細については，53ページ「暗号鍵の生成または更新」を参照してください。

注： 監視間隔は 15 秒であるため，プローブ側の機密マネージャ・クライアントに 15 秒間，最新の設定が反映されない可能性があります。

UCMDB サーバとData Flow Probe 間の機密マネージャ通信および認証設定の自動同期を無効にする場合，UCMDB サーバ側で機密マネージャ通信および認証設定を更新するたびに，すべてのプローブも新しい設定で更新する必要があります。詳細については，46ページ「サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化」を参照してください。

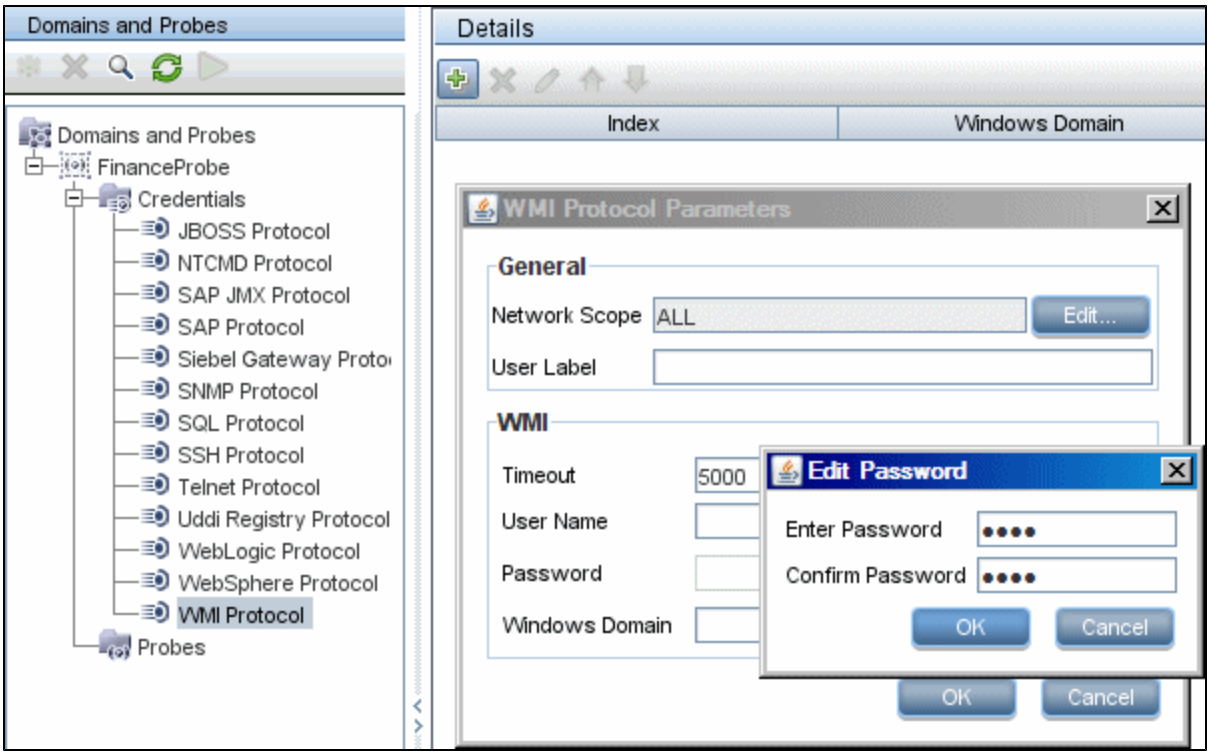
プローブ上の安全なストレージ

慎重に扱う必要があるすべての情報(機密マネージャ通信および認証設定と暗号化の鍵など)は，プローブで安全な保管場所の **C:\hp\UCMDB\DataFlowProbe\conf\security** にある **secured_storage.bin** ファイルに格納されます。このセキュリティ保護された保管場所は DPAPI を使用して暗号化されます。暗号化のプロセスでは Windows のユーザ・パスワードに依存します。DPAPI は，Windows システム上で証明書と秘密鍵などの機密データを保護する標準の方法です。パスワードが変更されてもセキュリティ保護された保管場所からプローブが情報を読み出せるように，プローブは常に同じ Windows ユーザで実行してください。

資格情報の表示

注： 本項では，データの方向がCMDB からHP Universal CMDB である場合の資格情報の表示について説明します。

パスワードは，CMDBからアプリケーションへは送信されません。つまり，HP Universal CMDB は，パスワード・フィールドの内容に関係なくアスタリスク(*)を表示します。



資格情報のアップグレード

注： 本項では，データの方向がHP Universal CMDBからCMDBである場合の資格情報の更新について説明します。

- この方向の通信は暗号化されないため，UCMDBサーバへはHTTPS/SSLを使用して接続するか，信頼されたネットワークを通して接続する必要があります。

  通信は暗号化されませんが，パスワードはネットワークでクリア・テキストとしては送信されません。パスワードは標準設定の鍵で暗号化されるため，送信中の機密性を高めるためにSSLを使用することを強くお勧めします。

- パスワードには，特殊文字や英字以外の文字も使用できます。

機密マネージャ・クライアント認証と暗号化設定の設定

このタスクは，UCMDBサーバ上の機密マネージャのクライアント認証および暗号化設定について記述するものであり，次のステップが含まれます。

- 45ページ「LW-SSO設定の構成」
- 45ページ「機密マネージャ・コミュニケーション暗号化の設定 」

LW-SSO 設定の構成

次の手順では，UCMDB サーバで LW-SSO init 文字列を変更する方法について説明します。変更を自動送信しないようにUCMDB サーバが設定されている場合を除き，この変更はプローブに(暗号化文字列として) 自動的に送信されます。詳細については，46ページ「サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化」を参照してください。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**
2. **UCMDB-UI:name=LW-SSO Configuration** をクリックして［JMX MBEAN View］ページを開きます。
3. **setInitString** メソッドを見つけます。
4. 新しい LW-SSO init 文字列を入力します。
5. ［Invoke］をクリックします。

機密マネージャ・コミュニケーション暗号化の設定

次の手順では，UCMDB サーバで機密マネージャ・コミュニケーション暗号化の設定を変更する方法について説明します。この設定では，機密マネージャ・クライアントと機密マネージャ・サーバ間の通信を暗号化する方法を指定します。変更を自動送信しないようにUCMDB サーバが設定されている場合を除き，この変更はプローブに(暗号化文字列として) 自動的に送信されます。詳細については，46ページ「サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化」を参照してください。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**
2. **UCMDB:service=Security Services** をクリックして，［JMX MBEAN View］ページを開きます。
3. **CMGetConfiguration** メソッドをクリックします。
4. ［**Invoke**］をクリックします。

   現在の機密マネージャ設定の XML が表示されます。
5. 表示された XML の内容をコピーします。
6. **セキュリティ・サービス**の［JMX MBEAN View］ページに戻ります。
7. **CMSetConfiguration** メソッドをクリックします。
8. コピーした XML を［**値**］フィールドに貼り付けます。
9. 関連する転送関連の設定を更新します。

   更新可能な値の詳細については，56ページ「資格情報マネージャの暗号化設定」を参照してください。

   例:

```
<transport>
        <encryptTransportMode>true</encryptTransportMode>
```

```
            <CMEncryptionDecryption>
                    <encryptDecryptInitString>radiohead</encryptDecryptInitString>
                    <cryptoSource>lw</cryptoSource>
                    <lwJCEPBECompatibilityMode>true</lwJCEPBECompatibilityMode>
                    <cipherType>symmetricBlockCipher</cipherType>
                    <engineName>AES</engineName>
                    <algorithmModeName>CBC</algorithmModeName>
                    <algorithmPaddingName>PKCS7Padding</algorithmPaddingName>
                    <keySize>256</keySize>
                    <pbeCount>20</pbeCount>
                    <pbeDigestAlgorithm>SHA1</pbeDigestAlgorithm>
                    <encodingMode>Base64Url</encodingMode>
                    <useMacWithCrypto>false</useMacWithCrypto>
                    <macType>hmac</macType>
                    <macKeySize>256</macKeySize>
                    <macHashName>SHA256</macHashName>
            </CMEncryptionDecryption>
    </transport>
```

10. [**Invoke**]をクリックします。

プローブでの手動での機密マネージャ・クライアントの認証設定および暗号化設定

本項の内容

サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化

標準設定では，UCMDB サーバは機密マネージャ / LW-SSO 設定をすべてのプローブに自動送信するように設定されています。この情報は暗号化された文字列としてプローブに送信され，取得時に復号化されます。機密マネージャ/LW-SSO 設定ファイルをすべてのプローブに自動送信しないよう

に, UCMDB サーバを設定できます。この場合, ユーザが手動で, すべてのプローブを新しい機密マネージャ / LW-SSO 設定で更新する必要があります。

機密マネージャ / LWSSO 設定の自動同期を無効にするには, 次の手順を実行します。

1. UCMDB で, **[管理]>[インフラストラクチャ設定マネージャ]>[全般設定]**をクリックします。
2. [**Enable automatic synchronization of CM/LW-SSO configuration and init string with probe**]を選択します。
3. [**値**]フィールドをクリックして, [**True**]から[**False**]に変更します。
4. [**保存**]ボタンをクリックします。
5. UCMDB サーバを再起動します。

プローブでの機密マネージャ・クライアントの認証設定および暗号化設定

次の手順は, UCMDB サーバが LW-SSO / 機密マネージャ設定をプローブに自動送信しないように設定されている場合に使用します。詳細については, 46ページ「サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化」を参照してください。

1. プローブ・マシンで Web ブラウザを起動し, 次のアドレスを入力します。**http://localhost:1977**

   注: プローブ・マネージャとプローブ・ゲートウェイが個別のプロセスとして実行されている場合は, プローブ・マネージャが実行されているマシンでアドレス **http://localhost:1978/jmx** を入力する必要があります。

2. **type=CMClient** をクリックして, [JMX MBEAN View]ページを開きます。
3. **setLWSSOInitString** メソッドを見つけて, UCMDB の LW-SSO 設定で指定した init 文字列を指定します。
4. [**setLWSSOInitString**]ボタンをクリックします。

プローブでの資格情報マネージャ通信の暗号化の設定

次の手順は, UCMDB サーバが LW-SSO / 機密マネージャ設定をプローブに自動送信しないように設定されている場合に使用します。詳細については, 46ページ「サーバとプローブ間の機密マネージャ・クライアント認証設定および暗号化設定の自動同期を無効化」を参照してください。

1. プローブ・マシンで Web ブラウザを起動し, 次のアドレスを入力します。**http://localhost:1977**

   注: プローブ・マネージャとプローブ・ゲートウェイが個別のプロセスとして実行されている場合は, プローブ・マネージャが実行されているマシンでアドレス **http://localhost:1978/jmx** を入力する必要があります。

2. **type=CMClient** をクリックして, [JMX MBEAN View]ページを開きます。
3. 次の転送関連の設定を更新します。

> **注：** UCMDB サーバで更新した設定と同じ設定を更新する必要があります。その際，プローブで更新するメソッドの中には，複数のパラメータが必要なものもあります。プローブの現在の設定を確認するには，[JMX MBEAN View]ページで **displayTransportConfiguration** をクリックします。詳細については，45ページ「機密マネージャ・コミュニケーション暗号化の設定」を参照してください。更新可能な値の詳細については，56ページ「資格情報マネージャの暗号化設定」を参照してください。

a. **setTransportInitString** は **encryptDecryptInitString** の設定を変更します。

b. **setTransportEncryptionAlgorithm** は次のマップに従ってプローブの機密マネージャ設定を変更します。

   - **エンジン名** は <engineName> エントリを指す
   - **鍵サイズ**は <keySize> エントリを指す
   - **アルゴリズム・パディング名** は <algorithmPaddingName> エントリを指す
   - **PBE カウント**は <pbeCount> エントリを指す
   - **PBE ダイジェスト・アルゴリズム**は <pbeDigestAlgorithm> エントリを指す

c. **setTransportEncryptionLibrary** は次のマップに従ってプローブの機密マネージャ設定を変更します。

   - **暗号化ライブラリ名** は <cryptoSource> エントリを指す
   - **軽量暗号化方式の以前のバージョンのサポート**は <lwJCEPBECompatibilityMode> エントリを指す

d. **setTransportMacDetails** は次のマップに従ってプローブの機密マネージャ設定を変更します。

   - **暗号化方式で MAC を使用**は <useMacWithCrypto> エントリを指す
   - **MAC 鍵サイズ**は <macKeySize> エントリを指す

4. [**reloadTransportConfiguration**]ボタンをクリックしてプローブで変更を有効にします。

さまざまな設定とその値の詳細については，56ページ「資格情報マネージャの暗号化設定」を参照してください。

資格情報マネージャ・クライアントのキャッシュの設定

本項の内容

- 48ページ「プローブでの機密マネージャ・クライアントのキャッシュ・モードの設定」
- 49ページ「プローブでの機密マネージャ・クライアントのキャッシュ暗号化文字列の設定」

プローブでの機密マネージャ・クライアントのキャッシュ・モードの設定

機密マネージャ・クライアントは資格情報をキャッシュに保存し，サーバ上で情報が変更されると情報を更新します。キャッシュはファイル・システムまたはメモリ内に保存できます。

- **ファイル・システムに保存する場合**，プローブが再起動されサーバに接続できない場合であっても資格情報を利用できます。
- **メモリ内に保存する場合**，プローブを再起動すると，キャッシュはクリアされすべての情報がサーバから再度取得されます。サーバを使用できない場合，プローブに資格情報は含まれないため，ディスカバリまたは統合は実行できません。

この設定を変更するには，次の手順を実行します。

1. **DataFlowProbe.properties** ファイルをテキスト・エディタで開きます。このファイルは，**c:\hp\UCMDB\DataFlowProbe\conf** フォルダにあります。
2. 次の属性を見つけます。**com.hp.ucmdb.discovery.common.security.storeCMData=true**
   - ファイル・システムに情報を保存するには，標準設定(**true**)をそのまま使用します。
   - 情報をメモリに保存するには，「**false**」と入力します。
3. **DataFlowProbe.properties** ファイルを保存します。
4. Probeを再起動します。

プローブでの機密マネージャ・クライアントのキャッシュ暗号化文字列の設定

次の手順では，機密マネージャ・クライアントのファイル・システム・キャッシュ・ファイルの暗号化設定を変更する方法について説明します。機密マネージャ・クライアントのファイル・システム・キャッシュの暗号化設定を変更すると，ファイル・システム・キャッシュ・ファイルが再作成されます。再作成プロセスでは，プローブを再起動してUCMDBサーバと完全同期する必要があります。

1. プローブ・マシンでWebブラウザを起動し，次のアドレスを入力します。**http://localhost:1977**

   注： プローブ・マネージャとプローブ・ゲートウェイが個別のプロセスとして実行されている場合は，プローブ・マネージャが実行されているマシンでアドレス**http://localhost:1978/jmx**を入力する必要があります。

2. **type=CMClient** をクリックして，[JMX MBEAN View]ページを開きます。
3. 次のキャッシュ関連の設定を更新します。

   注： プローブで更新するメソッドの中には，複数のパラメータが必要なものもあります。プローブの現在の設定を確認するには，[JMX MBEAN View]ページで**displayCacheConfiguration** をクリックします。

   a. **setCacheInitString** はファイル・システム・キャッシュの <encryptDecryptInitString> の設定を変更します。

   b. **setCacheEncryptionAlgorithm** は次のマップに従ってファイル・システム・キャッシュの設定を変更します。

      - **エンジン名** は <engineName> エントリを指す
      - **鍵サイズ**は <keySize> エントリを指す

- **アルゴリズム・パディング名** は <algorithmPaddingName> エントリを指す
- **PBE カウント** は <pbeCount> エントリを指す
- **PBE ダイジェスト・アルゴリズム** は <pbeDigestAlgorithm> エントリを指す

c. **setCacheEncryptionLibrary** は次のマップに従ってキャッシュ・ファイル・システムの設定を変更します。

- **暗号化ライブラリ名** は <cryptoSource> エントリを指す
- **軽量暗号化方式の以前のバージョンのサポート** は <lwJCEPBECompatibilityMode> エントリを指す

d. **setCacheMacDetails** は次のマップに従ってキャッシュ・ファイル・システムの設定を変更します。

- **暗号化方式で MAC を使用** は <useMacWithCrypto> エントリを指す
- **MAC 鍵サイズ** は <macKeySize> エントリを指す

4. [**reloadCacheConfiguration**]ボタンをクリックしてプローブで変更を有効にします。これによりプローブが再起動されます。

> **注：** この操作中は，プローブでジョブを実行しないようにしてください。

さまざまな設定とその値の詳細については，56ページ「資格情報マネージャの暗号化設定」を参照してください。

暗号化された形式による，資格情報および範囲情報のエクスポートとインポート

UCMDB サーバ間で資格情報をコピーするために，資格情報およびネットワーク範囲情報を暗号化形式でエクスポートおよびインポートできます。この操作は，システム・クラッシュ後のリカバリ，またはアップグレードのときなどに実行します。

- **資格情報をエクスポートする場合**，パスワードを入力または選択する必要があります。情報はこのパスワードで暗号化されます。
- **資格情報をインポートする場合**，DSD ファイルをエクスポートするときに設定したパスワードを使用する必要があります。

> **注：** エクスポートした資格情報ドキュメントには，ドキュメントのエクスポート元システムで指定された範囲情報も含まれます。資格情報ドキュメントのインポート時には，範囲情報もインポートされます。

> **注意：** UCMDB バージョン 8.02 domainScopeDocument から資格情報をインポートするには，バージョン 8.02 システムにある key.bin ファイルを使用する必要があります。

UCMDB サーバから資格情報をエクスポートするには，次の手順を実行します。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**ユーザ名とパスワードを使用してログインする必要がある場合もあります。
2. **UCMDB:service=DiscoveryManager** をクリックして，［JMX MBEAN View］ページを開きます。
3. **exportCredentialsAndRangesInformation** 操作を見つけます。次の操作を実行します。
   - 顧客 ID を入力します(標準設定は 1)。
   - エクスポートしたファイルの名前を入力します。
   - パスワードを入力します。
   - エクスポートしたファイルを指定したパスワードで暗号化する場合は **isEncrypted=True** を設定し，暗号化しない場合は **isEncrypted=False** を設定します(この場合，パスワードおよびその他の機密情報はエクスポートされません)。
4. ［**Invoke**］をクリックしてエクスポートします。

   エクスポート・プロセスが正常に完了すると，ファイルは次の場所に保存されます :**c:\hp\UCMDB\UCMDBServer\conf\discovery\<customer_dir>**。

UCMDB サーバから資格情報をインポートするには，次の手順を実行します。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。
2. **UCMDB:service=DiscoveryManager** をクリックして，［JMX MBEAN View］ページを開きます。
3. 次のいずれかの操作を見つけます。
   - インポートするファイルが，8.02 以降のバージョンのUCMDB サーバからエクスポートされた場合，**importCredentialsAndRangesInformation** 操作を見つけます。
   - インポートするファイルが，バージョン 8.02 の UCMDB サーバからエクスポートされた場合，**importCredentialsAndRangesWithKey** 操作を見つけます。
4. 顧客 ID を入力します(標準設定は 1)。
5. インポートするファイルの名前を入力します。このファイルは **c:\hp\UCMDB\UCMDBServer\conf\discovery\<customer_dir>** にあります。
6. パスワードを入力します。ファイルのエクスポート時に使用したパスワードを入力する必要があります。
7. ファイルが UCMDB バージョン 8.02 のシステムからエクスポートされた場合，**key.bin** ファイル名を入力します。このファイルは，インポートするファイルとともに **c:\hp\UCMDB\UCMDBServer\conf\discovery\<customer_dir>** にあります。
8. ［**Invoke**］をクリックして資格情報をインポートします。

資格情報マネージャ・クライアントのログ・ファイル・メッセージ・レベルの変更

Probeには，機密マネージャ・サーバと機密マネージャ・クライアント間の機密マネージャ関連通信についての情報を含む2つのログ・ファイルがあります。ファイルは次のとおりです。

- 52ページ「機密マネージャ・クライアント・ログ・ファイル」
- 52ページ「LW-SSO ログ・ファイル」

機密マネージャ・クライアント・ログ・ファイル

security.cm.log ファイルは **c:\hp\UCMDB\DataFlowProbe\runtime\log** フォルダにあります。

ログには，機密マネージャ・サーバと機密マネージャ・クライアント間でやり取りされた情報メッセージが含まれます。標準設定では，これらのメッセージのログ・レベルは INFO に設定されています。

メッセージのログ・レベルを DEBUG レベルに変更するには，次の手順を実行します。

1. Data Flow Probe Manager サーバで **c:\hp\UCMDB\DataFlowProbe\conf\log** に移動します。
2. **security.properties** ファイルをテキスト・エディタで開きます。
3. 次の行

   ```
   loglevel.cm=INFO
   ```

   を次に変えます。

   ```
   loglevel.cm=DEBUG
   ```

4. ファイルを保存します。

LW-SSO ログ・ファイル

security.lwsso.log ファイルは **c:\hp\UCMDB\DataFlowProbe\runtime\log** フォルダにあります。

このログには，LW-SSO に関する情報メッセージが含まれます。標準設定では，これらのメッセージのログ・レベルは INFO に設定されています。

メッセージのログ・レベルを DEBUG レベルに変更するには，次の手順を実行します。

1. Data Flow Probe Manager サーバで **c:\hp\UCMDB\DataFlowProbe\conf\log** に移動します。
2. security.properties ファイルをテキスト・エディタで開きます。
3. 次の行

   ```
   loglevel.lwsso=INFO
   ```

   を次に変えます。

   ```
   loglevel.lwsso=DEBUG
   ```

4. ファイルを保存します。

暗号鍵の生成または更新

UCMDB サーバとData Flow Probe 間の機密マネージャ通信および認証設定の暗号化または復号化に使用する暗号鍵を，生成または更新できます。生成または更新のいずれの場合も，UCMDB サーバは指定したパラメータ(鍵の長さ，追加 PBE サイクル，JCE プロバイダなど) に基づいて新しい暗号鍵を作成し，プローブに配布します。

generateEncryptionKey メソッドを実行すると，暗号鍵が新しく生成されます。この暗号鍵は安全なストレージにのみ保存され，名前などの詳細は認識されません。既存の Data Flow Probe を再インストールまたは新しいプローブを UCMDB サーバに接続する場合，新たに生成されたこの暗号鍵は新しいプローブには認識されません。このような場合，**changeEncryptionKey** メソッドを使用して暗号鍵を変更することをお勧めします。このようにすることで，Probe を再インストールまたは新しい Probe をインストールするときに，Probe JMX コンソールで **importEncryptionKey** メソッドを実行して (名前と場所を知っている) 既存の鍵をインポートできます。

注：

- 鍵の作成に使用するメソッド (**generateEncryptionKey**) と鍵の更新に使用するメソッド (**changeEncryptionKey**) の違いは，**generateEncryptionKey** は新しいランダム暗号鍵を作成し，**changeEncryptionKey** は指定した名前の暗号鍵をインポートするという点です。
- インストールされた Probe の数に関わらず，システムで使用できる暗号鍵は 1 つのみです。

本項の内容

新規暗号鍵の生成

UCMDB サーバとData Flow Probe で暗号化または復号化に使用される新しい鍵を生成できます。UCMDB サーバは，古い鍵を新しく生成した鍵と置き換え，この鍵をプローブに配布します。

JMX コンソールを使用して新しい暗号鍵を生成するには，次の手順を実行します。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

2. **UCMDB:service=DiscoveryManager** をクリックして，［JMX MBEAN View］ページを開きます。
3. generateEncryptionKey 操作を見つけます。
   a. ［**顧客 ID**］パラメータ・ボックスに 1(標準設定) を入力します。
   b. **keySize** に，暗号鍵の長さを指定します。有効な値は 128，192，256 です。

c. **usePBE** に，**True** または **False** を指定します。

- **True にすると**，追加 PBE ハッシュ・サイクルを使用します。
- **False にすると**，追加 PBE ハッシュ・サイクルを使用しません。

d. **jceVendor** には，標準設定でない JCE プロバイダの使用を選択できます。このボックスが空のときは，標準設定のプロバイダが使用されます。

e. **autoUpdateProbe** に，**True** または **False** を指定します。

- **true の場合 :**サーバは新しい鍵を自動的に Probe に配布します。
- **false の場合 :**新しい鍵を手動で Probe に配置してください。

f. **exportEncryptionKey** に，**True** または **False** を指定します。

- **true の場合 :**新しいパスワードを作成して安全なストレージに保存するほかに，サーバは新しいパスワードをファイル・システム (**c:\hp\UCMDB\UCMDBServer\conf\discovery\key.bin**) にエクスポートします。このオプションにより，新しいパスワードを使用して Probe を手動で更新できます。
- **false の場合 :**新しいパスワードはファイル・システムにエクスポートされません。Probe を手動で更新するには，**autoUpdateProbe** を False に設定し，**exportEncryptionKey** を True に設定します。

> **注 :** Probe が稼動していて，サーバに接続されていることを確認します。IProbe が停止している場合，鍵は Probe に接続できません。プローブが停止する前に鍵を変更した場合は，プローブが再起動するとそのプローブに鍵が再送信されます。ただし，Probe が停止する前に鍵を複数回変更した場合は，JMX コンソールを通して手動でその鍵を変更する必要があります(**exportEncryptionKey** に **False** を選択します)。

4. [**Invoke**]をクリックして，暗号鍵を生成します。

UCMDB サーバでの暗号鍵の更新

changeEncryptionKey メソッドを使用して独自の暗号鍵を UCMDB サーバにインポートし，すべてのプローブに配布します。

JMX コンソールを通して暗号鍵を更新するには，次の手順を実行します。

1. UCMDB サーバで，Web ブラウザを起動して次のアドレスを入力します。**http://localhost:8080/jmx-console**

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

2. **UCMDB:service=DiscoveryManager** をクリックして，[JMX MBEAN View]ページを開きます。
3. **changeEncryptionKey** 操作を見つけます。
   a. [**顧客 ID**]パラメータ・ボックスに **1** (標準設定) を入力します。
   b. **newKeyFileName** に，新しい鍵の名前を入力します。
   c. **keySizeInBits** に，暗号鍵の長さを指定します。有効な値は 128，192，256 です。

d. **usePBE** に，**True** または **False** を指定します。

- **true の場合 :**追加 PBE ハッシュ・サイクルを使用します。
- **false の場合 :**追加 PBE ハッシュ・サイクルを使用しません。

e. **jceVendor** には，標準設定でない JCE プロバイダの使用を選択できます。このボックスが空のときは，標準設定のプロバイダが使用されます。

f. **autoUpdateProbe** に，**True** または **False** を指定します。

- **true の場合 :**サーバは新しい鍵を自動的に Probe に配布します。
- **false の場合 :**プローブ JMX コンソールを使用して，新しい鍵を手動で配布する必要があります。

> **注 :** Probe が稼動していて，サーバに接続されていることを確認します。IProbe が停止している場合，鍵は Probe に接続できません。プローブが停止する前に鍵を変更した場合は，プローブが再起動するとそのプローブに鍵が再送信されます。ただし，Probe が停止する前に鍵を複数回変更した場合は，JMX コンソールを通して手動でその鍵を変更する必要があります(**autoUpdateProbe** に **False** を選択します) 。

4. [**Invoke**]をクリックして，暗号鍵を生成および更新します。

プローブでの暗号鍵の更新

セキュリティを考慮して，暗号鍵を UCMDB サーバからすべてのプローブに自動配布しない場合，新しい暗号鍵をすべてのプローブにダウンロードし，プローブで **importEncryptionKey** メソッドを実行する必要があります。

1. 暗号鍵ファイルを **C:\hp\UCMDB\DataFlowProbe\conf\security** ディレクトリに置きます。
2. プローブ・マシンで Web ブラウザを起動し，次のアドレスを入力します。**http://localhost:1977**

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

   > **注 :** プローブ・マネージャとプローブ・ゲートウェイが個別のプロセスとして実行されている場合は，プローブ・マネージャが実行されているマシンでアドレスを入力する必要があります。**http://localhost:1978**

3. プローブのドメイン上で **type=SecurityManagerService** をクリックします。
4. **importEncryptionKey** メソッドを見つけます。
5. **C:\hp\UCMDB\DataFlowProbe\conf\security** に置かれる暗号鍵ファイルの名前を入力します。このファイルには，インポートされる鍵が含まれています。
6. [**importEncryptionKey**]ボタンをクリックします。
7. プローブの再起動を実行します。

プローブ・マネージャとプローブ・ゲートウェイが別々のマシンにインストールされている場合に暗号鍵を手動で変更

1. プローブ・マネージャ・マシンで，プローブ・マネージャ・サービスを起動します(**[スタート]>[プログラム]>[HP UCMDB]>[プローブ マネージャ]**を選択します)。
2. プローブ・マネージャ JMX を使用して，サーバから鍵をインポートします。詳細については，53ページ「新規暗号鍵の生成」を参照してください。
3. 暗号鍵のインポートが完了したら，プローブ・マネージャおよびプローブ・ゲートウェイのサービスを再起動します。

複数の JCE プロバイダの定義

JMX コンソールを使用して暗号鍵を生成する場合，**changeEncryptionKey** と **generateEncryptionKey** メソッドを使用して複数の JCE プロバイダを定義できます。

標準設定の JCE プロバイダを変更するには，次の手順を実行します。

1. **$JRE_HOME/lib/ext** の JCE プロバイダ jar ファイルを登録します。
2. jar ファイルを $JRE_HOME フォルダにコピーします。
   - UCMDB サーバの場合 :次のディレクトリにある $JRE_HOME へコピーします。**c:\hp\UCMDB\UCMDBServer\bin\jre**
   - Data Flow Probe の場合 :次のディレクトリにある $JRE_HOME へコピーします。**c:\hp\UCMDB\DataFlowProbe\bin\jre**
3. **$JRE_HOME\lib\security\java.security** ファイルのプロバイダ・リストの最後にプロバイダ・クラスを追加します。
4. 無制限 JCE ポリシーを含めるように，**local_policy.jar** と **US_export_policy.jar** ファイルを更新します。これらの jar ファイルは Sun の Web サイトからダウンロードできます。
5. UCMDB サーバと Data Flow Probe を再起動します。
6. **changeEncryptionKey** または **generateEncryptionKey** メソッド用の JCE ベンダ・フィールドを見つけて，JCE プロバイダの名前を追加します。

資格情報マネージャの暗号化設定

次の表に，さまざまな JMX メソッドを使用して変更可能な暗号化設定を示します。これらの暗号化設定は，構成マネージャ・クライアントと構成マネージャ・サーバ間の通信の暗号化，および構成マネージャ・クライアントのキャッシュの暗号化に使用されます。

資格情報マネージャの設定名	プローブ資格情報マネージャの設定名	設定の詳細	利用可能な値	標準設定値
cryptoS-ource	暗号化ライブラリ名	この設定では，使用する暗号化ライブラリ	lw，jce，windowsDPAPI，lwJ-	lw

資格情報マネージャの設定名	プローブ資格情報マネージャの設定名	設定の詳細	利用可能な値	標準設定値
		を指定します。	CECompatible	
lwJCEP-BE 互換性モード	軽量暗号化方式の以前のバージョンのサポート	この設定では，軽量暗号化方式の以前のバージョンをサポートするかどうかを指定します。	true，false	true
engineN-ame	エンジン名	暗号化メカニズム名	AES，DES，3DES，Blowfish	AES
keySize	鍵サイズ	暗号鍵の長さ(ビット)	AES の場合：128，192，256。 DES の場合：64。 3DES の場合：192。 Blowfish の場合：32 から 448 までの任意の数値	256
アルゴリズム パディング 名前	アルゴリズム・パディング名	パディングの標準	PKCS7Padding，PKCS5Paddi-ng	PKCS7Pa-dding
pbeCou-nt	PBE カウント	パスワード(init 文字列)から鍵を作成するためにハッシュを実行する回数	任意の正数	20
pbeDige-st アルゴリズム	PBE ダイジェスト・アルゴリズム	ハッシュ・タイプ	SHA1，SHA256，MD5	SHA1
useMac-With 暗号化	暗号化方式で MAC を使用	暗号化方式で MAC を使用するかどうかを指定	true，false	false
macKey-Size	MAC 鍵サイズ	MAC アルゴリズムによって異なる	256	256

トラブルシューティングおよび制限事項

UCMDB サーバ上のデフォルトのドメイン名を変更する場合，まず Data Flow Probe が実行されていないことを確認する必要があります。デフォルトのドメイン名を適用した後は，Data Flow Probe 側で **DataFlowProbe\tools\clearProbeData.bat** スクリプトを実行する必要があります。

注： clearProbeData.bat スクリプトを実行すると, プローブがアップされたときにプローブ側でディレクトリ・サイクルが生じます。

第5章

データ・フロー・プローブの強化

本章の内容

MySQL データベースに暗号化パスワードの修正

本項では，MySQL データベース・ユーザの暗号化されたパスワードの修正方法について説明します。

1. パスワードの暗号化形式(AES，192 ビット鍵)を作成します。

   a. Data Flow Probe JMX コンソールにアクセスします。Web ブラウザを起動し，アドレスに**「http://<Data Flow Probe machine name or IP address<:1977」と入力します。**Data Flow Probe をローカルで実行している場合は，**http://localhost:1977** と入力します。

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

   > **注：** ユーザを作成していない場合は，標準設定のユーザ名 sysadmin とパスワード sysadmin を使用してログインします。

b. **Type=MainProbe** サービスを見つけ，リンクをクリックして［操作］ページを開きます。

c. **getEncryptedDBPassword** 操作を見つけます。

d. ［**DB パスワード**］フィールドに，暗号化するパスワードを入力します。

e. ［**getEncryptedDBPassword**］ボタンをクリックして操作を呼び出します。

この呼び出しの結果は，次のような暗号化されたパスワード文字列となります。

```
66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,114,
112,65,61,61
```

2. Data Flow Probe を停止します。

［スタート］>［すべてのプログラム］>［HP UCMDB］>［Data Flow Probe の停止］を選択します。

3. set_dbuser_password.cmd スクリプトを実行します。

このスクリプトは，次のフォルダにあります。**C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\set_dbuser_password.cmd**

新しいパスワードを第 1 引数，MySQL ルート・アカウントのパスワードを第 2 引数として **set_dbuser_password.cmd** スクリプトを実行します(MySQL ルート・アカウントがパスワード保護されていない場合は空白)。

例：

set_dbuser_password <my_password><root_password>

パスワードは，暗号化していない形式 (平文) で入力する必要があります。

4. Data Flow Probe の構成ファイルにあるパスワードを更新します。

a. 構成ファイルに書き込むパスワードは暗号化する必要があります。暗号化された形式のパスワードを取得するには，ステップ 1 の説明に従って **getEncryptedDBPassword** JMX メソッドを使用します。

b. 暗号化されたパスワードを，**C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** ファイルの次のプロパティに追加します。

- **appilog.agent.probe.jdbc.pwd**

  例：

  ```
  appilog.agent.probe.jdbc.user = mamprobe
  appilog.agent.probe.jdbc.pwd =
  66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,
  114,112,65,61,61
  ```

- **appilog.agent.local.jdbc.pwd**
- **appilog.agent.normalization.jdbc.pwd**

5. Data Flow Probe を起動します。

［**スタート**］>［**すべてのプログラム**］>［**HP UCMDB**］>［**Data Flow Probe を開始**］の順に選択します。

clearProbeData.bat スクリプト :使用法

clearProbeData.bat スクリプトは，現在のパスワードを変更せずにデータベース・ユーザを再作成します。

スクリプトは MySQL ルート・アカウントのパスワードを第 1 引数として受け取ることが想定されます。転送されるパラメータがない場合，MySQL ルート・アカウントのパスワードが空白であると見なされます。

スクリプトの実行後：

- C:
  C:\hp\UCMDB\DataFlowProbe\runtime\log\probe_setup.log
- データベースのパスワードが記録されているため，**C:\hp\UCMDB\DataFlowProbe\runtime\log\probe_setup.log**

JMX コンソールに暗号化パスワードを設定

本項では，JMX ユーザのパスワードの暗号化方法について説明します。暗号化されたパスワードは DataFlowProbe.properties ファイルに保存されます。JMX コンソールにアクセスするには，ログインする必要があります。

1. **パスワードの暗号化形式(AES，192 ビット鍵)を作成します。**
   a. Data Flow Probe JMX コンソールにアクセスします。Web ブラウザを起動し，アドレスに**「http://<Data Flow Probe machine name or IP address<:1977」と入力します。**Data Flow Probe をローカルで実行している場合は，**http://localhost:1977** と入力します。

      ユーザ名とパスワードを使用してログインする必要がある場合もあります。

      > **注：** ユーザを作成していない場合は，標準設定のユーザ名 sysadmin とパスワード sysadmin を使用してログインします。

   b. **Type=MainProbe** サービスを見つけ，リンクをクリックして[操作] ページを開きます。
   c. **getEncryptedKeyPassword** 操作を見つけます。
   d. [**Key Password**]フィールドに，暗号化するパスワードを入力します。
   e. [**getEncryptedKeyPassword**]ボタンをクリックして操作を呼び出します。

      この呼び出しの結果は，次のような暗号化されたパスワード文字列となります。

      ```
      85,-9,-61,11,105,-93,-81,118
      ```

2. **Data Flow Probe を停止します。**

   [スタート]>[すべてのプログラム]>[HP UCMDB]>[Data Flow Probe の停止]を選択します。

3. **暗号化されたパスワードを，**

   暗号化されたパスワードを，**C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** ファイルの次のプロパティに追加します。

   appilog.agent.Probe.JMX.BasicAuth.Pwd

例：

```
appilog.agent.Probe.JMX.BasicAuth.User=sysadmin
appilog.agent.Probe.JMX.BasicAuth.Pwd=12,-35,-37,82,-2,20,57,-40,
38,80,-111,-99,-64,-5,35,-122
```

> **注：** 認証を無効にするには，これらのフィールドを空白のままにします。認証を無効にすると，ユーザは認証情報を入力せずにプローブのJMXコンソールのメイン・ページを開くことができます。

4. **Data Flow Probeを起動します。**

   [スタート]>[すべてのプログラム]>[HP UCMDB]>[Data Flow Probeの起動]を選択します。

   Webブラウザで結果をテストします。

UpLoadScanFileのパスワード設定

本項では，オフサイト・スキャン保存に使用する**UpLoadScanFile**のパスワード設定方法について説明します。暗号化されたパスワードは**DataFlowProbe.properties**ファイルに保存されます。JMXコンソールにアクセスするには，ログインする必要があります。

1. **パスワードの暗号化形式(AES，192ビット鍵)を作成します。**

   a. Data Flow Probe JMXコンソールにアクセスします。Webブラウザを起動し，アドレスに**「http://<Data Flow Probe machine name or IP address<:1977」と入力します。**Data Flow Probeをローカルで実行している場合は，**http://localhost:1977**と入力します。

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

   > **注：** ユーザを作成していない場合は，標準設定のユーザ名 sysadminとパスワード sysadminを使用してログインします。

   b. **Type=MainProbe**サービスを見つけ，リンクをクリックして[操作]ページを開きます。

   c. **getEncryptedKeyPassword**操作を見つけます。

   d. [**Key Password**]フィールドに，暗号化するパスワードを入力します。

   e. [**getEncryptedKeyPassword**]ボタンをクリックして操作を呼び出します。

   この呼び出しの結果は，次のような暗号化されたパスワード文字列となります。

   ```
   85,-9,-61,11,105,-93,-81,118
   ```

2. **Data Flow Probeを停止します。**

   [スタート]>[すべてのプログラム]>[HP UCMDB]>[Data Flow Probeの停止]を選択します。

3. **暗号化されたパスワードを，**

   暗号化されたパスワード
   を，**C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties**ファイルの次のプロパティに追加します。

   appilog.agent.Probe.JMX.BasicAuth.Pwd

例：

```
com.hp.ucmdb.discovery.Probe.JMX.UploadAuth.User=UploadScanFile
com.hp.ucmdb.discovery.Probe.JMX.UploadAuth.Pwd=116,116,21,34,-59,
77,-108,14,127,4,-89,101,-33,-31,116,53
```

4. **Data Flow Probe を起動します。**

   [スタート]>[すべてのプログラム]>[HP UCMDB]>[Data Flow Probe の起動]を選択します。

   Web ブラウザで結果をテストします。

MySQL サーバへのリモート・アクセス

本項では，リモートのマシンからの MySQL Data Flow Probe アカウントへのアクセスを許可 / 制限する方法について説明します。

> **注：**
> - 標準設定では，アクセスは制限されています。
> - リモートのマシンから MySQL ルート・アカウントにはアクセスできません。

MySQL アクセスを許可するには，次の手順を実行します。

1. コマンド・プロンプト・ウィンドウで次のスクリプトを実行します。

   C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\enable_remote_user_access.cmd

2. パスワードを要求されたら MySQL ルート・アカウントのパスワードを第 1 引数として入力します(このパスワードはプローブのインストール時に入力したパスワードと同じです)。

MySQL アクセスを制限するには，次の手順を実行します。

1. コマンド・プロンプト・ウィンドウで次のスクリプトを実行します。

   C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\remove_remote_user_access.cmd

2. パスワードを要求されたら MySQL ルート・アカウントのパスワードを第 1 引数として入力します(このパスワードはプローブのインストール時に入力したパスワードと同じです)。

UCMDB サーバとデータ・フロー・プローブ間で，相互認証による SSL を有効化

Data Flow Probe とUCMDB サーバの両方で，証明書による認証を設定できます。設定すると，接続を確立する前に各コンポーネントの証明書が送信されて認証されます。

> **注：** Data Flow Probe でSSL を使用して相互認証を有効化する方法は最も安全で，推奨される通信モードです。この方法は基本認証の手順に代わるものです。

本項の内容

概要

UCMDBは，UCMDBサーバとData Flow Probeの間の通信で次のモードをサポートしています。

- **サーバ認証:** このモードでは，SSLを使用し，プローブはUCMDBサーバの証明書を認証します。詳細については，64ページ「サーバ認証(一方向)でのSSLの有効化」を参照してください。
- **相互認証:** このモードでは，SSLを使用し，プローブによるサーバ認証とサーバによるクライアント認証の両方を実行できます。詳細については，67ページ「サーバ証明書認証(双方向)の有効化」を参照してください。
- **標準 HTTP:** SSL通信は行われません。これは標準設定モードで，UCMDBのData Flow Probeコンポーネントに証明書は必要ありません。Data Flow Probeは標準HTTPプロトコルを使用してサーバと通信します。

注： SSLでの作業時にはディスカバリは証明書チェーンを使用できません。そのため，証明書チェーンを使用している場合，UCMDBサーバとの通信を行うにはData Flow Probe用のセルフ署名証明書を生成する必要があります。

キー・ストアとトラスト・ストア

UCMDBサーバとData Flow Probeは，キー・ストアとトラスト・ストアを使用して動作します。

- **キー・ストア:** キー・エントリ(証明書および一致する秘密鍵)を保持するファイル。
- **トラスト・ストア:** リモート・ホストを検証するために使用する証明書を保持するファイル(たとえば，サーバ認証で使用する場合，Data Flow Probeのトラスト・ストアにはUCMDBサーバの証明書が含まれている必要があります)。

相互認証の制限事項

C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties で定義したData Flow Probeキー・ストアには，キー・エントリが1つだけ含まれている必要があります。

サーバ認証(一方向)でのSSLの有効化

ここではSSLを使用し，プローブはサーバの証明書を認証します。

本項の内容

前提条件

1. UCMDBとData Flow Probeの両方が実行されていることを確認します。

> **注:** プローブが別々のモードでインストールされている場合，次の手順はプローブ・ゲートウェイを示します。

2. UCMDB または Data Flow Probe が標準設定フォルダにインストールされていない場合，正しい場所を確認してコマンドを適宜変更します。

UCMDB サーバの構成

1. **UCMDB 証明書のエクスポート**

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias
   <keystore alias> -keystore <＜キーストアのファイル・パス＞> -file
   C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   詳細:

   - **keystore alias** はキーストアに与えられた名前です。
   - **キーストアのファイル・パス**はキーストア・ファイルの場所のフル・パスです。

   例えば，追加設定なしの server.keystore の場合は次のコマンドを使用します。

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -
   alias hpcert -keystore
   C:\hp\ucmdb\ucmdbserver\conf\security\server.keystore -file
   C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   b. キーストアのパスワードを入力します。例えば，設定変更なしのキーストア・パスワードは **hppass** です。

   c. 証明書が次のディレクトリに作成されていることを確認します。**C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**

2. **UCMDB の Data Flow Probe コネクタのセキュリティ強化**

   a. UCMDB JMX コンソールにアクセスします。Web ブラウザで URL(**http://＜ucmdb マシンの名前または IP アドレス＞:8080/jmx-console**) を入力します。ユーザ名とパスワードを使用してログインする必要がある場合もあります。

   b. [**Ports Management Services**] のサービスを選択します。

   c. **PortsDetails** メソッドを呼出し，HTTPS のポート番号を確認します。(標準設定 :8443) [**有効**] 絡むの値が[**True**] であることを確認します。

   d. [**Ports Management Services**] に戻ります。

   e. Data Flow Probe コネクタをサーバ認証モードにマップするには，次のパラメータを指定して **mapComponentToConnectors** メソッドを呼び出します。

   - **componentName**:mam-collectors
   - **isHTTPSWithClientAuth**:true
   - **ほかのすべてのフラグ** : false

   次のメッセージが表示されます。

> **Operation succeeded.Component mam-collectors is now mapped to: HTTPS ports.**

f. [**Ports Management Services**]に戻ります。

g. 資格情報マネージャ・コネクタをサーバ認証モードにマップするには，次のパラメータを指定して **mapComponentToConnectors** メソッドを呼び出します。

- **componentName**:cm
- **isHTTPSWithClientAuth**:true
- **ほかのすべてのフラグ** : false

次のメッセージが表示されます。

> **Operation succeeded.Component cm is now mapped to: HTTPS ports.**

3. **UCMDB 証明書を各プローブ・マシンにコピー**

UCMDB サーバ・マシンの証明書ファイル (**C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**) を各 Data Flow Probe マシン上の次のフォルダにコピーします **C:\HP\UCMDB\DataFlowProbe\conf\security**

Data Flow Probe 設定

> **注：** 各 Data Flow Probe マシンに設定を行う必要があります。

1. 64ページ「サーバ認証(一方向)でのSSLの有効化」 **で作成された server.cert ファイルをプローブのトラスト・ストアにインポートします。**

a. コマンド・プロンプトを開いて次のコマンドを実行します。

```
C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -import -v -
keystore
C:\HP\UCMDB\DataFlowProbe\conf\security\MAMTrustStoreExp.jks -
file C:\HP\UCMDB\DataFlowProbe\conf\security\server.cert -alias
ucmdbcert
```

b. キーストアのパスワード，logomaniaを入力します。

c. [**Trust this certificate?**]と表示された場合は，[**y**]を押して **Enter** キーを押します。

次のメッセージが表示されます。

> **証明書がキーストアに追加されました。**

2. **C:\HP\UCMDB\DataFlowProbe\conf\ にある DiscoveryProbe.properties ファイルを開きます**

a. **appilog.agent.probe.protocol** プロパティを **HTTPS** に更新します。

b. **serverPortHttps** プロパティを，ポート番号に更新します。(65ページ「UCMDB サーバの構成」のステップ 2c のポート番号を使用します。)

マシンの再起動

UCMDB サーバとプローブ・マシンを両方再起動します。

サーバ証明書認証(双方向)の有効化

このモードでは，SSL を使用し，プローブによるサーバ認証とサーバによるクライアント認証の両方を実行できます。サーバとプローブの両方が，認証のために証明書をほかのエンティティに送信します。

本項の内容

前提条件

1. UCMDB と Data Flow Probe の両方が実行されていることを確認します。

   注： プローブが別々のモードでインストールされている場合，次の手順はプローブ・ゲートウェイを示します。

2. UCMDB または Data Flow Probe が標準設定フォルダにインストールされていない場合，正しい場所を確認してコマンドを適宜変更します。

UCMDB サーバの初期設定

1. **UCMDB 証明書のエクスポート**

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias
   <keystore alias> -keystore <＜キーストアのファイル・パス＞> -file
   C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   詳細：

   - **keystore alias** はキーストアに与えられた名前です。
   - **キーストアのファイル・パス**はキーストア・ファイルの場所のフル・パスです。

   例えば，追加設定なしの server.keystore の場合は次のコマンドを使用します。

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -
   alias hpcert -keystore
   C:\hp\ucmdb\ucmdbserver\conf\security\server.keystore -file
   C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   b. キーストアのパスワードを入力します。例えば，設定変更なしのキーストア・パスワードは **hppass** です。

c. 証明書が次のディレクトリに作成されていることを確認します。**C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**

2. **UCMDB の Data Flow Probe コネクタのセキュリティ強化**

a. UCMDB JMX コンソールにアクセスします。Web ブラウザで URL(**http://<<ucmdb マシンの名前または IP アドレス>>:8080/jmx-console**) を入力します。ユーザ名とパスワードを使用してログインする必要がある場合もあります。

b. [**Ports Management Services**] のサービスを選択します。

c. **PortsDetails** メソッドを呼出し，クライアント認証で HTTPS のポート番号を確認します。(標準設定 :8444) [**有効**] 絡むの値が[**True**]であることを確認します。

d. [**Ports Management Services**]に戻ります。

e. Data Flow Probe コネクタを相互認証モードにマップするには，次のパラメータを指定して **mapComponentToConnectors** メソッドを呼び出します。

- **componentName**:mam-collectors
- **isHTTPSWithClientAuth**: true
- **ほかのすべてのフラグ** : false

次のメッセージが表示されます。

Operation succeeded.Component mam-collectors is now mapped to: HTTPS_CLIENT_AUTH ports.

f. [**Ports Management Services**]に戻ります。

g. 資格情報マネージャ・コネクタを相互認証モードにマップするには，次のパラメータを指定して **mapComponentToConnectors** メソッドを呼び出します。

- **componentName**:cm
- **isHTTPSWithClientAuth**: true
- **ほかのすべてのフラグ** : false

次のメッセージが表示されます。

Operation succeeded.Component cm is now mapped to: HTTPS_CLIENT_AUTH ports.

3. **UCMDB 証明書を各プローブ・マシンにコピー**

UCMDB サーバ・マシンの証明書ファイル (**C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**) を各 Data Flow Probe マシン上の次のフォルダにコピーします。**C:\HP\UCMDB\DataFlowProbe\conf\security**

Data Flow Probe 設定

注： 各 Data Flow Probe マシンに設定を行う必要があります。

1. 67ページ「UCMDB 証明書のエクスポート」 **で作成された server.cert ファイルをプローブのトラスト・ストアにインポートします。**

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -import -v -
   keystore
   C:\HP\UCMDB\DataFlowProbe\conf\security\MAMTrustStoreExp.jks -
   file C:\HP\UCMDB\DataFlowProbe\conf\security\server.cert -alias
   ucmdbcert
   ```

   b. キーストアのパスワード，logomaniaを入力します。

   c. [**Trust this certificate?**]と表示された場合は，[**y**]を押して **Enter** キーを押します。

   次のメッセージが表示されます。

   証明書がキーストアに追加されました。

2. **client.keystore ファイルの新規作成**

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool -genkey -alias
   <ProbeName> -keyalg RSA -keystore
   C:\HP\UCMDB\DataFlowProbe\conf\security\client.keystore
   ```

   ここでは，**ProbeName** は Data Flow Probe の一意のエイリアスを示します。

   注： このエイリアスが一意であることを確認するには，プローブ定義時にプローブに付与されたプローブ名識別子を使用します。

   b. キーストアのパスワードを最低 6 文字で入力し，それを記録しておきます。

   c. 確認のために再度パスワードを入力します。

   d. **Enter** を押して次の質問それぞれに回答してください。

   あなたの氏名を教えてください。[不明]:

   あなたの部署名を教えてください。[不明]:

   あなたの組織を教えてください。[不明]:

   あなたの住所(市以下)を教えてください。[不明]:

   あなたの住所(都道府県)を教えてください。[不明]:

   このユニットの 2 文字の国コードを教えてください。[不明]:

   e. 「**CN=不明，OU=不明，O=不明，L=不明，ST=不明，C=不明でよろしいですか?**」と表示されたら**はい**と入力します。

   f. **Enter** を押して次の質問に回答してください。

   <probekey> のキー・パスワードを入力します(キーストア・パスワードと同じ場合は戻る)。

g. ファイルが次のフォルダに作成されており，そのファイル・サイズが0以上であることを確認します。**C:\hp\UCMDB\DataFlowProbe\conf\security\client.keystore**

3. **新しいクライアント証明書のエクスポート**

a. コマンド・プロンプトを開いて次のコマンドを実行します。

```
C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -export -alias
<ProbeName> -keystore
C:\hp\UCMDB\DataFlowProbe\conf\security\client.keystore -file
C:\hp\UCMDB\DataFlowProbe\conf\security\<ProbeName>.cert
```

b. 求められたらキーストアのパスワードを入力します。(上記ステップ 2b のパスワード。)

次のメッセージが表示されます。

> **<C:\hp\UCMDB\DataFlowProbe\conf\security\<ProbeName>.cert> のファイルに保存されている証明書**

4. **C:\HP\UCMDB\DataFlowProbe\conf\ にある DiscoveryProbe.properties ファイルを開きます**

a. **appilog.agent.probe.protocol** プロパティを **HTTPS** に更新します。

b. **serverPortHttps** プロパティを，ポート番号に更新します。(67ページ「UCMDB サーバの初期設定」のステップ 2c のポート番号を使用します。)

5. **C:\HP\UCMDB\DataFlowProbe\conf\security にあるプローブの ssl.properties ファイルを開きます。**

a. **javax.net.ssl.keyStore** プロパティを **client.keystore** に更新します。

b. 上記ステップ 2b のパスワードを暗号化します。

i. Data Flow Probe を起動するか，すでに実行されていることを確認します。

ii. Probe JMX にアクセスします。次の場所を参照します。**http://<probe_hostname>:1977**

例えば，プローブをローカルで実行している場合は次の場所を参照します。**http://localhost:1977**

iii. **type=MainProbe** のリンクをクリックします。

iv. **getEncryptedKeyPassword** の操作まで下にスクロールします。

v. **[Key Password]** フィールドにパスワードを入力します。

vi. [**getEncryptedKeyPassword**]ボタンを押します。

c. 暗号化したパスワードをコピーおよび貼り付けして **javax.net.ssl.keyStorePassword** プロパティを更新します。

> **注：** 数字はカンマで区切ります。例：-20,50,34,-40,-50.)

6. **プローブ証明書を UCMDB マシンにコピー**

C:\HP\UCMDB\DataFlowProbe\conf\security\client.cert のファイルを Data Flow Probe マシンから **C:\HP\UCMDB\UCMDBServer\conf\security\<ProbeName>.cert** の UCMDB マシンにコピーします。

追加の UCMDB サーバの構成

1. **各プローブ証明書の UCMDB トラスト・ストアへの追加**

   注： 各プローブ証明書ごとに次のステップを実施する必要があります。

   a. コマンド・プロンプトを開いて次のコマンドを実行します。

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -import -v -
   keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.truststore
   -file C:\hp\UCMDB\UCMDBServer\conf\security\<ProbeName>.cert -
   alias <ProbeName>
   ```

   b. キーストアのパスワードを入力します。例えば，設定変更なしのキーストア・パスワードは **hppass** です。

   c. [**Trust this certificate?**]と表示された場合は，[**y**]を押して **Enter** キーを押します。

   次のメッセージが表示されます。

   証明書がキーストアに追加されました

マシンの再起動

UCMDB サーバとプローブ・マシンを両方再起動します。

domainScopeDocument ファイルの場所を管理

Probe のファイル・システムは，暗号鍵と **domainScopeDocument** ファイルの両方を(標準設定で)保持しています。Probe は，起動するたびに，サーバから **domainScopeDocument** ファイルを取得してファイル・システムに格納します。承認されていないユーザがそれらの資格情報を取得するのを防ぐために，**domainScopeDocument** ファイルが Probe のメモリに保持され，Probe のファイル・システムには格納されないように Probe を設定できます。

domainScopeDocument ファイルの場所を制御するには，次の手順を実行します。

1. **C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** を開き，を次に変えます。

   ```
   appilog.collectors.storeDomainScopeDocument=true
   ```

   を次に変えます。

   ```
   appilog.collectors.storeDomainScopeDocument=false
   ```

   これで，Probe Gateway と Probe Manager の serverData フォルダに **domainScopeDocument** ファイルが存在しなくなります。

   domainScopeDocument ファイルを使用して DFM のセキュリティを強化する方法の詳細については，40ページ「データ・フロー資格情報管理」を参照してください。

2. Probe を再起動します。

データ・フロー・プローブのキー・ストアの作成

1. プローブ・マシンで，次のコマンドを実行します。

   ```
   C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool -genkey -alias
   probekey -keyalg RSA -keystore
   C:\HP\UCMDB\DataFlowProbe\conf\security\client.keystore
   ```

2. 新しいキー・ストアのパスワードを入力します。
3. 必要に応じて情報を入力します。
4. [**Is CN=... C=... Correct?**]と表示された場合は，[**はい**]と入力して **Enter** キーを押します。
5. もう一度 **Enter** キーを押して，そのキー・ストア・パスワードをキー・パスワードとして受け入れます。
6. **client.keystore** が次のディレクトリに作成されていることを確認します。**C:\HP\UCMDB\DataFlowProbe\conf\security**

プローブのキー・ストアとトラスト・ストアのパスワードを暗号化

プローブのキー・ストアとトラスト・ストアのパスワードは，暗号化されて **C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties** に保存されます。次の手順では，パスワードの暗号化方法について説明します。

1. Data Flow Probe を起動するか，すでに実行されていることを確認します。
2. Data Flow Probe JMX コンソールへのアクセス:Web ブラウザを起動し，アドレスに「http://<Data Flow Probe machine name or IP address>:1977」と入力します。Data Flow Probe をローカルで実行している場合は，http://localhost:1977 と入力します。

   > **注：** ユーザ名とパスワードを使用してログインする必要がある場合もあります。ユーザを作成していない場合は，標準設定のユーザ名 sysadmin とパスワード sysadmin を使用してログインします。

3. **Type=MainProbe** サービスを見つけ，リンクをクリックして[操作]ページを開きます。
4. **getEncryptedKeyPassword** 操作を見つけます。
5. [**キー パスワード**]フィールドにキー・ストアまたはトラスト・ストアのパスワードを入力し，[**getEncryptedKeyPassword**]をクリックして操作を呼び出します。
6. この呼び出しの結果は，次のような暗号化されたパスワード文字列となります。

   ```
   66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,114,
   112,65,61,61
   ```

7. 暗号化したパスワードをコピーし，**C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties** ファイルのキー・ストアまたはトラスト・ストアの関連する行に貼り付けます。

サーバとデータ・フロー・プローブのデフォルトのキー・ストアとトラスト・ストア

本項の内容

UCMDB サーバ

ファイルは次のディレクトリにあります。**C:\HP\UCMDB\UCMDBServer\conf\security**

エンティティ	ファイル名 / 用語	パスワード / 語句	エイリアス
サーバのキー・ストア	server.keystore (sKeyStoreFile)	hppass (sKeyStorePass)	hpcert
サーバのキー・ストア	server.truststore (sTrustStoreFile)	hppass (sTrustStorePass)	clientcert(標準設定の信頼されるエントリ)
クライアントのキー・ストア	client.keystore (cKeyStoreFile)	clientpass (cKeyStorePass)	clientcert

Data Flow Probe

ファイルは次のディレクトリにあります。**C:\HP\UCMDB\DataFlowProbe\conf\security**

エンティティ	ファイル名 / 用語	パスワード / 語句	エイリアス
プローブのキー・ストア	MAMKeyStoreExp.jks (pKeyStoreFile)	logomania (pKeyStorePass)	mam
相互認証手順で Data Flow Probe は，**cKeyStoreFile** キー・ストアを標準設定キー・ストアとして使用します。これは UCMDB インストールの一部であるクライアント・キー・ストアです。			
プローブのトラスト・ストア	MAMTrustStoreExp.jks (pTrustStoreFile)	logomania (pTrustStorePass)	mam(標準設定の信頼されるエントリ)
cKeyStorePass パスワードは **cKeyStoreFile** の標準設定のパスワードです。			

第6章

Lightweight シングル・サインオン認証 (LW-SSO) - 全般的な参照情報

本章の内容

LW-SSO 認証の概要

LW-SSO とは，一度ログオンしたユーザであれば，再びログオンせずに複数のソフトウェア・システムのリソースにアクセスできるようにするアクセス制御方法の 1 つです。設定されたソフトウェア・システムのグループに属するアプリケーションは，認証されていることを信用しているため，アプリケーションから別のアプリケーションに移動するときにさらに認証処理を行う必要がありません。

このセクションの情報は，LW-SSO バージョン 2.2 および 2.3 に適用されます。

- **LW-SSO トークンの期限**

  LW-SSO トークンの期限の数値により，アプリケーションのセッションの有効性が判断されます。そのため，期限の数値は，アプリケーションのセッション期限の数値と同じか，またはそれよりも大きな値にする必要があります。

- **LW-SSO トークンの期限の推奨設定**

  LW-SSO を使用するアプリケーションごとに，トークンの期限を設定する必要があります。推奨値は 60 分です。高度なセキュリティを必要としないアプリケーションの場合，値を 300 分に設定できます。

- **GMT 時間**

  LW-SSO に統合されているアプリケーションでは，すべて同じ GMT 時間を使用し，最大誤差を 15 分に抑える必要があります。

- **マルチドメイン機能**

  マルチドメイン機能では，LW-SSO 統合を行うアプリケーションを，異なる DNS ドメインのアプリケーションと統合する必要がある場合，それらすべてのアプリケーションで trustedHosts 設定 (または **protectedDomains** 設定) を行う必要があります。さらに，設定の **lwsso** 要素に正しいドメインを追加する必要があります。

- **URL 機能での SecurityToken の取得**

  ほかのアプリケーションから **URL に対する SecurityToken** として送信された情報を取得するには，ホスト・アプリケーション設定の **lwsso** 要素で，正しいドメインを設定する必要があります。

LW-SSO のシステム要件

アプリケーション	バージョン	コメント
Java	1.5 以降	
HTTP サーブレット API	2.1 以降	
Internet Explorer	6.0 以降	ブラウザで，HTTP セッション cookie とHTTP 302 リダイレクト機能を有効にする必要あり。
Firefox	2.0 以降	ブラウザで，HTTP セッション cookie とHTTP 302 リダイレクト機能を有効にする必要あり。
JBoss 認証	JBoss 4.0.3 JBoss 4.3.0	
Tomcat 認証	スタンドアロン Tomcat 5.0.28 スタンドアロン Tomcat 5.5.20	
Acegi 認証	Acegi 0.9.0 Acegi 1.0.4	
Web サービス・エンジン	Axis 1 - 1.4 Axis 2 - 1.2 JAX-WS-RI 2.1.1	

LW-SSO のセキュリティに関する警告

本項では，LW-SSO 設定に関するセキュリティの警告について説明します。

- **LW-SSO の InitString 機密パラメータ :** LW-SSO では，対称暗号化方式を使用して LW-SSO トークンを検証および作成します。設定内にある **initString** パラメータは，秘密鍵の初期化に使用します。アプリケーションでトークンが作成され，同じ initString パラメータを使用するアプリケーションにより，トークンが検証されます。

  注意 :

  - **initString** パラメータの設定を行わずに LW-SSO を使用することはできません。
  - **initString** パラメータは機密情報なので，公開，転送，永続性などの点で慎重に扱う必

要があります。

- **initString** パラメータは，LW-SSO を使用して相互に統合されたアプリケーション間でのみ共有する必要があります。.
- **initString** パラメータは，12 文字以上の長さである必要があります。

- **必要な場合のみ LW-SSO を有効化**：特に必要な場合を除き，LW-SSO を無効にする必要があります。
- **認証セキュリティのレベル**:最も弱いタイプの認証フレームワークを使用し，他の統合アプリケーションで信頼されている LW-SSO トークンを発行するアプリケーションにより，アプリケーション全体の認証セキュリティ・レベルが決まります。

  強力で安全な認証フレームワークを使用するアプリケーションの場合のみ，LW-SSO トークンを発行することをお勧めします。.
- **対称暗号化方式の影響**：LW-SSO では，対称暗号化方式を使用して LW-SSO トークンを発行および検証します。そのため，LW-SSO を使用するアプリケーションから，同じ initString パラメータを共有する，その他すべてのアプリケーションに信頼されたトークンを発行できます。これにより，initString を共有するアプリケーションが，信頼できない場所に置かれているか，またはそのような場所からアクセスできる場合，リスクが発生する場合があります。
- **ユーザ・マッピング(同期)**：LW-SSO フレームワークでは，統合アプリケーション間のユーザ・マッピングが保証されていません。そのため，統合アプリケーションでユーザ・マッピングを監視する必要があります。すべての統合アプリケーションで，同じユーザ・レジストリ (LDAP/AD など) を共有することをお勧めします。

  ユーザのマッピングに失敗すると，セキュリティ違反が発生し，アプリケーションの動作不良が起こる場合があります。たとえば，さまざまなアプリケーションで複数の実際のユーザに同じユーザ名が割り当てられることがあります。

  さらに，ユーザのマッピングに失敗した場合，ユーザがあるアプリケーション(AppA) にログオンしてから，コンテナ認証またはアプリケーション認証を使用する別のアプリケーション(AppB) にアクセスすると，そのユーザが手動で AppB にログオンしてユーザ名を入力することになります。ユーザが AppA へのログオンに使用していたのとは別のユーザ名を入力した場合，次のような予期せぬ動作が発生する場合があります。ユーザが AppA へのログオンに使用していたのとは別のユーザ名を入力した場合，次のような動作が発生する場合があります。その後ユーザが AppA または AppB から第 3 のアプリケーション (AppC) にアクセスすると，AppA または AppB へのログオンに使用していたユーザ名で，AppC にアクセスしてしまいます。
- **ID マネージャ**:認証のために使用します。ID マネージャ内の保護されていないリソースはすべて，LW-SSO 構成ファイル内で **nonsecureURLs** に設定する必要があります。
- **LW-SSO のデモ・モード**：
  - デモ・モードは，デモ目的にのみ使用してください。
  - デモ・モードは，保護されていないネットワークでのみ使用してください。
  - デモ・モードは，実運用で使用しないでください。実運用モードとデモ・モードは，どのような形でも併用しないでください。

トラブルシューティングおよび制限事項

本項では，LW-SSO 認証作業時の既知の問題および制限について説明します。

既知の問題

本項では，LW-SSO 認証の既知の問題について説明します。

- **セキュリティ・コンテキスト：**LW-SSO のセキュリティ・コンテキストでは，1 つの属性名につき 1 つの属性値のみがサポートされています。

  そのため，SAML2 トークンから，同じ属性名の値が複数送信されても，LW-SSO フレームワークで許可される値は 1 つのみです。

  同様に，同じ属性名に対して値を複数送信するように IdM トークンが設定されていても，LW-SSO フレームワークで許可される値は 1 つのみです。

- **Internet Explorer 7 を使用したマルチドメインのログアウト機能：**マルチドメインのログアウト機能は，次の条件下で失敗することがあります。

  - ブラウザに Internet Explorer 7 を使用していて，アプリケーションのログアウト手順で HTTP 302 リダイレクトの動作が 3 回を超え連続して呼び出されたとき

  この場合，Internet Explorer 7 で HTTP 302 リダイレクトの応答が正しく処理されず，[**Internet Explorer ではこのページは表示できません**] というエラー・ページが表示される場合があります。

  回避策としては，アプリケーションのログアウト手順で，できるだけリダイレクト・コマンドの数を少なくすることを推奨します。

制限事項

LW-SSO 認証を行う際，次の制限に注意してください。

- **アプリケーションへのクライアント・アクセス：**

  ドメインが LW-SSO 設定で定義されている場合：

  - アプリケーションのクライアントは，ログイン URL に FQDN(完全修飾ドメイン名) を使用してアプリケーションにアクセスする必要があります(http://myserver. **企業ドメイン名** .com/WebApp など) 。
  - LW-SSO では，IP アドレスを使用した URL はサポートされていません (http://192.168.12.13/WebApp など) 。
  - LW-SSO では，ドメインのない URL はサポートされていません(http://myserver/WebApp など) 。

  ドメインが LW-SSO 設定で定義されていない場合：クライアントは，ログイン URL で FQDN を持たないアプリケーションにアクセスできます。この場合，LW-SSO のセッション cookie は，一切のドメイン情報を持たない単一のマシン専用に作成されます。そのため，この cookie がほかのブラウザに委譲されたり，同じ DNS ドメインにある別のコンピュータに渡されることはありません。つまり，LW-SSO は同じドメインで機能しないことを意味します。

- **LW-SSO フレームワークの統合：**アプリケーションで LW-SSO の機能を活用できるのは，あらかじめ LW-SSO フレームワーク内に統合されている場合のみです。

- **マルチドメインのサポート：**

  - マルチドメイン機能は，HTTP リファラに基づいています。そのため LW-SSO では，アプリケーション間のリンクはサポートされていますが，2 つのアプリケーションが同じドメインにある場合を除き，ブラウザ・ウィンドウへの URL の入力はサポートされていません。
  - **HTTP POST** を使用したドメイン間のリンクはサポートされていません。

最初のクロス・ドメイン・リンクに **HTTP POST** を使用することはサポートされていません。(**HTTP GET** 要求のみサポートされています)。たとえば，最初のアプリケーションから2番目のアプリケーションへのHTTPリンクがある場合，**HTTP GET** 要求はサポートされていますが，**HTTP FORM** 要求はサポートされていません。2回目以降の要求は，すべて **HTTP POST** か **HTTP GET** のいずれかです。

- LW-SSOトークンのサイズ:

  LW-SSOが，あるドメインのアプリケーションから別のドメインのアプリケーションに転送できる情報量は，15グループ/ロール/属性までに制限されています (各項目は平均 15 文字長)。

- マルチドメイン・シナリオでの，保護されたページ(HTTPS) から保護されていないページ(HTTP) へのリンク:

  保護されたページ(HTTPS) から保護されていないページ(HTTP) にリンクする場合，マルチドメインは機能しません。これはブラウザの制限事項の1つです。この場合，保護されたリソースから保護されていないリソースにリンクするときに，リファラ・ヘッダが送信されません。具体例については，`http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP` を参照してください。

- サードパーティcookieのInternet Explorerでの動作：

  Microsoft Internet Explorer 6には，「P3P(Platform for Privacy Preferences) プロジェクト」をサポートするモジュールが含まれています。そのため，サードパーティ・ドメインのcookieは，[インターネット]セキュリティ・ゾーンの標準設定でブロックされています。IEでは，セッションのcookieもサードパーティcookieとみなされるため，セッションのcookieもブロックされてしまい，LW-SSOが機能しません。詳細については，`http://support.microsoft.com/kb/323752/en-us` を参照してください。

  この問題を解決するには，起動したアプリケーション(または *.mydomain.com などのDNSドメイン・サブセット)を，コンピュータの[イントラネット]または[信頼済みサイト]ゾーンに追加します(Microsoft Internet Explorerで，**メニュー>[ツール]>[インターネット オプション]>[セキュリティ]>[ローカル イントラネット]>[サイト]>[詳細]**を選択します)。こうすることで，cookieが許可されます。

  注意： LW-SSOのセッションcookieは，ブロックされているサードパーティアプリケーションで使用するcookieの1つにすぎません

- **SAML2トークン**
  - SAML2トークンを使用する場合，ログアウト機能がサポートされません。

    そのため，SAML2トークンを使用して2番目のアプリケーションにアクセスすると，最初のアプリケーションからログアウトするユーザが，2番目のアプリケーションからログアウトされません。

  - **TSAML2トークンの期限切れは，アプリケーションのセッション管理に反映されません。**

    そのため，SAML2トークンを使用して2番目のアプリケーションにアクセスする場合，各アプリケーションのセッション管理が個別に処理されます。

- **JAAS Realm**: TomcatのJAAS Realmはサポートされていません。
- **Tomcat ディレクトリでのスペースの使用：**TomcatのJAAS Realmはサポートされていません。

Tomcat のインストール・パス(フォルダ) にスペースが含まれており(「Program Files」など), かつ LW-SSO 構成ファイルが Tomcat の **common\classes** フォルダに置かれている場合は, LW-SSO を使用できません。

- **ロード・バランサの設定**: LW-SSO によりデプロイされたロード・バランサは, セッション維持を使用するよう設定する必要があります。
- **デモ・モード**: デモ・モードでは, LW-SSO ではアプリケーション間のリンクはサポートされますが, この場合は HTTP リファラ・ヘッダが存在しないため, ブラウザ・ウィンドウへの URL 入力はサポートされません。

第7章

HP Universal CMDB ログイン認証

本章の内容

認証メソッドの設定

認証は次の方法で行うことができます。

- **内部 HP Universal CMDB サービスに対して。**
- **Lightweight Directory Access Protocol (LDAP) を使用 詳細については，23 ページ「LDAP 認証メソッドの有効化と定義」を参照してください。** 内部 HP Universal CMDB サービスを使用する代わりに，専用の外部 LDAP サーバを使用して認証情報を格納できます。LDAP サーバは，すべてのHP Universal CMDB サーバと同じサブネット上になければなりません。

  LDAP の詳細については，『HP Universal CMDB 管理ガイド』のLDAP マッピングに関する項を参照してください。

  標準設定の認証メソッドは内部 HP Universal CMDB サービスを使用します。標準設定のメソッドを使用する場合には，システムに変更を加える必要はありません。

  これらのオプションは，Web サービスのほかユーザ・インタフェースを使用したログインにも適用されます。

- **LW-SSO を使用 ：** HP Universal CMDB にはLW-SSO を設定します。HP Universal CMDB にログインをすることで，同じドメインで実行されているほかの設定済みのアプリケーションにログインしなくても自動的にアクセスできるようになっています。

  LW-SSO 認証サポートを有効にする場合(標準設定は無効)，シングル・サインオン環境のほかのアプリケーションもLW-SSO が有効になっており，同じ initString パラメータで機能することを確認してください。

HP Universal CMDB への LW-SSO によるログインを有効化

HP Universal CMDB で LW-SSO を有効にするには，次のいずれかの手順を使用します。

1. Web ブラウザのアドレスに**http://<サーバ名 :>:8080/jmx-console<サーバ名** > と入力します。< サーバ名 >には HP Universal CMDB がインストールされているマシンの名前が入ります。
2. **UCMDB-UI** の下で，**name=LW-SSO Configuration** をクリックして［操作］ページを開きます。
3. **setInitString** メソッドを使用して init 文字列を設定します。
4. **setDomain** メソッドを使用して，UCMDB をインストールするマシンのドメイン名を設定します。
5. パラメータを「**True**」に設定して **setEnabledForUI** メソッドを呼び出します。
6. **任意指定** :複数ドメイン機能を使用して作業する場合，**addTrustedDomains** メソッドを選択してドメイン値を入力し，［**Invoke**］をクリックします。
7. **任意指定。**リバース・プロキシを使用して作業する場合は，**updateReverseProxy** メソッドを選択し，［**Is reverse proxy enabled**］パラメータを［**True**］に設定して［**Reverse proxy full server URL**］パラメータの URL を入力して［Invoke］をクリックします。UCMDB に直接およびリバース・プロキシを使用してアクセスする場合は，次の追加構成を設定してください。**setReverseProxyIPs** メソッドを選択し，リバース・プロキシ ip/s パラメータの IP アドレスを入力して［**Invoke**］をクリックします。
8. **任意指定** :外部認証ポイントを使用して UCMDB にアクセスする場合は，**setValidationPointHandlerEnable** メソッドを選択して［**Is validation point handler enabled**］パラメータを［**True**］に設定し，［**Authentication point server**］パラメータの認証ポイントの URL を入力して［**Invoke**］をクリックします。
9. 設定メカニズムに保存されているとおりに LW-SSO 設定を表示するには，**retrieveConfigurationFromSettings** メソッドを呼び出します。
10. 実際ロードされた LW-SSO 設定を表示するには，**retrieveConfiguration** メソッドを呼び出します。

注： ユーザ・インターフェースから LW-SSO を有効化することはできません。

SSL(Secure Sockets Layer) プロトコルによるセキュア接続の設定

ログイン処理では，HP Universal CMDB とLDAP サーバの間で機密情報がやり取りされるため，その内容に対して一定のレベルのセキュリティを適用するとよいでしょう。それには，LDAP サーバ上で SSL 通信を有効にして，SSL を使用できるように HP Universal CMDB を設定します。

HP Universal CMDB では，信頼できる認証局 (CA) から発行された証明書を使用する SSL をサポートしています。

Active Directory を含む大半の LDAP サーバは，SSL ベースの接続を対象とするセキュリティ保護されたポートを公開できます。プライベート CA を利用する Active Directory を使用している場合，当該 CA を JRE の信頼できる CA に追加する必要があります。

SSL 通信をサポートするように HP Universal CMDB プラットフォームを設定する方法の詳細については，17ページ「Secure Sockets Layer(SSL) 通信の有効化」を参照してください。

SSL ベースの接続を対象とするセキュリティ保護されたポートを公開するために，信頼できる CA に CA を追加するには，次の手順を実行します。

1. CA から証明書をエクスポートし，次の手順に従って, HP Universal CMDB で使用される JVM にインポートしてください。
   a. UCMDB サーバ・マシンで，**UCMDBServer\bin\JRE\bin** フォルダにアクセスします。
   b. 次のコマンドを実行します。

   ```
   Keytool -import -file ＜自分の証明書ファイル＞ -keystore
   C:\hp\UCMDB\UCMDBServer\bin\JRE\lib\security\cacerts
   ```

   例：

   ```
   Keytool -import -file c:\ca2ss_ie.cer -keystore
   C:\hp\UCMDB\UCMDBServer\bin\JRE\lib\security\cacerts
   ```

2. [**管理**]>[**インフラストラクチャ設定**]>[**LDAP 全般**]カテゴリを選択します。

   注： JMX コンソールを使用してこれらを設定することも可能です。詳細については，83ページ「JMX コンソールを使用した LDAP 設定の構成」を参照してください。

3. [**LDAP サーバ URL**]を見つけ，次の形式で値を入力します。

   ```
   ldaps://<ldapHost>[:<port>]/[<baseDN>][??scope]
   ```

   例：

   ```
   ldaps://my.ldap.server:389/ou=People,o=myOrg.com??sub
   ```

   ldaps の **s** に注意してください。

4. 新しい値を保存するには[**保存**]を，エントリを標準設定値(空白の URL) で置き換えるには[**標準設定に戻す**]をクリックします。

JMX コンソールを使用した LDAP 接続のテスト

本項では，JMX コンソールを使用して LDAP 認証設定をテストする方法について説明します。

1. Web ブラウザを起動して**http://<サーバ名 :>:8080/jmx-console<<サーバ名 >>** と入力します。< サーバ名 >には HP Universal CMDB がインストールされているマシンの名前が入ります。

   ユーザ名およびパスワードでのログインが必要な場合もあります。

2. [**UCMDB**]から[**UCMDB-UI:name=LDAP Settings**]をクリックして，[操作]ページを開きます。
3. **testLDAPConnection** を見つけます。
4. [**顧客 ID**]パラメータの[**値**]ボックスに，顧客 ID を入力します。
5. [**Invoke**] をクリックします。

LDAP 接続が成功したかどうかが JMX MBEAN 操作結果ページに示されます。接続が成功した場合は，LDAP ルート・グループもこのページに表示されます。

JMX コンソールを使用した LDAP 設定の構成

本項では，JMX コンソールを使用して LDAP 認証を設定する方法について説明します。

LDAP 認証設定を行うには，次の手順を実行します。

1. Web ブラウザを起動して**http://<サーバ名 :>:8080/jmx-console<<サーバ名 >>** と入力します。< サーバ名 >には HP Universal CMDB がインストールされているマシンの名前が入ります。

   ユーザ名およびパスワードでのログインが必要な場合もあります。

2. [**UCMDB**]から[**UCMDB-UI:name=LDAP Settings**]をクリックして，[操作]ページを開きます。

3. 現在の LDAP 認証設定を表示するには，**getLDAPSettings メソッド**を見つけます。[**Invoke**]をクリックします。表に，すべての LDAP 設定とその値が表示されます。

4. LDAP 認証設定の値を変更するには，**configureLDAP メソッド**を見つけます。関連する設定の値を入力して，[**Invoke**]をクリックします。LDAP 認証設定の更新が成功したかどうかが[JMX MBEAN Operation Result]ページに示されます。

   注： 設定に値を入力しない場合は，現在の値が保持されます。

5. LDAP 設定を行った後，LDAP ユーザの資格情報を確認できます。**verifyLDAPCredentials** メソッドを見つけます。顧客 ID，ユーザ名，およびパスワードを入力して，[Invoke]をクリックします。LDAP 認証が渡されたかどうかが[JMX MBEAN Operation Result]ページに示されます。

LDAP 認証メソッドの有効化と定義

HP Universal CMDB システムを対象に LDAP 認証メソッドの有効化と定義を行うことができます。

LDAP 認証メソッドを有効化して定義するには，次の手順を実行します。

1. [**管理**]>[**インフラストラクチャ設定**]>[**LDAP 全般**]カテゴリを選択します。

2. [**LDAP サーバ URL**]を選択して，次の形式で LDAP URL の値を入力します。

   ```
   ldap://<ldapHost>[:<port>]/[<baseDN>][??scope]
   ```

   例：

   ```
   ldap://my.ldap.server:389/ou=People,o=myOrg.com??sub
   ```

3. [**LDAP グループ定義**]カテゴリを選択し，[**グループのベース DN**]で，一般グループの識別名を入力します。

4. [**ルート グループのベース DN**]に，ルート・グループの識別名を入力します。

5. [**LDAP 全般**]カテゴリを選択し，[**ユーザ権限の同期化を有効化**]で，値が[**True**]に設定されていることを確認します。

6. [**LDAP 一般認証**]カテゴリを選択して，[**検索権限のあるユーザのパスワード**]を見つけ，パスワードを入力します。
7. [**クラスと属性のための LDAP オプション**]カテゴリを選択し，[**グループ クラス オブジェクト**]を見つけ，オブジェクト・クラス名を入力します(Microsoft Active Directory の場合は **group**，Oracle Directory Server の場合は **groupOfUniqueNames**)。
8. [**グループのメンバ属性**]を見つけて，属性名を入力します(Microsoft Active Directory の場合は **member**，Oracle Directory Server の場合は **uniqueMember**)。
9. [**ユーザのオブジェクト クラス**]を見つけて，オブジェクト・クラス名を入力します(Microsoft Active Directory の場合は **user**，Oracle Directory Server の場合は **inetOrgPerson**)。
10. [**UUID 属性**]を見つけ，使用しているディレクトリ・サーバのユーザの一意の識別属性を入力します。必ず使用しているディレクトリ・サーバ内で一意の属性を選択してください。例えば，SunOne/Oracle ディレクトリ・サーバを使用する場合は UID 属性は一意ではありません。このような場合，電子メール・アドレス属性と識別名のいずれかを使用してください。一意ではない属性を UCMDB の一意の識別属性として使用すると，ログイン時に不整合な動作が起こる場合があります。
11. 新しい値を保存します。エントリを標準設定値で置き換えるには，[**標準設定に戻す**]をクリックします。
12. [**LDAP 全般**]のインフラストラクチャ設定[**Is case-sensitivity enforced when authenticating with LDAP**]が[**True**]に設定されている場合，認証は大文字と小文字を区別します。

    注意： このインフラストラクチャ設定の値が変更されると，すべての外部ユーザを UCMDB 管理者によって手動で削除する必要があります。
13. LDAP ユーザ・グループを UCMDB ユーザ・グループにマッピングします。詳細については，80ページ「HP Universal CMDB ログイン認証」を参照してください。
14. グループ・マッピングのない LDAP グループのユーザに一連の標準権限を定義する場合，[**LDAP 全般**]カテゴリを選択し，[**自動的に割り当てられたユーザ グループ**]を見つけてグループ名を入力します。

LDAP サーバとの通信に使用される標準設定のプロトコルは TCP ですが，これを SSL に変更できます。詳細については，81ページ「SSL(Secure Sockets Layer)プロトコルによるセキュア接続の設定」を参照してください。

注： 全ての LDAP ユーザの氏名および電子メールはローカル・リポジトリに保存されます。LDAP サーバに保存されている場合これらパラメータ値のいずれかがローカル・リポジトリのものと異なる場合，ログイン時に LDAP サーバ値でローカル値が上書きされます。

分散化環境における現在の LW-SSO 設定の取得

UCMDB が分散環境に組み込まれている場合(BSM デプロイメントの場合など)，次の手順を実行して処理マシン上の現在の LW-SSO 設定を取得します。

現在の LW-SSO 設定を取得するには，次の手順を実行します。

1. Web ブラウザを起動し，アドレスとして`http://localhost.<domain_name>:8080/jmx-console`。

   ユーザ名とパスワードの入力を求められる場合もあります。

2. **UCMDB:service=Security Services** を見つけ，リンクをクリックして［操作］ページを開きます。
3. **retrieveLWSSOConfiguration** 操作を見つけます。
4. ［**Invoke**］をクリックして設定を取得します。

第8章

機密マネージャ

本章の内容

機密マネージャの概要

機密マネージャ・フレームワークは，HP Universal CMDB やほかの HP ソフトウェア製品の機密データの管理および配布に関する問題を解決します。

機密マネージャは，クライアントとサーバの 2 つのメイン・コンポーネントで構成されています。これらの 2 つのコンポーネントは，データを安全に転送する役割を担います。

- 機密マネージャ・クライアントは，機密データにアクセスするためにアプリケーションによって使用されるライブラリです。
- 機密マネージャ・サーバは，機密マネージャ・クライアントまたはサードパーティのクライアントから要求を受信し，必要なタスクを実行します。機密マネージャ・サーバは，データを安全に保存する役割を担います。

機密マネージャは，転送，クライアント・キャッシュ，永続的な保存場所，メモリの資格情報を暗号化します。機密マネージャは，対称暗号化方式を使用して機密マネージャ・クライアントと機密マネージャ・サーバ間で資格情報を転送します。この転送には共有秘密鍵が使用されます。機密マネージャは，設定に応じてキャッシュ，永続的な保存場所，転送の暗号化にさまざまな秘密鍵を使用します。

Data Flow Probe で資格情報の暗号化を管理するためのガイドラインの詳細については，40ページ「データ・フロー資格情報管理」を参照してください。

セキュリティの考慮事項

- セキュリティ・アルゴリズムに使用できる鍵のサイズは，128，192，256 ビットです。鍵が小さいほどアルゴリズムの実行速度は速くなりますが，セキュリティは弱くなります。通常，128 ビットのサイズで十分なセキュリティを確保できます。
- システムをより安全にするには，MAC を使用します(**useMacWithCrypto** を **true** に設定)。詳細については，88ページ「暗号化プロパティ」を参照してください。
- 強力な顧客セキュリティ・プロバイダを活用するには，JCE モードを使用できます。

HP Universal CMDB Server の設定

HP Universal CMDB で作業する場合，次の JMX メソッドを使用して暗号化の秘密鍵と暗号化プロパティを設定する必要があります。

1. HP Universal CMDB サーバ・マシンで Web ブラウザを起動し，次のサーバ・アドレスを入力します。**http://<UCMDB サーバのホスト名または IP>:8080/jmx-console** のように入力します。

   ユーザ名とパスワードを使用してログインする必要がある場合もあります。

2. UCMDB で，**UCMDB:service=Security Services** をクリックして［操作］ページを開きます。

3. 現在の設定を取得するには，**CMGetConfiguration** 操作を見つけます。

   ［**Invoke**］をクリックして，機密マネージャ・サーバ設定の XML ファイルを表示します。

4. 設定を変更するには，前の手順で起動した XML をテキスト・エディタにコピーします。88ページ「暗号化プロパティ」の表に従って変更します。

   CMSetConfiguration 操作を見つけます。更新された設定を［**値**］ボックスにコピーし，［**Invoke**］をクリックします。新しい設定が UCMDB サーバに書き込まれます。

5. 認証およびレプリケーションを行うためにユーザを機密マネージャに追加するには，**CMAddUser** 操作を見つけます。このプロセスは，レプリケーション・プロセスでも有効です。レプリケーションでは，スレーブ・サーバは権限のあるユーザを使用してマスタ・サーバと通信する必要があります。

   - **ユーザ名**：ユーザ名です。
   - **顧客**：標準設定は ALL_CUSTOMERS です。
   - **リソース**：リソース名です。標準設定は ROOT_FOLDER です。
   - **権限**：ALL_PERMISSIONS，CREATE，READ，UPDATE，DELETE のいずれかを選択します。標準設定は ALL_PERMISSIONS です。

   ［**Invoke**］をクリックします。

6. 必要に応じて，HP Universal CMDB を再起動します。

   注：通常，サーバを再起動する必要はありません。次のいずれかのリソースを変更した場合にサーバの再起動が必要になることがあります。

   - ストレージ・タイプ
   - データベースのテーブル名またはカラム名
   - データベース接続の作成者
   - データベースの接続プロパティ(URL，ユーザ，パスワード，ドライバ・クラス名)
   - データベース・タイプ

注：

- UCMDB サーバとそのクライアントで転送用暗号化プロパティを同じにすることが重要です。これらのプロパティが UCMDB サーバで変更されている場合，すべてのクライアントで変更する必要があります。(Data Flow Probe で実行されるプロセスは UCMDB サーバと同じなので転

送用の暗号化設定が必要ないため，これは Data Flow Probe には関係ありません)。

- 機密マネージャ・レプリケーションは標準設定では設定されていませんが，必要に応じて設定できます。
- 機密マネージャ・レプリケーションが有効になっている場合，マスタの転送用の **initString** またはほかの暗号化プロパティが変更されると，すべてのスレーブでその変更が採用されます。

定義

ストレージ用暗号化プロパティ: サーバでデータを保持および暗号化する方法(データベースまたはファイルや，データを暗号化/復号化する暗号化プロパティなど)，資格情報を安全に保存する方法，暗号化の処理方法，準拠する設定を定義する設定です。

転送用暗号化プロパティ: 転送用の設定では，サーバとクライアント間の転送を暗号化する方法，使用する設定，安全に資格情報を転送する方法，暗号化の処理方法，準拠する設定を定義します。サーバとクライアントの両方で転送の暗号化と復号化に同じ暗号化プロパティを使用する必要があります。

レプリケーションおよびレプリケーション用暗号化プロパティ: 機密マネージャで安全に保持されているデータが複数のサーバ間で安全に複製されます。これらのプロパティでは，スレーブ・サーバとマスタ・サーバ間でどのようにデータを転送するのかを定義します。

注:

- 機密マネージャ・サーバの設定を保持しているデータベース・テーブルの名前は**CM_CONFIGURATION です。**
- 構成マネージャ・サーバの標準設定の構成ファイルは，**defaultCMServerConfig.xml** という名前で app-infra.jar にあります。

暗号化プロパティ

次の表に，暗号化プロパティを示します。これらのパラメータの使用の詳細については，87ページ「HP Universal CMDB Server の設定」を参照してください。

パラメータ	詳細	推奨値
encryptTransportMode	転送データの暗号化: true false	true
encryptDecrypt InitString	暗号化のパスワード	9文字以上
cryptoSource	使用する暗号化実装ライブラリ • lw • jce • windowsDPAPI	lw

パラメータ	詳細	推奨値
	• lwJCECompatible	
lwJCEPBE CompatibilityMode	軽量暗号化方式の以前のバージョンのサポート : • true • false	true
cipherType	機密マネージャで使用する暗号化のタイプ。機密マネージャでサポートされる値は 1 つだけです。 **symmetricBlockCipher**	symmetric BlockCipher
engineName	• AES • Blowfish • DES • 3DES • Null(暗号化しない)	AES
algorithmModeName	ブロック暗号化アルゴリズムのモード : • CBC	CBC
algorithmPaddingName	パディングの標準 : • PKCS7Padding • PKCS5Padding	PKCS7Padding
keySize	アルゴリズムによって異なる(どの **engineName** をサポートするのかによる)	256
pbeCount	**encryptDecryptInitString** から鍵を作成するためにハッシュを実行する回数。 任意の正数。	1000
pbeDigestAlgorithm	ハッシュ・タイプ : • SHA1 • SHA256 • MD5	SHA256
encodingMode	暗号化するオブジェクトの ASCII 表現 : • Base64 • Base64Url	Base64Url
useMacWithCrypto	暗号化方式で MAC が使用されるかどうかの定義 : • true	false

パラメータ	詳 細	推 奨 値
	• false	
macType	メッセージ認 証 コード (MAC) のタイプ : • hmac	hmac
macKeySize SHA256	Mac アルゴリズムによって異 なる	256
macHashName	ハッシュ用 の Mac アルゴリズム : • SHA256	SHA256


Java™
COMPATIBLE