Oracle® Enterprise Performance Management System

Security Configuration Guide

リリース 11.1.2.3

ORACLE

ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

EPM System Security Configuration Guide, 11.1.2.3

著者: EPM 情報開発チーム

目次

ドキュメントのアクセシビリティについて

Oracle のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc を参照してください。

Access to Oracle Support

Oracle サポート・サービスでは、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info か、聴覚に障害のあるお客様は http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs を参照してください。

1 EPM Systemセキュリティについて

この章の内容

EPM System について

Oracle Enterprise Performance Management System 製品は、財務管理アプリケーションおよびプランニング・アプリケーションのモジュール式スイートと、レポートおよび分析の最も総合的なビジネス・インテリジェンス機能を統合する、総合的なエンタープライズ・システムを形成できます。EPM System 製品の主なコンポーネントは次のとおりです:

- Oracle Hyperion Foundation Services
- Oracle Essbase
- Oracle Hyperion Financial Management
- Oracle Hyperion Planning
- Oracle Hyperion Reporting and Analysis

各製品ファミリの製品とコンポーネントについては、Oracle Enterprise Performance Management System Installation Start Here を参照してください。

必要な知識

このガイドは、EPM System コンポーネントを構成、保護および管理するシステム管理者を対象にしています。前提条件となる知識は次のとおりです:

- 次のような、所属組織のセキュリティ・インフラストラクチャに関する十分な理解:
  - Oracle Internet Directory、Sun Java System Directory Server および Microsoft Active Directory などのディレクトリ・サーバー
  - 通信チャネルを保護する Secure Socket Layer (SSL)の使用

    - Oracle Access Manager および SiteMinder などのアクセス管理システム
    - Kerberos などシングル・サインオン(SSO)・インフラストラクチャ。
- 所属組織に関連する EPM System セキュリティの概念に関する知識。

セキュリティ・インフラストラクチャのコンポーネント

EPM System では、多くのセキュリティ・コンポーネントが統合されており、堅牢なアプリケーション・セキュリティを保証しています。EPM System は、セキュアなインフラストラクチャと統合することにより、データおよびアクセスのセキュリティを保証する高度なセキュア・アプリケーション・スイートを提供します。EPM System の保護に使用できるインフラストラクチャ・コンポーネントは次のとおりです:

- オプションのアクセス管理システム(EPM System コンポーネントに SSO アクセスを提供する Oracle Access Manager など)
- 統合 SSO インフラストラクチャ(Kerberos など)の使用。

  Kerberos 認証をアクセス管理システム(SiteMinder)とともに使用すると、Windows ユーザーは、SiteMinder および EPM System コンポーネントに透過的にログインできます。
- EPM System コンポーネントおよびクライアント間の通信チャネルを保護する Secure Socket Layer (SSL)の使用

ユーザー認証

ユーザー認証により、各ユーザーのログイン情報を検証することで EPM System コンポーネント全体でシングル・サインオン(SSO)機能が使用可能になり、認証済ユーザーが判別されます。コンポーネント固有の認可とともにユーザー認証は、EPM System コンポーネントへユーザー・アクセスを認めます。権限を付与するプロセスは、プロビジョニングと呼ばれます。

認証コンポーネント

次の項では、SSO をサポートするコンポーネントについて説明します。

- 12 ページの「ネイティブ・ディレクトリ」
- 13 ページの「外部ユーザー・ディレクトリ」

ネイティブ・ディレクトリ

ネイティブ・ディレクトリとは、Oracle Hyperion Shared Services がプロビジョニングのサポート、およびデフォルト・ユーザー・アカウントなどのシード・データの保管に使用するリレーショナル・データベースを指します。

ネイティブ・ディレクトリ機能:

- デフォルトの EPM System ユーザー・アカウントの維持と管理
- 全 EPM System プロビジョニング情報(ユーザー、グループおよび役割間の関係)の保管

ネイティブ・ディレクトリは、Oracle Hyperion Shared Services Console を使用してアクセスおよび管理します。Oracle Enterprise Performance Management System User Security Administration Guide のネイティブ・ディレクトリの管理に関する項を参照してください。

外部ユーザー・ディレクトリ

ユーザー・ディレクトリとは、EPM System コンポーネントと互換性のある、企業ユーザーおよび ID の管理システムを指します。

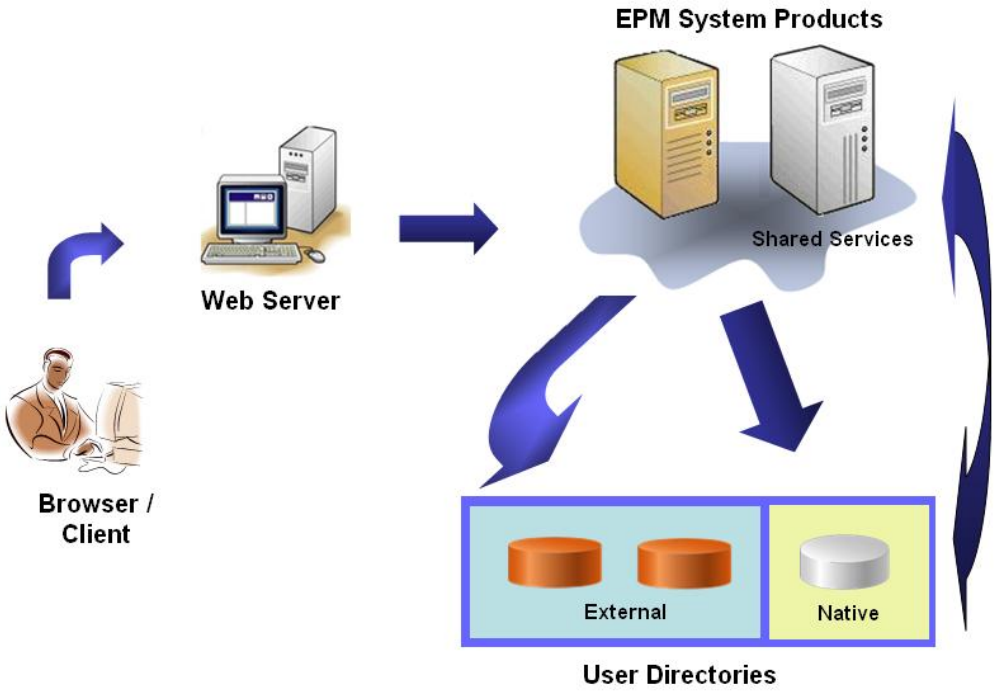
EPM System コンポーネントは、Oracle Internet Directory、Sun Java System Directory Server (旧 SunONE Directory Server)、Microsoft Active Directory などの LDAP ベースのユーザー・ディレクトリを含むいくつかのユーザー・ディレクトリでサポートされています。リレーショナル・データベースもユーザー・ディレクトリとしてサポートされています。このドキュメントでは、ネイティブ・ディレクトリ以外のユーザー・ディレクトリを外部ユーザー・ディレクトリと呼びます。サポートされているユーザー・ディレクトリのリストについては、Oracle Enterprise Performance Management 製品 - サポートされるプラットフォームのメトリックを参照してください。

Shared Services Console から、多くの外部ユーザー・ディレクトリを EPM System ユーザーおよびグループのソースとして構成できます。各 EPM System ユーザーは、構成済ユーザー・ディレクトリのいずれか 1 つで一意のアカウントを持っている必要があります。通常、EPM System ユーザーは、プロビジョニングを促進するためにグループに割り当てられます。

デフォルトの EPM System シングル・サインオン

EPM System では、あるアプリケーションの認証済ユーザーは、ログイン情報を再入力することなく別のアプリケーションにシームレスに移動でき、EPM System Web アプリケーション全体で SSO がサポートされます。SSO は、ユーザー認証およびプロビジョニング(役割ベースの認証)を処理する共通のセキュリティ環境を EPM System コンポーネント全体で統合することによって実装されます。

デフォルトの SSO プロセスを次の図に示します。



1. ユーザーは、ブラウザ経由で EPM System コンポーネントのログイン画面にアクセスし、ユーザー名とパスワードを入力します。

   EPM System コンポーネントにより、構成済ユーザー・ディレクトリ(ネイティブ・ディレクトリなど)への問合せが行われ、ユーザー・ログイン情報が確認されます。ユーザー・ディレクトリで一致するユーザー・アカウントが見つかると、検索は終了し、ユーザー情報が EPM System コンポーネントに戻されます。

   ユーザー・アカウントがどの構成済ユーザー・ディレクトリにもない場合、アクセスは拒否されます。
2. 取得したユーザー情報を使用して、EPM System コンポーネントにより、ネイティブ・ディレクトリへの問合せが行われ、ユーザーのプロビジョニングの詳細が入手されます。
3. EPM System コンポーネントにより、コンポーネントのアクセス制御リスト(ACL)がチェックされ、ユーザーがアクセスできるアプリケーション・アーティファクトが決定されます。

ネイティブ・ディレクトリからプロビジョニング情報を受け取ると、EPM System コンポーネントはユーザーに対して使用可能になります。この時点で、SSO は、ユーザーがプロビジョニングされているすべての EPM System コンポーネントで使用可能です。

アクセス管理システムからのシングル・サインオン

EPM System コンポーネントのセキュリティをさらに強化するには、Oracle Access Manager または SiteMinder など、サポートされているアクセス管理システムを実装できます。これらの製品では、認証済ユーザー・ログイン情報を EPM System コ

ンポーネントに提供し、事前定義済のアクセス権限に基づいてアクセスを制御できます。

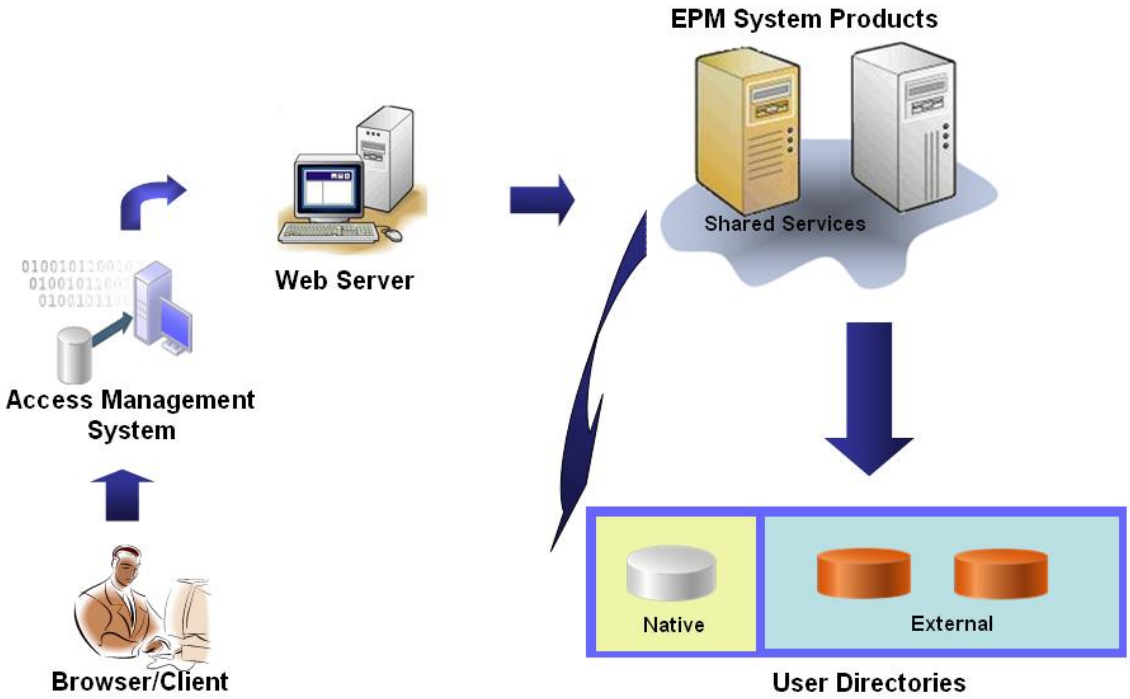
セキュリティ・エージェントからの SSO は EPM System Web アプリケーションでのみ使用可能です。このシナリオでは、EPM System コンポーネントは、セキュリティ・エージェントから提供されるユーザー情報を使用して、ユーザーのアクセス権を判別します。セキュリティを強化するには、すべての要求が SSO ポータルを経由するように、サーバーへの直接のアクセスをファイアウォールでブロックすることをお薦めします。

アクセス管理システムからの SSO は、条件を満たした SSO メカニズム経由で認証済のユーザー・ログイン情報を受け入れることによりサポートされます。49 ページの「サポートされている SSO メソッド」を参照してください。アクセス管理システムにより、ユーザーが認証され、ユーザーのログイン名が EPM System に渡されます。EPM System により、構成済のユーザー・ディレクトリに対してログイン名が確認されます。

次のトピックを参照してください。

- 52 ページの「Oracle Access Manager からのシングル・サインオン」
- 53 ページの「OracleAS シングル・サインオン」
- 65 ページの「SiteMinder SSO」
- 69 ページの「Kerberos シングル・サインオン」

概念を図で示します:



1. ブラウザを使用して、ユーザーはアクセス管理システム(Oracle Access Manager、SiteMinder など)で保護されているリソースへのアクセスを要求します。

注： EPM System コンポーネントは、アクセス管理システムで保護されているリソースとして定義されます。

アクセス管理システムは、要求をインターセプトし、ログイン画面を表示します。ユーザーはユーザー名とパスワードを入力します。これらは、ユーザーの信頼性を確認するためにアクセス管理システムで構成済ユーザー・ディレクトリに対して検証されます。EPM System コンポーネントは、これらのユーザー・ディレクトリと連動するようにも構成されています。

認証済ユーザーに関する情報は、EPM System コンポーネントに渡され、そのコンポーネントで有効なものとして受け入れられます。

アクセス管理システムは、条件を満たした SSO メカニズムを使用して、ユーザーのログイン名(ログイン属性の値)を EPM System コンポーネントに渡します。49 ページの「サポートされている SSO メソッド」を参照してください。

2. ユーザー・ログイン情報を確認するために、EPM System コンポーネントにより、ユーザー・ディレクトリでユーザーの検索が試みられます。一致するユーザー・アカウントが見つかると、ユーザー情報が EPM System コンポーネントに戻されます。EPM System セキュリティにより、EPM System コンポーネント全体で SSO を使用可能にする SSO トークンが設定されます。
3. 取得したユーザー情報を使用して、EPM System コンポーネントにより、ネイティブ・ディレクトリへの問合せが行われ、ユーザーのプロビジョニングの詳細が入手されます。

   ユーザー・プロビジョニング情報を受け取ると、EPM System コンポーネントはユーザーに対して使用可能になります。SSO は、ユーザーがプロビジョニングされているすべての EPM System コンポーネントで使用可能です。

プロビジョニング(役割ベースの認証)

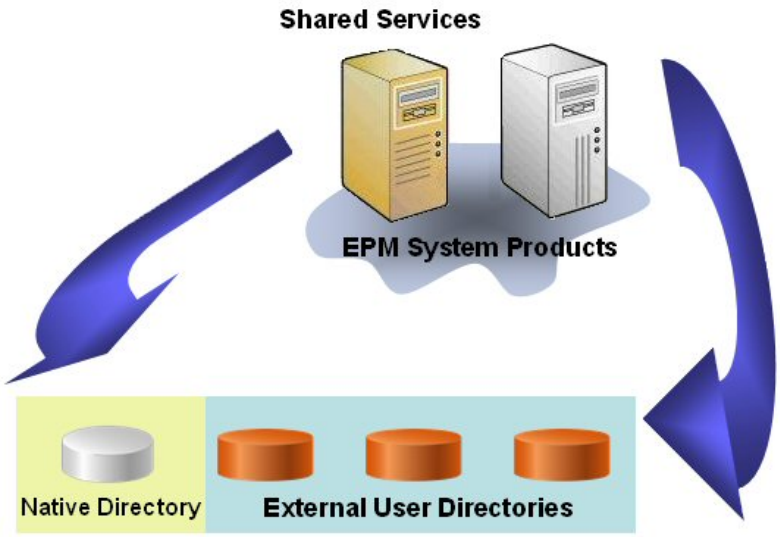
EPM System セキュリティでは、役割のコンセプトを使用してアプリケーションへのユーザー・アクセスが判別されます。役割とは、アプリケーション機能へのユーザー・アクセスを判別する権限です。一部の EPM System コンポーネントでは、レポートおよびメンバーなどのアーティファクトへのユーザー・アクセスをさらに詳細に制限するために、オブジェクトレベルの ACL が使用されます。

各 EPM System コンポーネントでは、様々な業務上の必要に対して調整された数個のデフォルトの役割が提供されます。EPM System コンポーネントに属する各アプリケーションはこの役割を継承します。Shared Services に登録されたアプリケーションからの事前定義済役割は、Shared Services Console から使用可能です。特定の要件に合うように、デフォルトの役割を集約する追加の役割も作成できます。この役割はプロビジョニングに使用されます。EPM System アプリケーションおよびそのリソースに属する固有の役割をユーザーおよびグループに付与するプロセスをプロビジョニングと呼びます。

ネイティブ・ディレクトリおよび構成済ユーザー・ディレクトリは、プロビジョニング・プロセス用のユーザーとグループ情報のソースです。Shared Services Console から、すべての構成済ユーザー・ディレクトリのユーザーとグループを参照およ

びプロビジョニングできます。また、プロビジョニング・プロセスでは、ネイティブ・ディレクトリで作成されたアプリケーション固有の集約役割も使用できます。

承認プロセスの概要図:



1. ユーザーが認証されたら、EPM System コンポーネントにより、ユーザー・ディレクトリへの問合せが行われ、ユーザーのグループが判別されます。
2. EPM System コンポーネントにより、グループ情報とユーザー情報を使用して、Shared Services からユーザーのプロビジョニング・データが取得されます。このデータを使用してユーザーがアクセスできるリソースが決定されます。

   製品固有のアクセス制御を設定するなどの製品固有のプロビジョニング・タスクは、各製品向けに完成されます。このデータは、プロビジョニング・データと組み合されて、ユーザーの製品アクセスを決定します。

EPM System 製品の役割ベースのプロビジョニングでは、これらのコンセプトが使用されます。

役割

役割は、EPM System リソースで機能を実行するユーザーとグループに許可されるアクセス権を定義するコンストラクト(アクセス制御リストに類似)です。役割は、リソースまたはリソース・タイプ(レポートなどのユーザーがアクセスできるもの)と、ユーザーがリソースで実行できるアクション(表示や編集など)の組合せです。

EPM System アプリケーション・リソースへのアクセスは制限されています。アクセスを提供する役割がユーザー、またはユーザーが属するグループに割り当てられてからのみ、ユーザーはこれらのリソースにアクセスできます。役割に基づいたアクセス制限では、管理者は、アプリケーション・アクセスを制御および管理できます。

グローバルな役割

グローバルな役割、つまり複数の製品に及ぶ Shared Services の役割により、ユーザーは EPM System 製品間で特定のタスクを実行できます。たとえば、Shared

Services 管理者は、すべての EPM System アプリケーションについてユーザーをプロビジョニングできます。

事前定義済役割

事前定義済役割は、EPM System 製品における組込みの役割です。これらを削除することはできません。EPM System 製品に属する各アプリケーション・インスタンスは、EPM System 製品の事前定義済役割を継承します。各アプリケーションのこれらの役割は、アプリケーションの作成時に Shared Services に登録されます。

集約役割

カスタム役割という名でも知られる集約役割では、アプリケーションに属する複数の事前定義済役割が集約されます。集約役割には、他の集約役割を含めることができます。たとえば、Shared Services 管理者またはプロビジョニング・マネージャは、Planning アプリケーションのプランナと表示ユーザーの役割を組み合せる集約役割を作成できます。役割を集約することにより、複数の細かい役割を持つアプリケーションの管理を簡略化できます。グローバル Shared Services の役割は、集約役割に含めることができます。複数のアプリケーションまたは製品に及ぶ集約役割は作成できません。

ユーザー

ユーザー・ディレクトリには、EPM System 製品にアクセスできるユーザーに関する情報が保管されています。認証および承認プロセスの双方でユーザー情報が使用されます。ネイティブ・ディレクトリ・ユーザーを作成して管理できるのは、Shared Services Console からのみです。

すべての構成済ユーザー・ディレクトリからのユーザーは、Shared Services Console から確認できます。これらのユーザーは、Shared Services に登録された EPM System アプリケーションでアクセス権を許可するように個別にプロビジョニングできます。個別ユーザーへのプロビジョニングはお薦めしません。

デフォルトの EPM System 管理者

管理者アカウント(デフォルト名 `admin`)は、配置プロセス中にネイティブ・ディレクトリに作成されます。これは最も強力な EPM System アカウントで、EPM System セキュリティおよび環境の管理の責任を負う情報テクノロジーの専門家であるシステム管理者の設定にのみ使用される必要があります。

EPM System 管理者のユーザー名およびパスワードは Foundation Services の配置中に設定されます。このアカウントは企業のアカウント・パスワード・ポリシーの対象にできないため、システム管理者アカウントを作成した後に非アクティブにすることをお薦めします。

通常、デフォルトの EPM System 管理者アカウントは次のタスクを実行するために使用します:

- 企業ディレクトリを外部ユーザー・ディレクトリとして構成します。第 4 章「ユーザー・ディレクトリの構成」を参照してください。

- 企業の情報テクノロジの専門家に Shared Services 管理者の役割をプロビジョニングして、システム管理者アカウントを作成します。Oracle Enterprise Performance Management System User Security Administration Guide のユーザーとグループのプロビジョニングに関する項を参照してください。

システム管理者

システム管理者は通常、企業の情報テクノロジの専門家で、EPM System 配置に含まれるすべてのサーバーに対する読取り、書込みおよび実行アクセス権を持っています。

一般に、システム管理者は次のタスクを実行します:

- デフォルトの EPM System 管理者アカウントを無効にします
- 機能の管理者を少なくとも 1 つ作成します
- Shared Services Console を使用して EPM System 用のセキュリティ構成を設定します
- オプションで、ユーザー・ディレクトリを外部ユーザー・ディレクトリとして構成します。
- ログ分析ツールを定期的に実行して EPM System を監視します。

  機能の管理者が実行するタスクは、このガイドに記載されています。

機能の管理者を作成する手順:

- 企業ディレクトリを外部ユーザー・ディレクトリとして構成します。第 4 章「ユーザー・ディレクトリの構成」を参照してください。
- ユーザーまたはグループに機能の管理者の作成に必要な役割をプロビジョニングします。Oracle Enterprise Performance Management System User Security Administration Guide のユーザーとグループのプロビジョニングに関する項を参照してください。

  機能の管理者には、次の役割がプロビジョニングされている必要があります:

  - Shared Services の LCM 管理者の役割
  - 配置されている各 EPM System コンポーネントの管理者およびプロビジョニング・マネージャの役割

機能の管理者

機能の管理者は、EPM System の専門家である企業ユーザーです。通常、このユーザーは外部ユーザー・ディレクトリとして Shared Services に構成されている企業ディレクトリで定義されます。

機能の管理者は、他の機能の管理者の作成、委任された管理の設定、アプリケーションやアーティファクトの作成およびプロビジョニング、EPM System 監査の設定などの EPM System 管理タスクを実行します。機能の管理者が実行するタスクは、Oracle Enterprise Performance Management System User Security Administration Guide に記載されています。

グループ

グループは、ユーザーまたは他のグループのコンテナです。Shared Services Console からネイティブ・ディレクトリ・グループを作成して、管理できます。すべての構成済ユーザー・ディレクトリからのグループは、Shared Services Console に表示されます。これらのグループをプロビジョニングして、Shared Services に登録された EPM System 製品の権限を許可できます。

Shared Services Console の起動

Oracle Hyperion Enterprise Performance Management Workspace のメニュー・オプションを使用して、Shared Services Console にアクセスします。

➤ Shared Services Console を起動するには:

1 次に移動します:

```
http://Web_server_name:port_number/workspace
```

URL の中で、`Web_server_name` は Foundation Services が使用する Web サーバーが実行されているコンピュータの名前を示し、`port_number` は、Web サーバー・ポートを示します。たとえば、`http://myWebserver:19000/workspace` のようになります。

注： セキュアな環境の EPM Workspace にアクセスする場合、プロトコルとして `https` (`http` ではなく)を使用し、セキュアな Web サーバー・ポート番号を使用します。たとえば、`https://myserver:19043/workspace` のような URL を使用します。

2 「アプリケーションの起動」をクリックします。

注： ポップアップ・ブロッカが原因で EPM Workspace が開かない場合があります。

3 「ログオン」で、ユーザー名とパスワードを入力します。

最初は、Shared Services Console へアクセスできる唯一のユーザーは、ユーザー名とパスワードが配置プロセス中に指定された EPM System 管理者です。

4 「ログオン」をクリックします。

5 「ナビゲート」>「管理」>「Shared Services Console」を選択します。

2 EPM Systemコンポーネントの SSL使用可能化

この章の内容

前提

- 配置トポロジを判別し、SSL を使用する保護された通信リンクを識別しています。
- よく知られている証明機関(CA)または独自の CA から必要な証明書を取得しているか、自己署名証明書を作成しているものとします。23 ページの「必須の証明書」を参照してください。
- SSL の概念および証明書のインポートなどの手順に精通しています。

  参照ドキュメントのリストについては、21 ページの「情報ソース」を参照してください。

情報ソース

EPM System を SSL 使用可能にするには、SSL を使用して通信するアプリケーション・サーバー、Web サーバー、データベース、ユーザー・ディレクトリなどのコンポーネントを準備している必要があります。このドキュメントでは、これらのコンポーネントを SSL 使用可能にするタスクに精通していることを前提としています。

- **Oracle WebLogic Server:** Securing WebLogic Server Guide の SSL の構成に関する説明を参照してください。
- **Oracle HTTP Server:** 『Oracle HTTP Server 管理者ガイド』の次のトピックを参照してください:

    - セキュリティの管理に関する説明
    - Oracle HTTP Server の SSL の使用可能化に関する説明
- **ユーザー・ディレクトリ**: ユーザー・ディレクトリ・ベンダーのドキュメントを参照してください。次のリンクが役に立ちます:
    - **Oracle Internet Directory:** 『Oracle Internet Directory 管理者ガイド』を参照してください。
    - **Sun Java System Directory Server:** Sun Java System Directory Server Administration Guide の Directory Server Security を参照してください
    - **Active Directory:** 次のドキュメントを参照してください:
        - Microsoft Windows Server 2008 Active Directory のドキュメント
        - Microsoft Windows Server 2003 Active Directory のドキュメント
    - **Novell eDirectory:** Novell eDirectory のドキュメントを参照してください。
- **データベース**: データベース・ベンダーのドキュメントを参照してください。
- **Internet Information Services:** IIS への SSL の実装方法を参照してください。

場所のリファレンス

このドキュメントでは、次のインストールおよび配置の場所を参照します:

- `MIDDLEWARE_HOME` は、WebLogic Server などのミドルウェア・コンポーネントの場所、オプションで 1 つ以上の `EPM_ORACLE_HOME` を参照します。`MIDDLEWARE_HOME` は、EPM System 製品のインストール中に定義されます。デフォルトの `MIDDLEWARE_HOME` ディレクトリは、`Oracle/Middleware` です。
- `EPM_ORACLE_HOME` は、EPM System 製品をサポートするのに必要なファイルを含むインストール・ディレクトリを参照します。`EPM_ORACLE_HOME` は `MIDDLEWARE_HOME` 内にあります。デフォルトの `EPM_ORACLE_HOME` は `MIDDLEWARE_HOME/EPMSystem11R1` で、たとえば、`Oracle/Middleware/EPMSystem11R1` となります。

  EPM System 製品は、`EPM_ORACLE_HOME/products` ディレクトリ(たとえば、`Oracle/Middleware/EPMSystem11R1/products`)にインストールされます。

  また、EPM System 製品の構成中に、一部の製品によってコンポーネントが `MIDDLEWARE_HOMEuser_projects/epmsystem1`(たとえば、`Oracle/Middleware/user_projects/epmsystem1`)に配置されます。
- `EPM_ORACLE_INSTANCE` は、一部の製品によってコンポーネントが配置される構成プロセス時に定義される場所を表します。`EPM_ORACLE_INSTANCE` のデフォルトの場所は、`MIDDLEWARE_HOME/user_projects/epmsystem1` (`Oracle/Middleware/user_projects/epmsystem1` など)です。

EPM System 製品の SSL 使用可能化について

EPM System 配置プロセスでは、自動的に Oracle EPM System 製品が SSL モードおよび非 SSL モードの両方で機能するよう配置されます。

EPM System の共通設定を指定する場合、配置内のすべてのサーバー間通信で SSL を使用可能にするかどうかを指定します。

配置プロセス中に SSL 設定を選択しても、各自の環境が自動的に SSL 用に構成されることはありません。この操作では、Oracle Hyperion Shared Services レジストリでフラグが設定されるだけであり、このフラグは、Shared Services レジストリを使用するすべての EPM System コンポーネントでサーバー間通信に安全なプロトコル (HTTPS)を使用する必要があることを示します。各自の環境を SSL 使用可能にするには、さらに手順を実行する必要があります。このドキュメントでは、このような手順について説明します。

サポートされている SSL シナリオ

次の SSL シナリオがサポートされています:

- SSL オフローダでの SSL 停止。24 ページの「SSL オフローダでの SSL 停止」を参照してください。
- 完全な SSL 配置。26 ページの「EPM System の完全な SSL 配置」を参照してください。

注： このドキュメントでは、WebLogic Server を使用して Web アプリケーションをホストしていることを前提としています。WebSphere を使用している場合、WebSphere アプリケーション・サーバーおよび IBM HTTP Server プラグインの SSL 使用可能化の詳細は、WebSphere のドキュメントを参照してください。

必須の証明書

SSL 通信では、コンポーネント間の信頼の確立に証明書が使用されます。よく知られたサードパーティ CA からの証明書を使用して、本番環境の EPM System を SSL 使用可能にすることをお薦めします。

注： EPM System では、1 つの SSL 証明書で複数のサブドメインをセキュアにできるワイルドカード証明書の使用をサポートしています。ワイルドカード証明書を使用すると、管理の時間とコストを削減できます。

ワイルドカード証明書を使用して通信を暗号化している場合、WebLogic Server でホスト名の確認を無効化する必要があります。

EPM System コンポーネントのホストの各サーバーに次の証明書が必要です:

- ルート CA 証明書。

注： ルート証明書が Java キーストアにすでにインストールされた、よく知られたサードパーティ CA からの証明書を使用している場合、Java キーストアにルート CA 証明書をインストールする必要はありません。

Firefox および Internet Explorer には、よく知られたサードパーティ CA の証明書があらかじめロードされています。CA として機能するには、これらのブラウザからアクセスされるクライアントで使用されるキーストアに CA ルート証明書をインポートする必要があります。たとえば、CA として機能する場合、CA ルート証明書が Web Analysis にアクセスするブラウザで使用できないと、Oracle Hyperion Web Analysis クライアントでサーバーとの SSL ハンドシェイクを確立できません。

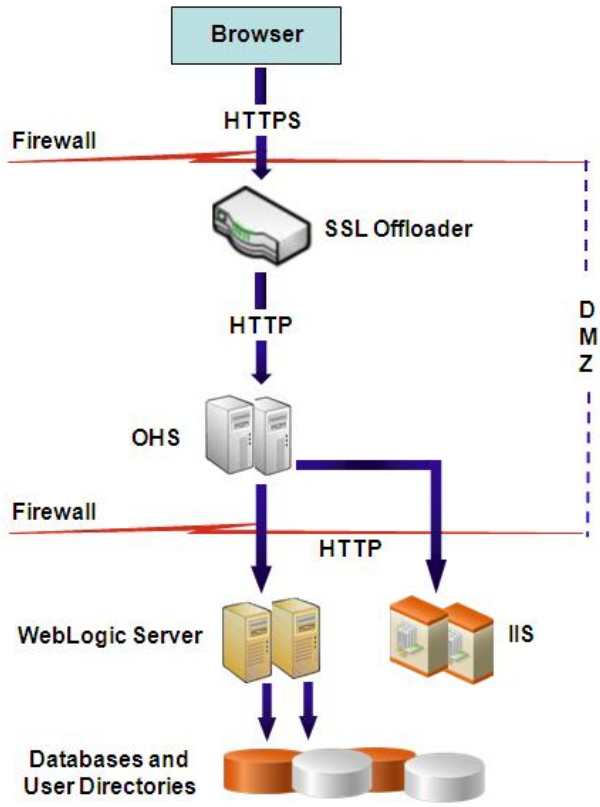
- 配置内の各 Oracle HTTP Server の署名付き証明書。
- WebLogic Server ホスト・マシンの署名付き証明書。このマシンの管理対象サーバーも、この証明書を使用できます。
- SSL オフローダおよびロード・バランサの 2 つの証明書。これらの証明書の 1 つは外部通信用で、もう 1 つは内部通信用です。

SSL オフローダでの SSL 停止

サブトピック

配置アーキテクチャ

このシナリオでは、SSL は EPM System クライアント(ブラウザなど)と SSL オフローダの間の通信リンクを保護するために使用されます。概念を図で示します:



前提

サブトピック

- SSL オフローダおよびロード・バランサ
- 仮想ホスト

SSL オフローダおよびロード・バランサ

完全に構成された SSL オフローダが、ロード・バランサとともに配置環境に存在している必要があります。

ロード・バランサは、仮想ホストから受信したすべての要求を Oracle HTTP Server に転送するように構成されている必要があります。

仮想ホスト

SSL オフローダで停止する SSL の構成では、SSL オフローダ/ロード・バランサで 2 つのサーバー別名(たとえば、`epm.myCompany.com`、`empinternal.myCompany.com`)が使用され、1 つはオフローダとブラウザの間の外部通信用、もう 1 つは EPM System サーバー間の内部通信用です。サーバー別名がマシンの IP アドレスを示し、DNS を介して解決可能であることを確認してください。

オフローダとブラウザ間の外部通信(`epm.myCompany.com` を使用)をサポートする署名付き証明書が、オフローダ/ロード・バランサにインストールされている必要があります。

EPM System の構成

EPM System コンポーネントのデフォルト配置は、SSL オフローダでの SSL 停止をサポートしています。追加のアクションは必要ありません。

EPM System を構成する際、論理 Web アプリケーションが、内部通信用に作成された別名(`empinternal.myCompany.com`)をポイントしていることを確認します。EPM System をインストールし、構成するには、次の情報ソースを参照してください:

- Oracle Enterprise Performance Management System Installation and Configuration Guide
- Oracle Enterprise Performance Management System Installation Start Here
- Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide

配置のテスト

配置プロセスが完了したら、次のセキュアな EPM Workspace の URL に接続して、すべてが機能していることを確認します:

```
https://
virtual_host_external
:
SSL_PORT
/workspace/index.jsp
```

たとえば、`https://epm.myCompany.com:443/workspace/index.jsp`(443 は SSL ポート)などです。

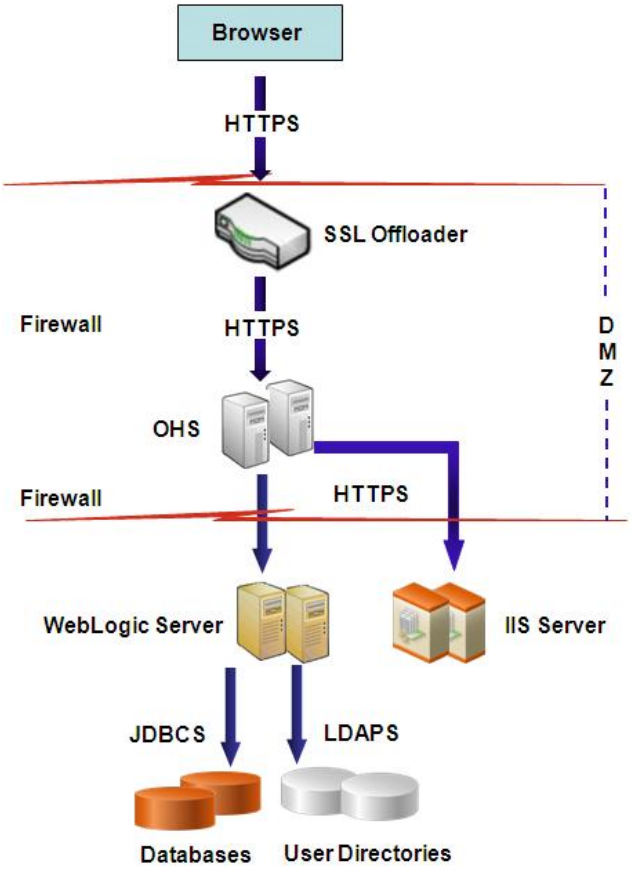
EPM System の完全な SSL 配置

サブトピック

配置アーキテクチャ

完全 SSL モードでは、すべてのセキュリティ保護可能なチャンネル間の通信は、SSL を使用して保護されています。この EPM System 配置シナリオは最も安全です。

概念を図で示します:



注： SSL 使用可能にはできない EPM System コンポーネントもあります。通常、バックエンド・サーバー(Oracle Hyperion Strategic Finance Server、Financial Management サーバーなど)は、SSL 使用可能にはできません。

前提

サブトピック

- データベース
- EPM System
- SSL オフローダおよびロード・バランサ

データベース

データベース・サーバーおよびクライアントは SSL 使用可能です。データベース・サーバーおよびクライアントの SSL 使用可能化の詳細は、データベースのドキュメントを参照してください。

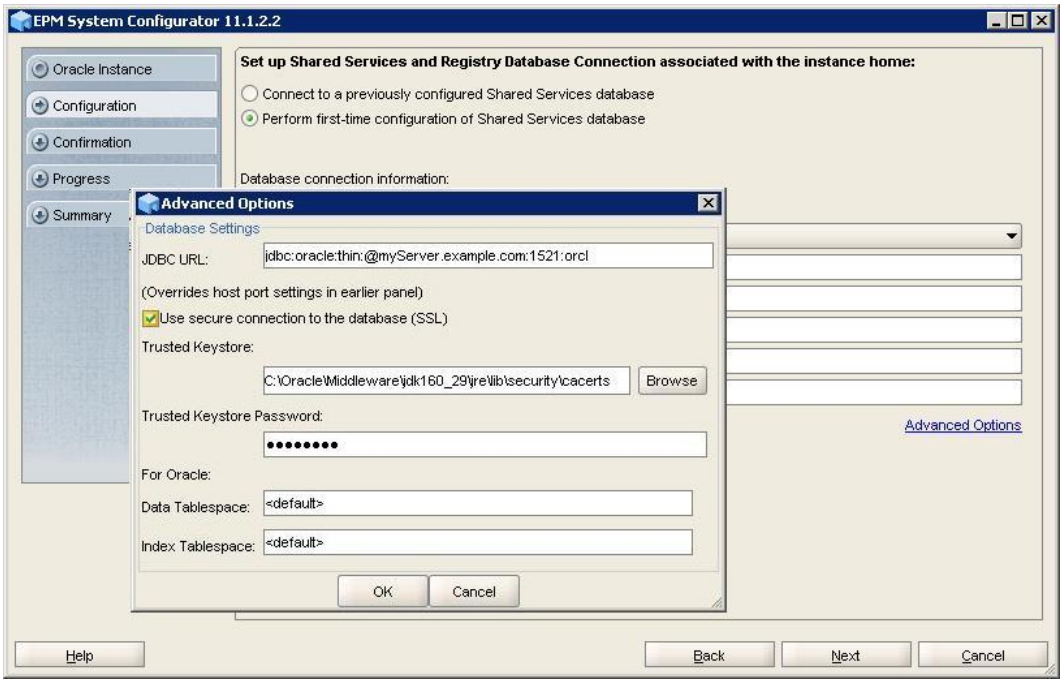
EPM System

WebLogic Server および Oracle HTTP Server を含む EPM System コンポーネントがインストールされ配置されます。さらに、EPM System 環境は、すべてが非 SSL モードで動作していることがテストされています。次の情報ソースを参照してください:

- Oracle Enterprise Performance Management System Installation and Configuration Guide
- Oracle Enterprise Performance Management System Installation Start Here
- Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide

データベースの接続を SSL 使用可能化する場合、構成プロセス中、各データベースの構成画面で「詳細設定オプション」リンクを選択し、次に示す必須の設定を指定する必要があります:

- 「データベースに対して保護された接続を使用(SSL)」を選択し、安全なデータベース URL を入力します。たとえば、
  `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=myDBhost)(PORT=1529)(CONNECT_DATA=(SERVICE NAME=myDBhost.myCompany.com)))` です。
- 「信頼できるキーストア」
- 「信頼できるキーストア・パスワード」



詳細は、Oracle Enterprise Performance Management System Installation and Configuration Guide を参照してください。

SSL オフローダおよびロード・バランサ

完全に構成された SSL オフローダが、ロード・バランサとともに配置環境に存在している必要があります。

完全 SSL 構成では、2 つのサーバー別名(`epm.myCompany.com` と `empinternal.myCompany.com` など)を SSL オフローダで使用します。1 つはオフローダとブラウザ間の外部通信用、もう 1 つは EPM System サーバー間の内部通信

用です。サーバー別名がマシンの IP アドレスを示し、DNS を介して解決可能であることを確認してください。

ロード・バランサは、仮想ホストから受信したすべての要求を Oracle HTTP Server に転送するように構成されている必要があります。

2 つの署名付き証明書(1 つは、`epm.myCompany.com` を介したオフローダとブラウザ間の外部通信のサポート用、もう 1 つは、`empinternal.myCompany.com` を介したアプリケーション間の内部通信のサポート用)がオフローダ/ロード・バランサにインストールされている必要があります。サーバー名の公開を防ぎ、セキュリティを強化するために、これらの証明書はサーバー別名に結び付けることをお薦めします。

完全 SSL 用 EPM System の構成

サブトピック

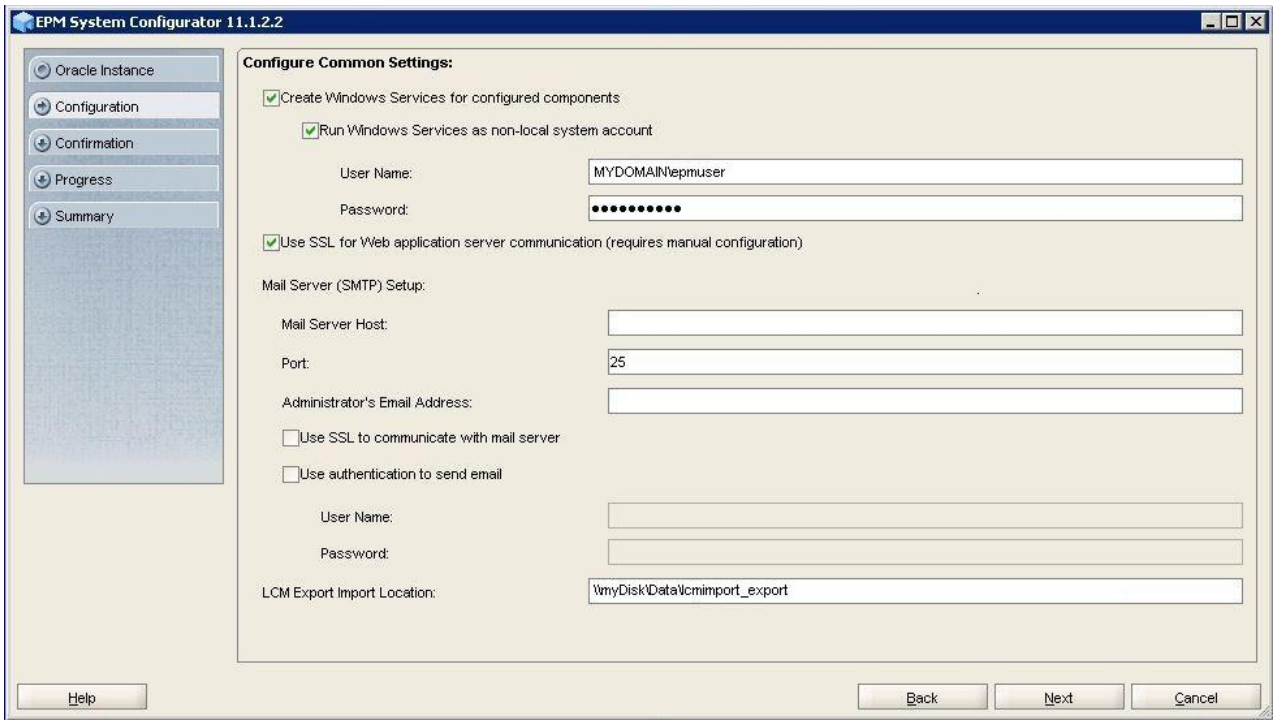
EPM System の共通設定の再構成

このプロセス中、EPM System コンポーネントに、SSL 通信を使用させる設定を選択します。

➤ SSL 用に EPM System を再構成するには:

1. **Oracle Hyperion Enterprise Performance Management System コンフィグレータを起動します。**
2. **「すべて選択解除」をクリアします。**
3. **「Hyperion Foundation」構成タスクを展開し、「共通設定の構成」を選択します。**
4. **「次へ」をクリックします。**
5. **「共通設定の構成」で、次の設定を選択します:**

> **注意** 電子メール・サーバーとの通信に SSL を使用する設定を選択する前に、電子メール・サーバーが SSL 用に構成されていることを確認します。



1. 「Java Web アプリケーション・サーバー通信に SSL を使用(手動構成が必要)」を選択し、EPM System が通信に SSL を使用する必要があることを指定します。
2. **オプション:** 「メール・サーバー・ホスト」および「ポート」に情報を入力します。SSL 通信をサポートするには、SMTP メール・サーバーで使用されるセキュアなポートを指定する必要があります。
3. **オプション:** SMTP メール・サーバーとの SSL 通信をサポートするには、「メール・サーバーとの通信に SSL を使用」を選択します。
4. 残りのフィールドに、設定を選択または入力します。

6 **後続の「EPM System コンフィグレータ」画面で「次へ」をクリックします。**

7 **配置プロセスが完了すると、「要約」画面が表示されます。すべてのタスクが正常に完了したことを確認し、「終了」をクリックします。**

オプション: WebLogic Server に対するルート CA 証明書のインストール

ほとんどのよく知られたサードパーティ CA のルート CA 証明書は、Sun および JRockit JVM キーストアにすでインストールされています。よく知られたサードパーティ CA の証明書を使用してインストールしていない場合、この項の手順を実行します(非推奨)。デフォルト JVM キーストアの場所:

- **Sun JVM キーストア:** `MIDDLEWARE_HOME/jdk160_29/jre/lib/security/cacerts`
- **JRockit JVM キーストア:** `MIDDLEWARE_HOME/jrockit_160_29/jre/lib/security/cacerts`

注: 各 EPM System サーバーで、この手順を実行します。

➤ ルート CA 証明書をインストールするには:

1 **ルート CA 証明書を WebLogic Server がインストールされているマシンのローカル・ディレクトリにコピーします。**

2 **コンソールから、ディレクトリを** `MIDDLEWARE_HOME/jdk160_29/jre/bin` **に変更します。**

3 **次のような keytool コマンドを実行して、署名付き証明書を Sun JVM キーストアにインストールします:**

```
keytool -import -alias
ALIAS
 -file
CA_CERT_FILE
 -keystore
KEYSTORE
 -storepass
KEYSTORE_PASSWORD
 -trustcacerts
```

たとえば、次のコマンドを使用して現在のディレクトリに格納されている証明書 `CAcert.crt` をキーストア内の証明書別名として `Blister` を使用して Sun JVM キーストアに追加できます。デフォルトのストアパス(`changeit`)を仮定します。

```
keytool -import -alias Blister -file CAcert.crt -keystore ../lib/security/
cacerts -storepass changeit -trustcacerts
```

注 : 前述のコマンドと例では、keytool を使用した証明書のインポートに構文の一部が使用されます。インポート構文の完全なリストは、keytool のドキュメントを参照してください。

4 **次のようなコマンドを実行して、ルート CA 証明書を JRockit JVM キーストアにインストールします:**

```
keytool -import -alias
ALIAS
 -file
CERT_FILE
 -keystore
KEYSTORE
 -storepass
KEYSTORE_PASSWORD
 -trustcacerts
```

たとえば、次のコマンドを使用して現在のディレクトリに格納されている証明書 `CAcert.crt` を証明書別名として `Blister` を使用して JRockit JVM キーストアに追加できます。デフォルトのストアパス(`changeit`)を仮定します。

```
keytool -import -alias Blister -file CAcert.crt -keystore
```

```
MIDDLEWARE_HOME
/jrockit_160_05/jre/lib/security/cacerts -storepass changeit -trustcacerts
```

注： MIDDLEWARE_HOME をディレクトリ・パスに置き換えていることを確認します。

WebLogic Server に対する証明書のインストール

デフォルトの WebLogic Server インストールでは、SSL をサポートするデモ用の証明書が使用されます。ご使用の環境を強化するために、よく知られたサードパーティが発行した証明書をインストールすることをお薦めします。

WebLogic Server のホストとなる各マシンで、ツール(keytool など)を使用して、WebLogic Server および EPM System Web コンポーネントの署名付き証明書を格納するカスタム・キーストアを作成します。

➤ カスタム・キーストアを作成して証明書をインポートするには:

1 **コンソールから、ディレクトリを** MIDDLEWARE_HOME/jdk160_29/jre/bin **に変更します。**

2 **次のような keytool コマンドを実行して、既存のディレクトリで(コマンドの-**keystore **ディレクティブで識別される)カスタム・キーストアを作成します:**

```
keytool -genkey -dname "cn=
myserver
, ou=
EPM
, o=
myCompany
, c=US" -alias
epm_ssl
 -keypass
password
 -keystore
C:\oracle\Middleware\EPMSystem11R1\ssl\keystore
 -storepass
password
 -validity 365 -keyalg RSA
```

注： 設定する共通名(cn)はサーバー名と一致する必要があります。cn に完全修飾ドメイン名(FQDN)を使用する場合、Web コンポーネントの配置の際に FQDN を使用する必要があります。

3 **証明書要求を生成します。**

```
keytool -certreq -alias
epm_ssl
 -file
C:/certs/epmssl_csr
```

```
 -keypass
password
 -storetype jks -keystore
C:\oracle\Middleware\EPMSystem11R1\ssl\keystore
 -storepass
password
```

4 WebLogic Server マシンの署名付き証明書を取得します。

5 署名付き証明書をキーストアにインポートします:

```
keytool -import -alias
epm_ssl
 -file
C:/certs/epmssl_crt
 -keypass
password
 -keystore
C:\Oracle\Middleware\EPMSystem11R1\ssl\keystore
 -storepass
password
```

WebLogic Server の構成

EPM System Web コンポーネントの配置後、SSL 通信用に構成する必要があります。

➤ SSL 用に Web コンポーネントを構成するには:

1 `MIDDLEWARE_HOME/user_projects/domains/EPMSystem/bin` に格納されているファイルを実行して、WebLogic Server を起動します:

- `startWebLogic.cmd` (Windows)
- `startWebLogic.sh` (UNIX)

2 次の URL にアクセスして WebLogic Server 管理コンソールを起動します:

```
http://
SERVER_NAME:Port
/console
```

たとえば、`myServer` のデフォルト・ポートに配置されている WebLogic Server コンソールにアクセスするには、`http://myServer:7001/console` を使用する必要があります。

3 「ようこそ」画面で、EPM System コンフィグレータで指定した WebLogic Server 管理者のユーザー名とパスワードを入力します。

4 「チェンジ・センター」で、「ロックして編集」をクリックします。

5 コンソールの左側のペインで、「環境」を展開して、「サーバー」を選択します。

6 「サーバーのサマリー」画面で、SSL 使用可能にするサーバーの名前をクリックします。

たとえば、Foundation Services コンポーネントを SSL 使用可能にするには、EPMServer0 サーバーを使用します。

7 「リスニング・ポートの有効化」を選択解除して、HTTP リスニング・ポートを使用不可にします。

8 「SSL リスニング・ポートの有効化」が選択されていることを確認します。

9 「SSL リスニング・ポート」に、このサーバーが要求をリスニングする SSL リスニング・ポートを入力します。

10 使用する ID と信頼キーストアを指定するには、「キーストア」を選択し、「キーストア」タブを開きます。

11 「変更」をクリックします。

12 次のいずれかのオプションを選択します:

- よく知られたサードパーティ CA からのサーバー証明書を使用していない場合、「カスタム ID とカスタム信頼」を選択します
- よく知られたサードパーティ CA からのサーバー証明書を使用している場合、「カスタム ID と Java 標準信頼」を選択します

13 「保存」をクリックします。

14 「カスタム ID キーストア」で、署名付き WebLogic Server 証明書がインストールされているキーストアのパスを入力します。

15 「カスタム ID キーストアのタイプ」で、jks と入力します。

16 「カスタム ID キーストアのパスフレーズ」および「カスタム ID キーストアのパスフレーズを確認」に、キーストアのパスワードを入力します。

17 「キーストア」で「カスタム ID とカスタム信頼」を選択した場合、次を実行します:

1. 「カスタム信頼キーストア」で、サーバー証明書に署名した CA のルート証明書が使用できるカスタム・キーストアのパスを入力します。
2. 「カスタム信頼キーストアのタイプ」で、jks と入力します。
3. 「カスタム信頼キーストアのパスフレーズ」および「カスタム信頼キーストアのパスフレーズを確認」に、キーストアのパスワードを入力します。

18 「保存」をクリックします。

19 SSL 設定を指定します。

1. 「SSL」を選択します。
2. 「秘密鍵の別名」で、署名付き WebLogic Server 証明書のインポートの際に指定した別名を入力します。
3. 「秘密鍵のパスフレーズ」および「秘密鍵のパスフレーズを確認」に、秘密鍵の取得に使用するパスワードを入力します。
4. 「保存」をクリックします。

20 このホストに属している各管理対象サーバーに対して、手順 6 から手順 19 を実行します。

21 「チェンジ・センター」で、「変更のアクティブ化」をクリックします。

Oracle HTTP Server に関する手順

サブトピック

ウォレットの作成および Oracle HTTP Server の証明書のインストール

デフォルトのウォレットは、Oracle HTTP Server とともに自動的にインストールされます。配置内の各 Oracle HTTP Server に、実際のウォレットを構成する必要があります。

➤ Oracle HTTP Server の証明書を作成し、インストールするには:

1 **Oracle HTTP Server のホストとなる各マシンで、Wallet Manager を起動します。**

- **Windows:** 「スタート」、「すべてのプログラム」、「Oracle-OHxxxxxx」、「統合管理ツール」、「Wallet Manager」を選択します。

  xxxxxx は Oracle HTTP Server インスタンス番号です。

- **UNIX:** `MIDDLEWARE_HOME/ohs/bin/owm` を実行して Wallet Manager をコマンドラインから起動します。

  注： Wallet Manager ではグラフィック環境が必要です。

2 **新規で空のウォレットを作成します。**

1. Oracle Wallet Manager で、「ウォレット」、「新規」を選択します。
2. 「はい」をクリックしてデフォルトのウォレット・ディレクトリを作成するか、「いいえ」をクリックして選択した場所にウォレット・ファイルを作成します。
3. 「新規ウォレット」画面の「ウォレット・パスワード」および「パスワードの確認」に、使用するパスワードを入力します。
4. 「OK」をクリックします。
5. 確認のダイアログボックスで、「いいえ」をクリックします。

3 **オプション: Oracle HTTP Server にとって既知の CA を使用していない場合、ルート CA 証明書をウォレットにインポートします。**

1. Oracle Wallet Manager で、「信頼できる証明書」を右クリックして「信頼できる証明書のインポート」を選択します。
2. ルート CA 証明書を参照して選択します。
3. 「オープン」を選択します。

4 **証明書要求を作成します。**

1. Oracle Wallet Manager で、「証明書: [空]」を右クリックして「証明書リクエストの追加」を選択します。
2. 証明書要求の作成で、必要な情報を入力します。

共通名について、システムの hosts ファイルで使用可能な完全修飾サーバー別名(たとえば、epm.myCompany.com または epminternal.myCompany.com)を入力します。

3. 「OK」をクリックします。
4. 確認のダイアログボックスで、「OK」をクリックします。
5. 作成した証明書要求を右クリックして、「証明書リクエストのエクスポート」を選択します。
6. 証明書要求ファイルの名前を指定します。

5 証明書要求ファイルを使用して、署名付き証明書を CA から取得します。

6 署名付き証明書をインポートします。

1. Oracle Wallet Manager で、署名付き証明書の取得に使用した証明書要求を右クリックし、「ユーザー証明書のインポート」を選択します。
2. 「証明書のインポート」で、「OK」をクリックして証明書をファイルからインポートします。
3. 「証明書のインポート」で、証明書ファイルを選択して「オープン」をクリックします。

7 ウォレットを便利な場所(EPM_ORACLE_INSTANCEhttpConfig/ohs/config/OHS/ohs_component/keystores/epmsystem など)に保存します。

8 「ウォレット」、「自動ログイン」を選択して自動ログインをアクティブ化します。

Oracle HTTP Server の SSL 使用可能化

Oracle HTTP Server をホストする各マシンで Web サーバーを再構成した後、作成したウォレットの場所で、デフォルト・ウォレットの場所を置き換え、Oracle HTTP Server 構成ファイルを更新します。

➤ SSL 用に Oracle HTTP Server を構成するには:

1 配置内の各 Oracle HTTP Server ホスト・マシン上の Web サーバーを再構成します。

1. このインスタンスの EPM System コンフィグレータを開始します。
2. 構成タスクの選択画面で、次の手順を完了し、「次へ」をクリックします。
    1. 「すべて選択解除」で、選択をクリア(チェック解除)します。
    2. 「Hyperion Foundation」タスク・グループを展開し、「Web サーバーの構成」を選択(チェック)します。
3. 「Web サーバーの構成」で、「次へ」をクリックします。
4. 「確認」で、「次へ」をクリックします。
5. 「要約」で、「終了」をクリックします。

2 配置内の各 Oracle HTTP Server の構成設定を更新します。

1. テキスト・エディタを使用して、EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/ssl.conf を開きます。

2. SSLWallet ディレクティブを見つけ、その値を、証明書をインストールしたウォレットを示すように変更します。ウォレットを EPM_ORACLE_INSTANCEhttpConfig/ohs/config/OHS/ohs_component/keystores/epmsystem に作成した場合、SSLWallet ディレクティブは次のようになります:

```
SSLWallet "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/keystores/epmsystem"
```

3. ssl.conf を保存して閉じます。

3 配置内の各 Oracle HTTP Server の mod_wl_ohs.conf を更新します。

1. テキスト・エディタを使用して、EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/mod_wl_ohs.conf を開きます。
2. WLSSLWallet ディレクティブが、SSL 証明書が格納されている Oracle Wallet を示していることを確認します。

```
WLSSLWallet
MIDDLEWARE_HOME
/ohs/bin/wallets/myWallet
```

例: C:/Oracle/Middleware/ohs/bin/wallets/myWallet

3. mod_wl_ohs.conf を保存して閉じます。

サーバーおよび EPM System の再起動

配置内のすべてのサーバーを再起動し、その後各サーバーの EPM System を開始します。

配置のテスト

SSL 配置が完了したら、すべてが機能していることを確認します。

➤ 配置をテストするには:

1 ブラウザを使用して、次のセキュアな EPM Workspace の URL にアクセスします:

外部通信用のサーバー別名に epm.myCompany.com を、SSL ポートに 4443 を使用した場合、EPM Workspace の URL は次のようになります:

```
https://epm.myCompany.com:4443/workspace/index.jsp
```

2 「ログオン」画面で、ユーザー名とパスワードを入力します。

3 「ログオン」をクリックします。

4 配置された EPM System コンポーネントにセキュアにアクセスできていることを確認します。

SSL 使用可能な外部ユーザー・ディレクトリの構成

サブトピック

前提

- Shared Services Console で構成する予定の外部ユーザー・ディレクトリは SSL 使用可能です。
- ユーザー・ディレクトリを SSL 使用可能にするために、よく知られたサードパーティ CA からの証明書を使用しなかった場合、サーバー証明書に署名した CA のルート証明書のコピーがあります。

ルート CA 証明書のインポート

ユーザー・ディレクトリを SSL 使用可能にするために、よく知られたサードパーティ CA からの証明書を使用しなかった場合、サーバー証明書に署名した CA のルート証明書を次の JVM にインポートする必要があります:

keytool などのツールを使用して、ルート CA 証明書をインポートします。

- すべての EPM System サーバー:
  - **Sun JVM キーストア:** `MIDDLEWARE_HOME/jdk160_11/jre/lib/security/cacerts`
  - **JRockit JVM キーストア:** `MIDDLEWARE_HOME/jrockit_160_05/jre/lib/security/cacerts`
- 各 EPM System コンポーネント・ホスト・マシンの JVM に使用されるキーストア。デフォルトでは、EPM System コンポーネントは次のキーストアを使用します:

  `MIDDLEWARE_HOME /jdk160_11/jre/lib/security/cacerts`

外部ユーザー・ディレクトリの構成

Shared Services Console を使用して、ユーザー・ディレクトリを構成します。ユーザー・ディレクトリの構成時、「SSL使用可能」オプションを選択し、EPM System セキュリティでセキュア・プロトコルを使用してユーザー・ディレクトリと通信するよう指定する必要があります。EPM System セキュリティと LDAP 対応のユーザー・ディレクトリ(Oracle Internet Directory、Microsoft Active Directory など)間の接続で SSL 使用可能にすることができます。

Oracle Enterprise Performance Management System User Security Administration Guide のユーザー・ディレクトリの構成に関する項を参照してください。

Financial Reporting Studio の暗号化の有効化

暗号化された RMI 通信用の Oracle Hyperion Financial Reporting Studio を構成するには、JVM スタートアップ・パラメータ(UNIX サーバーのシェル・スクリプト・ファイル)または JVMOption Windows レジストリ・エントリ(Windows サーバー)に次を追加します。

```
-Djavax.net.ssl.trustStore=
TRUSTSTORE_LOCATION
```

TRUSTSTORE_LOCATION を、CA のルート証明書をインストールしたキーストアの絶対的な場所と置き換えます。

Windows サーバーの Financial Reporting Studio にこのパラメータを追加するレジストリの場所は、HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Reports\HReports\JVM です。

Financial Reporting Web アプリケーションに JVM パラメータを追加する場所は、HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\FinancialReporting0\HyS9FRReports です。

Essbase 用 SSL

サブトピック

概要

Essbase では、一方向 SSL のみサポートされます。一方向 SSL では、Essbase インスタンス(サーバーとエージェント)は証明書を使用してセキュリティ保護されます。

この項では、Essbase インスタンスとコンポーネント(MaxL、Oracle Essbase Administration Services サーバー、Oracle Essbase Studio サーバー、Oracle Hyperion Provider Services、Foundation Services、Planning、Financial Management および Shared Services レジストリなど)間の通信の保護に使用されるデフォルト証明書を置き換える手順を説明します。

デフォルトの配置

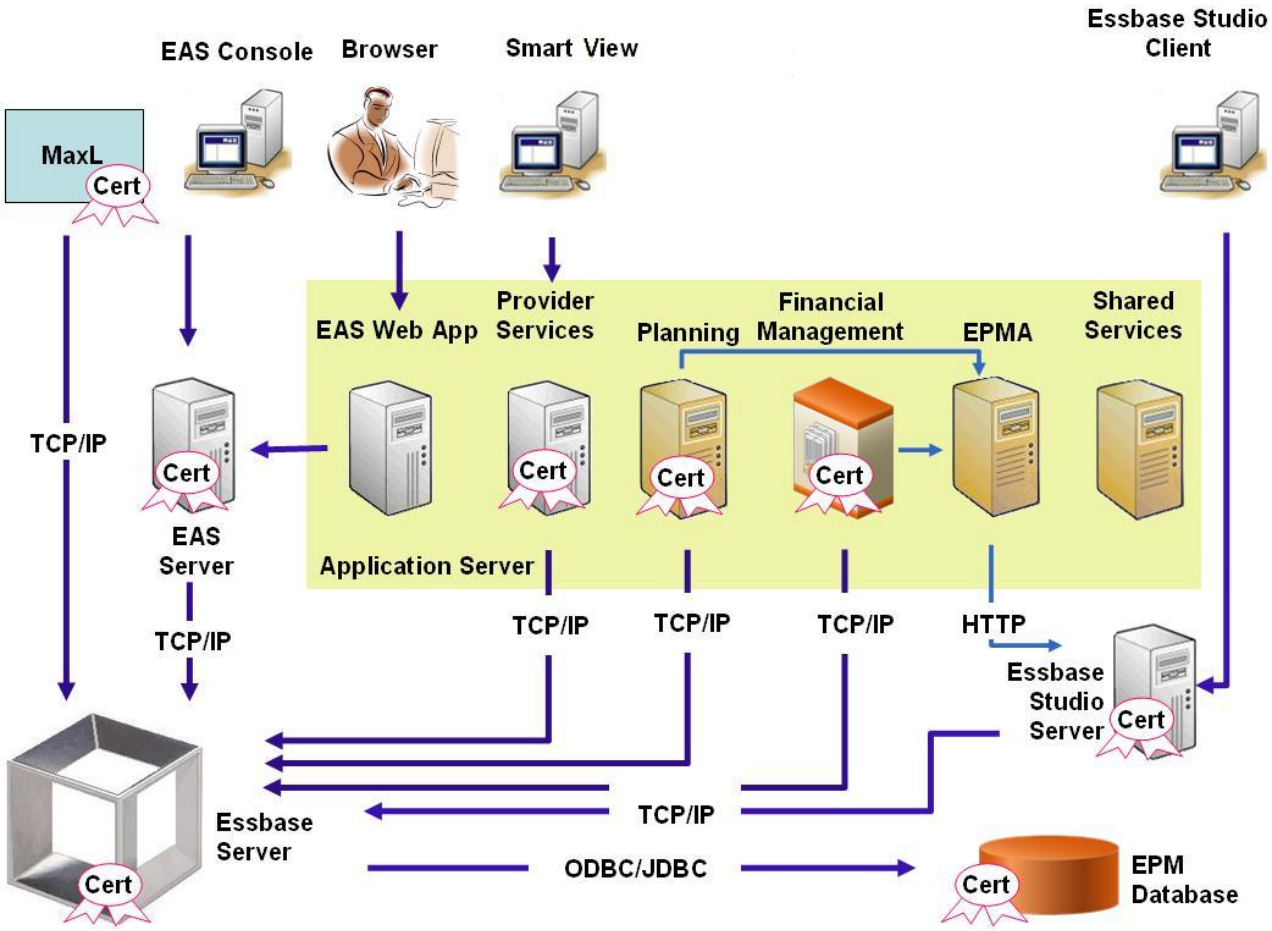
Essbase は、SSL モードおよび非 SSL モードで機能するよう配置できます。Essbase エージェントは、セキュアでないポートでリスニングしますが、セキュアなポー

トでリスニングするよう構成することもできます。セキュアなポートにアクセスするすべての通信は SSL 接続として処理されます。クライアントが Essbase エージェントに非 SSL ポートで接続すると、接続は非 SSL 接続として処理されます。コンポーネントは、Essbase エージェントに対して非 SSL 接続と SSL 接続を同時に確立できます。

ログイン時にセキュアなプロトコルとポートを指定することで、セッションごとに SSL を制御できます。47 ページの「セッションごとの SSL 接続の確立」を参照してください。

SSL が有効な場合、Essbase インスタンス内の通信はすべて暗号化され、データのセキュリティが保障されます。

セキュアなモードでの Essbase コンポーネントのデフォルトの配置では、主にテストを目的とする場合は、自己署名された証明書を使用して SSL 通信を有効にします。本番環境で Essbase を SSL 使用可能にするには、よく知られたサードパーティ CA から発行された証明書を使用することをお薦めします。



通常、Oracle Wallet に、Essbase RTC (C API)を使用するクライアントとの SSL 通信を有効にする証明書が保管され、Java キーストアに、通信に JAPI を利用するコンポーネントとの SSL 通信を有効にする証明書が保管されます。SSL 通信を確立するために、Essbase クライアントとツールは、Essbase サーバーの証明書に署名した CA のルート証明書を保管します。41 ページの「必要な証明書とその場所」を参照してください。

必要な証明書とその場所

よく知られたサードパーティ CA からの証明書を使用して、本番環境の Essbase を SSL 使用可能にすることをお薦めします。デフォルトの自己署名付き証明書は、テスト目的で使用します。

注： Essbase では、1 つの SSL 証明書で複数のサブドメインをセキュアにできるワイルドカード証明書の使用をサポートしています。ワイルドカード証明書を使用すると、管理の時間とコストを削減できます。

ホスト名チェックが有効な場合、ワイルドカード証明書は使用できません。

次の証明書が必要です:

- ルート CA 証明書。

  Essbase RTC (C API)を使用して Essbase との接続を確立するコンポーネントの場合、ルート CA 証明書を Oracle Wallet に保管する必要があります。JAPI を使用して接続を確立するコンポーネントの場合、ルート CA 証明書を Java キーストアに保管する必要があります。必要な証明書とその場所を次の表に示します。

  注： ルート証明書が Oracle Wallet にすでにインストールされた、よく知られたサードパーティ CA からの証明書を使用している場合、ルート CA 証明書をインストールする必要はありません。

- Essbase サーバーと Essbase エージェント用の署名付き証明書。

表 1　必要な証明書とその場所

コンポーネント [1]	キーストア	証明書 [2]
MaxL	Oracle Wallet	ルート CA 証明書
Administration Services サーバー	Oracle Wallet	ルート CA 証明書
Provider Services	Oracle Wallet	ルート CA 証明書
EPM システム・データベース	Oracle Wallet	ルート CA 証明書
Essbase Studio サーバー	Java キーストア	ルート CA 証明書
Planning	• Oracle Wallet • Java キーストア	ルート CA 証明書
Financial Management	Java キーストア	ルート CA 証明書
Essbase (サーバーおよびエージェント)[3]	• Oracle Wallet • Java キーストア	• ルート CA 証明書 • Essbase サーバーとエージェント用の署名付き証明書
Shared Services リポジトリ		

[1] 同様のキーストアを使用する複数のコンポーネントのサポートには、1 つのキーストアのインスタンスのみ必要です。

[2] 複数のコンポーネントで、キーストアにインストールされているルート証明書を使用できます。
[3] 証明書は、デフォルトの Oracle Wallet および Java キーストアにインストールされている必要があります。

Essbase と SSL 使用可能な EPM System

SSL を使用して EPM System をセキュリティ保護しても、Essbase は SSL 使用可能になりません。

SSL 使用可能な EPM System に配置されている Essbase インスタンスに影響を及ぼす設定は、Shared Services レジストリにある JDBC 接続設定のみです。EPM System Web コンポーネントが、セキュアな JDBC 接続を使用して Foundation Services データベースと通信するよう構成されている場合、Shared Services レジストリには、セキュアな JDBC 接続文字列が含まれます。このシナリオでは、Essbase で使用されるルート CA 証明書をデータベース・サーバーに手動でインストールします。

データベース・サーバーおよびクライアントの SSL 使用可能化の詳細は、データベースのドキュメントを参照してください。

Essbase コンポーネントのインストールと配置

構成プロセスで、セキュアなエージェント・ポート(デフォルトは 6423)を選択できます。このポートは Essbase サーバーの構成時に変更できます。デフォルトでは、配置プロセスで自己署名された必要な証明書がインストールされ、テスト用に機能上セキュアな配置が作成されます。

Oracle HTTP Server がインストールされている場合、Oracle Hyperion Enterprise Performance Management System インストーラで、Oracle Wallet と自己署名された証明書が Essbase インスタンスをホストするマシンの ARBOR_PATH 内にインストールされます。単一ホストの配置では、この証明書がすべての Essbase コンポーネントで共有されます。

信頼できるサードパーティ CA 証明書の Essbase への使用

サブトピック

証明書要求の作成と証明書の取得

証明書要求を生成して、Essbase サーバーと Essbase エージェントをホストするサーバー用の証明書を取得します。証明書要求には、識別名(DN)に固有の暗号化された情報が含まれます。証明書要求を署名機関に送信して SSL 証明書を取得します。

keytool や Oracle Wallet Manager などのツールを使用して証明書要求を作成します。証明書要求の作成の詳細は、使用しているツールのドキュメントを参照してください。

keytool を使用する場合、次のようなコマンド使用して証明書要求を作成します:

```
   keytool -certreq -alias essbase_ssl -file C:/certs/essabse_server_csr -keypass
password -storetype jks -keystore C:\oracle\Middleware\EPMSystem11R1\Essbase_ssl
\keystore -storepass password
```

ルート CA 証明書の取得とインストール

ルート CA 証明書は、SSL をサポートするのに使用される証明書の妥当性を証明します。これには、証明書を確認するために証明書の署名に使用された秘密鍵を照合する対象の公開鍵が含まれます。SSL 証明書に署名した認証局からルート CA 証明書を取得できます。

Essbase サーバーまたはエージェントに接続するクライアントに、Essbase サーバー証明書に署名した CA のルート証明書をインストールします。ルート証明書は、必ずクライアントに適したキーストアにインストールします。41 ページの「必要な証明書とその場所」を参照してください。

注： 複数のコンポーネントで、サーバー・マシンにインストールされているルート CA 証明書を使用できます。

Oracle Wallet

Oracle Wallet 内に CA ルート証明書が必要なコンポーネントのリストは、表 1 を参照してください。ウォレットを作成するか、デフォルトの自己署名付き証明書がインストールされているデモ・ウォレットに証明書をインストールします。

ウォレットの作成とルート CA 証明書のインポートの詳細な手順については、Oracle Wallet Manager のドキュメントを参照してください。

Java キーストア

Java キーストア内にルート CA 証明書が必要なコンポーネントのリストは、表 1 を参照してください。デフォルトの自己署名付き証明書がインストールされているキーストアに証明書を追加するか、証明書を保管するためのキーストアを新たに作成します。

注： 多くのよく知られたサードパーティ CA のルート CA 証明書は、Java キーストアにすでインストールされています。

詳細な手順については、使用するツールのドキュメントを参照してください。keytool を使用している場合、次のようなコマンドを使用してルート証明書をインポートします:

```
   keytool -import -alias blister_CA -file c:/certs/CA.crt -keypass
password -trustcacerts -keystore C:\Oracle\Middleware\EPMSystem11R1\Essbase_ssl
\keystore -storepass password
```

署名付き証明書のインストール

Essbase サーバーと Essbase エージェントをホストするサーバーに署名付き SSL 証明書をインストールします。Essbase RTC (C API)を使用して Essbase サーバーまたはエージェントとの接続を確立するコンポーネントの場合、証明書をルート CA 証明書とともに Oracle Wallet に保管する必要があります。JAPI を使用して Essbase サーバーまたはエージェントとの接続を確立するコンポーネントの場合、ルート CA 証明書と署名付き SSL 証明書を Java キーストアに保管する必要があります。詳細は、次の情報ソースを参照してください:

- Oracle Wallet Manager のドキュメント
- 証明書のインポートに使用するツール(keytool など)のドキュメントまたはオンライン・ヘルプ

keytool を使用する場合、次のようなコマンドを使用して証明書をインポートします:

```
   keytool -import -alias essbase_ssl -file C:/certs/essbase_ssl_crt -keypass
password -keystore
 C:\Oracle\Middleware\EPMSystem11R1\Essbase_ssl\keystore -storepass password
```

デフォルト設定の更新

サブトピック

- Essbase の SSL 設定の更新
- JAPI クライアント用の SSL プロパティのカスタマイズ
- Essbase C API を使用するコンポーネントに使用可能な暗号スイート

C API を使用するコンポーネント(Essbase サーバーとクライアント)の SSL 設定は、`essbase.cfg` に値を指定してカスタマイズします。

Essbase サーバーの SSL 設定は、`essbase.cfg` に値を指定してカスタマイズします。

Essbase の SSL 設定の更新

`essbase.cfg` を編集して、次のような Essbase の SSL 設定をカスタマイズします:

- セキュア・モードを有効にする設定
- クリア・モードを有効にする設定
- クライアントとの通信で優先されるモード(クライアントでのみ使用)
- セキュアなポート

- 暗号スイート
- Oracle Wallet のパス

➤ essbase.cfg を更新するには:

1 テキスト・エディタを使用して、EPM_ORACLE_INSTANCE/EssbaseServer/essbaseserver1/bin/essbase.cfg を開きます。

2 必要に応じて設定を入力します。表 2 を参照してください。

表 2　Essbase の SSL 設定

設定	説明 [1]
EnableClearMode [2]	Essbase アプリケーションと Essbase エージェントとの間で暗号化されていない通信を有効にします。このプロパティが FALSE に設定されている場合、Essbase は SSL 要求を処理できません。 **デフォルト:** TRUE **例:** EnableClearMode FALSE
EnableSecureMode	Essbase クライアントと Essbase エージェントとの間で SSL 暗号化通信を有効にします。SSL をサポートするには、このプロパティを TRUE に設定する必要があります。 **デフォルト:** FALSE **例:** EnableSecureMode TRUE
SSLCipherSuites	SSL 通信で使用される暗号スイートの優先順のリスト。47 ページの「Essbase C API を使用するコンポーネントに使用可能な暗号スイート」を参照してください。Essbase エージェントで、これらの暗号スイートの 1 つが SSL 通信に使用されます。エージェントが暗号スイートを選択する際、リスト内の最初の暗号スイートに最も高い優先度が適用されます。 **デフォルト:** SSL_RSA_WITH_RC4_128_MD5 **例:** SSLCipherSuites SSL_RSA_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_DES_CBC_SHA
AgentSecurePort	エージェントがリスニングするセキュアなポート。 **デフォルト:** 6423 **例:** AgentSecurePort 16001
WalletPath	ルート CA 証明書と署名付き証明書を保管する Oracle Wallet の場所(1024 文字未満)。 **デフォルト:** ARBORPATH/bin/wallet **例:** WalletPath/usr/local/wallet

設定	説明 [1]
ClientPreferredMode [3]	クライアント・セッションのモード(セキュアまたはクリア)。このプロパティが Secure に設定されている場合、SSL モードがすべてのセッションに使用されます。 このプロパティが Clear に設定されている場合、クライアント・ログイン要求にセキュアなトランスポート・キーワードが含まれているかどうかに基づいてトランスポートが選択されます。47 ページの「セッションごとの SSL 接続の確立」を参照してください。 **デフォルト:** CLEAR **例:** ClientPreferredMode SECURE

[1] essbase.cfg にプロパティがない場合、デフォルト値が適用されます。

[2] EnableClearMode と EnableSecureMode が FALSE に設定されると、Essbase は動作不能になります。

[3] クライアントはこの設定を使用して、Essbase でセキュアな通信を確立するかセキュアでない通信を確立するかを決定します。

3 essbase.cfg を保存して閉じます。

JAPI クライアント用の SSL プロパティのカスタマイズ

JAPI を使用する Essbase コンポーネントを配置すると、様々なデフォルト・プロパティが設定されます。これらのカスタマイズ可能なプロパティは、essbase.properties に外在します。

➤ JAPI クライアントの SSL プロパティを更新するには:

1 テキスト・エディタを使用して、EPM_ORACLE_HOMEcommon\EssbaseJavaAPI\11.1.2.0\bin\essbase.properties を開きます。

2 必要に応じてプロパティを更新します。カスタマイズ可能な JAPI クライアントのプロパティの詳細は、表 3 を参照してください。

表 3 JAPI クライアントのデフォルト SSL プロパティ

プロパティ	説明
olap.server.ssl.alwaysSecure	すべての Essbase インスタンスに対してクライアントで使用されるモードを設定します。このプロパティ値を true に設定すると、SSL モードが適用されます。 **デフォルト:** false
olap.server.ssl.securityHandler	プロトコルの処理用パッケージ名。この値を変更して別のハンドラを指定できます。 **デフォルト:** java.protocol.handler.pkgs
olap.server.ssl.securityProvider	Oracle では Sun SSL プロトコル実装が使用されます。この値を変更すると、別のプロバイダを指定できます。 **デフォルト:** com.sun.net.ssl.internal.www.protocol

プロパティ	説明
olap.server.ssl.supportedCiphers	セキュアな通信用に有効にされる追加の暗号のカンマ区切りリスト。Essbase でサポートされる暗号のみを指定する必要があります。47 ページの「Essbase C API を使用するコンポーネントに使用可能な暗号スイート」を参照してください。 **例**: SSL_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_WITH_AES_128_CBC_SHA
olap.server.ssl.trustManagerClass	署名の確認と証明書の有効期限のチェックによる SSL 証明書の検証に使用する TrustManager クラス。 デフォルトでは、すべての検証チェックを実施するようにはこのプロパティは設定されません。 誤りチェックが実施されないようにするには、このパラメータの値を com.essbase.services.olap.security.EssDefaultTrustManager に設定します。これは、すべての検証チェックを成功とするデフォルトの TrustManager クラスです。 カスタム TrustManager を実装するには、javax.net.ssl.X509TrustManager インタフェースを実装する TrustManager クラスの完全修飾クラス名を指定します。 **例**:com.essbase.services.olap.security.EssDefaultTrustManager
olap.server.ssl.keyManagerClass	このパラメータは、このリリースでは使用されません。

3 essbase.properties を保存して閉じます。

4 すべての Essbase コンポーネントを再起動します。

Essbase C API を使用するコンポーネントに使用可能な暗号スイート

次の暗号スイートが、Essbase サーバーの SSL 実装でサポートされます。

- SSL_RSA_WITH_AES_256_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_RC4_128_MD5

セッションごとの SSL 接続の確立

MaxL などの Essbase コンポーネントでは、トランスポート・キーワードとして secure を使用して Essbase エージェントに接続すると、セッション・レベルで SSL を制御できます。たとえば、次のいずれかのコマンドを MaxL Console から実行すると、MaxL と Essbase エージェントとの間にセキュアな接続を確立できます。

```
login admin welcome1 on hostA:
PORT
```

```
:secure
```

```
login admin welcome1 on hostA:secure
```

セッションごとの制御は、essbase.cfg に指定された構成設定より優先されます。トランスポート・キーワードが指定されていない場合、Essbase クライアントでは、ClientPreferredMode に設定された値を使用して、Essbase とのセキュアな接続を開始するかどうかを決定します。ClientPreferredMode 設定が secure に設定されていない場合、通信は非セキュアなチャネルで常に行われます。

3 セキュリティ・エージェントでのSSOの使用可能

この章の内容

サポートされている SSO メソッド

サブトピック

- HTTP ヘッダー
- カスタム・ログイン・クラス
- HTTP 認証ヘッダー
- HTTP 要求からリモート・ユーザーを取得

SSO では、Web アイデンティティ管理ソリューションで、認証済ユーザーのログイン名が EPM System 製品に渡される必要があります。次の標準的な EPM System の方法を使用して、EPM System と市販あるいはカスタムの Web ベースの SSO ソリューションを統合できます。

- 50 ページの「HTTP ヘッダー」
- 50 ページの「カスタム・ログイン・クラス」
- 51 ページの「HTTP 認証ヘッダー」
- 52 ページの「HTTP 要求からリモート・ユーザーを取得」

注意 セキュリティ対策として、組織で ID 伝播用のヘッダーにユーザー ID を持つメソッドを使用する場合、Web サーバーとアプリケーション・サーバー間でクライアント証明書認証(双方向 SSL)を実装することをお薦めします。

HTTP ヘッダー

Oracle Single Sign-on (OSSO)、SiteMinder または Oracle Access Manager を Web アイデンティティ管理ソリューションとして使用中の場合、EPM System セキュリティでは自動的にカスタム HTTP ヘッダーを選択して、認証済ユーザーのログイン名を EPM Systemコンポーネントに渡します。

EPM System 製品ユーザーのログイン名は、Shared Services でユーザー・ディレクトリを構成する際に指定されるログイン属性によって決定されます。ログイン属性の簡単な説明については、Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。

HTTP ヘッダーには、ログイン属性として設定される属性の値が含まれている必要があります。たとえば、uid がログイン属性値である場合、HTTP ヘッダーは、uid 属性の値を持っている必要があります。

カスタム HTTP ヘッダーの定義および発行の詳細は、Web アイデンティティ管理ソリューションのドキュメントを参照してください。

EPM System セキュリティにより、HTTP ヘッダーが分析され、Shared Services で構成されたユーザー・ディレクトリに対して持っているログイン名が検証されます。

カスタム・ログイン・クラス

ユーザーがログインすると、Web アイデンティティ管理ソリューションでは、ユーザーがディレクトリ・サーバーに対して認証され、SSO メカニズムで認証済ユーザーのログイン情報がカプセル化されて、下流のシステムで SSO が使用可能になります。Web アイデンティティ管理ソリューションで、EPM System 製品によってサポートされないメカニズムが使用されるか、ログイン属性の値が SSO メカニズムで使用できない場合、カスタム・ログイン・クラスを使用し、ログイン属性の値を導いて EPM System 製品に渡すことができます。

ログイン・クラスを使用することによって、X509 証明書ベースの認証を使用するセキュリティ・エージェントと EPM System を統合できます。この認証メカニズムを使用するには、EPM System コンポーネント間の SSO インタフェースを定義するための標準 Shared Services API の実装と、Web アイデンティティ管理ソリューションが必要です。カスタム・ログイン・クラスでは、ログイン属性の値が EPM System 製品に渡される必要があります。ログイン属性の簡単な説明については、Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。サンプル・コードおよび実装手順については、付録 B「カスタム・ログイン・クラスの実装」を参照してください。

カスタム・ログイン・クラス(デフォルト名 com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl)を使用するには、com.hyperion.css.CSSSecurityAgentIF インタフェースの実装をこのクラスパスで使用できる必要があります。CSSSecurityAgentIF では、ユーザー名とパスワードを取得するゲッター・メソッドが定義されます(オプション)。インタフェースで null のパスワードが戻される場合、セキュリティ認証ではプロバイダが信頼済として扱われ、構成済プロバイダにおけるユーザーの存在が確認さ

れます。インタフェースでパスワードの null 以外の値が戻される場合、EPM System では、この実装により戻されるユーザー名とパスワードを使用して要求の認証が試みられます。

CSSSecurityAgentIF は、getUserName と getPassword の 2 つのメソッドから構成されています。

getUserName メソッド

このメソッドでは、認証用のユーザー名が戻されます。

```
java.lang.String getUserName(
        javax.servlet.http.HttpServletRequest req,
        javax.servlet.http.HttpServletResponse res)
        throws java.lang.Exception
```

req パラメータでは、ユーザー名を判別するために使用される情報を持つ HTTP 要求が識別されます。res パラメータは使用されません(下位互換性にプリセット)。

getPassword メソッド

このメソッドでは、認証用のクリアテキストのパスワードが戻されます。パスワードの取得はオプションです。

```
java.lang.String getPassword(
        javax.servlet.http.HttpServletRequest req,
        javax.servlet.http.HttpServletResponse res)
        throws java.lang.Exception
```

req パラメータでは、パスワードを判別するために使用される情報を持つ HTTP 要求が識別されます。res パラメータは使用されません(下位互換性にプリセット)。

HTTP 認証ヘッダー

EPM System セキュリティは HTTP 認証ヘッダーの使用をサポートし、ログイン属性の値を Web アイデンティティ管理ソリューションから EPM System 製品に渡します。EPM System 製品は、認証ヘッダーを分析して、ユーザーのログイン名を取得します。

HTTP 要求からリモート・ユーザーを取得

EPM System セキュリティは HTTP 要求の使用をサポートし、ログイン属性の値を Web アイデンティティ管理ソリューションから EPM System 製品に渡します。Web アイデンティティ管理ソリューションがログイン属性(`setRemoteUser` 関数を使用して設定される)の値を含む HTTP 要求を渡す場合、この SSO メソッドを使用します。

Oracle Access Manager からのシングル・サインオン

EPM System は、ログイン属性値を含むカスタム HTTP ヘッダー(デフォルト名は `HYPLOGIN`)を受け入れることで Oracle Access Manager と統合されます。ログイン属性は、Shared Services で外部ユーザー・ディレクトリを構成する際に設定されます。ログイン属性の簡単な説明については、Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。

EPM System にログイン属性を提供する任意のヘッダー名を使用できます。ヘッダー名は、Oracle Access Manager からの SSO 用に Shared Services を構成する際に使用します。

EPM System は、ログイン属性の値を使用して、構成されているユーザー・ディレクトリ(この場合は、Oracle Access Manager がユーザーの認証に使用するユーザー・ディレクトリ)に対してユーザーを認証し、EPM System 全体で SSO を有効にする EPM SSO トークンを生成します。ユーザーのプロビジョニング情報がネイティブ・ディレクトリで確認され、ユーザーに EPM System リソースが許可されます。

注： シック・クライアントである Administration Services コンソールは、Oracle Access Manager からの SSO をサポートしていません。

Oracle Access Manager の構成および HTTP ヘッダーおよびポリシー・ドメインの設定などのタスクの実行に関する情報は、Oracle Access Manager のドキュメントに記載されています。このガイドは、次のタスクが完了し、機能している Oracle Access Manager の配置を想定しています。

- EPM System コンポーネントに必要なポリシー・ドメインを設定します。
- ログイン属性値を EPM System に渡す HTTP ヘッダーの構成。
- 61 ページの「保護するリソース」にリストされた EPM System リソースの保護。保護されたリソースへのアクセス要求は Oracle Access Manager によって処理されます。
- 62 ページの「保護しないリソース」にリストされた EPM System リソースの保護解除。保護されないリソースへのアクセス要求は Oracle Access Manager によって処理されません。

➤ EPM System に Oracle Access Manager からの SSO を構成するには:

1 **Oracle Access Manager が EPM System で外部ユーザー・ディレクトリとしてユーザーを認証するために使用するユーザー・ディレクトリを追加します。Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。**

注 : 「接続情報」画面で、ユーザー・ディレクトリが信頼できる SSO ソースであることを示す「信頼済」チェック・ボックスが選択されていることを確認します。

2 **EPM System に SSO を構成します。84 ページの「SSO 用の EPM System の構成」を参照してください。**

「SSO プロバイダ/エージェント」リストから、Oracle Access Manager を選択します。Oracle Access Manager からの HTTP ヘッダーで HYPLOGIN 以外の名前を使用する場合、「SSO メカニズム」リストの隣にあるテキスト・ボックスにカスタム・ヘッダーの名前を入力します。

3 **Oracle Data Relationship Management のみ:**

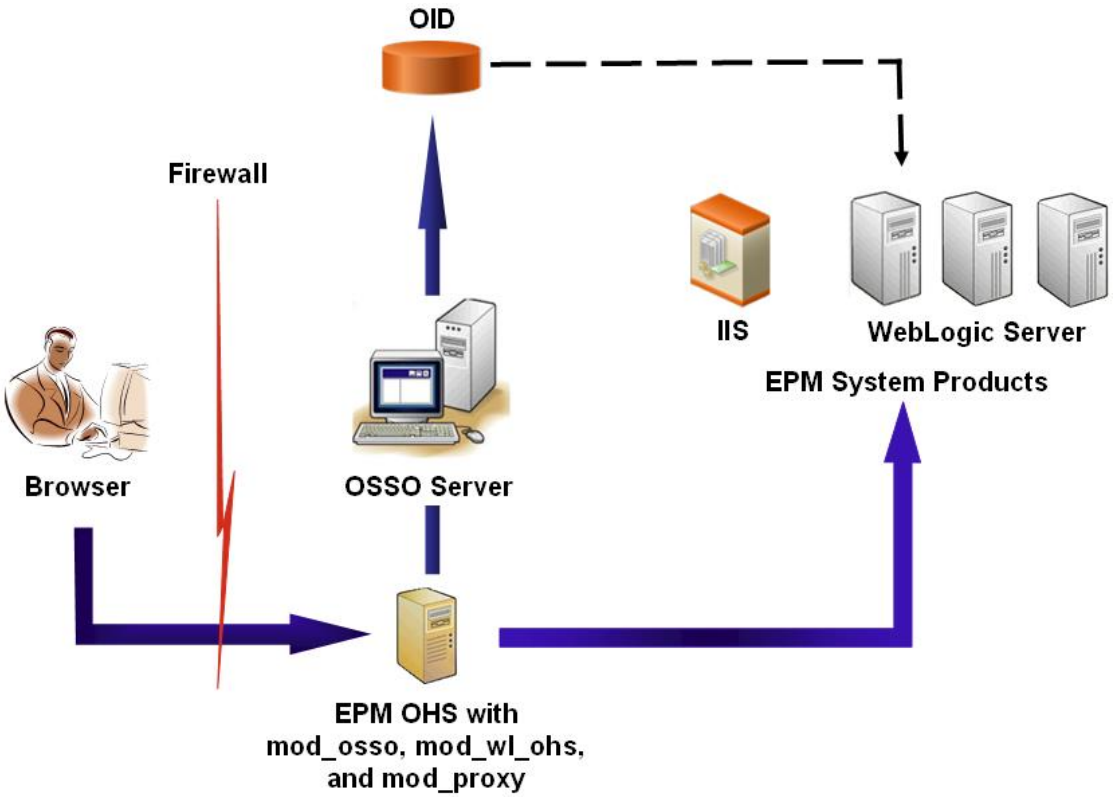
1. Shared Services の認証に Data Relationship Management を構成します。
2. Data Relationship Management コンソールで SSO を使用可能にします。

   詳細は、Data Relationship Management のドキュメントを参照してください。

OracleAS シングル・サインオン

OracleAS Single Sign-on (OSSO)ソリューションでは、Oracle Internet Directory (OID)をユーザー・ディレクトリとして使用して、Web アプリケーションへの SSO アクセスを提供します。ユーザーは、OID で定義されたユーザー名とパスワードを使用して、EPM System 製品にログインします。

プロセス・フロー



OSSO プロセス:

1. EPM System URL(`http://OSSO_OHS_Server_NAME:OSSO_OHS_Server_PORT/interop/index.jsp` など)を使用して、OSSO で保護されたアプリケーションとして定義される EPM System コンポーネントにアクセスします。
2. URL が OSSO で保護されているため、Oracle HTTP Server に配置された `mod_osso` は要求をインターセプトします。`mod_osso` を使用して、Oracle HTTP Server は有効な cookie を確認します。有効な cookie が要求で使用不可能な場合、Oracle HTTP Server は、OID に対して認証されるログイン情報を要求する OSSO サーバーにユーザーをリダイレクトします。
3. OSSO サーバーは obSSOCookie を作成し、ブラウザに obSSOCookie を設定する Oracle HTTP Server 上の `mod_osso` モジュールに制御を返します。また、`mod_wl_ohs`(WebLogic Server)または `mod_proxy`(IIS Server)を介して、EPM System リソースに要求をリダイレクトします。EPM System リソースに要求を転送する前に、Oracle HTTP Server は、EPM System セキュリティで SSO 使用可能にするのに使用するプロキシ・リモート・ユーザー・ヘッダーを設定します。
4. EPM System コンポーネントは、プロキシ・リモート・ユーザーからユーザー ID を取得するユーザーが OID に存在することを確認します。このプロセスが機能するには、OSSO サーバーを使用して構成される OID を Shared Services の外部ユーザー・ディレクトリとして構成する必要があります。

前提条件

1. 完全な機能の Oracle Application Server インフラストラクチャ。

Oracle Application Server インフラストラクチャを確立するには、Oracle Identity Management Infrastructure 10.1.4 をインストールおよび構成します。OSSO が使用可能であることを確認します。Oracle Identity Management Infrastructure 10.1.4 をインストールすると、次のコンポーネントが含まれ、OSSO をサポートします。

- Oracle 10g OSSO Server。
- OSSO サーバーでログイン情報を検証するのに使用する OID。次のガイドを参照してください:
  - Oracle Fusion Middleware Oracle Identity Management インストレーション・ガイド
  - Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド
- OSSO サーバーへのフロントエンドとしての Oracle HTTP Server。このインストールには、OSSO のパートナ・アプリケーションを定義できる `mod_osso` が含まれます。

  注： この Oracle HTTP Server インスタンスは、OSSO インフラストラクチャの一部です; EPM System コンポーネントの OSSO の構成に直接は使用されません。

  インストール・プロセス中に、`mod_osso` がパートナ・アプリケーションとして OSSO サーバーに登録されていることを確認します。

2. 完全な機能の EPM System 配置。

   EPM System コンポーネントに Web サーバーを構成する場合、EPM System コンフィグレータは Oracle HTTP Server に次を構成し、要求をアプリケーション・サーバーにプロキシします:

   - WebLogic Server に要求をプロキシする `mod_wl_ohs.conf`
   - IIS Server に要求をプロキシする `mod_proxy`

EPM System 向けの OSSO の使用可能化

サブトピック

このセクションでは、完全に構成された OSSO インフラストラクチャがあることを前提としています。『Oracle Application Server 管理者ガイド』を参照してください。

パートナ・アプリケーションとしての EPM System Web サーバーの登録

Oracle Identity Manager の SSO 登録ツール(ssoreg.sh または ssoreg.bat)を使用して、OSSO サーバーをフロントエンドする Oracle HTTP Server のパートナ・アプリケーションとして、EPM System Web サーバーを登録します。

OSSO サーバーをフロントエンドする Oracle HTTP Server のホストとなるサーバー上で、この手順を実行します。このプロセスでは、選択した場所に難読化された osso.conf を生成および格納します。

➤ パートナ・アプリケーションとして EPM System Web サーバーを登録するには:

1 **OSSO サーバーをフロントエンドする Oracle HTTP Server のホストとなるサーバー上のコンソールを開き、Oracle HTTP Server の** ORACLE_HOME/sso/bin **ディレクトリ (Windows の場合、**C:/OraHome_1/sso/bin **など)に移動します。**

2 -remote_midtier **オプションで次のようなコマンドを実行します:**

```
     ssoreg.bat -site_name epm.myCompany.com
-mod_osso_url http://epm.myCompany.com:19400
-config_mod_osso TRUE
-update_mode CREATE
-remote_midtier
-config_file C:\OraHome_1\myFiles\osso.conf
```

次に、このコマンドで使用されるパラメータについて説明します。この説明では、パラメータ・アプリケーションは、EPM System Web サーバーとして使用される Oracle HTTP Server を参照します。

- -site_name は、パートナ・アプリケーションの Web サイト (epm.myCompany.com など)を識別します。

- -mod_osso_url は、パートナ・アプリケーションの URL を PROTOCOL://HOST_NAME:PORT 形式で示します。これは、EPM System Web サーバーが受信クライアント要求を受け入れる URL(http://epm.myCompany.com:19000 など)です。
- -config_mod_osso は、パートナ・アプリケーションで mod_osso を使用することを示します。config_mod_osso パラメータを含めて osso.conf を生成する必要があります。
- -update_mode は、更新モードを示します。デフォルトの CREATE を使用して、新規レコードを生成します。
- -remote_midtier は、mod_osso パートナ・アプリケーションが離れた中間層にあることを示します。パートナ・アプリケーションが OSSO サーバーとは異なる ORACLE_HOME にある場合に、このオプションを使用します。
- -virtualhost は、パートナ・アプリケーションの URL が仮想ホストであることを示します。仮想ホストを使用していない場合は、このパラメータを使用しないでください。

  仮想ホストに結び付けられたパートナ・アプリケーションの URL を登録している場合、httpd.conf に仮想ホストを定義する必要があります。57 ページの「オプション: 仮想ホストの定義」を参照してください。
- -config_file は、osso.conf ファイルを生成するパスを示します。

オプション: 仮想ホストの定義

パートナ・アプリケーションの登録時に仮想ホスト URL を使用する場合、EPM System Web サーバーとして使用される Oracle HTTP Server の httpd.conf を更新することによって、仮想ホストを定義する必要があります。

➤ 仮想ホストを定義するには:

1 **テキスト・エディタを使用して、**EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/httpd.conf **を開きます。**

2 **次の記述に類似した定義を追加します。この定義は、Web サーバーが、仮想サーバー** epm.myCompany.com**、ポート** epm.myCompany.com:19400 **で実行されていることを前提としています。各自の要件に合うように設定を変更してください。**

```
     NameVirtualHost epm.myCompany.com:19400
Listen 19400
 <VirtualHost epm.myCompany.com:19400>
DocumentRoot "C:/Oracle/Middleware/user_projects/epmsystem1/httpConfig/ohs
    /config/OHS/ohs_component/private-docs"
   include "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}
    /${COMPONENT_NAME}/mod_osso.conf"
</VirtualHost>
```

mod_osso.conf の作成

EPM System Web サーバーをフロントエンドする Oracle HTTP Server 上に、mod_osso.conf を作成します。

➤ mod_osso.conf を作成するには:

1 テキスト・エディタを使用して、ファイルを作成します。

2 次のコンテンツをファイルにコピーして、使用する環境に合せてファイルを変更します。

```
     LoadModule osso_module C:/Oracle/Middleware/ohs/ohs/modules/mod_osso.so
<IfModule mod_osso.c>
 OssoIpCheck off
 OssoIdleTimeout off
 OssoSecureCookies off
 OssoConfigFile C:/Oracle/Middleware/user_projects/epmsystem1/httpConfig/
  ohs/config/OHS/ohs_component/osso/osso.conf
```

3 <IfModule mod_osso.c 定義内に、次の記述に類似した場所定義を含めて、OSSO を使用して保護する予定の各リソースを識別します。

```
      <Location /interop/>
  require valid user
  AuthType Osso
 </Location>
</IfModule>
```

4 mod_osso.conf という名前でファイルを保存します。

osso.conf の再配置

EPM System Web サーバーをパートナ・アプリケーションとして登録するプロセス(56 ページの「パートナ・アプリケーションとしての EPM System Web サーバーの登録」を参照)では、-config_file ディレクティブで識別された場所に、難読化された osso.conf を作成します。

➤ osso.conf を再配置するには:

1 EPM System Web サーバーをパートナ・アプリケーションとして登録(56 ページの「パートナ・アプリケーションとしての EPM System Web サーバーの登録」を参照)したときに作成された osso.conf を検索します。

2 mod_osso.conf(58 ページの「mod_osso.conf の作成」を参照)に定義された OssoConifgFile プロパティで識別されたディレクトリ(OSSO サーバーをフロントエンドする Oracle HTTP Server 上)に、osso.conf をコピーします。

Reporting and Analysis のキャッシュ管理構成の追加

Oracle HTTP Server の `httpd.conf` を編集して、Reporting and Analysis のキャッシュ管理構成設定を追加します。

➤ キャッシュ管理構成設定を追加するには:

1 テキスト・エディタを使用して、`EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/httpd.conf` を開きます。

2 Reporting and Analysis キャッシュ管理に関する次のディレクティブを追加します:

```
     <Location /WebAnalysis/>
OssoSendCacheHeaders off
</Location>
<Location /workspace/>
OssoSendCacheHeaders off
</Location>
<Location /hr/>
OssoSendCacheHeaders off
</Location>
<Location /HReports/>
OssoSendCacheHeaders off
</Location>
```

3 `httpd.conf` を保存して閉じます。

OSSO 用 EPM System の構成

OSSO ソリューションと統合される OID を、外部ユーザー・ディレクトリとして EPM System で構成し、SSO を使用可能にします。

➤ OSSO 用 EPM System を構成するには:

1 OSSO ソリューションで使用する OID を外部ユーザー・ディレクトリとして構成します。Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。

2 EPM System で SSO を使用可能にします。84 ページの「SSO 用の EPM System の構成」

注: OSSO をアイデンティティ管理ソリューションとして構成するには、「SSO プロバイダ/エージェント」で「その他」を選択し、「SSO メカニズム」で「カスタムHTTPヘッダー」を選択します。プロキシ・リモート・ユーザーをカスタム HTTP ヘッダーの名前として入力します。

3 少なくとも 1 つの OID ユーザーを Shared Services 管理者としてプロビジョニングします。

4 EPM System 製品、および Shared Services セキュリティ API を使用するカスタム・アプリケーションを再起動します。

注： Shared Services で構成済の OID が EPM System 製品を開始する前に必ず実行しているようにします。

オプション: OSSO サーバー上のデバッグ・メッセージの使用可能化

OSSO サーバー上のデバッグ・メッセージを記録するには、policy.properties を変更します。デバッグ・メッセージは ORACLE_HOME/sso/log/ssoServer.log に書き込まれます。

➤ デバッグ・メッセージを記録するには:

1. テキスト・エディタを使用して、OSSO サーバー上の ORACLE_HOME/sso/conf/policy.properties(C:\OraHome_1\sso\conf\policy.properties など)を開きます。
2. debugLevel プロパティの値を DEBUG に設定します。

```
debugLevel = DEBUG
```

3. policy.properties を保存して閉じます。

オプション: 保護されたリソースのデバッグ・メッセージの使用可能化

mod_osso.conf を使用して保護されたリソースの OSSO デバッグ・メッセージを記録するには、EPM System Web サーバー上の httpd.conf を変更します。デバッグ・メッセージは、EPM_ORACLE_INSTANCE/httpConfig/ohs/diagnostics/logs/OHS/ohs_component/ohs_component.log に書き込まれます。

➤ 保護されたリソースのデバッグ・メッセージを記録するには:

1. テキスト・エディタを使用して、EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/httpd.conf を開きます。
2. OraLogSeverity プロパティの値を TRACE に設定します。

```
OraLogSeverity TRACE:32
```

3. httpd.conf を保存して閉じます。

SSO 用の EPM System 製品の保護

サブトピック

ユーザーからの SSO 要求がセキュリティ・エージェント(OAM、OSSO または SiteMinder)にリダイレクトされるように、EPM System リソースを保護する必要があります。

Oracle HTTP サーバーでは、mod_osso を使用して OSSO サーバーにユーザーがリダイレクトされます。ユーザーは、要求する URL が保護される mod_osso で構成される場合にのみ、リダイレクトされます。『Oracle HTTP Server 管理者ガイド』のセキュリティの管理に関する項を参照してください。

SiteMinder SSO のリソース保護は、SiteMinder のドキュメントを参照してください。

保護するリソース

表 4 に、保護される必要のあるコンテキストをリストします。OSSO 用にリソースを保護する構文(例として interop を使用)は、次のとおりです:

```
   <Location /interop>
Require valid-user
AuthType Basic
order deny,allow
deny from all
allow from
   myServer.myCompany.com

satisfy any
</Location>
```

allow from パラメータでは、コンテキストの保護をバイパスできる開始サーバーを指定します。

EPM Workspace、Financial Reporting および Web Analysis の場合、次の例に示されたパラメータのみを設定する必要があります:

```
   <Location /workspace>
Require valid-user
AuthType Basic
</Location>
```

表 4　保護する EPM System リソース

EPM System 製品	保護するコンテキスト
Shared Services	/interop

EPM System 製品	保護するコンテキスト
Oracle Hyperion Reporting and Analysis Framework	• /raframework • /biplus_webservices
EPM Workspace	/workspace
Financial Reporting	/hr
Web Analysis	/WebAnalysis
Oracle Hyperion EPM Architect	/awb
Planning	/HyperionPlanning
Oracle Hyperion Performance Scorecard	• /HPSWebReports • /HPSAlerter
Oracle Hyperion Strategic Finance	/HSFWebServices
Oracle Integrated Operational Planning	/interlace
Financial Management	• /hfmadf • /hfmofficeprovider • /hfmsmartviewprovider
Data Relationship Management	/drm-web-client
Administration Services	/hbrlauncher
Oracle Hyperion Financial Data Quality Management	/HyperionFDM
Oracle Hyperion Calculation Manager	/calcmgr
Oracle Hyperion Provider Services	/aps
Oracle Hyperion Profitability and Cost Management	/profitability
Account Reconciliation Manager	/arm
Oracle Hyperion Financial Close Management	/fcc
Oracle Hyperion Disclosure Management [1]	/mappingtool
Oracle Hyperion Financial Data Quality Management Enterprise Edition	/aif

[1]SSL で保護された Web サービスによる Disclosure Management クライアントの使用をサポートするには、クライアント・マシンに(ルート証明書から始まる)証明書チェーン全体が必要です。

保護しないリソース

表 5 では、保護する必要のないコンテキストがリストされます。OSSO 用にリソースを保護しない構文(例として/interop/framework(.*)を使用)は、次のとおりです。

```
<LocationMatch /interop/framework(.*)>
```

```
 Require valid-user
 AuthType Basic
 allow from all
 satisfy any
</LocationMatch>
```

表 5　保護しない EPM System リソース

EPM System 製品	保護しないコンテキスト
Shared Services	● /interop/framework(.*) ● /interop/Audit(.*) ● /interop/taskflow* ● /interop/WorkflowEngine/* ● /interop/TaskReceiver ● /framework/lcm/HSSMigration
Performance Management Architect	● /awb/ces.executeAction.do ● /awb/lcm.executeAction.do ● /awb/appmanager.deployStatusUpdate.do ● /awb/jobstask.updateJobStatus.do ● /hyperion-bpma-server
EPM Workspace	/workspace/browse/listXML*
Planning	/HyperionPlanning/Smartview
Oracle Hyperion Reporting and Analysis Framework	● /raframework/browse/listXML ● /raframework/wsrp4j(.*) ● /raframework/ResourceProxy(.*)
Oracle Hyperion Web Analysis *	● /WebAnalysis/wsrp4j(.*) ● /WebAnalysis/ResourceProxy(.*)
Oracle Hyperion Financial Reporting *	● /hr/common/HRLogon.jsp ● /hr/wsrp4j(.*) ● /hr/ResourceProxy(.*) ● /hr/services/* ● /hr/modules/com/hyperion/reporting/web/reportViewer/HRStaticReport.jsp
Oracle Data Relationship Management	/drm-migration-client
Oracle Hyperion Calculation Manager	/calcmgr/common.performAction.do (Performance Management Architect 用)
Oracle Essbase Administration Services	● /eas ● /easconsole ● /easdocs

EPM System 製品	保護しないコンテキスト
Financial Management	● /hfm/EIE/EIEListener.asp ● /hfmapplicationservice ● /oracle-epm-fm-webservices ● /hfmlcmservice
Planning	● /HyperionPlanning/servlet/HspLCMServlet ● /HyperionPlanning/servlet/HspAppManagerServlet (Performance Management Architect 用)
Oracle Hyperion Performance Scorecard	● /HPSWebReports/wsrp4j(.*) ● /HPSWebReports/ResourceProxy(.*) ● /HPSWebReports/action/lcmCallBack
Performance Management Architect データの同期	/DataSync/services*
Oracle Hyperion Strategic Finance	● /HSFWebServices/HSFWebService.asmx ● /HSFWebServices/HSFEntityWebService.asmx
Oracle Integrated Operational Planning	● /interlace/services/(.*) ● /interlace/anteros/(.*) ● /interlace/interlace/(.*) ● /interlace/WebHelp/(.*) ● /interlace/html/(.*) ● /interlace/email-book/(.*)
Profitability and Cost Management	● /profitability/cesagent ● /profitability/lcm ● /profitability/control ● /profitability/ApplicationListener
Oracle Hyperion Financial Data Quality Management Enterprise Edition	● /aif/services/FDMRuleService ● /aif/services/RuleService
Oracle Hyperion Disclosure Management	● /discmanwebservices ● /mappingtool/MappingToolWS

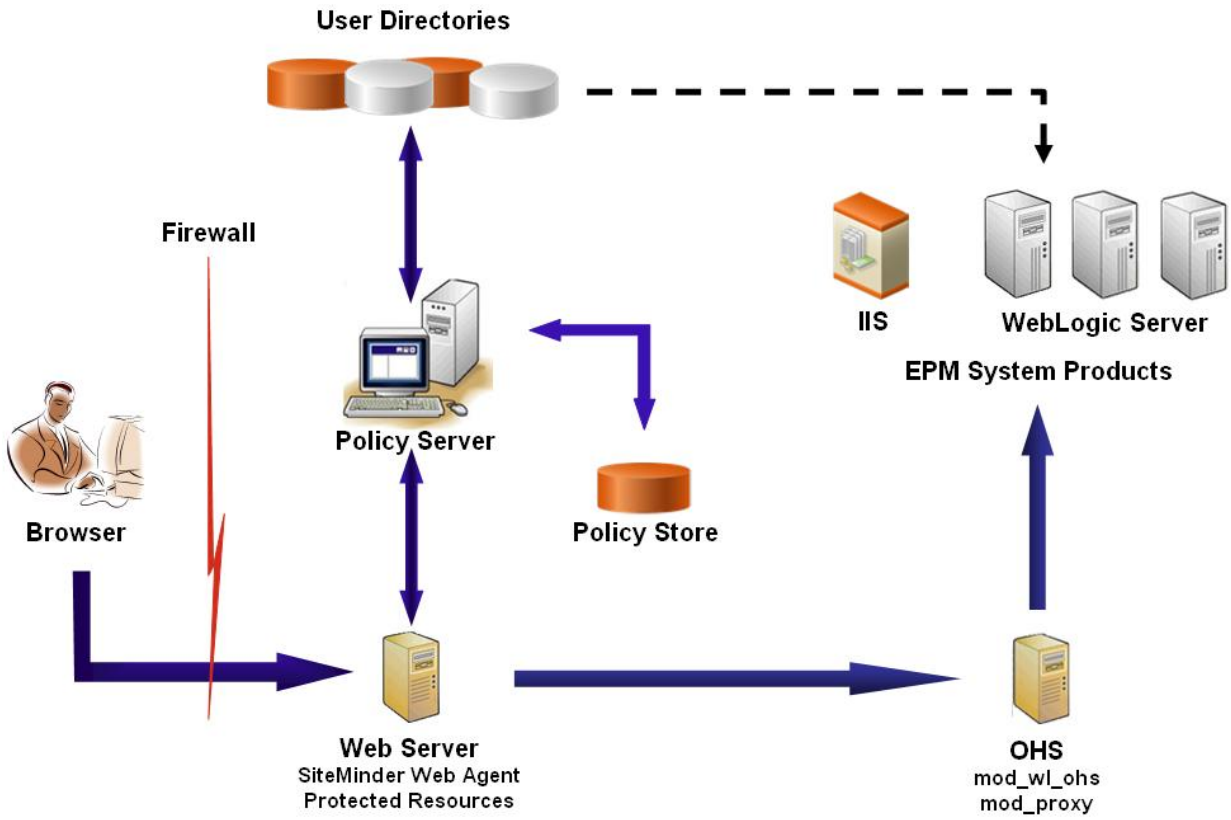
SiteMinder SSO

サブトピック

SiteMinder は Web 専用のソリューションです。デスクトップ・アプリケーションおよびそのアドイン(たとえば、Microsoft Excel や Report Designer)は、SiteMinder からの認証を使用できません。ただし、Oracle Hyperion Smart View for Office では、SiteMinder 認証を使用できます。

プロセス・フロー

SiteMinder 使用可能な SSO の概要を図で示します:



SiteMinder の SSO プロセス:

1. ユーザーは、SiteMinder で保護された EPM System リソースへのアクセスを試行します。SiteMinder ポリシー・サーバーをフロントエンドする Web サーバーに接続する URL(`http://`

`WebAgent_Web_Server_Name:WebAgent_Web_ServerPort/interop/index.jsp` など)を使用します。

2. Web サーバーは、ログイン情報を要求するポリシー・サーバーにユーザーをリダイレクトします。構成済ユーザー・ディレクトリに対するログイン情報の検証後、ポリシー・サーバーは、SiteMinder Web エージェントのホストとなる Web サーバーにログイン情報を渡します。
3. SiteMinder Web エージェントのホストとなる Web サーバーは、EPM System をフロントエンドする Oracle HTTP Server に要求をリダイレクトします。Oracle HTTP Server は、WebLogic Server または IIS Server 上に配置されている、要求されたアプリケーションにユーザーをリダイレクトします。
4. EPM System コンポーネントは、プロビジョニング情報を確認し、コンテンツを提供します。このプロセスが機能するには、SiteMinder でユーザーの認証に使用するユーザー・ディレクトリを、EPM System の外部ユーザー・ディレクトリとして構成する必要があります。これらのディレクトリは信頼済として構成する必要があります。

注意事項

SiteMinder は Web 専用のソリューションです。デスクトップ・アプリケーションおよびそのアドイン(たとえば、Microsoft Excel や Report Designer)は、SiteMinder からの認証を使用できません。ただし、Smart View では、SiteMinder 認証を使用できます。

前提条件

1. 完全な機能の SiteMinder インストールは、次のコンポーネントで構成されています:
   - ポリシーおよびエージェント・オブジェクトが定義された SiteMinder ポリシー・サーバー
   - SiteMinder ポリシー・サーバーをフロントエンドする Web サーバーにインストールされた SiteMinder Web エージェント
2. 完全な機能の EPM System 配置。

   EPM System コンポーネントに Web サーバーを構成する場合、EPM System コンフィグレータは Oracle HTTP Server に次を構成し、要求をアプリケーション・サーバーにプロキシします:
   - WebLogic Server に要求をプロキシする `mod_wl_ohs.conf`
   - IIS に要求をプロキシする `mod_proxy`

SiteMinder Web エージェントの使用可能化

Web エージェントは、EPM System リソースに対する要求をインターセプトする Web サーバー上にインストールされます。認証されていないユーザーが保護された EPM System リソースにアクセスしようとすると、Web エージェントではユー

ザーに対して SSO 認証情報を要求します。ユーザーが認証されると、ポリシー・サーバーは認証されたユーザーのログイン名を追加し、そのログイン名はヘッダーで渡されます。その後、HTTP 要求が EPM System Web サーバーに渡され、要求をリダイレクトします。EPM System コンポーネントはヘッダーから認証済ユーザー・ログイン情報を抽出します。

SiteMinder は、異種の Web サーバー・プラットフォーム上で実行されている EPM System 製品全体の SSO をサポートしています。EPM System 製品が異なる Web サーバーを使用する場合、SiteMinder Cookie を同じドメイン内の Web サーバーに確実に渡せるようにする必要があります。各 Web サーバーの `WebAgent.conf` ファイルの `Cookiedomain` の値として適切な EPM System アプリケーション・ドメインを指定して、これを行います。

Netegrity SiteMinder エージェント・ガイドの Web エージェントの構成に関する項を参照してください。

注： Shared Services はそのコンテンツを保護するために基本認証を使用するため、Shared Services への要求をインターセプトする Web サーバーは、SiteMinder を使用した SSO をサポートできるように基本認証を使用可能にする必要があります。

SiteMinder Web エージェント構成ウィザードを実行して、Web エージェントを構成します(これを行うには、`WEBAGENT_HOME/install_config_info/nete-wa-config` を実行します。たとえば、Windows の場合、`C:\netegrity\webagent\install_config_info\nete-wa-config.exe` になります)。構成プロセスでは、SiteMinder Web サーバーの `WebAgent.conf` が作成されます。

➤ SiteMinder Web エージェントを使用可能にするには:

1 **テキスト・エディタを使用して、`WebAgent.conf` を開きます。このファイルの場所は、使用している Web サーバーによって異なります。IIS Server を SiteMinder Web サーバーとして構成している場合、`WebAgent.conf` の場所は、`WEB_AGENT_HOME/bin/IIS`(`C:\SiteMinder\webagent\bin\iis\WebAgent.conf` など)になります。**

2 `enableWebAgent` **プロパティの値を**はい**に設定します。**

```
enableWebAgent=”YES”
```

3 **Web エージェント構成ファイルを保存して閉じます。**

SiteMinder ポリシー・サーバーの構成

SiteMinder 管理者は、EPM System 製品に SSO が使用可能になるようにポリシー・サーバーを構成する必要があります。

構成プロセスは次のとおりです:

- SiteMinder Web エージェントの作成、および SiteMinder Web サーバーに適した構成オブジェクトの追加

- 保護する必要がある各 EPM System リソースのレルムの作成、および Web エージェントのレルムへの追加。61 ページの「保護するリソース」を参照してください
- 保護する EPM System リソース用に作成されたレルム内で、保護しないリソース用のレルムを作成します。62 ページの「保護しないリソース」を参照してください
- HTTP ヘッダー参照の作成。ヘッダーは、EPM System アプリケーションにログイン属性の値を提供する必要があります。ログイン属性の簡単な説明については、Oracle Enterprise Performance Management System User Security Administration Guide の OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成に関する項を参照してください。
- Web エージェント・アクションとして、取得、ポストおよび配置を使用したレルム内のルールの作成
- 値が `hyplogin=<%userattr="SM_USERLOGINNAME"%>`のレスポンス属性の作成
- ポリシーの作成、ユーザー・ディレクトリ・アクセスの割当て、および EPM System 用に作成したルールの現在のメンバー・リストへの追加
- EPM System コンポーネント用に作成したルールに対する応答の設定

EPM System Web サーバーに要求を転送するための SiteMinder Web サーバーの構成

SiteMinder Web エージェントのホストとなる Web サーバーを構成して、認証済ユーザー(ユーザーを識別するヘッダーを含む)から EPM System Web サーバーに要求を転送します。

Apache ベースの Web サーバー用に、次の記述に類似したディレクティブを使用して、認証済要求を転送します:

```
   ProxyPass / http://
   EPM_WEB_SERVER
   :
   EPM_WEB_SERVER_PORT
   /
ProxyPassReverse / http://
   EPM_WEB_SERVER
   :
   EPM_WEB_SERVER_PORT
   /
ProxyPreserveHost On
#If SiteMinder Web Server is using HTTPS but EPM Web Server is using HTTP
RequestHeader set WL-Proxy-SSL true
```

このディレクティブで、`EPM_WEB_SERVER` および `EPM_WEB_SERVER_PORT` を、各自の環境の実際の値に置き換えます。

EPM System で SiteMinder を使用可能にする

SiteMinder との統合により、EPM System 製品の SiteMinder 認証を使用可能にする必要があります。84 ページの「SSO 用の EPM System の構成」を参照してください。

Kerberos シングル・サインオン

サブトピック

概要

EPM System 製品は、EPM System 製品をホストするアプリケーション・サーバーが Kerberos 認証用に設定されている場合は、Kerberos SSO をサポートします。

Kerberos は信頼できる認証サービスで、各 Kerberos クライアントは他の Kerberos クライアント(ユーザー、ネットワーク・サービスなど)の ID を有効なものとして信頼します。

EPM System 製品にユーザーがアクセスする場合に行われる処理は、次のとおりです:

- Windows コンピュータで、ユーザーが Kerberos レルムにログインします。
- 統合 Windows 認証を使用するように構成されているブラウザを使用して、ユーザーはアプリケーション・サーバー上で実行されている EPM System 製品にログインします。
- アプリケーション・サーバー(ネゴシエート ID アサーション・プロバイダ)は要求をインターセプトし、ブラウザの認証ヘッダーから Kerberos 情報とともに Simple and Protected Generic Security Services API (GSSAPI) Negotiation Mechanism (SPNEGO)トークンを取得します。
- アサーション・プロバイダは、EPM System 製品にユーザーに関する情報を渡すために、その ID ストアに対してトークンに含まれるユーザーの ID の妥当性を確認します。EPM System 製品は Active Directory に対してユーザー名を検証します。EPM System 製品は、すべての EPM System 製品間で SSO をサポートする SSO トークンを発行します。

サポート制約事項

Kerberos SSO は、すべての EPM System 製品に対してサポートされていますが、次の例外があります:

- Kerberos SSO は、Smart View 以外のシック・クライアントに対してサポートされていません。
- Smart View は、Oracle Essbase、Planning および Financial Management プロバイダに対してのみ Kerberos 統合をサポートします
- IIS が埋め込まれた EPM System 製品の Kerberos SSO サポート(Financial Management など)は、EPM Workspace を介してのみ使用可能です。Oracle Hyperion Financial Data Quality Management への SSO アクセスは、Financial Management を介して提供されます。

前提

このドキュメントにはアプリケーション・レベルの Kerberos 構成手順が記載されていますが、システム・レベルでの Kerberos 構成に関する知識があることを前提としています。このドキュメントに記載されている手順を開始する前に、次のタスクの前提条件が満たされていることを確認してください。

このドキュメントでは、Windows クライアント・マシンが Kerberos 認証用に構成されている、フル機能の Kerberos 対応ネットワーク環境で作業していることを前提としています。

- 企業の Active Directory が Kerberos 認証用に構成されています。Microsoft Windows Server のドキュメントを参照してください
- EPM System 製品へのアクセスに使用されるブラウザは、Kerberos チケットを使用してネゴシエートするように構成されています。
  - Firefox: https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/sso-config-firefox.html
  - Internet Explorer: http://docs.oracle.com/cd/E12839_01/web.1111/e13707/sso.htm#i1102444
- KDC とクライアント・マシン間で、時間同期の誤差は 5 分以内です。http://technet.microsoft.com/en-us/library/cc780011(WS.10).aspx の「Authentication Errors are Caused by Unsynchronized Clocks」を参照してください。
- Internet Information System (IIS)が EPM System 製品の Web サーバーとして使用される場合、IIS での統合 Windows 認証は無効になります。

WebLogic Server を使用した Kerberos SSO

WebLogic Server Kerberos SSO は、Microsoft クライアントを使用した SSO が使用可能になるように、SPNEGO トークンをネゴシエートおよびデコードするためネゴシエート ID アサーション・プロバイダを使用します。WebLogic Server は Kerberos チケットを取得するために SPNEGO トークンをデコードし、そのチケットを検証して WebLogic Server ユーザーにマップします。WebLogic Server の Active Directory 認証プロバイダは、WebLogic Server ユーザーのユーザー・ディレクトリとして Active Directory を構成するためにネゴシエート ID アサーション・プロバイダとともに使用できます。

ブラウザが EPM System 製品へのアクセスを要求する場合、KDC はそのブラウザに Kerberos チケットを発行し、それによって、サポートされる GSS トークン・タイプを含む SPNEGO トークンが作成されます。ネゴシエート ID アサーション・プロバイダは SPNEGO トークンをデコードし、GSSAPI を使用して、セキュリティ・コンテキストを受け入れます。要求を開始したユーザーの ID はユーザー名にマップされ、WebLogic Server に渡されます。また、WebLogic Serve は、ユーザーが属するグループを決定します。この段階で、要求された EPM System 製品はユーザーに使用できるようになります。

注： ユーザーは SPNEGO をサポートするブラウザ(たとえば、Internet Explorer や Firefox など)を使用して、WebLogic Server で実行している EPM System 製品にアクセスできます。WebLogic Server は UNIX または Windows プラットフォームで実行できます。

認証プロセスから取得されたユーザー ID を使用して、EPM System 製品認証プロセスはプロビジョニング・データをチェックします。EPM System 製品へのアクセスは、プロビジョニング・データに基づいて制限されます。

『Oracle Fusion Middleware Oracle WebLogic Server の保護』の Microsoft クライアントでのシングル・サインオンの構成に関する項を参照してください。

Kerberos 認証をサポートするための WebLogic Server での手順

Kerberos 認証をサポートするには、管理者は次のタスクを完了する必要があります:

- EPM System の WebLogic ドメインを作成します。72 ページの「EPM System の WebLogic ドメインの作成」を参照してください。
- 認証プロバイダを作成します。72 ページの「WebLogic Server での LDAP 認証プロバイダの作成」を参照してください。
- ネゴシエート ID アサーション・プロバイダを作成します。72 ページの「ネゴシエート ID アサーション・プロバイダの作成」を参照してください。
- Kerberos 識別を作成します。73 ページの「WebLogic Server 用の Kerberos 識別の作成」を参照してください。
- WebLogic 起動スクリプトを更新します。75 ページの「WebLogic 起動スクリプトの更新」を参照してください。
- 認可ポリシーを構成します。75 ページの「認可ポリシーの構成」を参照してください。
- SSODiag を配置して使用し、WebLogic Server が EPM System に対して Kerberos SSO をサポートする準備が整っているかを確認します。76 ページの「SSODiag を使用した Kerberos 環境のテスト」を参照してください。

EPM System の WebLogic ドメインの作成

通常、EPM System コンポーネントは、EPMSystem WebLogic ドメインに配置されます。

➤ Kerberos 認証用に EPM System WebLogic ドメインを構成するには:

1 **EPM System コンポーネントをインストールします。**

2 **Foundation Services のみを配置します。**

Foundation Services の配置により、デフォルトの EPM System WebLogic ドメインが作成されます。

3 **Shared Services Console にログインして Foundation Services の配置が成功したことを確認します。****20 ページの「Shared Services Console の起動」を参照してください。**

WebLogic Server での LDAP 認証プロバイダの作成

WebLogic Server 管理者は、LDAP 認証プロバイダを作成して、ユーザー情報およびグループ情報を外部 LDAP サーバーに格納します。LDAP v2-または v3-に準拠した LDAP サーバーは、WebLogic Server と連携して機能します。次のリファレンス・ソースを参照してください:

- 『Oracle Fusion Middleware Oracle WebLogic Server の保護』の LDAP 認証プロバイダの構成に関する項。
- Oracle Fusion Middleware Oracle WebLogic Server Administration Console オンライン・ヘルプの認証および ID アサーション・プロバイダの構成に関する項

ネゴシエート ID アサーション・プロバイダの作成

ネゴシエート ID アサーション・プロバイダは、Microsoft クライアントによる SSO の使用を可能にします。SPNEGO トークンをデコードして Kerberos トークンを取得し、Kerberos トークンを検証してトークンを WebLogic ユーザーにマップします。ネゴシエート ID アサーション・プロバイダは、WebLogic Security フレームワークで定義されているように Security Service Provider Interface (SSPI)の実装で、クライアントの SPNEGO トークンに基づいたクライアントの認証に必要なロジックを提供します。

- 『Oracle Fusion Middleware Oracle WebLogic Server の保護』の>ネゴシエート ID アサーション・プロバイダの構成に関する項
- Oracle Fusion Middleware Oracle WebLogic Server Administration Console オンライン・ヘルプの認証および ID アサーション・プロバイダの構成に関する項

ネゴシエート ID アサーション・プロバイダの作成時、すべての認証プロバイダに対して JAAS 制御フラグ・オプションを OPTIONAL に設定します。Oracle Fusion Middleware Oracle WebLogic Server 管理コンソール・オンライン・ヘルプの JAAS 制御フラグの設定に関する項を参照してください。

WebLogic Server 用の Kerberos 識別の作成

Active Directory ドメイン・コントローラ・マシンで、WebLogic Server および EPM System Web サーバーを表すユーザー・オブジェクトを作成し、Kerberos レルムの WebLogic Server および Web サーバーを表すサービス・プリンシパル名(SPN)にマップします。クライアントでは、SPN がないサービスを検索できません。SPN は、ログイン・プロセスで使用する WebLogic Server ドメインにコピーする Keytab ファイルに格納します。

手順の詳細は、『Oracle Fusion Middleware Oracle WebLogic Server の保護』の WebLogic Server 用の Kerberos 識別の作成に関する項を参照してください。

➤ WebLogic Server 用の Kerberos 識別を作成するには:

1 **Active Directory ドメイン・コントローラ・マシンで、WebLogic Server ドメインおよび EPM System コンポーネントで使用される IIS をホストするコンピュータについて、ユーザー・アカウント(`epmHost` など)を作成します。**

注: マシンではなく、ユーザー・オブジェクトとして識別を作成します。

コンピュータの簡易名を使用します。たとえば、ホスト名が `epmHost.example.com` の場合、`epmHost` を使用します。

ユーザー・オブジェクトの作成時に使用したパスワードを書き留めます。これは、SPN の作成に必要です。

パスワード・オプション、特に「ユーザーは次回ログオン時にパスワードの変更が必要」オプションを選択しないでください。

2 **Kerberos プロトコルに準拠するようにユーザー・オブジェクトを変更します。オブジェクトの暗号化タイプは DES にする必要があり、アカウントは Kerberos 事前認証を必要とします。**

- **「アカウント」**タブで、「このアカウントに`DES`暗号化を使う」チェック・ボックスを選択します。
- 他のアカウント・オプション(特に「`Kerberos`事前認証を必要としない」)が選択されていないことを確認します。
- 暗号化タイプを設定すると、オブジェクトのパスワードが破損する可能性があるため、パスワードをオブジェクトの作成時に設定したパスワードにリセットします。

3 **Active Directory ドメイン・コントローラをホストするコンピュータで、コマンド・プロンプト・ウィンドウを開き、Active Directory サポート・ツールがインストールされているディレクトリに移動します。**

4 **必要な SPN を作成して構成します。**

1. setspn ユーティリティを使用して、手順 1 で作成したユーザー・アカウント(`epmHost`)の SPN を作成します。

   たとえば、次のコマンドを使用します:

```
setspn -a host/epmHost.example.com
epmHost
```

```
setspn -a HTTP/epmHost.example.com
epmHost
```

2. 次のようなコマンドを使用して、手順 1 で作成したユーザー・オブジェクト(epmHost)に SPN が関連付けられていることを確認します。

```
setspn -L
epmHost
```

3. 次のようなコマンドを使用して、Active Directory ドメイン・サービス(AD DS)で WebLogic Server の SPN を構成し、共有秘密鍵を含む keytab ファイルを生成します。

```
ktpass -princ HTTP/
epmHost.example.com
@
epmHost.example.com
 -pass
password
 -mapuser epmHost -out c:\epmHost.keytab
```

これも同様

5 WebLogic Server をホストするコンピュータで keytab ファイルを作成します。

1. コマンド・プロンプトを開きます。
2. MIDDLEWARE_HOME/jdk160_29/bin に移動します。
3. 次のようなコマンドを実行します:

```
ktab -k
keytab_filename
 -a epmHost@example.com
```

4. パスワードの入力を求められたら、手順 1 でユーザーの作成時に設定したパスワードを入力します。

6 WebLogic ドメイン内の起動ディレクトリ(C:\Oracle\Middleware\user_projects\domains\EPMSystem など)に keytab ファイルをコピーします。

7 Kerberos 認証が正しく機能していることを確認します。

```
kinit -k -t keytab-file account-name
```

このコマンドからの出力は次のようになります:

```
New ticket is stored in cache file C:\Documents and Settings\Username
\krb5cc_MachineB
```

WebLogic 起動スクリプトの更新

『Oracle Fusion Middleware Oracle WebLogic Server の保護 11g リリース 1 (10.3.1)』の WebLogic Server での Kerberos 認証における起動引数の使用に関する項および JAAS ログイン・ファイルの作成に関する項を参照してください。

EPM System 管理対象サーバーが Windows のサービスとして稼働している場合、Windows レジストリを更新して、JVM 起動オプションを設定します。

➤ Windows レジストリで JVM 起動オプションを更新するには:

1 **Windows レジストリ・エディタを開きます。**

2 **「マイ コンピュータ」、「HKEY_LOCAL_MACHINE」、「Software」、「Hyperion Solutions」、「EPMServer0」、「HyS9EPMServer_epmsystem1」の順に選択します。**

3 **次の文字列値を作成します:**

注： 表 6 に示されている名前は例です。

表 6 Kerberos 認証用の JVM 起動オプション

名前	タイプ	データ
JVMOption44	REG_SZ	`-Djava.security.krb5.realm=Active Directory Realm Name`
JVMOption45	REG_SZ	`-Djava.security.krb5.kdc=Active Directory host name or IP address`
JVMOption46	REG_SZ	`-Djava.security.auth.login.config=location of Kerberos login configuration file`
JVMOption47	REG_SZ	`-Djavax.security.auth.useSubjectCredsOnly=false`

4 **追加した JVMOption を反映するように JVMOptionCount DWord の値を更新します(現在の 10 進数値に 4 を加えます)。**

認可ポリシーの構成

EPM System にアクセスする Active Directory ユーザー用の認可ポリシーの構成の詳細は、Oracle Fusion Middleware Oracle WebLogic Server ロールおよびポリシーによるリソースの保護の Web アプリケーションと EJB リソースの保護オプションに関する項を参照してください。

ポリシーの構成手順の例については、78 ページの「SSODiag 用のポリシーの作成」を参照してください。

SSODiag を使用した Kerberos 環境のテスト

サブトピック

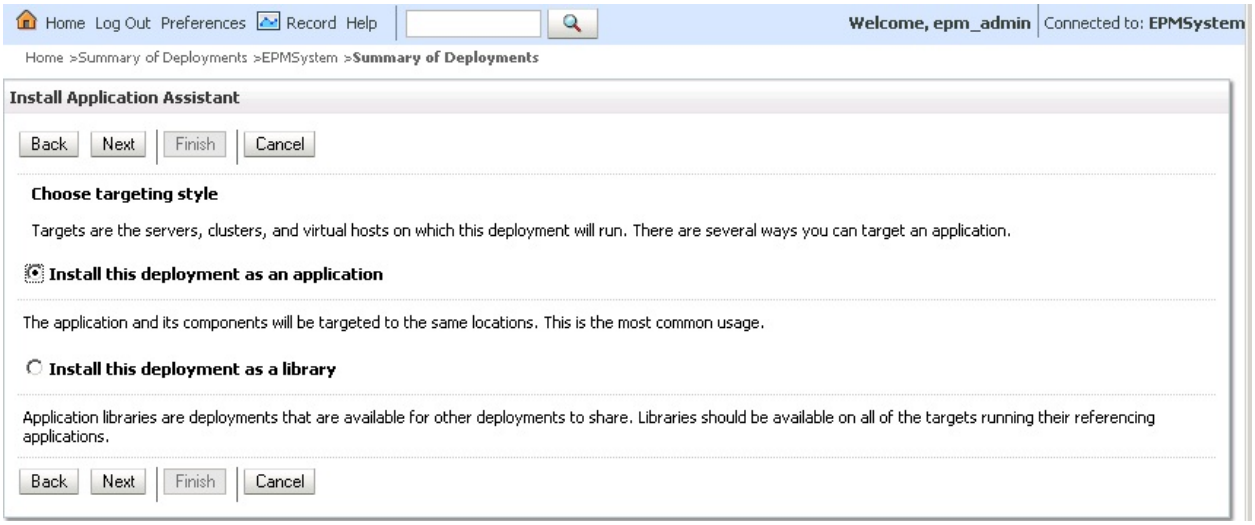
SSODiag は、Kerberos 環境で WebLogic Server が EPM System をサポートする準備が整っているかをテストする診断 Web アプリケーションです。

SSODiag の配置

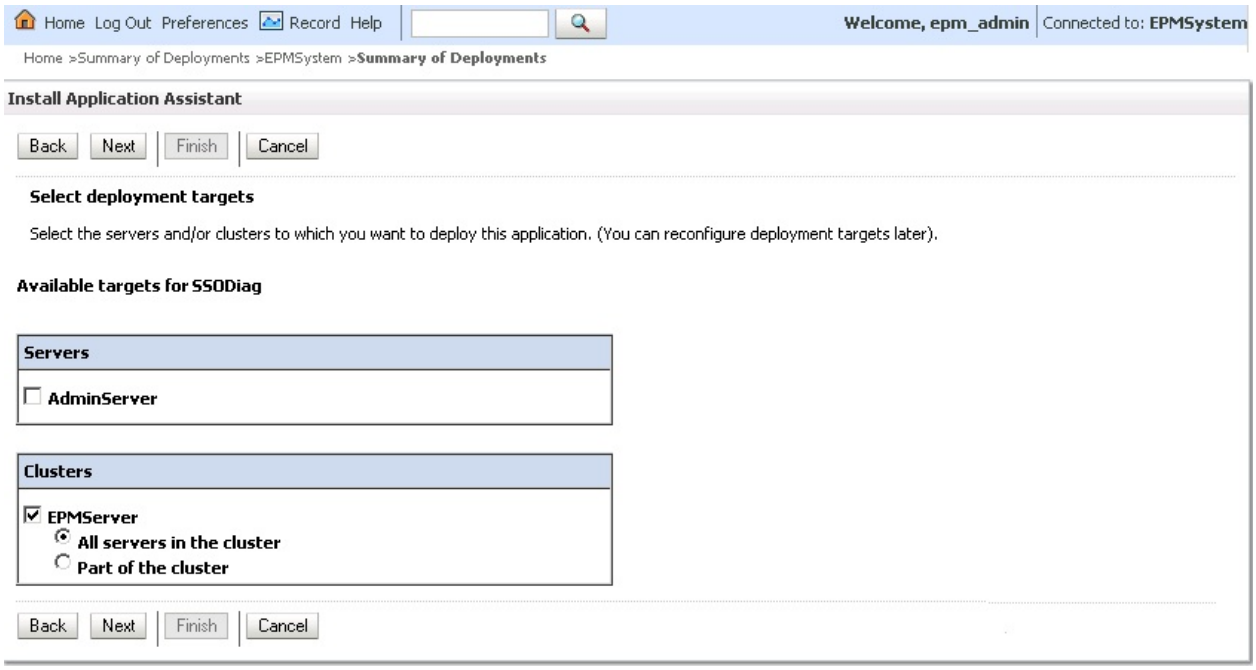
Foundation Services の配置時に指定した WebLogic Server 管理者のログイン情報(デフォルトのユーザー名は `epm_admin`)を使用して、SSODiag を配置します。

➤ SSODiag を配置して構成するには:

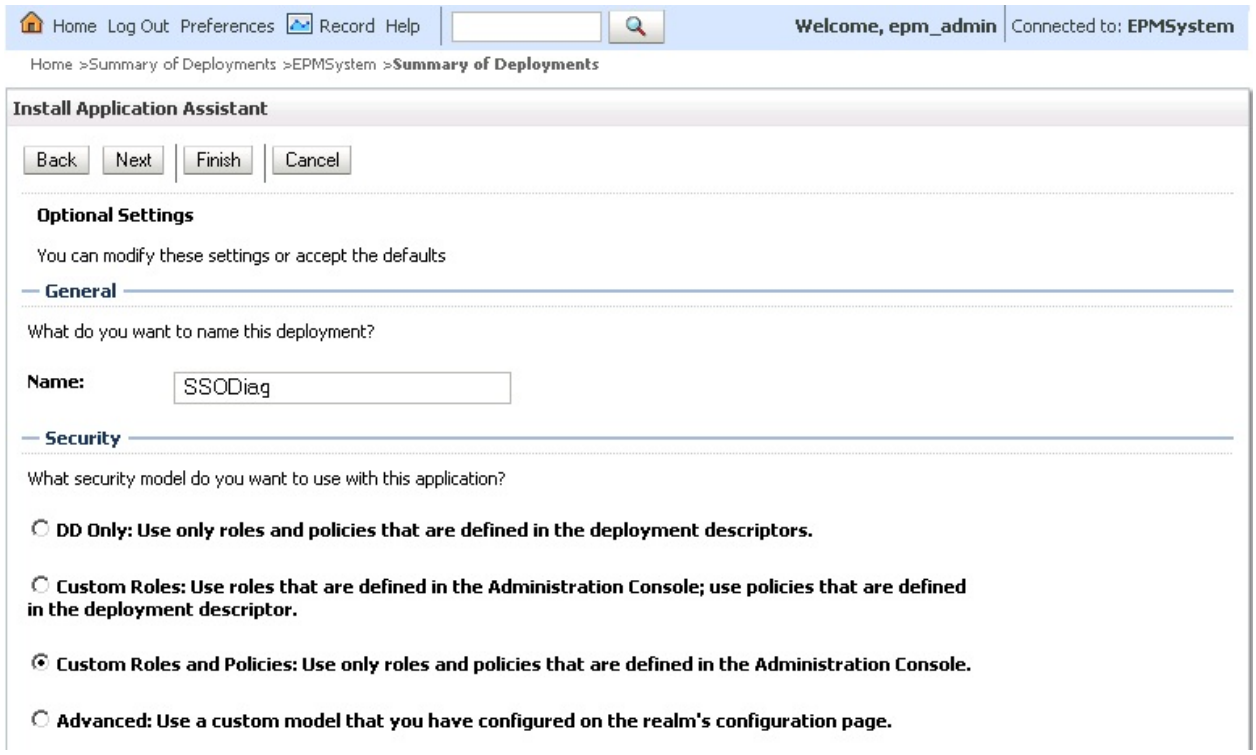
1. **EPM System ドメインに対する WebLogic Server 管理コンソールにログオンします。**
2. チェンジ・センターで、「ロックして編集」をクリックします。
3. 「ドメイン構造」の「EPMSystem」から、「デプロイメント」をクリックします。
4. 「デプロイメント」の「サマリー」で、「インストール」をクリックします。
5. 「パス」で、`EPM_ORACLE_HOME/products/Foundation/AppServer/InstallableApps/common/SSODiag.war` を選択します。
6. 「次へ」をクリックします。
7. 「ターゲット指定スタイルの選択」で、「このデプロイメントをアプリケーションとしてインストールする」が選択されていることを確認し、「次へ」をクリックします。



8. 「デプロイ・ターゲットの選択」で、次を選択し、「次へ」をクリックします。
   - 「EPMServer」
   - 「クラスタのすべてのサーバー」



9 「オプション設定」で、「カスタム・ロールおよびポリシー: 管理コンソール内に定義されたロールとポリシーのみを使用します。」をセキュリティ・モデルとして選択します。



10 「次へ」をクリックします。

11 確認画面で、「いいえ、後で構成を確認します。」を選択します。

12 「終了」をクリックします。

13 チェンジ・センターで、「変更のアクティブ化」を選択します。

SSODiag 用の Oracle HTTP Server の構成

mod_wl_ohs.conf を更新して、SSODiag URL 要求を WebLogic Server に転送するよう Oracle HTTP Server を構成します。

➤ Oracle HTTP Server で URL 転送を構成するには:

1 テキスト・エディタを使用して、EPM_ORACLE_INSTANCE/httpConfig/ohs/config/OHS/ohs_component/mod_wl_ohs.conf を開きます。

2 SSODiag の LocationMatch 定義を追加します:

```
    <LocationMatch /SSODiag/>
  SetHandler weblogic-handler
  WeblogicCluster myServer:28080
</LocationMatch>
```

前述の例で、myServer は Foundation Services ホスト・マシンを表し、28080 は Shared Services が要求をリスニングするポートを表します。

3 mod_wl_ohs.conf を保存して閉じます。

4 Oracle HTTP Server を再起動します。

SSODiag 用のポリシーの作成

WebLogic Server 管理コンソールでポリシーを作成し、次の SSODiag URL を保護します。

```
http://
OHS_HOST_NAME
:
PORT
/SSODiag/krbssodiag
```

この例では、OHS_HOST_NAME は Oracle HTTP Server をホストするサーバーの名前を表し、PORT は Oracle HTTP Server が要求をリスニングするポートを表します。

➤ SSODiag を保護するポリシーを作成するには:

1 WebLogic Server 管理コンソールのチェンジ・センターで、EPM System に対して「ロックして編集」を選択します。

2 「デプロイメント」、「SSODiag」、「ロール」、「ポリシー」の順に選択します。

3 次の URL パターンを作成します:

- /
- /index.jsp

4 作成した各 URL パターンを変更します:

1. 「スタンドアロン Web アプリケーションの URL パターン」の URL パターンのリストから、作成したパターン(/)をクリックして開きます。

2. 「条件の追加」を選択します。
3. 「述部リスト」から「ユーザー」を選択します。
4. 「次へ」を選択します。
5. 「ユーザー引数名」で、アカウントが Kerberos 認証用に構成されているクライアント・デスクトップへのアクセスに使用される Active Directory ユーザー(krbuser1 など)を入力し、「追加」を選択します。
6. 「終了」を選択します。

5 「保存」を選択します。

SSODiag を使用した Kerberos 認証用の WebLogic Server 構成のテスト

Kerberos 認証用の WebLogic Server 構成が正しく機能する場合、Oracle Hyperion Kerberos SSO 診断ユーティリティ V 1.0 に次のメッセージが表示されます:

```
   Retrieving Kerberos User principal name... Success.
Kerberos principal name retrieved...
   SOME_USER_NAME
```

注意 SSODiag が Kerberos プリンシパル名を取得できない場合、Kerberos 認証用に EPM System コンポーネントを構成しないでください。

➤ Kerberos 認証用の WebLogic Server 構成をテストするには:

1 Foundation Services と Oracle HTTP Server を起動します。

2 WebLogic Server 管理コンソールを使用して、すべての要求を処理する SSODiag Web アプリケーションを起動します。

3 有効な Active Directory ログイン情報を使用して、Kerberos 認証用に構成されているクライアント・マシンにログオンします。

4 ブラウザを使用して、次の SSODiag URL に接続します:

```
http://
OHS_HOST_NAME
:
PORT
/SSODiag/krbssodiag
```

この例では、OHS_HOST_NAME は Oracle HTTP Server をホストするサーバーの名前を表し、PORT は Oracle HTTP Server が要求をリスニングするポートを表します。

Kerberos 認証が適切に機能する場合、SSODiag は次の情報を表示します:

```
     Retrieving Kerberos User principal name... Success.
Kerberos principal name retrieved...
```

```
SOME_USER_NAME
```

Kerberos 認証が適切に機能しない場合、SSODiag は次の情報を表示します:

```
Retrieving Kerberos User principal name... failed.
```

EPM System コンポーネントの構成

EPM System コンフィグレータを使用して、Foundation Services が配置されている WebLogic ドメインに他の EPM System コンポーネントを構成および配置します。

Kerberos 認証用の EPM System 管理対象サーバーの構成

Microsoft Windows 環境では、EPM System 管理対象サーバーは Windows サービスとして実行されます。WebLogic 管理対象サーバーごとに JVM 起動オプションを変更する必要があります。次に、非コンパクト配置モードの管理対象サーバーの包括的なリストを示します。

- AnalyticProviderServices0
- CalcMgr0
- DisclosureManagement0
- EpmaDataSync0
- EpmaWebReports0
- ErpIntegrator0
- EssbaseAdminServer0
- FinancialReporting0
- FMWebServices0
- FoundationServices0
- HpsAlerter0
- HpsWebReports0
- hsfweb0
- Planning0
- Profitability0
- RaFramework0
- WebAnalysis0

Web アプリケーションがコンパクト配置モードで配置されている場合は、EPMSystem0 管理対象サーバーの JVM 起動オプションのみを更新する必要があります。複数のコンパクト管理対象サーバーがある場合、すべての管理対象サーバーについて JVM 起動オプションを更新する必要があります。

『Oracle Fusion Middleware Oracle WebLogic Server の保護』の WebLogic Server での Kerberos 認証における起動引数の使用に関する項を参照してください。

注： 次の手順は、FoundationServices 管理対象サーバーの JVM 起動オプションを設定する方法を示しています。このタスクは、配置内の WebLogic 管理対象サーバーごとに実行する必要があります。

➤ WebLogic Server 起動スクリプトで JVM オプションを構成する手順

1 **管理対象サーバーをホストする Windows サーバーで、Windows レジストリ・エディタを起動します。**

2 `HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\EPMServer0\HyS9EPMServer_epmsystem1` **に移動します。**

3 **次の文字列値を作成します:**

注： 表 6 に示されている名前は例です。

表 7 Kerberos 認証用の JVM 起動オプション

名前	タイプ	データ
JVMOption44	REG_SZ	`-Djava.security.krb5.realm=Active Directory Realm Name`
JVMOption45	REG_SZ	`-Djava.security.krb5.kdc=Active Directory host name or IP address`
JVMOption46	REG_SZ	`-Djava.security.auth.login.config=location of Kerberos login configuration file`
JVMOption47	REG_SZ	`-Djavax.security.auth.useSubjectCredsOnly=false`
JVMOption48	REG_SZ	`-Djavax.security.enableNegotiate=true`

4 **追加した JVMOption を反映するように JVMOptionCount DWord の値を更新します(現在の 10 進数値に 5 を加えます)。**

認可ポリシーの構成

Foundation Services 以外の EPM System コンポーネントにアクセスする Active Directory ユーザー用の認可ポリシーを構成します。WebLogic 管理コンソールからセキュリティ・ポリシーを構成する手順は、75 ページの「認可ポリシーの構成」を参照してください。

EPM System コンポーネントのデフォルトのセキュリティ・モデルの変更

EPM System 構成ファイルを編集して、デフォルトのセキュリティ・モデルを変更します。非コンパクト EPM System 配置の場合、`config.xml` に記載されている各 EPM System Web アプリケーションのデフォルトのセキュリティ・モデルを変更す

る必要があります。次に、EPM System Web アプリケーションのリストを示します。

- AIF
- APS
- CALC
- DISCLOSUREMANAGEMENT
- EAS
- EPMADATASYNCHRONIZER
- EPMAWEBTIER
- FINANCIALREPORTING
- HPSAlerter
- HPSWebReports
- HSFWEB
- PLANNING
- PROFITABILITY
- RAFRAMEWORK
- SHAREDSERVICES
- WEBANALYSIS
- WORKSPACE

➤ セキュリティ・モデルを変更するには:

1 テキスト・エディタを使用して、`MIDDLEWARE_HOME/user_projects/domains/EPMSystem/config/config.xml` を開きます。

2 各 EPM System コンポーネントの app-deployment 定義で、次の例に示すように、`<security-dd-model>`の値を `CustomRolesAndPolicies` に設定します:

```
     <app-deployment>
  <name>SHAREDSERVICES#11.1.2.0</name>
  <target>EPMServer</target>
  <module-type>ear</module-type>
  <source-path>C:\Oracle\Middleware\EPMSystem11R1/products/Foundation/AppServer/
InstallableApps/common/interop.ear</source-path>
  <security-dd-model>CustomRolesAndPolicies</security-dd-model>
  <staging-mode>nostage</staging-mode>
</app-deployment>
```

3 `config.xml` を保存して閉じます。

4 WebLogic Server を再起動します。

EPM System コンポーネントの URL 保護ポリシーの作成

各 EPM System コンポーネントの URL を保護するには、WebLogic Server 管理コンソールで URL 保護ポリシーを作成します。詳細は、『Oracle Fusion Middleware Oracle WebLogic Server ロールおよびポリシーによるリソースの保護』の「Web アプリケーションおよび EJB リソースの保護のオプション」を参照してください。

➤ URL 保護ポリシーを作成するには:

1 **EPM System ドメインに対する WebLogic Server 管理コンソールのチェンジ・センターで、「ロックして編集」をクリックします。**

2 **「デプロイメント」をクリックします。**

3 **配置内の EPM System エンタープライズ・アプリケーション(`PLANNING` など)を展開し、その Web アプリケーション(`HyperionPlanning` など)をクリックします。EPM System コンポーネントのリストは、81 ページの「EPM System コンポーネントのデフォルトのセキュリティ・モデルの変更」を参照してください。**

   注 : 一部のエンタープライズ・アプリケーション(EAS など)は、URL パターンの定義が必要な複数の Web アプリケーションから構成されます。

4 **Web アプリケーションの URL パターン・スコープのポリシーを作成します。**

   1. 「セキュリティ」、「ポリシー」、「新規」の順にクリックします。
   2. 「URL パターン」に、`/*`を入力します。
   3. 「OK」をクリックします。
   4. 作成した URL パターン(`/*`)をクリックします。
   5. 「条件の追加」をクリックします。
   6. 「述部リスト」で、ポリシー条件を選択して「次へ」をクリックします。

      指定したグループのすべてのメンバーにこのセキュリティ・ポリシーを付与する「グループ」条件を使用することをお薦めします。
   7. 選択した述部に関連する引数を指定します。たとえば、前の手順で「グループ」を選択した場合、次の手順を実行する必要があります:
      1. 「グループ引数名」に、Web アプリケーションへのアクセスを許可するユーザーを含むグループの名前を入力します。入力する名前は、Active Directory グループ名と完全一致する必要があります。
      2. 「追加」をクリックします。
      3. さらにグループを追加するには、前述の手順を繰り返します。
   8. 「終了」をクリックします。

      Active Directory でグループが見つからない場合は、WebLogic Server にエラー・メッセージが表示されます。続行する前に、このエラーを解決する必要があります。
   9. 「保存」を選択します。

5 **配置内の他の EPM System コンポーネントについて、手順 3 および手順 4 を繰り返します。**

6 チェンジ・センターで、「構成の解放」をクリックします。

7 WebLogic Server を再起動します。

EPM System セキュリティ構成の更新

SSO を順守するように EPM System セキュリティを構成します。84 ページの「SSO 用の EPM System の構成」を参照してください。

Kerberos 用の IIS サーバーの構成

アプリケーション・サーバーとして IIS を使用する EPM System コンポーネント (Financial Management など)を使用している場合は、この項を実行します。

➤ Kerberos 用に EPM System IIS サーバーを構成するには:

1 IIS マネージャを起動します。

2 「Web サイト」、「既定の Web サイト」の順に展開します。

3 EPM System コンポーネントの Web サイト(Financial Management の `hfm` など)を右クリックし、「プロパティ」を選択します。

4 「ディレクトリ セキュリティ」タブで、「認証とアクセス制御」の「編集」をクリックします。

5 「認証方法」で、「統合 Windows 認証」を選択します。

6 「アプリケーション プール」を展開します。

7 前述の手順で認証およびアクセス制御方法を変更した EPM System コンポーネントのアプリケーション・プールを右クリックし、「プロパティ」をクリックします。たとえば、`hfm` Web サイトの認証およびアクセス制御方法を変更した場合、`hfmAppPool` を右クリックして「プロパティ」を選択します。

8 「アプリケーション プール ddentity」の「識別」タブで、次の手順を実行します。

   1. 「ユーザー名」に、作成したサービス・プリンシパルを入力します(73 ページの「WebLogic Server 用の Kerberos 識別の作成」を参照)。
   2. 「パスワード」に、サービス・プリンシパルのパスワードを入力します。
   3. 「OK」をクリックします。

9 手順 3 から手順 8 を繰り返し、残りの Web サイトおよびアプリケーション・プールについて Kerberos 認証を構成します。

10 IIS を再起動します。

SSO 用の EPM System の構成

EPM System 製品は、SSO 用にセキュリティ・エージェントをサポートするために構成する必要があります。Shared Services で指定した構成により、すべての EPM System 製品に次のことが決定されます。

- セキュリティ・エージェントから SSO を受け入れるかどうか

- SSO を受け入れる認証メカニズム

SSO を使用可能な環境において、ユーザーが最初にアクセスする EPM System 製品では、SSO メカニズムが分析され、ここに含まれている認証済ユーザー ID が取得されます。EPM System 製品では、Shared Services で構成されたユーザー・ディレクトリに対してユーザー ID がチェックされ、ユーザーが有効な EPM System ユーザーであることが決定されます。また、EPM System 製品全体で SSO を使用可能にするトークンも発行されます。

Shared Services で指定される構成により、SSO が使用可能になり、すべての EPM System 製品に対して SSO を受け入れる認証メカニズムが決定されます。

➤ Web アイデンティティ管理ソリューションから SSO を使用可能にするには:

1. **Shared Services Console を Shared Services 管理者として起動します。****20 ページの「Shared Services Console の起動」を参照してください。**
2. **「管理」、「ユーザー・ディレクトリの構成」の順に選択します。**
3. **Web アイデンティティ管理ソリューションにより使用されるユーザー・ディレクトリが Shared Services で外部のユーザー・ディレクトリとして構成されることを確認します。**

   たとえば、Kerberos SSO を使用可能にする場合、Kerberos 認証用に構成されている Active Directory を外部ユーザー・ディレクトリとして構成する必要があります。

   第 4 章「ユーザー・ディレクトリの構成」を参照してください。
4. **「セキュリティ・オプション」を選択します。**
5. **「詳細オプションの表示」を選択します。**
6. **「定義済ユーザー・ディレクトリ」画面の「シングル・サインオン構成」で、次の手順に従います:**

   1. 「SSO の使用可能」を選択します。
   2. 「SSO プロバイダ/エージェント」から、Web アイデンティティ管理ソリューションを選択します。Kerberos で SSO を構成している場合、「その他」を選択します。

      推奨される SSO メカニズムが自動的に選択されます。表 8 を参照してください。49 ページの「サポートされている SSO メソッド」を参照してください。

      注： 推奨される SSO メカニズムを使用していない場合、「SSO プロバイダ/エージェント」で「その他」を選択する必要があります。たとえば、SiteMinder の HTTP ヘッダー以外のメカニズムを使用するには、「SSO プロバイダ/エージェント」の「その他」を選択してから、「SSO メカニズム」で使用する SSO メカニズムを選択します。

表 8　Web アイデンティティ管理ソリューションに適した SSO メカニズム

Web アイデンティティ管理ソリューション	推奨 SSO メカニズム
Oracle Access Manager	カスタムHTTPヘッダー [1]
OSSO	カスタムHTTPヘッダー
SiteMinder	カスタムHTTPヘッダー
Kerberos	HTTP要求からリモート・ユーザーを取得

[1] デフォルトの HTTP ヘッダー名は、HYPLOGIN です。カスタム HTTP ヘッダーを使用中の場合、名前を置き換えます。

7　「OK」をクリックします。

Smart View に対するシングル・サインオンのオプション

Smart View はシック・クライアントであり、ブラウザではありませんが、HTTP を使用してサーバー・コンポーネントに接続し、システム的にはブラウザのように動作します。Smart View では、ブラウザ・インタフェースでサポートされるすべての標準的な Web ベースの統合方法がサポートされます。ただし、いくつかの制限事項があります:

- Smart View は、Kerberos が使用可能な環境ではサポートされません。
- SSO メカニズムは、共有コンポーネントに対してのみサポートされます。主に下位互換として使用されるプライベート接続では、SSO メカニズムはサポートされません。
- Smart View が、EPM System コンポーネントに接続されている既存のブラウザ・セッションから起動される場合、既存のセッションからの cookie が共有されないため、ユーザーは Smart View に再度サイン・インする必要があります。

4 ユーザー・ディレクトリの構成

この章の内容

EPM System セキュリティのユーザー・ディレクトリ

EPM System 製品は、ユーザー・ディレクトリと総称される多くのユーザーおよびアイデンティティ管理システムでサポートされています。その中には、Sun Java System Directory Server (旧 SunONE Directory Server)、Active Directory など、Lightweight Directory Access Protocol (LDAP)対応のユーザー・ディレクトリが含まれています。また、EPM System は、外部ユーザー・ディレクトリとしてリレーショナル・データベースもサポートします。

通常、EPM System 製品では、プロビジョニングにネイティブ・ディレクトリおよび外部ユーザー・ディレクトリが使用されます。サポートされているユーザー・ディレクトリのリストについては、Oracle Enterprise Performance Management 製品 - サポートされるプラットフォームのメトリックを参照してください。

EPM System 製品では、製品にアクセスする各ユーザーにユーザー・ディレクトリ・アカウントが必要です。これらのユーザーは、プロビジョニングを円滑にするようグループに割り当てることができます。ユーザーおよびグループには、EPM System の役割とオブジェクト ACL をプロビジョニングすることができます。管理のオーバーヘッドのため、個別ユーザーのプロビジョニングはお薦めしません。

すべての構成済ユーザー・ディレクトリからのユーザーおよびグループは、Shared Services Console に表示されます。

デフォルトで、EPM System コンフィグレータにより、EPM System 製品をサポートする Shared Services リポジトリがネイティブ・ディレクトリとして構成されます。ディレクトリ・マネージャは Shared Services Console を使用して、ネイティブ・ディレクトリにアクセスして管理します。

ユーザー・ディレクトリ構成に関連する操作

SSO と承認をサポートするには、システム管理者が外部ユーザー・ディレクトリを構成する必要があります。Shared Services Console から、システム管理者はユーザー・ディレクトリの構成と管理に関連する複数のタスクを実行できます。これらのトピックは、次の手順に示されています。

- ユーザー・ディレクトリの構成:
  - 90 ページの「OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成」
  - 102 ページの「リレーショナル・データベースをユーザー・ディレクトリとして構成する」
- 105 ページの「ユーザー・ディレクトリの接続のテスト」
- 105 ページの「ユーザー・ディレクトリ設定の編集」
- 106 ページの「ユーザー・ディレクトリ構成の削除」
- 106 ページの「ユーザー・ディレクトリの検索順の管理」
- 108 ページの「セキュリティ・オプションの設定」

Oracle Identity Manager と EPM System

Oracle Identity Manager は、エンタープライズ・リソース全体でユーザー・アカウントと属性レベルの権限の両方を追加、更新および削除するプロセスを自動化する、役割およびユーザーの管理ソリューションです。Oracle Identity Manager は、スタンドアロン製品として、あるいは Oracle Identity and Access Management Suite Plus の一部として使用できます。

EPM System は、LDAP グループであるエンタープライズ・ロールの使用によって Oracle Identity Manager と統合されます。EPM System コンポーネントの役割は、エンタープライズ・ロールに割り当てることができます。Oracle Identity Manager エンタープライズ・ロールに追加されたユーザーまたはグループは、割り当てられている EPM System の役割を自動的に継承します。

たとえば、Budget Planning という名前の Planning アプリケーションがあるとします。このアプリケーションをサポートするには、Budget Planning インタラクティブ・ユーザー、Budget Planning エンド・ユーザー、Budget Planning 管理者の 3 つの役割を Oracle Identity Manager で作成します。EPM System の役割をプロビジョニングする際、プロビジョニング・マネージャは Oracle Identity Manager のエンタープライズ・ロールを、Budget Planning と、Shared Services などのその他の EPM

System コンポーネントの必須役割に必ずプロビジョニングします。Oracle Identity Manger のエンタープライズ・ロールに割り当てられているユーザーとグループはすべて、EPM System の役割を継承します。Oracle Identity Manager の配置と管理の詳細は、Oracle Identity Manager のドキュメントを参照してください。

Oracle Identity Manager と EPM System を統合するには、管理者は次の手順を実行する必要があります。

- EPM System プロビジョニングに使用する予定の Oracle Identity Manager エンタープライズ・ロールのメンバー(ユーザーとグループ)が LDAP 対応のユーザー・ディレクトリ(OID、Active Directory など)で定義されていることを確認します。
- EPM System で、エンタープライズ・ロールのメンバーが定義されている LDAP 対応のユーザー・ディレクトリを外部ユーザー・ディレクトリとして構成します。90 ページの「OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成」を参照してください。

Active Directory の情報

この項では、このドキュメントで使用される Microsoft Active Directory の概念について説明します。

DNS 検索とホスト名検索

システム管理者は Shared Services が静的ホスト名検索または DNS 検索を行って Active Directory を識別できるように Active Directory を構成できます。静的ホスト名検索は Active Directory フェイルオーバーをサポートしません。

DNS 検索を使用すると、Active Directory が確実に高可用性を実現するように複数のドメイン・コントローラ上で構成されるシナリオでは、Active Directory の高可用性が確実に実現されます。DNS 検索を実行するように構成されている場合、Shared Services は登録されているドメイン・コントローラを識別するクエリーを DNS サーバーに対して行い、最大の重みのドメイン・コントローラに接続します。Shared Services が接続されるドメイン・コントローラで障害が発生すると、Shared Services は次に使用可能な最大の重みのドメイン・コントローラに動的に切り替えます。

注： DNS 検索は、フェイルオーバーをサポートする冗長 Active Directory 設定が使用可能な場合のみ構成できます。詳細は、Microsoft のドキュメントを参照してください。

グローバル・カタログ

グローバル・カタログは、フォレスト内のすべての Active Directory オブジェクトのコピーを保管するドメイン・コントローラです。そのホスト・ドメインのディレクトリ内のその他すべてのドメインのすべてのオブジェクトの完全なコピーおよびフォレスト内のその他すべてのドメインのすべてのオブジェクトの部分コピー

を保管し、これらは通常のユーザー検索操作で使用されます。グローバル・カタログの設定については、Microsoft のドキュメントを参照してください。

組織でグローバル・カタログを使用する場合、次のメソッドのいずれかを使用して、Active Directory を構成します。

- 外部ユーザー・ディレクトリとしてグローバル・カタログ・サーバーを構成する(推奨)。
- 個別の外部ユーザー・ディレクトリとして各 Active Directory ドメインを構成する。

個々の Active Directory ドメインではなく、グローバル・カタログを構成することにより、EPM System 製品がフォレスト内のローカルおよびユニバーサル・グループにアクセスできます。

OID、Active Directory およびその他の LDAP ベースのユーザー・ディレクトリの構成

この項で示す手順を使用して、管理者は、OID、Sun Java System Directory Server、Oracle Virtual Directory、Active Directory、IBM Tivoli Directory Server などの LDAP ベースの企業ユーザー・ディレクトリを構成するか、あるいは構成画面に示されない LDAP ベースのユーザー・ディレクトリを構成します。

➤ OID、Active Directory および他の LDAP ベースのユーザー・ディレクトリを構成するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

「プロバイダ構成」タブが開きます。この画面には、ネイティブ・ディレクトリを含むすでに構成済のすべてのユーザー・ディレクトリが一覧表示されます。

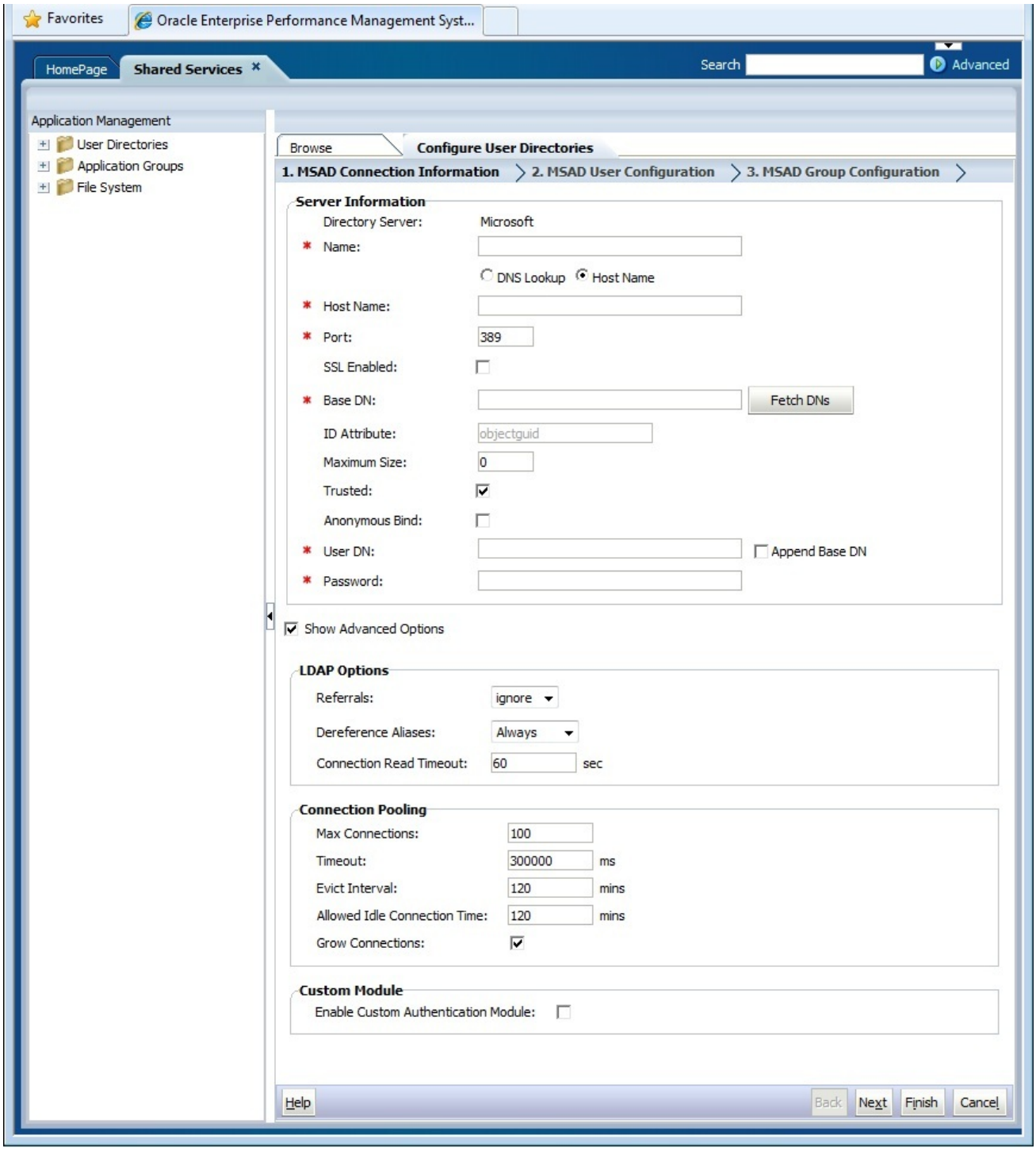
3 「新規」をクリックします。

4 「ディレクトリ・タイプ」で、次のいずれかのオプションを選択します:

- **Lightweight Directory Access Protocol(LDAP)**: Active Directory 以外の LDAP 対応ユーザー・ディレクトリを構成します。Oracle Virtual Directory を構成するには、このオプションを選択します。
- **Microsoft Active Directory (MSAD)**: Active Directory を構成します。

  Active Directory および Active Directory Application Mode (ADAM)のみ: カスタム ID 属性(`ObjectGUID` 以外の属性、たとえば `sAMAccountName`)を Active Directory または ADAM で使用する場合、**Lightweight Directory Access Protocol (LDAP)**を選択し、ディレクトリ・タイプ「その他」として構成します。

5 「次へ」をクリックします。



6 必要なパラメータを入力します。

表 9 「接続情報」画面

ラベル	説明
ディレクトリ・サーバー	ユーザー・ディレクトリを選択します。**ID 属性**値が、選択した製品の推奨される一定の一意の ID 属性に変わります。 手順 4 で Active Directory を選択した場合、このプロパティは自動的に選択されます。 次のシナリオで、「その他」を選択します。 ● リストされていないユーザー・ディレクトリ・タイプ(Oracle Virtual Directory など)を構成している ● リストされている LDAP 対応ユーザー・ディレクトリ(たとえば OID)を構成していますが、カスタム ID 属性は使用しません。 ● カスタム ID 属性を使用する Active Directory または ADAM を構成しています。 **注：** Oracle Virtual Directory では、LDAP ディレクトリと RDMBS データ・リポジトリの抽象化が仮想化されて 1 つのディレクトリ・ビューで提供されるため、Oracle Virtual Directory でサポートされるユーザー・ディレクトリの数やタイプに関係なく、EPM System では 1 つの外部ユーザー・ディレクトリとみなされます。 **例:** `Oracle Internet Directory`
名前	ユーザー・ディレクトリのわかりやすい名前。複数のユーザー・ディレクトリが構成されている場合は、特定のユーザー・ディレクトリを識別するために使用します。 **例:** `Corporate_OID`
DNS 検索	**Active Directory のみ:** このオプションを選択して DNS 検索を使用可能にします。89 ページの「DNS 検索とホスト名検索」を参照してください。DNS 検索は、接続が失敗しないように、本番環境での Active Directory への接続方法として構成することをお薦めします。 **注：** グローバル・カタログを構成している場合は、このオプションを選択しないでください。 このオプションを選択すると、次のフィールドが表示されます: ● **ドメイン**: Active Directory フォレストのドメイン名です。 **例:** `example.com` または `us.example.com` ● **AD サイト**: Active Directory サイト名で、通常は Active Directory 構成コンテナに保管されているサイト・オブジェクトの相対的な識別名です。一般的に AD サイトにより、市、都道府県、地域や国などの地理的な場所が識別されます。 **例:** `Santa Clara` または `US_West_region` ● **DNS サーバー**: ドメイン・コントローラの DNS サーバー検索をサポートするサーバーの DNS 名。
ホスト名	**Active Directory のみ:** このオプションを選択して静的なホスト名検索を使用可能にします。89 ページの「DNS 検索とホスト名検索」を参照してください。 **注：** Active Directory グローバル・カタログを構成している場合は、このオプションを選択します。
ホスト名	ユーザー・ディレクトリ・サーバーの DNS 名。SiteMinder から SSO をサポートするためにユーザー・ディレクトリを使用する場合は、完全修飾のドメイン名を使用します。ホスト名は、テスト目的で Active Directory 接続を確立する場合にのみ使用することをお薦めします。 **注：** Active Directory グローバル・カタログを構成している場合は、グローバル・カタログ・サーバーのホスト名を指定します。89 ページの「グローバル・カタログ」を参照してください。 **例:** `MyServer`

ラベル	説明
ポート	ユーザー・ディレクトリが実行するポート番号。 **注：** Active Directory グローバル・カタログを構成している場合は、グローバル・カタログ・サーバーが使用するポート(デフォルトは 3268)を指定します。89 ページの「グローバル・カタログ」を参照してください。 **例:** `389`
SSL 使用可能	このユーザー・ディレクトリとのセキュア通信を使用可能にするチェック・ボックス。ユーザー・ディレクトリは、セキュア通信として構成する必要があります。
ベース DN	ユーザーおよびグループの検索を開始するノードの識別名(DN)。また、「DN のフェッチ」ボタンを使用して、使用可能なベース DN のリストを表示し、そのリストから適切なベース DN を選択できます。 **注：** グローバル・カタログを構成している場合は、フォレストのベース DN を指定します。 特殊文字の使用上の制限については、112 ページの「特殊文字の使用方法」を参照してください。 EPM System 製品のすべてのユーザーとグループを含む最下位の DN を選択することをお薦めします。 **例:** `dc=example,dc=com`
ID 属性	この属性値は、「ディレクトリ・タイプ」で「その他」が選択されている場合のみ変更できます。この属性はディレクトリ・サーバー上のユーザーおよびグループ・オブジェクトに存在する共通の属性である必要があります。 この属性の推奨値が自動的に、OID (`orclguid`)、SunONE (`nsuniqueid`)、IBM Directory Server (`Ibm-entryUuid`)、Novell eDirectory (`GUID`)および Active Directory (`ObjectGUID`)に設定されます。 **例:** `orclguid` 「ディレクトリ・サーバー」で「その他」を選択後、ID 属性値を手動で設定する場合(Oracle Virtual Directory を構成する場合など)、ID 属性値は次のようになります: ● 一意の属性を指します ● 場所に固有ではありません ● 時間の経過とともに変わりません
最大サイズ	検索が戻す結果の最大数。ユーザー・ディレクトリ設定でサポートする値よりもこの値が大きい場合は、ユーザー・ディレクトリ値がこの値をオーバーライドします。 Active Directory 以外のユーザー・ディレクトリの場合、このフィールドを空白にすると、検索条件を満たすすべてのユーザーとグループが取得されます。 Active Directory の場合、この値を `0` に設定すると、検索条件を満たすすべてのユーザーとグループが取得されます。 委任された管理モードで Shared Services を構成している場合は、この値を 0 に設定します。
信頼済	このプロバイダが信頼できる SSO ソースであることを示すチェック・ボックス。信頼できるソースからの SSO トークンにはユーザーのパスワードは含まれません。
匿名のバインド	Shared Services で匿名をユーザー・ディレクトリにバインドしてユーザーおよびグループを検索できることを示すチェック・ボックス。ユーザー・ディレクトリが匿名のバインドを許可する場合にのみ使用できます。このオプションを選択しない場合は、ユーザー情報が保管されたディレクトリを検索するのに十分なアクセス権を持つアカウントをユーザー DN に指定する必要があります。 匿名のバインドの使用はお薦めしません。 **注：** 匿名のバインドは OID ではサポートされません。

ラベル	説明
ユーザー DN	「匿名のバインド」が選択されている場合、このオプションは使用不可です。 Shared Services がユーザー・ディレクトリとのバインドに使用するユーザーの識別名。このユーザーには DN 内の RDN 属性に検索権限が必要です。たとえば、dn: `cn=John Doe, ou=people, dc=myCompany, dc=com` では、バインド・ユーザーには `cn` 属性への検索アクセス権が必要です。 ユーザー DN の値に特殊文字を指定する場合はエスケープ文字を使用する必要があります。制限については、112 ページの「特殊文字の使用方法」を参照してください。 **例:** `cn=admin,dc=myCompany,dc=com`
ベース DN の追加	ベース DN をユーザー DN に追加するためのチェック・ボックス。ディレクトリ・マネージャ・アカウントをユーザー DN として使用している場合は、ベース DN を追加しないでください。 「匿名のバインド」オプションが選択されている場合、このチェック・ボックスは使用不可です。
パスワード	ユーザー DN パスワード 「匿名のバインド」オプションが選択されている場合、このボックスは使用不可です。 **例:** `UserDNpassword`
詳細オプションの表示	詳細オプションを表示するチェック・ボックス。
参照	**Active Directory のみ:** Active Directory が構成されている場合は、「従う」を選択すると、LDAP 参照に自動的に従います。「無視」を選択すると、参照は使用されません。
別名の逆参照	Shared Services の検索で使用するメソッドを選択すると、ユーザー・ディレクトリの別名が逆参照されます。これにより、別名の DN が指すオブジェクトが検索で取得されます。選択: ● **常時**: 常に別名を逆参照します。 ● **なし**: 別名を逆参照しません。 ● **検索中**: 名前解決の間にのみ別名を逆参照します。 ● **検索中**: 名前解決の後にのみ別名を逆参照します。
接続読取りタイムアウト	この間隔(秒数)が経過した後も応答がない場合、LDAP プロバイダは LDAP 読取り試行を中止します。 **デフォルト:** 60 秒
最大接続数	接続プール内の最大接続数。LDAP ベースのディレクトリ(Active Directory を含む)の場合、デフォルトは 100 です。 **デフォルト:** 100
タイムアウト	プールから接続を取得するまでのタイムアウト。この期間が過ぎると例外が発生します。 **デフォルト:** 300000 ミリ秒(5 分)
削除間隔	**オプション:** 削除プロセスを実行してプールを消去するための間隔。削除プロセスによって、「アイドル状態の接続許容時間」を超えたアイドル状態の接続が除去されます。 **デフォルト:** 120 分

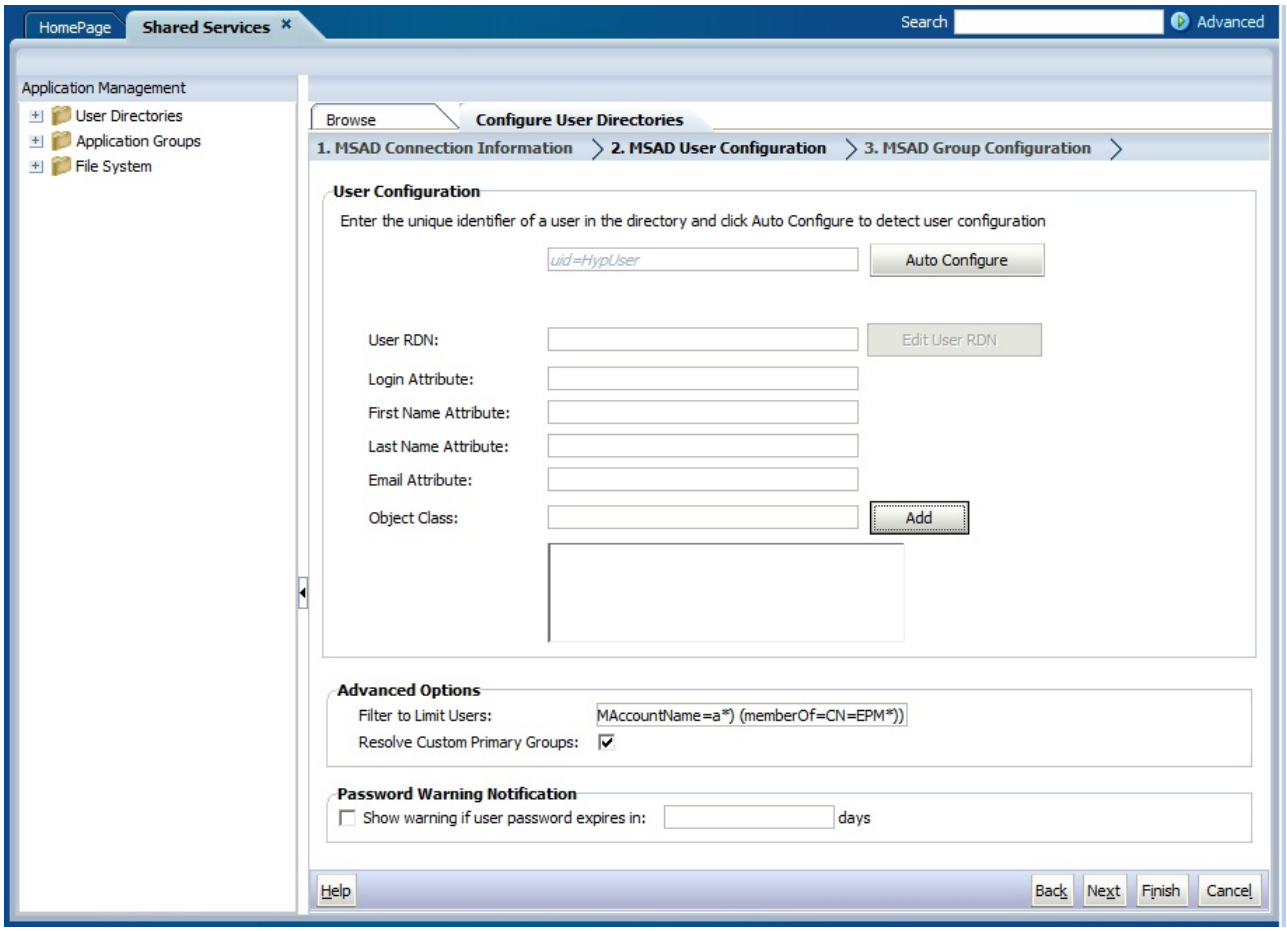
ラベル	説明
アイドル状態の接続許容時間	**オプション:** 削除プロセスがプール内のアイドル状態の接続を除去するまでの許容時間。 **デフォルト:** 120 分
接続の拡大	このオプションは、接続プールが最大接続数を超える接続を保持できるかどうかを示します。デフォルトで選択されています。接続プールの拡大を禁止した場合、タイムアウトで設定された時間内に接続が有効にならないとエラーが発生します。
カスタム認証モジュールを使用可能にする	カスタム認証モジュールの使用を使用可能にして、このユーザー・ディレクトリで定義されたユーザーを認証するためのチェック・ボックス。認証モジュールの完全修飾 Java クラス名も、「セキュリティ・オプション」画面で入力する必要があります。108 ページの「セキュリティ・オプションの設定」を参照してください。 カスタム認証モジュールの認証は、シン・クライアントおよびシック・クライアントに対して透過的で、クライアントの配置変更は必要ありません。Oracle Enterprise Performance Management System Security Configuration Guide のカスタム認証モジュールの使用に関する項を参照してください。

7 「次へ」をクリックします。

Shared Services は、「ユーザー構成」画面に設定されたプロパティを使用して、ユーザーの検索を開始するノードの特定に利用されるユーザー URL を作成します。この URL を使用すると、検索効率が向上します。

注意 ユーザー URL は別名をポイントできません。EPM System のセキュリティでは、ユーザー URL が実際のユーザーをポイントすることが求められます。

画面の「自動構成」領域を使用して、必要な情報を取得することをお薦めします。



注： ユーザー構成で使用できる特殊文字のリストについては、112 ページの「特殊文字の使用方法」を参照してください。

8 **「自動構成」に、フォーマット `attribute=identifier` を使用して、一意のユーザー識別子を入力します。例: `uid=jdoe`。**

ユーザーの属性は、「ユーザー構成」領域に表示されます。

OID を構成している場合は、OID のルート DSE がネーミング・コンテキスト属性内にエントリを含まないため、ユーザー・フィルタを自動的に構成できません。『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』のネーミング・コンテキストの管理に関する項を参照してください。

注： 「ユーザー構成」領域のテキスト・ボックスに、必要なユーザー属性を手動で入力できます。

表 10 「ユーザー構成」画面

ラベル	説明 [1]
ユーザー RDN	ユーザーの相対的な識別名。DN の各コンポーネントは RDN と呼ばれ、ディレクトリ・ツリー内の分岐を表します。ユーザーの RDN は一般に、`uid` または `cn` と同じです。 制限については、112 ページの「特殊文字の使用方法」を参照してください。 **例:** `ou=People`

ラベル	説明[1]
ログイン属性	ユーザーのログオン名を保管する一意の属性(カスタム属性も可能)。ユーザーは、EPM System 製品にログインするとき、この属性の値をユーザー名として使用します。 ユーザー ID (「ログイン属性」の値)は、すべてのユーザー・ディレクトリにわたって一意である必要があります。たとえば、SunONE 構成と Active Directory 構成の「ログイン属性」として、それぞれ `uid` と `sAMAccountName` を使用できます。これらの属性の値は、ネイティブ・ディレクトリを含むすべてのユーザー・ディレクトリにわたって一意である必要があります。 **注：** ユーザー ID では、大文字と小文字が区別されません。 **注：** Kerberos 環境の Oracle Application Server に配置された EPM System 製品の外部ユーザー・ディレクトリとして OID を構成している場合は、このプロパティを `userPrincipalName` に設定する必要があります。 **デフォルト** ● **Active Directory:** `cn` ● **Active Directory 以外の LDAP ディレクトリ:** `uid`
名の属性	ユーザーの名を保管する属性 **デフォルト:** `givenName`
姓の属性	ユーザーの姓を保管する属性 **デフォルト:** `sn`
電子メール属性	**オプション**: ユーザーの電子メール・アドレスを保管する属性 **デフォルト:** `mail`
オブジェクト・クラス	ユーザーのオブジェクト・クラス(ユーザーに関連付けられる必須とオプションの属性)。Shared Services は、この画面に表示されたオブジェクト・クラスを検索フィルタで使用します。これらのオブジェクト・クラスを使用して、Shared Services は、プロビジョニングされたすべてのユーザーを検索する必要があります。 **注：** ユーザー・ディレクトリ・タイプ「その他」として Active Directory または ADAM を、カスタム ID 属性を使用するように構成している場合、この値を `user` に設定する必要があります。 オブジェクト・クラスは、必要に応じて手動で追加できます。オブジェクト・クラスを追加するには、「オブジェクト・クラス」ボックスにオブジェクト・クラス名を入力し、「追加」をクリックします。 オブジェクト・クラスを削除するには、オブジェクト・クラスを選択し、「削除」をクリックします。 **デフォルト** ● **Active Directory:** `user` ● **Active Directory 以外の LDAP ディレクトリ:** `person, organizationalPerson, inetorgperson`
詳細オプションの表示	詳細な構成オプションが表示されるチェック・ボックス。

ラベル	説明 [1]
ユーザーを制限するフィルタ	EPM System 製品の役割がプロビジョニングされるユーザーのみを取得する LDAP クエリー。たとえば、LDAP クエリー(uid=Hyp*)は、名前が Hyp で始まるユーザーのみを取得します。 ユーザー構成画面はユーザー RDN を検証します。必要な場合は、ユーザー・フィルタの使用をお薦めします。 ユーザー・フィルタは、クエリーで戻されるユーザー数を制限します。ユーザー RDN によって識別されるノードが、プロビジョニングされる必要のない多くのユーザーを含む場合に特に重要です。ユーザー・フィルタは、プロビジョニングされる必要のないユーザーを除外するために使用できます。これにより、パフォーマンスが向上します。
複数属性の RDN 用のユーザー検索属性	**Active Directory 以外の LDAP 対応ユーザー・ディレクトリのみ:** ディレクトリ・サーバーが複数属性の RDN を使用するように構成される場合にのみ、この値を設定します。設定した値はいずれかの RDN 属性である必要があります。指定した属性の値は一意で、属性は検索可能である必要があります。 たとえば、SunONE ディレクトリ・サーバーが、cn (cn=John Doe)および uid (uid=jDoe12345)属性を組み合わせるように構成され、次のような複数属性の RDN を作成するとします: `cn=John Doe+uid=jDoe12345, ou=people, dc=myCompany, dc=com` この場合、これらの属性が次の条件を満たしている場合には、cn または uid のいずれかを使用できます: ● この属性は「接続情報」タブにファイルされたユーザー DN で識別されたユーザーにより検索可能です ● この属性はユーザー・ディレクトリ全体で一意の値に設定する必要があります
カスタム・プライマリ・グループの解決	**Active Directory のみ:** 効果的な役割を決定するためにユーザーのプライマリ・グループを識別するかどうかを示すチェック・ボックス。このチェック・ボックスはデフォルトで選択されています。この設定は変更しないことをお薦めします。
ユーザー・パスワードの期限が次の日数以内に切れる場合に警告を表示	**Active Directory のみ:** Active Directory ユーザーのパスワードが指定した日数以内に期限切れになる場合に警告メッセージを表示するかどうかを示すチェック・ボックス。

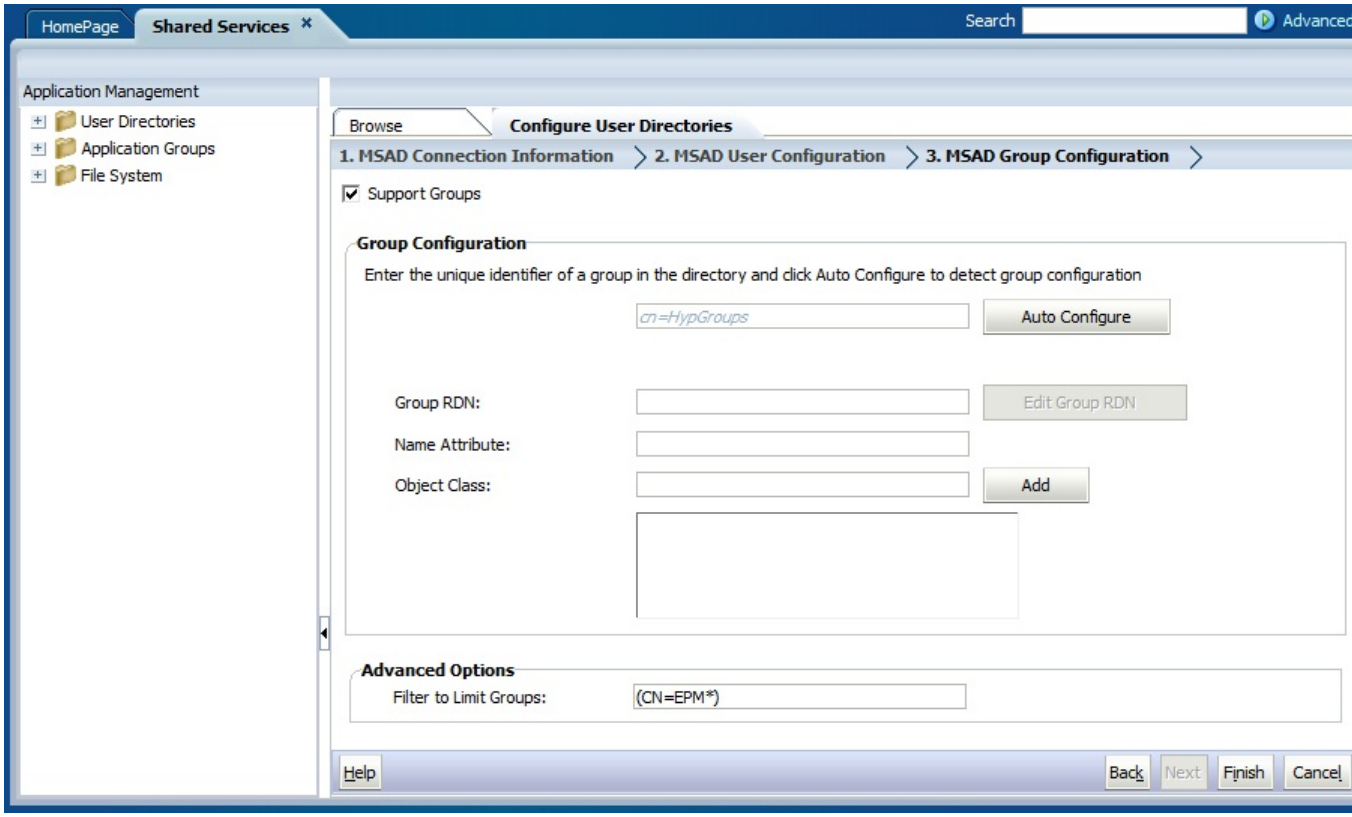
[1]EPM System セキュリティでは、構成値がオプションの一部のフィールドにデフォルト値が使用されます。そのようなフィールドに値を入力しない場合、デフォルト値が実行時に使用されます。

9 「次へ」をクリックします。

「グループ構成」画面が開きます。Shared Services は、この画面に設定されたプロパティを使用して、グループの検索を開始するノードを特定するグループ URL を作成します。この URL を使用すると、検索効率が向上します。

注意 グループ URL は別名をポイントできません。EPM System のセキュリティでは、グループ URL が実際のグループをポイントすることを求められます。グループの別名を使用する Novell eDirectory を構成している場合、グループ URL 内でグループの別名とグループ・アカウントを使用できる必要があります。

注： 「グループ構成」画面のデータ入力はオプションです。グループ URL の設定を入力しない場合、Shared Services は、ベース DN 内を検索してグループを見つけます。これは特に、ユーザー・ディレクトリに多くのユーザーが含まれている場合に悪影響をパフォーマンスに及ぼします。



10 組織で、グループのプロビジョニングを予定していない場合、またはユーザーがユーザー・ディレクトリでループに分類されない場合は、「グループのサポート」を選択解除します。このオプションを選択解除すると、この画面のフィールドは使用不可になります。

グループをサポートしている場合は、自動構成機能を使用して、必要な情報を取得することをお薦めします。

OID をユーザー・ディレクトリとして構成している場合は、自動構成機能を使用できません。理由は、OID のルート DSE がネーミング・コンテキスト属性内にエントリを含まないからです。『Oracle Fusion Middleware Oracle Internet Directory 管理者ガイド』のネーミング・コンテキストの管理に関する項を参照してください。

11 「自動構成」テキスト・ボックスに、一意のグループ識別子を入力し、「検索」をクリックします。

グループ識別子は、フォーマット `attribute=identifier` で指定する必要があります。例: `cn=western_region`。

グループの属性は、「グループ構成」領域に表示されます。

注： 必要なグループ属性は、「グループ構成」テキスト・ボックスに入力できます。

注意 ノード名に/ (スラッシュ)または\ (円記号(バックスラッシュ))を含むユーザー・ディレクトリにグループ URL が設定されていない場合、ユーザーおよびグループの検索は失敗します。たとえば、ユーザーおよびグループが存在するノード内(`OU=child\ou,OU=parent/ou` または `OU=child/ou,OU=parent \ ou` など)のユーザー・ディレクトリにグループ URL が指定されていない場合、ユーザーまたはグループを表示する操作は失敗します。

表 11 「グループ構成」画面

ラベル	説明 [1]
グループ RDN	グループの相対 DN。この値は、ベース DN の相対パスで、グループ URL として使用されます。 グループ RDN を指定します。これにより、プロビジョニングする予定のすべてのグループが使用可能な最下位のユーザー・ディレクトリ・ノードが識別されます。 プロビジョニングに Active Directory プライマリ・グループを使用する場合、プライマリ・グループがグループ RDN 下にあることを確認します。Shared Services では、グループ URL のスコープ外のプライマリ・グループは取得されません。 グループ RDN はログインと検索のパフォーマンスに重大な影響を及ぼします。グループ RDN はすべてのグループ検索の開始点であるため、EPM System 製品のすべてのグループが使用可能な最下位ノードを識別する必要があります。最適なパフォーマンスを保証するには、グループ RDN 内に存在するグループのメンバーが 10,000 を超えないようにする必要があります。これより多くのグループが存在する場合は、グループ・フィルタを使用して、プロビジョニングするグループのみを取得します。 **注：** グループ URL 内の使用可能なグループ数が 10,000 を超えると、Shared Services は警告を表示します。 制限については、112 ページの「特殊文字の使用方法」を参照してください。 **例:** `ou=Groups`
名前の属性	グループの名前を保管する属性 **デフォルト** ● **Active Directory を含む LDAP ディレクトリ:** `cn` ● **ネイティブ・ディレクトリ:** `cssDisplayNameDefault`

ラベル	説明[1]
オブジェクト・クラス	グループのオブジェクト・クラス。Shared Services は、この画面に表示されたオブジェクト・クラスを検索フィルタで使用します。これらのオブジェクト・クラスを使用して、Shared Services は、ユーザーに関連付けられたすべてのグループを検索する必要があります。 **注：** ユーザー・ディレクトリ・タイプ「その他」として Active Directory または ADAM を、カスタム ID 属性を使用するように構成している場合、この値を `group?member` に設定する必要があります。 オブジェクト・クラスは、必要に応じて手動で追加できます。オブジェクト・クラスを追加するには、「オブジェクト・クラス」テキスト・ボックスにオブジェクト・クラス名を入力し、「追加」をクリックします。 オブジェクト・クラスを削除するには、オブジェクト・クラスを選択し、「削除」をクリックします。 **デフォルト** ● **Active Directory:** `group?member` ● **Active Directory 以外の LDAP ディレクトリ:** `groupofuniquenames?uniquemember, groupOfNames?member` ● **ネイティブ・ディレクトリ:** `groupofuniquenames?uniquemember, cssGroupExtend?cssIsActive`
詳細オプションの表示	フィルタの使用を使用可能にして、検索操作時にグループを取得するためのチェック・ボックス。
グループを制限するフィルタ	EPM System 製品の役割がプロビジョニングされるグループのみを取得する LDAP クエリー。たとえば、LDAP クエリー `(|(cn=Hyp*)(cn=Admin*))` は、名前が `Hyp` または `Admin` で始まるグループのみを取得します。 グループ・フィルタは、クエリーで戻されるグループ数を制限するために使用します。グループ RDN によって識別されるノードが、プロビジョニングされる必要のない多くのグループを含む場合に特に重要です。フィルタは、プロビジョニングされる必要のないグループを除外するために使用できます。これにより、パフォーマンスが向上します。 プロビジョニングに Active Directory プライマリ・グループを使用する場合、設定したグループ・フィルタがグループ URL のスコープ内に含まれるプライマリ・グループを取得できることを確認します。たとえば、フィルタ `(|(cn=Hyp*)(cn=Domain Users))` は、名前が `Hyp` で始まるグループと `Domain Users` という名前のプライマリ・グループを取得します。

[1]EPM System セキュリティでは、構成値がオプションの一部のフィールドにデフォルト値が使用されます。そのようなフィールドに値を入力しない場合、デフォルト値が実行時に使用されます。

12 「保存」をクリックします。

Shared Services は構成を保存して、「定義済ユーザー・ディレクトリ」画面に戻ります。この画面には、今構成したユーザー・ディレクトリが表示されます。

13 構成をテストします。105 ページの「ユーザー・ディレクトリの接続のテスト」を参照してください。

14 必要に応じて、検索順の割当てを変更します。詳細は、106 ページの「ユーザー・ディレクトリの検索順の管理」を参照してください。

15 必要に応じて、セキュリティ・オプションを指定します。108 ページの「セキュリティ・オプションの設定」を参照してください。

16 Foundation Services とその他の EPM System コンポーネントを再起動します。

リレーショナル・データベースをユーザー・ディレクトリとして構成する

Oracle、SQL Server、および IBM DB2 リレーショナル・データベースのシステム・テーブルからのユーザーおよびグループ情報を使用して、プロビジョニングをサポートできます。グループ情報がデータベースのシステム・スキーマから取得できない場合、Shared Services はそのデータベース・プロバイダからのグループのプロビジョニングはサポートしません。たとえば、Shared Services は、データベースがオペレーティング・システム上で定義されているグループを使用するため、古いバージョンの IBM DB2 からグループ情報を抽出できません。ただし、プロビジョニング・マネージャはネイティブ・ディレクトリのグループにこれらのユーザーを追加して、このグループをプロビジョニングできます。サポートされているプラットフォームの情報は、Oracle Enterprise Performance Management 製品 - サポートされるプラットフォームのメトリックを参照してください。

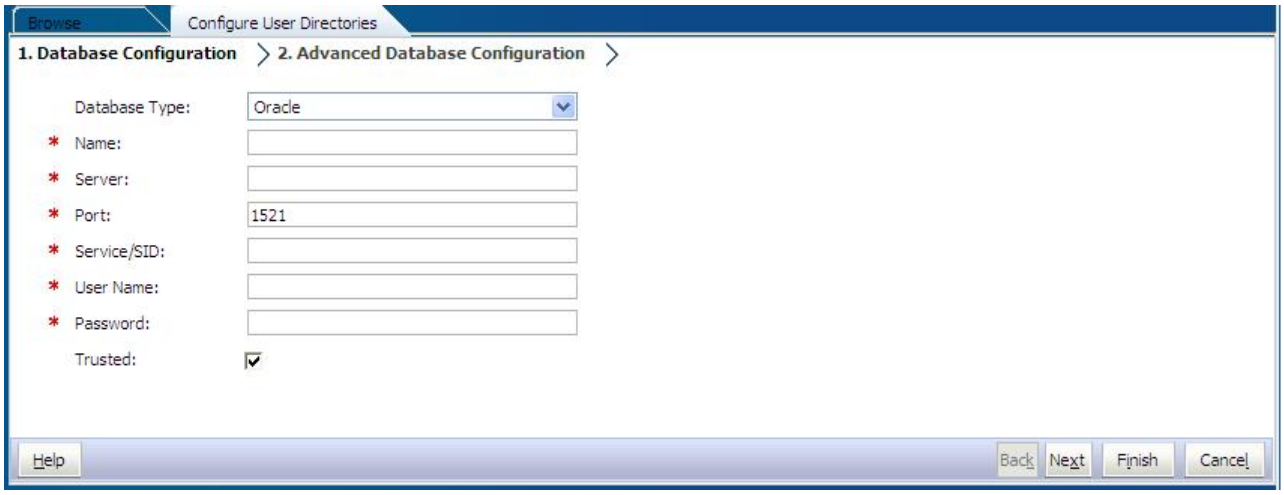
注： DB2 データベースを使用する場合、ユーザー名は 8 文字以上にする必要があります。Oracle および SQL Server データベースの場合は 256 文字、DB2 の場合は 1000 文字を超えないようにしてください。

ユーザーおよびグループのリストを取得するには、データベース管理者、たとえば、Oracle SYSTEM ユーザーとしてデータベースに接続できるように Shared Services を構成します。

注： Shared Services は、プロビジョニングに対してアクティブなデータベース・ユーザーのみ取得します。非アクティブでロックされているデータベース・ユーザー・アカウントは無視されます。

➤ データベース・プロバイダを構成するには:

1. システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。
2. 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。
3. 「追加」をクリックします。
4. 「ディレクトリ・タイプ」画面で、「リレーショナル・データベース(Oracle、DB2、SQL Server)」を選択します。
5. 「次へ」をクリックします。



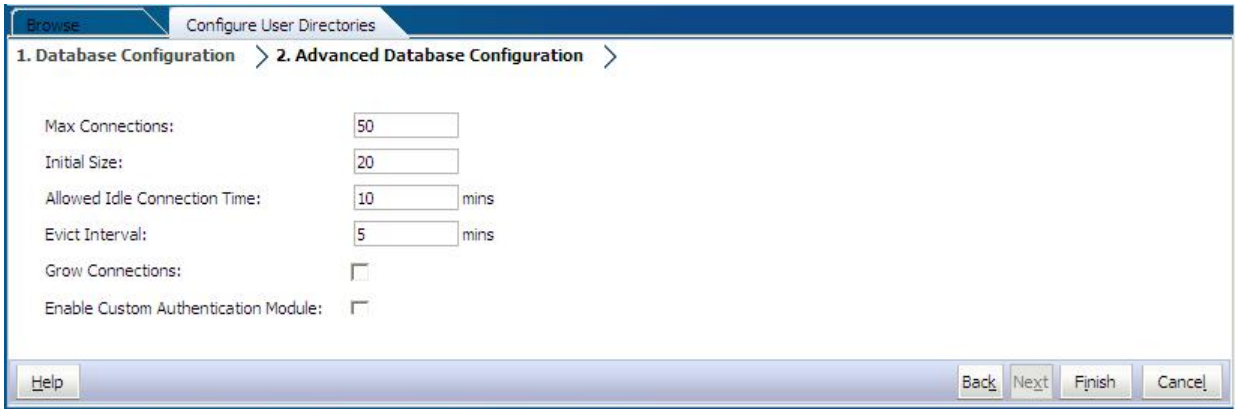
6 「データベースの構成」タブで、構成パラメータを入力します。

表 12 「データベースの構成」タブ

ラベル	説明
データベース・タイプ	リレーショナル・データベース・プロバイダ。Shared Services は、データベース・プロバイダとして Oracle、IBM DB2、および SQL Server データベースのみサポートしています。 **例:** `Oracle`
名前	データベース・プロバイダの一意の構成名。 **例:** `Oracle_DB_FINANCE`
サーバー	データベース・サーバーが稼働しているコンピュータの DNS 名。 **例:** `myserver`
ポート	データベース・サーバーのポート番号 **例:** `1521`
サービス/SID (Oracle のみ)	システム識別子(デフォルトは `orcl`) **例:** `orcl`
データベース (SQL Server および DB2 のみ)	Shared Services が接続する必要があるデータベース **例:** `master`
ユーザー名	Shared Services がデータベースへのアクセスに使用するユーザー名。このデータベース・ユーザーには、データベース・システム・テーブルへのアクセス権が必要です。Oracle データベースにはシステム・アカウント、SQL Server および IBM DB2 データベースにはデータベース管理者のユーザー名を使用することをお薦めします。 **例:** `SYSTEM`
パスワード	「ユーザー名」でユーザーを識別するパスワード。 **例:** `system_password`
信頼済	このプロバイダが信頼できる SSO ソースであることを指定するチェック・ボックス。信頼できるソースからの SSO トークンにはユーザーのパスワードは含まれません。

7 **オプション:** 接続プールを構成するには、「次へ」をクリックします。

「詳細なデータベース構成」タブが開きます。



8 「詳細なデータベース構成」タブで、接続プールのパラメータを入力します。

表 13 「詳細なデータベース構成」タブ

ラベル	説明
最大接続数	プールの最大接続数。デフォルトは 50 です。
初期サイズ	プールを初期化する場合に使用可能な接続数。デフォルトは 20 です。
アイドル状態の接続許容時間	**オプション:** 削除プロセスがプール内のアイドル状態の接続を除去するまでの許容時間。デフォルトは 10 分です。
削除間隔	**オプション:** プールを消去するために削除プロセスを実行する間隔。削除はアイドル状態の接続許容時間を超えたアイドル接続を除去します。デフォルトは 5 分です。
接続の拡大	接続プールが最大接続数を超える接続を保持できるかどうかを示します。デフォルトでは、このオプションは選択解除されており、接続は保持できないことを示します。接続プールが接続を保持できず、接続がタイムアウトに設定された時間内に使用できない場合、システムはエラーを返します。
カスタム認証モジュールを使用可能にする	カスタム認証モジュールの使用を使用可能にして、このユーザー・ディレクトリで定義されたユーザーを認証するためのチェック・ボックス。認証モジュールの完全修飾 Java クラス名も、「セキュリティ・オプション」画面で入力する必要があります。108 ページの「セキュリティ・オプションの設定」を参照してください。 カスタム認証モジュールの認証は、シン・クライアントおよびシック・クライアントに対して透過的に行われます。Oracle Enterprise Performance Management System Security Configuration Guide のカスタム認証モジュールの使用に関する項を参照してください。

9 「保存」をクリックします。

10 「定義済ユーザー・ディレクトリ」画面に戻るには、「OK」をクリックします。

11 データベース・プロバイダ構成をテストします。105 ページの「ユーザー・ディレクトリの接続のテスト」を参照してください。

12 必要に応じて、検索順の割当てを変更します。詳細は、106 ページの「ユーザー・ディレクトリの検索順の管理」を参照してください。

13 必要に応じて、セキュリティ設定を指定します。108 ページの「セキュリティ・オプションの設定」を参照してください。

14 Foundation Services とその他の EPM System コンポーネントを再起動します。

ユーザー・ディレクトリの接続のテスト

ユーザー・ディレクトリの構成後、Shared Services が現在の設定を使用してユーザー・ディレクトリに接続できることを確認するため、接続をテストします。

➤ ユーザー・ディレクトリ接続をテストするには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 ユーザー・ディレクトリのリストから、テストする外部ユーザー・ディレクトリ構成を選択します。

4 「テスト」、「OK」の順にクリックします。

ユーザー・ディレクトリ設定の編集

管理者は名前以外のユーザー・ディレクトリ構成のパラメータを変更できます。プロビジョニング用に使用されていたユーザー・ディレクトリの構成データは編集しないことをお薦めします。

注意 たとえば、ユーザー・ディレクトリ構成の ID属性などのいくつかの設定を編集すると、プロビジョニング・データが使用不可になります。プロビジョニングされたユーザー・ディレクトリの設定を変更する場合は、十分注意してください。

➤ ユーザー・ディレクトリ構成を編集するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「定義済ユーザー・ディレクトリ」画面から、編集するユーザー・ディレクトリを選択します。

4 「編集」をクリックします。

5 構成設定を変更します。

注: 構成名は変更できません。LDAP ユーザー・ディレクトリ構成を変更する場合、「ディレクトリ・サーバー」リストから別のディレクトリ・サーバーや「その他」(カスタム LDAP ディレクトリの場合)を選択できます。ネイティブ・ディレクトリ・パラメータは編集できません。

編集可能なパラメータの説明については、次のテーブルを参照してください。

- Active Directory およびその他の LDAP ベースのユーザー・ディレクトリ:
  - 表 9

    - 表 10
    - 表 11
- データベース: 表 12 を参照

6 「終了」をクリックして、変更を保存します。

ユーザー・ディレクトリ構成の削除

管理者はユーザー・ディレクトリ構成をいつでも削除できます。構成を削除すると、ユーザー・ディレクトリから取得されたユーザーおよびグループのプロビジョニング情報がすべて使用不可になり、検索順からディレクトリが除去されます。

ヒント： プロビジョニング用に使用された構成済のユーザー・ディレクトリを使用しない場合、ユーザーおよびグループの検索に使用されないように検索順から除去します。このアクションにより、プロビジョニング情報の整合性を維持し、後でユーザー・ディレクトリを使用できます。

➤ ユーザー・ディレクトリ構成を削除するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「定義済ユーザー・ディレクトリ」画面から、ディレクトリを選択します。

4 「削除」をクリックします。

5 「OK」をクリックします。

6 再度「OK」をクリックします。

7 Foundation Services とその他の EPM System コンポーネントを再起動します。

ユーザー・ディレクトリの検索順の管理

管理者が外部ユーザー・ディレクトリを構成すると、Shared Services により自動的にユーザー・ディレクトリが検索順に追加され、ネイティブ・ディレクトリの検索順より上位の次の使用可能な検索順が割り当てられます。検索順は、EPM System でユーザーとグループについて検索する際、構成されたユーザー・ディレクトリ間を循環するために使用されます。

管理者はユーザー・ディレクトリを検索順から除去できます。この場合、Shared Services により残りのディレクトリの検索順が自動的に再割当てされます。検索順に含まれないユーザー・ディレクトリは、認証およびプロビジョニングのサポートに使用されません。

注： Shared Services は、指定されたアカウントを検出するとユーザーまたはグループの検索を停止します。EPM System ユーザーの大部分が存在する企業ディレクトリを検索順の一番上に配置することをお薦めします。

デフォルトでは、ネイティブ・ディレクトリは検索順の最後のディレクトリとして設定されます。管理者は検索順を管理するために、次のタスクを実行できます。

- 107 ページの「ユーザー・ディレクトリの検索順への追加」
- 108 ページの「検索順の変更」
- 107 ページの「検索順の割当ての除去」

ユーザー・ディレクトリの検索順への追加

新規に構成されたユーザー・ディレクトリは、検索順に自動的に追加されます。検索順からディレクトリを除去した場合、検索順の最後にそれを追加できます。

➤ 検索順にユーザー・ディレクトリを追加するには:

1. システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。
2. 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。
3. 「定義済ユーザー・ディレクトリ」画面から、検索順に追加するユーザー・ディレクトリを選択します。
4. 「含む」をクリックします。

   このボタンは、検索順にないユーザー・ディレクトリを選択している場合のみ使用可能です。
5. 「定義済ユーザー・ディレクトリ」画面に戻るには、「OK」をクリックします。
6. Foundation Services とその他の EPM System コンポーネントを再起動します。

検索順の割当ての除去

検索順からユーザー・ディレクトリを除去してもディレクトリ構成が無効にならず、ユーザー認証のために検索されるディレクトリのリストからユーザー・ディレクトリが除去されます。検索順に含まれないディレクトリは、未使用のステータスに設定されます。管理者が検索順からユーザー・ディレクトリを除去すると、他のユーザー・ディレクトリに割り当てられている検索順は自動的に更新されます。

注: ネイティブ・ディレクトリは検索順から削除できません。

➤ 検索順からユーザー・ディレクトリを除去するには:

1. システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。
2. 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。
3. 「定義済ユーザー・ディレクトリ」画面から、検索順から削除するディレクトリを選択します。
4. 「除外」をクリックします。

5 「OK」をクリックします。

6 「ディレクトリの構成結果」画面で「OK」をクリックします。

7 Foundation Services とその他の EPM System コンポーネントを再起動します。

8

検索順の変更

各ユーザー・ディレクトリに割り当てられているデフォルトの検索順は、ディレクトリが構成されたシーケンスに基づきます。デフォルトでは、ネイティブ・ディレクトリは検索順の最後のディレクトリとして設定されます。

➤ 検索順を変更するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「定義済ユーザー・ディレクトリ」画面から、検索順を変更するユーザー・ディレクトリを選択します。

4 「上へ移動」または「下へ移動」をクリックします。

5 「保存」をクリックします。

6 Foundation Services とその他の EPM System コンポーネント、および Shared Services セキュリティ API を使用するカスタム・アプリケーションを再起動します。

セキュリティ・オプションの設定

セキュリティ・オプションは、検索順に含まれるすべてのユーザー・ディレクトリに適用可能なグローバル・パラメータから構成されています。

➤ セキュリティ・オプションを設定するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「セキュリティ・オプション」を選択します。

4 「セキュリティ・オプション」では、グローバル・パラメータを設定します。

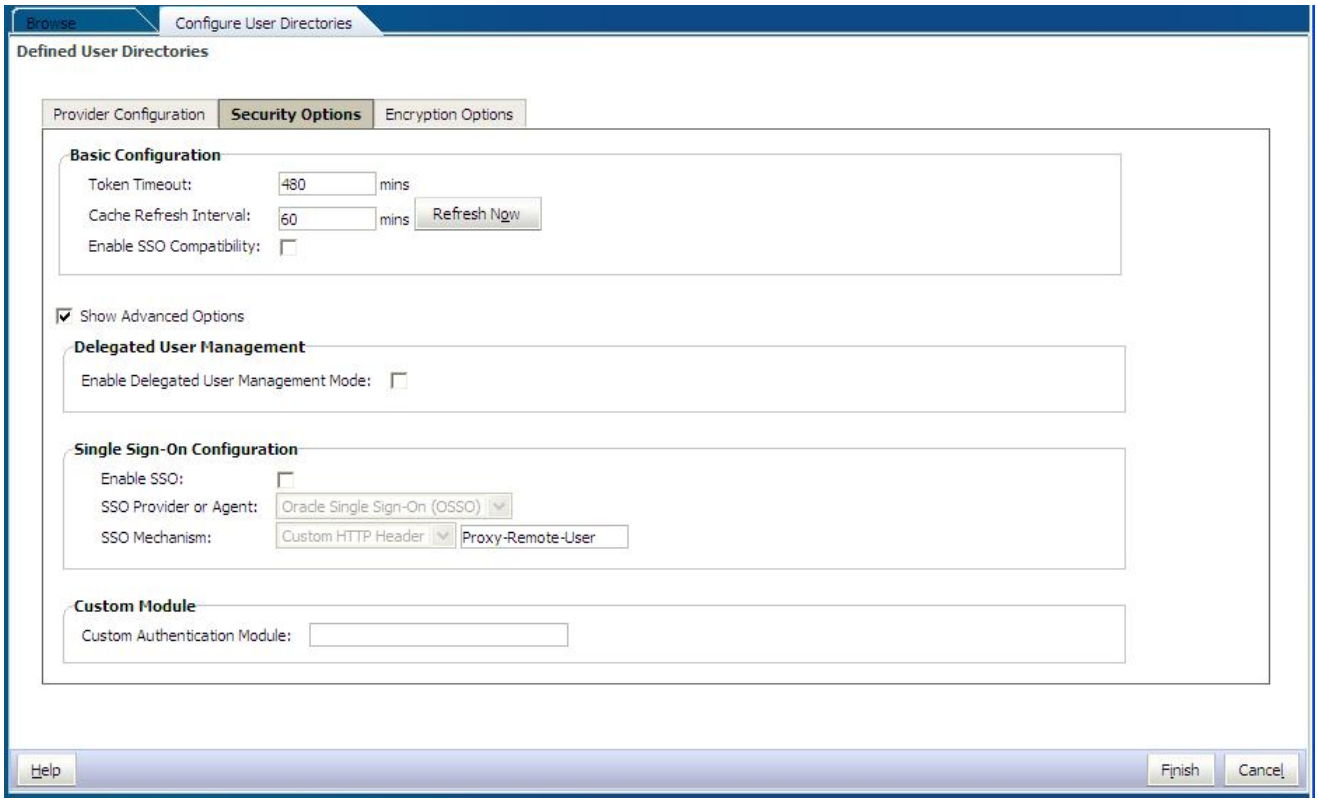


表 14　ユーザー・ディレクトリ用のセキュリティ・オプション

パラメータ	説明
トークンのタイムアウト	EPM System 製品または Web アイデンティティ管理ソリューションが発行元となる SSO トークンの期限が切れるまでの時間(分)。ユーザーは、この期間が過ぎてからログインする必要があります。トークンのタイムアウトは、サーバーのシステム・クロックに基づいて設定されます。デフォルトは 480 分です。 **注：**　トークンのタイムアウトは、セッションのタイムアウトとは異なります。
キャッシュのリフレッシュ間隔	Shared Services のグループ・キャッシュをユーザー関係データにリフレッシュする間隔(分)。デフォルトは 60 分です。 Shared Services では、次のキャッシュ・リフレッシュ後にのみ、新しい外部ユーザー・ディレクトリ・グループと、既存のグループに追加された新しいユーザーに関する情報がキャッシュされます。新規に作成された外部ユーザー・ディレクトリ・グループを通じてプロビジョニングされたユーザーの役割は、キャッシュがリフレッシュされるまでプロビジョニングされません。
今すぐリフレッシュ	グループを含む Shared Services キャッシュのユーザー関係データへのリフレッシュを手動で開始するには、このボタンをクリックします。外部ユーザー・ディレクトリに新規グループを作成し、それらをプロビジョニングした後、または新規ユーザーを既存のグループに追加した後に、キャッシュ・リフレッシュを開始することが必要な場合があります。キャッシュは、Shared Services によってキャッシュ内のデータを使用する呼出しが行われた後にのみリフレッシュされます。
SSO 互換性の有効化	配置が Oracle Business Intelligence Enterprise Edition リリース 11.1.1.5 以前と統合した場合は、このオプションを選択します。
詳細オプションの表示	委任された管理および SSO 構成、認証モジュール・クラスに関連する設定を表示できるオプション

パラメータ	説明
委任されたユーザー管理モードを使用可能にする	EPM System 製品の委任されたユーザー管理を使用可能にし、配布されたプロビジョニング・アクティビティの管理をサポートするオプション。Oracle Enterprise Performance Management System User Security Administration Guide の「委任されたユーザー管理」を参照してください。
SSO の使用可能	Oracle Access Manager などのセキュリティ・エージェントからの SSO のサポートを使用可能にするオプション
SSO プロバイダ/エージェント	EPM System 製品が SSO を受け入れる必要のある Web アイデンティティ管理ソリューションを選択します。Web アイデンティティ管理ソリューションがリストされていない場合(Kerberos などの場合)、「その他」を選択します。 SSO プロバイダを選択すると、希望する SSO メカニズムと名前が自動的に選択されます。必要に応じて、SSO メカニズム(HTTP ヘッダーまたはカスタム・ログイン・クラス)の名前を変更できます。 SSO プロバイダまたはエージェントとして「その他」を選択した場合、EPM System のサポートする SSO メカニズムをサポートすることを確認する必要があります。Oracle Enterprise Performance Management System Security Configuration Guide のサポートされる SSO メソッドに関する項を参照してください。
SSO メカニズム	ユーザーのログイン名を EPM System 製品に提供するために選択した Web アイデンティティ管理ソリューションで使用されるメソッド。受入れ可能な SSO メソッドの説明は、Oracle Enterprise Performance Management System Security Configuration Guide のサポートされている SSO メソッドに関する項を参照してください。 ● カスタムHTTPヘッダー: セキュリティ・エージェントが EPM System に渡すヘッダーの名前を設定します。 ● カスタム・ログイン・クラス: 認証用の HTTP 要求を処理するカスタム Java クラスを指定します。Oracle Enterprise Performance Management System Security Configuration Guide のカスタム・ログイン・クラスに関する項を参照してください。 **注：** カスタム・ログイン・クラスは、カスタム認証と同じではありません。 ● HTTP認証ヘッダー: 標準 HTTP メカニズム。 ● HTTP要求からリモート・ユーザーを取得: セキュリティ・エージェントによって HTTP 要求にリモート・ユーザーが挿入される場合、このオプションを選択します。
カスタム認証モジュール	認証モジュールで、カスタム認証モジュールが選択されているすべてのユーザー・ディレクトリでユーザーの認証に使用される必要があるカスタム認証モジュールの完全修飾 Java クラス名(たとえば、`com.mycompany.epm.CustomAuthenticationImpl`)。 認証モジュールは、ディレクトリ構成で使用可能(デフォルト)である場合にのみ、ユーザー・ディレクトリに使用されます。 Foundation Services では、カスタム認証 JAR ファイルの名前が `customAuth.jar` である必要があります。`customAuth.jar` は `EPM_ORACLE_HOME/common/jlib/11.1.2.0` にあります。JAR ファイル内では任意のパッケージ構造およびクラス名を使用できます。 詳細は、Oracle Enterprise Performance Management System Security Configuration Guide のカスタム認証モジュールの使用に関する項を参照してください。

5 「保存」をクリックします。

6 **Foundation Services とその他の EPM System コンポーネントを再起動します。**

暗号化鍵の再生成

EPM System では、次のキーを使用してセキュリティを保証します。

- シングル・サインオン暗号化鍵。EPM System SSO トークンの暗号化と復号化に使用されます。このキーは、Shared Services レジストリに保管されます。
- 信頼できるサービス・キー。EPM System コンポーネントで、SSO トークンを要求しているサービスの認証の確認に使用されます。
- プロバイダ構成暗号化鍵。EPM System セキュリティで、構成されている外部ユーザー・ディレクトリとのバインドに使用されるパスワード(LDAP 対応ユーザー・ディレクトリのユーザー DN パスワード)の暗号化に使用されます。このパスワードは、外部ユーザー・ディレクトリの構成時設定されます。

EPM System セキュリティを強化するために、これらのキーを定期的に変更します。

注意 シングル・サインオン暗号化鍵の再生成時、Financial Management、Performance Management Architect および Profitability and Cost Management で使用されるタスクフローは無効化されます。キーの再生成後、タスクフローを開いて保存し、再度有効にします。

➤ シングル・サインオン暗号化鍵、プロバイダ構成キーまたは信頼できるサービス・キーを再生成するには:

1 システム管理者として Shared Services Console にアクセスします。20 ページの「Shared Services Console の起動」を参照してください。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「暗号化オプション」を選択します。

4 「暗号化オプション」で、再生成するキーを選択します。

表 15 EPM System の暗号化オプション

オプション	説明
シングル・サインオン・トークン	EPM System SSO トークンの暗号化と復号化に使用される暗号化鍵を再生成する場合に選択します。 「セキュリティ・オプション」で「SSO 互換性の使用可能化」が設定されている場合、次のいずれかのボタンを選択します。 ● 新規 SSO トークン暗号化鍵を作成する場合、「新しいキーの生成」。 ● デフォルトの SSO トークン暗号化鍵をリストアする場合、「デフォルトにリセット」。 **注：** デフォルトの暗号化鍵に戻す場合は、既存のキーストア・ファイル(EPM_ORACLE_HOME/common/CSS/ssHandlerTK)を、すべての EPM System ホスト・マシンから削除する必要があります。
信頼できるサービス・キー	EPM System コンポーネントで、SSO トークンを要求しているサービスの認証の確認に使用される信頼できる認証キーを再生成する場合、このオプションを選択します。
プロバイダ構成キー	EPM System セキュリティで、構成されている外部ユーザー・ディレクトリとのバインドに使用されるパスワード(LDAP 対応ユーザー・ディレクトリのユーザー DN パスワード)の暗号化に使用されるキーを再生成する場合、このオプションを選択します。このパスワードは、外部ユーザー・ディレクトリの構成時設定されます。

5 「OK」をクリックします。

6 **SSO 暗号化キーを新たに生成する場合、この手順を完了させます。**

1. 「ダウンロード」をクリックします。
2. 「OK」をクリックして、ssHandlerTK(新規 SSO 暗号化鍵をサポートするキーストア・ファイル)を Foundation Services をホストするサーバーのフォルダに保存します。
3. ssHandlerTK をすべての EPM System ホスト・マシン上の EPM_ORACLE_HOME/common/CSS にコピーします。

7 **Foundation Services とその他の EPM System コンポーネントを再起動します。**

特殊文字の使用方法

Active Directory およびその他の LDAP ベースのユーザー・ディレクトリでは、DN、ユーザー名、役割およびグループ名などのエンティティで特殊文字が使用可能です。このような文字を理解させるには、Shared Services に対して特別な処理が必要になる場合があります。

通常、ユーザー・ディレクトリ設定(ベース DN やユーザーおよびグループの URL など)で特殊文字を指定する場合は、エスケープ文字を使用する必要があります。表 16 は、ユーザー名、グループ名、ユーザー URL、グループ URL、およびユーザー DN の OU の値で使用可能な特殊文字をリストしています。

表 16　サポートされる特殊文字

文字 [1]	名前または意味	文字	名前または意味
(	左カッコ	$	ドル
)	右カッコ	+	プラス
"	二重引用符	&	アンパサンド
'	一重引用符	\	円記号(バックスラッシュ)
,	カンマ	^	脱字記号
=	次と等しい	;	セミコロン
<	次より小さい	#	ポンド
>	次より大きい	@	アット記号

[1] ベース DN 内の組織単位名に/(スラッシュ)を使用しないでください

- 特殊文字はログイン・ユーザー属性の値には使用できません。
- アスタリスク(*)は、ユーザー名、グループ名、ユーザー URL、グループ URL、またはユーザー DN の OU 名には使用できません。
- 特殊文字の組合せを含んだ属性値は使用できません。
- アンパサンド(&)は、エスケープ文字なしで使用できます。Active Directory の設定では、&は&のように指定する必要があります。

- ユーザー名とグループ名には円記号(バックスラッシュ)(\)とスラッシュ(/)の両方を使用できません。たとえば、test/\user や new\test/user のような名前は使用できません。

表 17　エスケープする必要がない文字

文字	名前または意味	文字	名前または意味
(	左カッコ	'	一重引用符
)	右カッコ	^	脱字記号
$	ドル	@	アット記号
&[1]	アンパサンド		

[1]&のように記述されている必要があります。

これらの文字は、ユーザー・ディレクトリの設定(ユーザー名、グループ名、ユーザー URL、グループ URL およびユーザー DN)で使用する場合にエスケープされる必要があります。

表 18　ユーザー・ディレクトリ構成設定における特殊文字のエスケープ

特殊文字	エスケープ	サンプル設定	エスケープの例
カンマ(,)	円記号(バックスラッシュ)(\)	ou=test,ou	ou=test\,ou
プラス符号(+)	円記号(バックスラッシュ)(\)	ou=test+ou	ou=test\+ou
次と等しい(=)	円記号(バックスラッシュ)(\)	ou=test=ou	ou=test\=ou
ポンド(#)	円記号(バックスラッシュ)(\)	ou=test#ou	ou=test\#ou
セミコロン(;)	円記号(バックスラッシュ)(\)	ou=test;ou	ou=test\;ou
次より小さい(<)	\<	ou=test<ou	ou=test\<ou
次より大きい(>)	\>	ou=test>ou	ou=test\>ou
" (二重引用符)[1]	\\ (二重円記号(バックスラッシュ))	ou=test"ou	ou=test\\"ou
\ (円記号(バックスラッシュ))[2]	\\\ (三重円記号(バックスラッシュ))	ou=test\ou	ou=test\\\\ou

[1] ユーザー DN では、二重引用符(")は、1 つの円記号(バックスラッシュ)でエスケープされる必要があります。たとえば ou=test”ou は、ユーザー DN では ou=test\”ou と指定する必要があります。

[2] ユーザー DN では、円記号(バックスラッシュ)(\)は、1 つの円記号(バックスラッシュ)でエスケープされる必要があります。たとえば ou=test\ou は、ユーザー DN では ou=test\\ou と指定する必要があります。

注意　ユーザー URL が指定されていない場合、RDN ルート内で作成されるユーザーに/ (スラッシュ)または\ (円記号(バックスラッシュ))が含まれていてはいけません。同様に、グループ URL が指定されない場合、これらの文字は RDN ルート内に作成されたグループ名で使用してはいけません。たとえば、OU=child\ou,OU=parent/ou または OU=child/ou,OU=parent\ou などのグループ名は、サポートされません。この問題は、ユーザー・ディレクトリ構成の ID属性に一意の属性を使用している場合は該当しません。

5 カスタム認証モジュールの使用方法

この章の内容

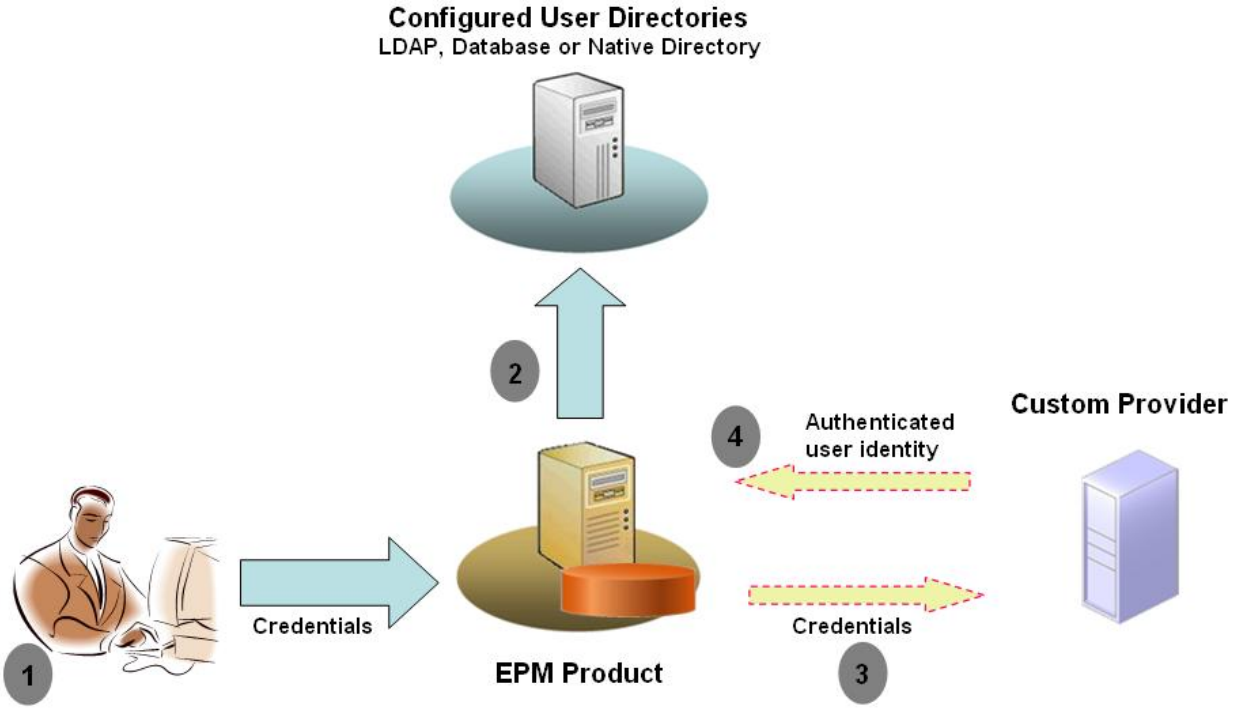
概要

カスタム認証モジュールは、EPM System ユーザーを認証するためにユーザーが開発および実行する Java モジュールです。通常、EPM System 製品では、ユーザー名とパスワードの取得にログオン画面が使用されます。ユーザー名とパスワードはユーザーの認証に使用されます。EPM System 認証を使用するかわりに、カスタム認証モジュールを使用してユーザーを認証し、その後の処理のために認証済ユーザー・ログイン情報を EPM System に渡すことができます。カスタム認証モジュールの実装には EPM System 製品の変更は含まれません。

カスタム認証モジュールは、シック・クライアント(Oracle Hyperion Smart View for Office および Oracle Essbase Studio など)とシン・クライアント(EPM Workspace など)の両方で使用できます。

カスタム認証モジュールは、ユーザーが EPM System 製品にログインする際に入力する情報を使用します。ユーザー・ディレクトリに対して使用可能な場合、カスタム認証モジュールを使用してユーザーを認証します。ユーザーを正しく認証できた場合、カスタム認証モジュールは EPM System にユーザー名を戻します。

次の図は、カスタム認証のシナリオの例を示しています:



たとえば、RSA SecurID インフラストラクチャをカスタム・プロバイダとして使用し、EPM System への透過的で強力な認証を確保します。概要:

1. ユーザーはログイン情報(通常、ユーザー名とパスワード)を入力して EPM System 製品にアクセスします。これらのログイン情報は、カスタム認証モジュールで使用されるプロバイダに対してユーザーを一意に識別する必要があります。たとえば、RSA SecurID インフラストラクチャを使用してユーザーを認証する場合、ユーザーは RSA ユーザー ID と PIN(EPM System ユーザー ID およびパスワードではなく)を入力します。
2. 検索順序(118 ページの「検索順序」を参照)を使用すると、 EPM System は構成済ユーザー・ディレクトリ内を循環し、ユーザーを検索します。
   - 現在のユーザー・ディレクトリがカスタム認証用に構成されていない場合、EPM System は、EPM System 認証を使用してユーザーを検索し、認証しようとします。
   - ユーザー・ディレクトリがカスタム認証用に構成されている場合、EPM System は認証プロセスをカスタム・モジュールに委任します。
3. EPM System が認証をカスタム・モジュールに委任した場合、カスタム認証モジュールはログイン情報を受け入れ、その独自のロジックを使用してカスタム・プロバイダ(RSA SecurID インフラストラクチャなど)に対してユーザー認証を送ります。
4. カスタム認証モジュールでそのプロバイダに対してユーザーが認証される場合、ユーザー名が EPM System に戻されるか、Java 例外が戻されます。

   カスタム認証モジュールで戻されるユーザー名は、カスタム認証で使用可能なユーザー・ディレクトリの 1 つのユーザー名と同一である必要があります。
   - カスタム認証モジュールでユーザー名が戻された場合、EPM System は、カスタム認証で使用可能なユーザー・ディレクトリでユーザーを検索しま

す。この段階では、EPM System はカスタム認証用に構成されていないユーザー・ディレクトリは検索しません。

- カスタム認証モジュールで例外がスローされるか、null ユーザーが戻された場合、EPM System は、カスタム認証が使用可能になっていない、検索順序の残りのユーザー・ディレクトリ内でユーザーの検索を続行します。ログイン情報が一致するユーザーが見つからない場合、EPM System にエラーが表示されます。

使用事例の例と制限

カスタム認証の実装シナリオには、次のようなものがあります:

- ワンタイム・パスワード・サポートの追加
- Resource Access Control Facility (RACF)に対する認証の実行
- 簡単な LDAP バインドのかわりに Simple Authentication and Security Layer (SASL) バインドを LDAP 対応ユーザー・ディレクトリに追加

チャレンジ/応答メカニズムの認証は、カスタム認証モジュールを実装している場合、うまく機能しない可能性があります。カスタム認証モジュールによってスローされたカスタム・メッセージは、クライアントに伝播されません。クライアント(EPM Workspace など)が通常のメッセージを表示するためにエラー・メッセージをオーバーライドするので、次のシナリオは有効ではありません:

- 2 つの連続する RSA SecurID PIN
- チャレンジのパスワード変形(パスワードの最初、最後および 3 番目の文字の入力など)

前提条件

- `CustomAuth.jar` という完全にテストされた Java アーカイブには、カスタム認証モジュール・ライブラリが含まれます。`CustomAuth.jar` は、標準 Shared Services API の一部として `com.hyperion.css` パッケージで定義される、パブリック・インタフェース `CSSCustomAuthenticationIF` を実装する必要があります。http://download.oracle.com/docs/cd/E12825_01/epm.111/epm_security_api_11111/client/com/hyperion/css/CSSCustomAuthenticationIF.html を参照してください。
- Shared Services 管理者としての Shared Services へのアクセス権

設計およびコーディングに関する考慮事項

サブトピック

検索順序

ネイティブ・ディレクトリ以外に、複数のユーザー・ディレクトリを Shared Services に構成することができます。デフォルトの検索順序の位置は、すべての構成済ユーザー・ディレクトリに割り当てられます。検索順序を Shared Services Console から変更できます。ネイティブ・ディレクトリを除き、構成済ユーザー・ディレクトリは検索順序から除去できます。EPM System では検索順序に含まれていないユーザー・ディレクトリは使用されません。Oracle Enterprise Performance Management System User Security Administration Guide を参照してください。

検索順序により、ユーザーの認証のために EPM System がユーザー・ディレクトリ内を循環する順序が決定されます。ユーザーがユーザー・ディレクトリ内で認証されている場合、EPM System は検索を停止し、ユーザーを戻します。ユーザーが検索順序内のユーザー・ディレクトリに対して認証されていない場合、EPM System は認証を拒否してエラーを戻します。

検索順序でのカスタム認証の影響

カスタム認証は、EPM System セキュリティによる検索順序の解釈に影響を及ぼします。

カスタム認証モジュールでユーザー名が戻された場合、EPM System は、カスタム認証で使用可能なユーザー・ディレクトリのみでユーザーを検索します。この段階では、EPM System はカスタム認証用に構成されていないユーザー・ディレクトリを無視します。

カスタム認証のフローについて

次の使用事例シナリオを、カスタム認証のフローを調査するために使用します:

- 118 ページの「使用事例シナリオ 1」
- 120 ページの「使用事例シナリオ 2」
- 120 ページの「使用事例シナリオ 3」

使用事例シナリオ 1

表 19 は、このシナリオで使用される EPM System ユーザー・ディレクトリの構成と検索順序の詳細です。このシナリオでは、カスタム認証モジュールが RSA インフラストラクチャを使用してユーザーを認証すると仮定します。

表 19　シナリオ 1 の設定

ユーザー・ディレクトリのタイプと名前	検索順序	カスタム認証	サンプル・ユーザー名	パスワード [1]
ネイティブ・ディレクトリ	1	使用不可	test_user_1 test_user_2 test_user_3	password
LDAP 対応 SunONE_West	2	使用不可	test_ldap1 test_ldap_2 test_user_3 test_ldap_4	ldappassword
LDAP 対応 SunONE_East	3	使用可能	test_ldap1 test_ldap_2 test_user_3	SunONE では ldappassword、カスタム・モジュールでは RSA PIN

[1] 単純化するため、すべてのユーザーが同じユーザー・ディレクトリ・パスワードを使用すると仮定します。

認証プロセスを開始するには、ユーザーは EPM System 製品のログオン画面でユーザー名とパスワードを入力します。

このシナリオでは、カスタム認証モジュールは次のアクションを実行します:

- ユーザー名と RSA PIN をユーザー・ログイン情報として受け入れます
- ユーザー名を username@providername 形式(たとえば、test_ldap_2@SunONE_East)で EPM System セキュリティに戻します。

表 20　ユーザーのやりとりと結果

ユーザー名およびパスワード	認証結果	ログイン・ユーザー・ディレクトリ
test_user_1/password	成功	ネイティブ・ディレクトリ
test_user_3/password	成功	ネイティブ・ディレクトリ
test_user_3/ldappassword	成功	SunONE_West (検索順序 2)[1]
test_user_3/RSA PIN	成功	SunONE_East (検索順序 3)[2]
test_ldap_2/ldappassword	成功	SunONE_West (検索順序 2)
test_ldap_4/RSA PIN	失敗 EPM System に認証エラーが表示されます。[3]	

[1] ユーザーは EPM System ログイン情報を入力したので、カスタム認証ではこのユーザーは認証できません。EPM System はカスタム認証で使用可能でないユーザー・ディレクトリでのみこのユーザーを識別できます。ユーザーはネイティブ・ディレクトリ(検索順序番号 1)にはなく、SunONE West(検索順序番号 2)で識別されます。

[2]EPM System は、このユーザーをネイティブ・ディレクトリ(検索順序番号 1)または SunONE West(検索順序番号 2)で見つけられません。カスタム認証モジュールでは RSA サーバーに対してユーザーを検証し、test_user_3@SunONE_EAST を EPM System に戻します。EPM System はユーザーを SunONE East(検索順序番号 3)で検索します。これはカスタム認証が有効なディレクトリです。

[3] カスタム・モジュールで認証されているユーザーはすべて、検索順序に含まれるカスタム認証が有効なユーザー・ディレクトリに含めることをお薦めします。カスタム認証モジュールで戻されるユーザー名が、検索順序に含まれるカスタム認証が有効なユーザー・ディレクトリにない場合、ログインは失敗します。

使用事例シナリオ 2

表 21 は、このシナリオで使用される EPM System ユーザー・ディレクトリの構成と検索順序の詳細です。このシナリオでは、カスタム認証モジュールが RSA インフラストラクチャを使用してユーザーを認証すると仮定します。

このシナリオでは、カスタム認証モジュールは次のアクションを実行します:

- ユーザー名と RSA PIN をユーザー・ログイン情報として受け入れます
- ユーザー名(たとえば、test_ldap_2)を EPM System セキュリティに戻します。

表 21　検索順序の例

ユーザー・ディレクトリ	検索順序	カスタム認証	サンプル・ユーザー名	パスワード [1]
ネイティブ・ディレクトリ	1	使用不可	test_user_1 test_user_2 test_user_3	password
LDAP 対応(たとえば、SunONE)	2	使用可能	test_ldap1 test_ldap2 test_user_3	SunONE では ldappassword、カスタム・モジュールでは RSA PIN

[1] 単純化するため、すべてのユーザーが同じユーザー・ディレクトリ・パスワードを使用すると仮定します。

認証プロセスを開始するには、ユーザーは EPM System 製品のログイン画面でユーザー名とパスワードを入力します。

表 22　ユーザーのやりとりと結果

ユーザー名およびパスワード	ログイン結果	ログイン・ユーザー・ディレクトリ
test_user_1/password	成功	ネイティブ・ディレクトリ
test_user_3/password	成功	ネイティブ・ディレクトリ
test_user_3/ldappassword	失敗	SunONE[1]
test_user_3/RSA PIN	成功	SunONE[2]

[1] ネイティブ・ディレクトリに対するユーザーの認証は、パスワードが一致していないために失敗します。カスタム認証モジュールを使用したユーザーの認証は、使用されたパスワードが有効な RSA PIN ではないため失敗します。EPM System は、カスタム認証設定がこのディレクトリの EPM System 認証をオーバーライドしたため、SunONE (検索順序 2)でこのユーザーの認証を試行しません。

[2] ネイティブ・ディレクトリに対するユーザーの認証は、パスワードが一致していないために失敗します。カスタム認証モジュールによりユーザーが認証され、ユーザー名 test_user_3 が EPM System に戻されます。

使用事例シナリオ 3

表 23 は、このシナリオで使用される EPM System ユーザー・ディレクトリの構成と検索順序の詳細です。このシナリオでは、カスタム認証モジュールが RSA インフラストラクチャを使用してユーザーを認証すると仮定します。

このようなシナリオの明確さのため、カスタム認証モジュールがユーザー名を username@providername 形式(たとえば、test_ldap_4@SunONE)で戻すことをお薦めします。

表 23　検索順序の例

ユーザー・ディレクトリ	検索順序	カスタム認証	サンプル・ユーザー名	パスワード[1]
ネイティブ・ディレクトリ	1	使用可能	test_user_1 test_user_2 test_user_3	RSA_PIN
LDAP 対応(たとえば、MSAD)	2	使用不可	test_ldap1 test_ldap4 test_user_3	ldappassword
LDAP 対応(たとえば、SunONE)	3	使用可能	test_ldap1 test_ldap4 test_user_3	SunONE では ldappassword、カスタム・モジュールでは RSA PIN

[1] 単純化するため、すべてのユーザーが同じユーザー・ディレクトリ・パスワードを使用すると仮定します。

認証プロセスを開始するには、ユーザーは EPM System 製品のログオン画面でユーザー名とパスワードを入力します。

表 24　ユーザーのやりとりと結果

ユーザー名およびパスワード	認証結果	ログイン・ユーザー・ディレクトリ
test_user_1/password	成功	ネイティブ・ディレクトリ
test_user_3/RSA_PIN	成功	ネイティブ・ディレクトリ
test_user_3/ldappassword	成功	MSAD (検索順序 2)
test_ldap_4/ldappassword	成功	MSAD (検索順序 2)
test_ldap_4/RSA PIN	成功	SunONE (検索順序 3)

ユーザー・ディレクトリおよびカスタム認証モジュール

カスタム認証モジュールを使用するには、EPM System ユーザーおよびグループ情報を含むユーザー・ディレクトリを、カスタム・モジュールに認証を委任するよう個別に構成できます。

カスタム・モジュールを使用して認証された EPM System ユーザーは、検索順序(118 ページの「検索順序」を参照)に含まれたユーザー・ディレクトリの 1 つに含まれている必要があります。また、ユーザー・ディレクトリは、認証をカスタム・モジュールに委任するよう構成されている必要があります。

カスタム・プロバイダのユーザーの ID(たとえば、RSA SecurID インフラストラクチャの 1357642)は、Shared Services で構成されるユーザー・ディレクトリのユーザー名(たとえば、Oracle Internet Directory の jDoe)と異なる場合があります。ユーザーの認証後、カスタム認証モジュールは、ユーザー名 jDoe を EPM System に戻す必要があります。

注： ベスト・プラクティスとして、EPM System で構成されるユーザー・ディレクトリのユーザー名は、カスタム認証モジュールで使用されるユーザー・ディレクトリで使用可能なものと同一にすることをお薦めします。

CSSCustomAuthenticationIF Java インタフェース

カスタム認証モジュールは、EPM System セキュリティ・フレームワークとの統合に CSSCustomAuthenticationIF Java インタフェースを使用する必要があります。カスタム認証が成功した場合はユーザー名の文字列を、認証が失敗した場合はエラー・メッセージを戻す必要があります。認証プロセスが完了した場合、カスタム認証モジュールによって戻されたユーザー名は、Shared Services 検索順序に含まれるユーザー・ディレクトリの 1 つに存在する必要があります。EPM System セキュリティ・フレームワークでは、username@providerName 形式がサポートされています。

注： カスタム認証モジュールが戻すユーザー名に* (アスタリスク)を含めないでください。EPM System セキュリティ・フレームワークがユーザーの検索中にワイルドカード文字と解釈します。

CSSCustomAuthenticationIF インタフェース・シグネチャについては、133 ページの「サンプル・コード 1」を参照してください。

使用するカスタム認証モジュールは、CustomAuth.jar に含める必要があるクラス・ファイルにできます。パッケージ構造は重要ではありません。

CSSCustomAuthenticationIF インタフェースの詳細は、セキュリティ API のドキュメントを参照してください。

CSSCustomAuthenticationIF の authenticate メソッドではカスタム認証がサポートされます。authenticate メソッドは、EPM System にアクセスしようとする際にユーザーが入力したログイン情報(ユーザー名とパスワード)を入力パラメータとして受け入れます。このメソッドは、カスタム認証が成功した場合に文字列(ユーザー名)を戻します。認証に失敗した場合は java.lang.Exception をスローします。メソッドにより戻されるユーザー名は、Shared Services 検索順序に含まれるユーザー・ディレクトリの 1 つでユーザーを一意に識別する必要があります。EPM System セキュリティ・フレームワークでは、username@providerName 形式がサポートされています。

注： リソース(たとえば、JDBC 接続プール)を初期化するには、クラス・コンストラクタを使用します。これにより、認証のたびにリソースをロードすることがなくなり、パフォーマンスが向上します。

カスタム認証モジュールの配置

サブトピック

EPM System 配置には、1 つのカスタム・モジュールのみサポートされます。検索順序の 1 つ以上のユーザー・ディレクトリに付いてカスタム認証を有効にできます。

カスタム認証モジュールは、`com.hyperion.css` パッケージで定義されるパブリック・インタフェース `CSSCustomAuthenticationIF` を実装する必要があります。このドキュメントでは、選択したユーザー・プロバイダに対してユーザーを認証するロジックを定義する完全な機能のカスタム・モジュールを持っていることを前提としています。カスタム認証モジュールを開発およびテストした後、EPM System 環境で実装する必要があります。

手順の概要

カスタム認証モジュールを実装するには、次の手順を実行します:

- EPM System 製品を停止します。これには、Shared Services と Shared Services API を使用するシステムが含まれます。
- カスタム認証モジュールの Java アーカイブ `CustomAuth.jar` を `EPM_ORACLE_HOME/common/jlib/11.1.2.0` にコピーします。

  注： カスタム認証コードではエラー・ロギングに log4j を使用しないでください。以前のリリースで使用したコードで log4j を使用している場合は、このリリースで使用する前に、コードから削除する必要があります。
- カスタム認証モジュール Java アーカイブ(`CustomAuth.jar`)を `MIDDLEWARE_HOME/user_projects/domains/WebLogic_DOMAIN/lib`(通常 `Oracle/Middleware/user_projects/domains/EPMSystem/lib`)にコピーします。
- Shared Services のユーザー・ディレクトリ設定を更新します。124 ページの「Shared Services での設定の更新」を参照してください。
- Shared Services を開始してからその他の EPM System 製品を開始します。
- 実装をテストします。125 ページの「配置のテスト」を参照してください。

Shared Services での設定の更新

サブトピック

デフォルトでは、カスタム認証は、すべてのユーザー・ディレクトリで使用不可です。デフォルトの動作をオーバーライドして、特定の外部ユーザー・ディレクトリまたはネイティブ・ディレクトリに対して、カスタム認証を使用可能にできます。

ユーザー・ディレクトリ構成の更新

カスタム認証を使用可能にするユーザー・ディレクトリの構成を更新する必要があります。

➤ ユーザー・ディレクトリ構成を更新するには:

1 Foundation Services を起動します。

2 システム管理者として Shared Services Console にアクセスします。

3 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

4 「定義済ユーザー・ディレクトリ」画面で、カスタム認証設定を変更するユーザー・ディレクトリを選択します。

注： EPM System は、検索順序に含まれたユーザー・ディレクトリのみを使用します。

5 「編集」をクリックします。

6 「詳細オプションの表示」を選択します。

7 「カスタム・モジュール」で、「認証モジュール」を選択し、現在のユーザー・ディレクトリに対してカスタム・モジュールを使用可能にします。

8 「終了」をクリックします。

9 この手順を繰り返して、検索順序に含まれる他のユーザー・ディレクトリの構成を更新します。

セキュリティ・オプションの更新

`CustomAuth.jar` が `EPM_ORACLE_HOME/common/jlib/11.1.2.0` にあることを確認してから次の手順を開始してください。

➤ セキュリティ・オプションを更新するには:

1 システム管理者として Shared Services Console にアクセスします。

2 「管理」、「ユーザー・ディレクトリの構成」の順に選択します。

3 「セキュリティ・オプション」を選択します。

4 「詳細オプションの表示」を選択します。

5 「認証モジュール」で、カスタム認証モジュールが選択されるすべてのユーザー・ディレクトリでユーザーの認証に使用されるカスタム認証モジュールの完全修飾クラス名を入力します。たとえば、`com.mycompany.epm.CustomAuthenticationImpl` です。

6 「OK」をクリックします。

配置のテスト

ネイティブ・ディレクトリがカスタム認証に対して構成されていない場合、カスタム認証のテストにネイティブ・ディレクトリ・ユーザーを使用しないでください。

注： カスタム認証モジュールの問題を識別して修正する必要があります。カスタム・モジュールで使用するユーザー・ディレクトリからのユーザーを EPM System 検索順序で使用可能なカスタム認証が有効なユーザー・ディレクトリのユーザーにマップするために、カスタム・モジュールがフレームレスに機能することを想定しています。

配置をテストするには、カスタム・モジュールで使用されるユーザー・ディレクトリ(たとえば、RSA SecurID インフラストラクチャ)からのユーザー・ログイン情報を使用して EPM System にログインします。これらのログイン情報は、EPM System のログイン情報と異なる場合があります。

EPM System 製品でリソースへのアクセスを許可された場合、実装は成功したと考えられます。ユーザーが見つからなかったというエラーが常に実装の失敗を示しているわけではありません。このような場合、入力したログイン情報がカスタム・ユーザー・ストアに存在するか、一致するユーザーが EPM System 検索順序のカスタム認証が有効なユーザー・ディレクトリの 1 つに存在するかを確認してください。

➤ カスタム認証をテストするには:

1 EPM System 製品が実行されていることを確認します。

2 EPM System コンポーネント、たとえば EPM Workspace にアクセスします。

3 カスタム認証が有効なユーザーディレクトリで定義されているユーザーとしてログインします。

   1. 「ユーザー名」に、ユーザー ID(たとえば、RSA ユーザー ID)を入力します。
   2. 「パスワード」に、パスワード(たとえば、RSA PIN)を入力します。
   3. 「ログイン」をクリックします。

4 EPM System 製品のリソースにアクセスできたことを確認します。

6 EPM Systemの保護のガイドライン

この章の内容

SSL の実装

SSL では、データを暗号化する暗号システムを使用します。SSL は、データを安全に送信できるクライアントとサーバー間の安全な接続を作成します。

EPM System 環境をセキュリティ保護するには、Web アプリケーションおよびユーザー・ディレクトリ接続で使用されるすべての通信チャネルを、SSL を使用して保護します。第 2 章「EPM System コンポーネントの SSL 使用可能化」を参照してください。

管理パスワードの変更

デフォルトのネイティブ・ディレクトリ管理ユーザー・アカウントでは、すべての Shared Services 機能へのアクセスが提供されます。このパスワードは、Foundation Services の配置時に設定されます。このアカウントのパスワードを定期的に変更する必要があります。

パスワードを変更するには、admin ユーザー・アカウントを編集します。Oracle Enterprise Performance Management System User Security Administration Guide のユーザー・アカウントの変更に関する項を参照してください。

暗号化鍵の再生成

Shared Services Console を使用して、次のものを定期的に再生成します:

- シングル・サインオン・トークン

> **注意** Financial Management、Oracle Hyperion EPM Architect、および Oracle Hyperion Profitability and Cost Management で使用されるタスクフローは、新しいキーストアの生成時には無効化されています。キーストアを再生成した後に、タスクフローを開いて保存すると、タスクフローは再度有効化されます。

- 信頼できるサービス・キー
- プロバイダ構成キー

Oracle Enterprise Performance Management System User Security Administration Guide の暗号化オプションの設定に関する項を参照してください。

データベース・パスワードの変更

EPM System 製品のすべてのデータベースのパスワードを定期的に変更します。Shared Services レジストリでデータベースのパスワードを変更する手順の詳細は、この項で説明します。

EPM System 製品のデータベース・パスワードを変更する詳細手順は、Oracle Enterprise Performance Management System Installation and Configuration Guide を参照してください。

➤ EPM System 製品のデータベースのパスワードを Shared Services レジストリで変更するには:

1 **データベース管理コンソールを使用して、EPM System 製品のデータベースの構成に使用したアカウントを持つユーザーのパスワードを変更します。**

2 **EPM System 製品(Web アプリケーション、サービスおよびプロセス)を停止します。**

3 **EPM System コンフィグレータを使用して、次の手順のいずれかを実行してデータベースを再構成します。**

Shared Services のみ:

> **注:** EPM System 製品が Shared Services と異なるマシンに存在する分散環境では、すべてのサーバーでこの手順を実行する必要があります。

1. EPM System コンフィグレータの Foundation タスクから「データベースの構成」を選択します。
2. 「Shared Services およびレジストリ・データベース構成」ページで、「前に構成された Shared Services データベースに接続」を選択します。
3. Shared Services データベースを構成するのに使用したアカウントを持つユーザーの新パスワードを指定します。他の設定は変更しないでください。

4. 構成を続行し、完了したら「終了」をクリックします。

Shared Services 以外の EPM System 製品:

注： 現在のサーバーに配置されている EPM System 製品に対しのみ、次の手順を行います。

1. EPM System コンフィグレータの製品の構成タスク・リストから、「データベースの構成」を選択します。
2. 「データベースの構成」ページで、「データベースの初回構成を実行」を選択します。
3. EPM System 製品のデータベースを構成するのに使用したアカウントを持つユーザーの新パスワードを指定します。他の設定は変更しないでください。
4. 「次へ」をクリックします。
5. 「既存のデータベースを再使用します」を選択します。
6. 構成を続行し、完了したら「終了」をクリックします。

詳細手順は、Oracle Enterprise Performance Management System Installation and Configuration Guide を参照してください。

4 **EPM System 製品およびサービスを開始します。**

Cookie の保護

EPM System の Web アプリケーションは、cookie を設定してセッションを追跡します。特にセッションの cookie を設定しているとき、サーバーは保護フラグを設定できます。これにより、ブラウザは保護チャネルを介して cookie を送信できます。この動作で、セッションが乗っ取られる危険性が低くなります。

注： EPM System 製品が SSL 使用可能の環境に配置される場合のみ Cookie を保護します。

WebLogic Server セッションの記述子を変更して、WebLogic Server の Cookie を保護します。`session-param` 要素内の `cookieSecure` 属性の値を `TRUE` に設定します。

SSO トークンのタイムアウトの低減

SSO トークンのデフォルトのタイムアウトは 480 分です。SSO トークンのタイムアウトを、たとえば 60 分に縮小すると、表示されている場合はトークンの再利用を最小限にできます。Oracle Enterprise Performance Management System User Security Administration Guide のセキュリティ・オプションの設定に関する項を参照してください。

セキュリティ・レポートの確認

セキュリティ・レポートには、監査を構成しているセキュリティ・タスクに関する監査情報が含まれています。特に EPM System 製品で失敗したログイン試行とプロビジョニングの変更を識別するために、このレポートを Shared Services Console で定期的に生成し確認します。レポート生成オプションとして「詳細ビュー」を選択し、変更された属性と新しい属性値に基づいてレポート・データをグループ化します。Oracle Enterprise Performance Management System User Security Administration Guide のレポートの生成に関する項を参照してください。

認証システムの強力な認証としてのカスタマイズ

カスタム認証モジュールを使用して、EPM System に強力な認証を追加できます。たとえば、RSA SecurID two-factor 認証を nonchallenge 応答モードで使用できます。カスタム認証モジュールは、シン・クライアントおよびシック・クライアントに対して透過的であり、クライアント側の配置変更は必要ありません。第 5 章「カスタム認証モジュールの使用方法」を参照してください。

Financial Management の詳細なエラー・メッセージの非表示

技術情報を含む Financial Management の詳細なエラー・メッセージは、Windows レジストリ・エントリを更新することで、ユーザーに非表示にすることができます。

➤ 詳細な技術情報を含むエラー・メッセージを非表示にするには:

1. **Financial Management のホストとなる Windows サーバーで、Windows レジストリ・エディタを起動します。**
2. `HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Financial Management` **に移動します。**
3. **これらの設定を使用して、新しい DWORD 値を作成します:**

   値名: `DisableTechnicalError`

   値データ: `1`(`0` に設定すると詳細メッセージが表示されます)
4. **Financial Management のホストとなる IIS サーバーのホストとなる Windows サーバーで、Windows レジストリ・エディタを起動します。**
5. `HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Financial Management\web` **に移動します。**
6. **これらの設定を使用して、新しい DWORD 値を作成します:**

   値名: `DisableAspTechnicalErrorMessage`

   値データ: `1`(`0` に設定すると詳細メッセージが表示されます)

UDL ファイルの暗号化(Financial Management)

Financial Management の構成中に、EPM System コンフィグレータはデフォルトで暗号化されていない UDL ファイルを作成します。このファイルは、Oracle Hyperion Enterprise Performance Management System コンフィグレータの詳細データベース・オプション・ページのオプションを選択するか、構成の完了後に `EncryptHFMUDL` ユーティリティを実行することによって、暗号化できます。

Oracle Enterprise Performance Management System Installation and Configuration Guide の UDL ファイルの暗号化に関する項を参照してください。

EPM Workspace のデバッグ・ユーティリティを使用不可にする

- トラブルシューティングの目的で、EPM Workspace は未処理の JavaScript ファイルとともに出荷されます。セキュリティの目的で、これらの未処理の JavaScript ファイルを本番環境から除去する必要があります:
  - `EPM_ORACLE_HOME/common/epmstatic/wspace/js/`ディレクトリのバックアップ・コピーを作成します。
  - ファイル `DIRECTORY_NAME.js` を除き、`EPM_ORACLE_HOME/common/epmstatic/wspace/js` の各サブディレクトリから `.js` ファイルを削除します。

    各サブディレクトリには、ディレクトリの名前を持つ `.js` ファイルが含まれています。たとえば、`EPM_ORACLE_HOME/common/epmstatic/wspace/js/com/hyperion/bpm/web/common` には `Common.js` が含まれています。ディレクトリの名前を持つファイル(この場合は `Common.js`)以外のすべての `.js` ファイルを除去します。
- EPM Workspace では、EPM Workspace がデバッグ・モードで配置された場合にアクセス可能になるデバッグ・ユーティリティおよびテスト・アプリケーションを提供します。セキュリティの目的で、管理者は EPM Workspace のクライアント側のデバッグをオフにする必要があります。

  デバッグ・モードをオフにするには:

  1. Oracle Hyperion Enterprise Performance Management Workspace に管理者としてログインします。
  2. 「ナビゲート」、「管理者」、「Workspace サーバー設定」の順に選択します。
  3. 「Workspace サーバー設定」の「ClientDebugEnabled」で、「いいえ」を選択します。
  4. 「OK」をクリックします。

デフォルトの Web サーバー・エラー・ページの変更

アプリケーション・サーバーが要求を受け入れられないとき、バックエンド・アプリケーション・サーバーの Web サーバー・プラグイン(Oracle WebLogic Server の Oracle HTTP Server プラグインなど)はプラグインの構築情報が表示されたデフォルトのエラー・ページを戻します。Web サーバーはその他の場合も同様にデフォルトのエラー・ページを表示します。攻撃者は、この情報から公共の Web サイトの既知の脆弱性を知ることができます。

Web アプリケーション・サーバー・プラグインおよび Web サーバーのエラー・ページをカスタマイズして、サーバーのバージョン、サーバー・タイプ、プラグインの作成日、プラグイン・タイプなどの本番環境用システム・コンポーネントに関する情報が含まれないようにできます。詳細は、ご使用のアプリケーション・サーバーおよび Web サーバーのベンダーのドキュメントを参照してください。

サードパーティ製ソフトウェアのサポート

オラクル社は、サードパーティ・ベンダーが明言している下位互換性を了承し、サポートします。したがって、ベンダーが下位互換を明言している場合、その後のメンテナンス・リリースやサービス・パックを使用できます。互換性がないことがわかると、オラクル社では、製品を配置すべきパッチ・リリースを指定(およびサポート・マトリックスから互換性のないバージョンを削除)するか、その Oracle 製品のメンテナンス・リリースまたはサービス・フィックスを提供します。

サーバー側の更新: サードパーティ製サーバー側コンポーネントのアップグレードに関するサポートは将来のメンテナンス・リリースに関する方針に従います。通常、Oracle では、サードパーティ製サーバー側コンポーネントについて、現在サポートしているリリースのサービス・パックの次回メンテナンス・リリースへのアップグレードをサポートします。次回の主要リリースへのアップグレードはサポートされません。

クライアント側の更新: Oracle ではクライアント・コンポーネントの自動更新をサポートしています。これには、サードパーティ製クライアント・コンポーネントの次回主要リリースへの更新が含まれます。たとえば、ブラウザの JRE バージョンを 1.5 から 1.6 に更新できます。

A カスタム認証サンプル・コード

この付録の内容

サンプル・コード 1

注： カスタム認証コードではエラー・ロギングに log4j を使用しないでください。以前のリリースで使用したカスタム認証コードで log4j を使用している場合は、このリリースで使用する前に、コードから削除する必要があります。

次のコード・スニペットは、カスタム・モジュールの空の実装です:

```
package com.hyperion.css.custom;

import java.util.Map;
import com.hyperion.css.CSSCustomAuthenticationIF;

public class CustomAuthenticationImpl implements CSSCustomAuthenticationIF {
   public String authenticate(Map context,String userName,
               String password) throws Exception{
    try{
     //Custom code to find and authenticate the user goes here.
     //The code should do the following:
     //if authentication succeeds:
        //set authenticationSuccessFlag = true
        //return authenticatedUserName
     // if authentication fails:
        //log an authentication failure
        //throw authentication exception
    }
    catch (Exception e){
     //Custom code to handle authentication exception goes here
     //Create a new exception, set the root cause
     //Set any custom error message
     //Return the exception to the caller
    }
    return authenticatedUserName;
   }
```

```
}
```

入力パラメータ:

- コンテキスト: ロケール情報のキーと値のペアを含むマップ
- ユーザー名: カスタム・モジュールがユーザーを認証するユーザー・ディレクトリにユーザーを一意に識別する識別子。ユーザーは、EPM System コンポーネントにログインする際にこのパラメータの値を入力します。
- パスワード: カスタム・モジュールがユーザーを認証するユーザー・ディレクトリのユーザーのパスワード・セット。ユーザーは、EPM System コンポーネントにログインする際にこのパラメータの値を入力します。

サンプル・コード 2

次のサンプル・コードは、フラット・ファイルに含まれるユーザー名とパスワードを使用したユーザーのカスタム認証を示します。カスタム認証を機能させるには、クラス・コンストラクタ内のユーザーとパスワードのリストを初期化する必要があります。

```
   package com.hyperion.css.security;

import java.util.Map;
import java.util.HashMap;
import com.hyperion.css.CSSCustomAuthenticationIF;
import java.io.*;

public class CSSCustomAuthenticationImpl implements CSSCustomAuthenticationIF{
 static final String DATA_FILE = "datafile.txt";

/**
 * authenticate method includes the core implementation of the
 * Custom Authentication Mechanism. If custom authentication is
 * enabled for the provider, authentication operations
 * are delegated to this method. Upon successful authentication,
 * this method returns a valid user name, using which EPM System
 * retrieves the user from a custom authentication enabled provider.
 * User name can be returned in the format username@providerName,
 * where providerName indicates the name of the underlying provider
 * where the user is available. authenticate method can use other
 * private methods to access various core components of the
 * custom authentication module.

 *  @param context
 *  @param userName
 *  @param password
 *  @return
 *  @throws Exception
*/

Map users = null;

public CSSCustomAuthenticationImpl(){
```

```
 users = new HashMap();
 InputStream is = null;
 BufferedReader br = null;
 String line;
 String[] userDetails = null;
 String userKey = null;
 try{
  is = CSSCustomAuthenticationImpl.class.getResourceAsStream(DATA_FILE);
  br = new BufferedReader(new InputStreamReader(is));
  while(null != (line = br.readLine())){
   userDetails = line.split(":");
    if(userDetails != null && userDetails.length==3){
     userKey = userDetails[0]+ ":" + userDetails[1];
     users.put(userKey, userDetails[2]);
    }
   }
 }
 catch(Exception e){
    // log a message
 }
 finally{
  try{
   if(br != null) br.close();
   if(is != null) is.close();
  }
  catch(IOException ioe){
   ioe.printStackTrace();
  }
 }
}

/* Use this authenticate method snippet to return username from a flat file */

public String authenticate(Map context, String userName, String password) throws
Exception{
 //userName : user input for the userName
 //password : user input for password
 //context  : Map, can be used to additional information required by
 //      the custom authentication module.

 String authenticatedUserKey = userName + ":" + password;

 if(users.get(authenticatedUserKey)!=null)
  return(String)users.get(authenticatedUserKey);
 else throw new Exception("Invalid User Credentials");
    }

/* Refer to this authenticate method snippet to return username in
 username@providername format */

public String authenticate(Map context, String userName, String password) throws
Exception{

 //userName : user input for userName
 //password : user input for password
 //context  : Map can be used to additional information required by
 //      the custom authentication module.
```

```
 //Your code should uniquely identify the user in a custom provider and in a
configured
 //user directory in Shared Services. EPM Security expects you to append the
provider
 //name to the user name. Provider name must be identical to the name of a custom
 //authentication-enabled user directory specified in Shared Services.

 //If invalid arguments, return null or throw exception with appropriate message
 //set authenticationSuccessFlag = false

 String authenticatedUserKey = userName + ":" + password;
 if(users.get(authenticatedUserKey)!=null)
  String userNameStr = (new StringBuffer())
            .append((String)users.get(authenticatedUserKey))
            .append("@").append(PROVIDER_NAME).toString();
             return userNameStr;
 else throw new Exception("Invalid User Credentials");
    }
}
```

サンプル・コード 2 のデータ・ファイル

データ・ファイルが datafile.txt という名前(サンプル・コードで使用される名前)で、作成する Java アーカイブに含まれていることを確認してください。

サンプル・コード 2 (134 ページの「サンプル・コード 2」を参照)で実装されるカスタム認証モジュールをサポートするためにカスタム・ユーザー・ディレクトリとして使用されるフラット・ファイルのコンテンツとして次を使用します:

```
xyz:password:admin
test1:password:test1@LDAP1
test1:password:test1
test1@LDAP1:password:test1@LDAP1
test1@1:password:test1
user1:Password2:user1@SunONE1
user1_1:Password2:user1
user3:Password3:user3
DS_User1:Password123:DS_User1@MSAD1
DS_User1:Password123:DS_User1
DS_User1@1:Password123:DS_User1
```

ユーザー名を username@providername 形式で戻す予定の場合にカスタム・ユーザー・ディレクトリとして使用されるフラット・ファイルのコンテンツとして次を使用します:

```
xyz:password:admin
test1:password:test1
test1@1:password:test1
user1_1:Password2:user1
```

```
user3:Password3:user3
DS1_1G100U_User61_1:Password123:DS1_1G100U_User61
DS1_1G100U_User61_1@1:Password123:DS1_1G100U_User61
TUser:password:TUser
```

B カスタム・ログイン・クラスの実装

この付録の内容

EPM System は、com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl を提供して、X509 証明書からユーザー ID (DN)を抽出します。

DN 以外の証明書にある属性からユーザー ID を取得する必要がある場合、この付録で説明しているように、com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl に類似したカスタム・ログイン・クラスを開発および実装する必要があります。

カスタム・ログイン・クラス・サンプル・コード

このサンプル・コードは、デフォルトの com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl の実装を示しています。通常、この実装の parseCertificate(String sCertificate) メソッドをカスタマイズして、DN 以外の証明書属性からユーザー名を取得する必要があります。

```
   package com.hyperion.css.sso.agent;

import java.io.ByteArrayInputStream;
import java.io.UnsupportedEncodingException;
import java.security.Principal;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import com.hyperion.css.CSSSecurityAgentIF;
import com.hyperion.css.common.configuration.*;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
```

```
 * X509CertificateAuthImpl implements the CSSSecurityAgentIF interface It accepts
 * the X509 certificate of the authenticated user from the Web Server via a
 * header, parses the certificate, extracts the DN of the User and
 * authenticates the user.
 */
public class X509CertificateSecurityAgentImpl implements CSSSecurityAgentIF
{
  static final String IDENTITY_ATTR = "CN";
  String g_userDN = null;
  String g_userName = null;
  String hostAdrress= null;
  /**
   * Returns the User name (login name) of the authenticated user,
   * for example demouser. See CSS API documentation for more information
   */
  public String getUserName(HttpServletRequest req, HttpServletResponse res)
      throws Exception
  {
    hostAdrress = req.getServerName();
    String certStr = getCertificate(req);

    String sCert = prepareCertificate(certStr);

    /* Authenticate with a CN */
    parseCertificate(sCert);

    /* Authenticate if the Login Attribute is a DN */
    if (g_userName == null)
    {
      throw new Exception("User name not found");
    }
    return g_userName;
  }

  /**
   * Passing null since this is a trusted Security agent authentication
   * See Security API documentation for more information on CSSSecurityAgentIF
   */
  public String getPassword(HttpServletRequest req, HttpServletResponse res)
      throws Exception
  {
    return null;
  }

  /**
   * Get the Certificate sent by the Web Server in the HYPLOGIN header.
   * If you pass a different header nane from the Web server, change the
   * name in the method.
   */
  private String getCertificate(HttpServletRequest request)
  {
    String cStr = (String)request
        .getHeader(CSSConfigurationDefaults.HTTP_HEADER_HYPLOGIN);
    return cStr;
  }

  /**
```

```
  * The certificate sent by the Web server is a String.
  * Put a "\n" in place of whitespace so that the X509Certificate
  * java API can parse the certificate.
  */
  private String prepareCertificate(String gString)
  {
    String str1 = null;
    String str2 = null;

    str1 = gString.replace("-----BEGIN CERTIFICATE-----", "");
    str2 = str1.replace("-----END CERTIFICATE-----", "");
    String certStrWithNL = "-----BEGIN CERTIFICATE-----"
        + str2.replace(" ", "\n") + "-----END CERTIFICATE-----";
    return certStrWithNL;
  }

  /**
  * Parse the certificate
  * 1. Create X509Certificate using the certificateFactory
  * 2. Get the Principal object from the certificate
  * 3. Set the g_userDN to a certificate attribute value (DN in this sample)
  * 4. Parse the attribute (DN in this sample) to get a unique username
  */
  private void parseCertificate(String sCertificate) throws Exception
  {
    X509Certificate cert = null;
    String userID = null;
    try
    {
      X509Certificate clientCert = (X509Certificate)CertificateFactory
          .getInstance("X.509")
          .generateCertificate(
                   new ByteArrayInputStream(sCertificate
                       .getBytes("UTF-8")));
      if (clientCert != null)
      {
        Principal princDN = clientCert.getSubjectDN();
        String dnStr = princDN.getName();
        g_userDN = dnStr;
        int idx = dnStr.indexOf(",");
        userID = dnStr.substring(3, idx);
        g_userName = userID;
      }

    }
    catch (CertificateException ce)
    {
       throw ce;

    }
    catch (UnsupportedEncodingException uee)
    {
      throw uee;
    }
  } //end of getUserNameFromCert
}// end of class
```

カスタム・ログイン・クラスの配置

カスタム・ログイン・クラスを実装するには、次の手順を実行します:

- カスタム・ログイン・クラスを作成およびテストします。コードに log4j への参照がないことを確認します。139 ページの「カスタム・ログイン・クラス・サンプル・コード」を参照してください。

  カスタム・クラスには任意の名前を使用できます。
- カスタム・ログイン・クラスを CustomAuth.jar にパッケージします
- CustomAuth.jar を MIDDLEWARE_HOME/user_projects/domains/WebLogic_DOMAIN/lib(通常 Oracle/Middleware/user_projects/domains/EPMSystem/lib)にコピーします。
- CustomAuth.jar を EPM_ORACLE_HOME/common/jlib/11.1.2.0/にコピーします。
- カスタム・ログイン・クラスを使用する場合、クライアント証明書認証を有効にすることをお薦めします。

C ネイティブ・ディレクトリの更新ユーティリティの使用方法

この付録の内容

ネイティブ・ディレクトリの更新ユーティリティについて

ネイティブ・ディレクトリには、様々な外部ユーザー・ディレクトリに定義されたユーザーとグループの ID を参照する情報が含まれています。たとえば、ネイティブ・ディレクトリ・グループには、外部ユーザー・ディレクトリに定義されたユーザーが含まれます。ユーザー・アカウントの削除またはユーザーの外部ユーザー・ディレクトリから別のディレクトリへの移行などの、外部ユーザー・ディレクトリでの変更(付録 D「ユーザー・ディレクトリ全体のユーザーとグループの移行」を参照)は、EPM System セキュリティがこのような変更を認識するように同期化されないため、ネイティブ・ディレクトリ内のデータの陳腐化を引き起こす可能性があります。このような場合、ネイティブ・ディレクトリの更新ユーティリティを使用して陳腐化したデータを特定し、ネイティブ・ディレクトリから削除します。

ネイティブ・ディレクトリの更新ユーティリティのインストール場所

ネイティブ・ディレクトリの更新ユーティリティは、Windows サーバーの `EPM_ORACLE_HOME/common/utilities/UpdateNativeDir` (`C:\Oracle\Middleware\EPMSystem11R1\common\utilities\UpdateNativeDir` など)にインストールされます。

ネイティブ・ディレクトリの更新ユーティリティのオプション

ネイティブ・ディレクトリの更新ユーティリティは、ログ・ファイルと CSS_MIGRATION_DELETE_LIST.csv を作成します。147 ページの「ネイティブ・ディレクトリの更新ユーティリティによって生成されるログ・ファイル」を参照してください。

注意 ネイティブ・ディレクトリの更新ユーティリティでは、検索順に含まれていないユーザー・ディレクトリからのユーザーおよびグループのプロビジョニング・データは陳腐化したデータとみなされます。このようなデータを保持する必要がある場合、データを CSS_MIGRATION_DELETE_LIST.csv から削除する必要があります。

表 25 ネイティブ・ディレクトリの更新ユーティリティのコマンドライン・オプション

オプション	説明
-noprompt	**オプション:** このオプションを使用して、サイレント・モードの操作を起動します。ネイティブ・ディレクトリの更新ユーティリティを使用するジョブのスケジュールに使用されます。 **例:** updateNativeDir -noprompt では、ネイティブ・ディレクトリがサイレント・モードで更新されます。
-delete all	**オプション:** このオプションを使用して、削除とマークされている陳腐化したネイティブ・ディレクトリ ID をすべて削除します。
-delete PATH_OF_DELETE_LIST	**オプション:** このオプションを使用して、CSS_MIGRATION_DELETE_LIST.csv にリストされている陳腐化したネイティブ・ディレクトリ ID を削除します。145 ページの「陳腐化したデータの特定」を参照してください。
-cssLocation	**オプション:** このオプションを使用して、EPM セキュリティ構成ファイルの絶対パスを指定します。このオプションを指定しない場合、ユーティリティは Shared Services レジストリで使用可能なセキュリティ構成ファイルを使用して初期化します。

ネイティブ・ディレクトリの更新ユーティリティの使用

通常、次の手順を実行して、陳腐化したネイティブ・ディレクトリ・データを管理します:

- 145 ページの「陳腐化したデータの特定」
- 146 ページの「陳腐化したデータの削除」
- 不明確な ID を解決します。不明確な ID は、ユーティリティで解決できなかった ID です。これらの ID は手動で解決する必要があります。

ネイティブ・ディレクトリの更新ユーティリティの設定の更新

EPM_ORACLE_HOME/common/utilities/UpdateNativeDir にある updateNativeDir.bat (Windows)または updateNativeDir.sh (UNIX)内のパラメータ値を変更します。

➤ ユーティリティの設定を更新するには:

1 テキスト・エディタを使用して、EPM_ORACLE_HOME/common/utilities/UpdateNativeDir にある updateNativeDir.bat (Windows)または updateNativeDir.sh (UNIX)を開きます。

2 環境内のインスタンスの場所を反映して EPM_ORACLE_INSTANCE の値を更新します。デフォルトでは、EPM_ORACLE_INSTANCE は C:\Oracle\Middleware\user_projects\epmsystem1 (Windows)です。

3 ファイルを保存して閉じます。

陳腐化したデータの特定

オプションを指定せずにネイティブ・ディレクトリの更新ユーティリティを実行し、ネイティブ・ディレクトリ内の削除可能な陳腐化した ID を示す CSS_MIGRATION_DELETE_LIST.csv を生成します。

➤ 陳腐化したネイティブ・ディレクトリ・データを特定するには:

1 ネイティブ・ディレクトリの更新ユーティリティの設定を変更します。145 ページの「ネイティブ・ディレクトリの更新ユーティリティの設定の更新」を参照してください。

2 EPM System コンポーネントをホストするサーバーでコマンド・プロンプト・ウィンドウまたはコンソールを使用し、Windows サーバーの EPM_ORACLE_HOME/common/utilities/UpdateNativeDir (C:\Oracle\Middleware\EPMSystem11R1\common\utilities\UpdateNativeDir など)に移動します。

3 コマンドを実行します:

- updateNativeDir -cssLocation LOCATION_OF_CSS.xml (Windows)
- updateNativeDir.sh -cssLocation LOCATION_OF_CSS.xml (UNIX)

前述のコマンドで、LOCATION_OF_CSS.xml は、Oracle Hyperion Shared Services レジストリから生成した CSS.xml の絶対パスを表します。たとえば、Windows サーバー上の C:\CSS.xml などです。

4 ネイティブ・ディレクトリの更新ユーティリティからの次の問合せに対し、1 と入力します:

```
Do you want to proceed? [0->No/1->Yes] :
```

陳腐化したデータの削除

陳腐化したデータを削除する前に、CSS_MIGRATION_DELETE_LIST.csv の内容を確認します。145 ページの「陳腐化したデータの特定」を参照してください。

注： ネイティブ・ディレクトリの更新ユーティリティでは、接続を確立できない外部ユーザー・ディレクトリから参照されている陳腐化したネイティブ・ディレクトリ ID は削除されません。

注意 削除処理では、Shared Services の検索順に含まれていない外部ユーザー・ディレクトリからのユーザーおよびグループを参照するプロビジョニング・データは削除されます。

➤ 陳腐化したネイティブ・ディレクトリ・データを削除するには:

1. ネイティブ・ディレクトリの更新ユーティリティの設定を変更します。145 ページの「ネイティブ・ディレクトリの更新ユーティリティの設定の更新」を参照してください。
2. EPM System コンポーネントをホストするサーバーでコマンド・プロンプト・ウィンドウまたはコンソールを使用し、Windows サーバーの EPM_ORACLE_HOME/common/utilities/UpdateNativeDir (C:\Oracle\Middleware\EPMSystem11R1\common\utilities\UpdateNativeDir など)に移動します。
3. コマンドを実行します。指定できるオプションのリストは、144 ページの「ネイティブ・ディレクトリの更新ユーティリティのオプション」を参照してください。

   注： -noprompt および-cssLocation ディレクティブを次のコマンドと組み合せることができます。

   - updateNativeDir -delete PATH_OF_DELETE_LIST
   - updateNativeDir -delete all

   このコマンドで、PATH_OF_DELETE_LIST は CSS_MIGRATION_DELETE_LIST.csv の絶対パスを指します;たとえば、Windows サーバー上の C:\Oracle\Middleware\EPMSystem11R1\common\utilities\UpdateNativeDir\logs\security-migration\CSS_MIGRATION_DELETE_LIST.csv などです。
4. ネイティブ・ディレクトリの更新ユーティリティからの次の問合せに対し、1 と入力します:

```
Do you want to proceed? 0->No/1->Yes] :
```

ネイティブ・ディレクトリの更新ユーティリティによって生成されるログ・ファイル

デフォルトでは、ネイティブ・ディレクトリの更新ユーティリティで`EPM_ORACLE_HOME/common/utilities/UpdateNativeDir/logs/security-migration`にログ・ファイルが作成されます。

- `CSSMigration-Ambiguous_time_stamp.log`には、ネイティブ・ディレクトリの更新ユーティリティで解決できなかった不明確なIDがリストされます。このファイルにリストされているIDは手動で更新する必要があります。
- `CSSMigration-Deleted_time_stamp.log`には、ネイティブ・ディレクトリの更新ユーティリティでネイティブ・ディレクトリから削除したIDがリストされます。
- `CSSMigration-Updated_time_stamp.log`には、外部ユーザー・ディレクトリでのIDに対する変更を反映してネイティブ・ディレクトリの更新ユーティリティで更新したネイティブ・ディレクトリIDがリストされます。
- `CSSMigration-Ignored_time_stamp.log`には、更新の必要がなかったため、アクションがなにもとられなかったエントリがリストされます。

D ユーザー・ディレクトリ全体のユーザーとグループの移行

この付録の内容

概要

プロビジョニングされた EPM System ユーザーのユーザー ID およびグループ ID を陳腐化させる可能性のあるシナリオが数多くあります。EPM System コンポーネントは、コンポーネントで使用可能なプロビジョニング情報が陳腐化すると、アクセスできなくなります。陳腐化したプロビジョニング・データが作成される可能性のあるシナリオは次のとおりです:

- ユーザー・ディレクトリの処分: 組織でユーザーを別のユーザー・ディレクトリに移動した後、元のユーザー・ディレクトリを処分する場合があります。
- バージョンのアップグレード: ユーザー・ディレクトリのバージョンをアップグレードすると、ホスト・マシン名またはオペレーティング・システム環境の要件が変わる場合があります。
- ベンダーの変更: 組織で別のベンダーのユーザー・ディレクトリを使用することにしたため、元のユーザー・ディレクトリの使用を打ち切る場合があります。たとえば、組織で Oracle Internet Directory を SunONE Directory Server に切り替えたとします。

注： この付録では、廃止予定のユーザー・ディレクトリをソース・ユーザー・ディレクトリと呼び、ユーザー・アカウントの移行先のユーザー・ディレクトリをターゲット・ユーザー・ディレクトリと呼びます。

前提条件

- プロビジョニング・データがユーザー・ディレクトリ間で移行される EPM System ユーザーとグループは、ターゲット・ユーザー・ディレクトリで使用可能である必要があります。

  ソース・ユーザー・ディレクトリ内にあるグループ関係は、ターゲット・ユーザー・ディレクトリで保持される必要があります。

- EPM System ユーザーのユーザー名は、ソース・ユーザー・ディレクトリとターゲット・ユーザー・ディレクトリで同一である必要があります。

移行手順

サブトピック

ネイティブ・ディレクトリ・データのエクスポート

Oracle Hyperion Enterprise Performance Management System ライフサイクル管理を使用して、ネイティブ・ディレクトリから次のデータをエクスポートします:

- ネイティブ・ディレクトリ・グループ
- 割り当てられている役割
- 委任リスト

ライフサイクル管理では、通常、`EPM_ORACLE_INSTANCE/import_export/USER_NAME/EXPORT_DIR/resource/Native Directory` に複数のエクスポート・ファイルを作成します(`USER_NAME` は、`admin@Native Directory` などのエクスポート操作を行うユーザーの ID で、`EXPORT_DIR` は、エクスポート・ディレクトリの名前)。通常、次のファイルが作成されます:

- `Groups.csv`
- `Roles.csv`
- `Delegated Lists.csv`
- 配置されているアプリケーションごとの `Assigned Roles/PROD_NAME.csv` (`PROD_NAME` は、`Shared Services` などの EPM System コンポーネントの名前)。

注: ライフサイクル管理を使用したデータのエクスポートの詳細な手順については、Oracle Enterprise Performance Management System Lifecycle Management Guide を参照してください。

➤ プロビジョニング・データをネイティブ・ディレクトリからエクスポートするには:

1. **Shared Services Console のビュー・ペインで、「Foundation」アプリケーション・グループ内の「Shared Services」アプリケーションを選択します。**

2 プロビジョニング情報をエクスポートするアーティファクトのタイプを選択します。

3 「移行の定義」を選択します。

4 ソース・オプションを設定し、「次へ」をクリックします。

5 エクスポート・ファイルを格納するファイル・システムの場所を入力し、「次へ」をクリックします。

6 「宛先オプション」で「次へ」をクリックします。

7 「移行の実行」をクリックします。

EPM System の移行の準備

サブトピック

外部ユーザー・ディレクトリとしてのターゲット・ユーザー・ディレクトリの追加

ソース・ユーザー・ディレクトリから別のユーザー・ディレクトリにユーザー・アカウントを移行した場合、EPM System でターゲット・ユーザー・ディレクトリを外部ユーザー・ディレクトリとして追加します。たとえば、Oracle Internet Directory から SunONE Directory Server にユーザー・アカウントを移行した場合、SunONE Directory Server を外部ユーザー・ディレクトリとして追加します。Oracle Enterprise Performance Management System User Security Administration Guide の第 3 章、ユーザー・ディレクトリの構成の章を参照してください。

注： データをソース・ユーザー・ディレクトリから移行する、すべての EPM System ユーザーのユーザー・アカウントとグループがターゲット・ユーザー・ディレクトリに含まれていることを確認します。

すでに外部ユーザー・ディレクトリとして定義されているユーザー・ディレクトリにユーザーを移行した場合、ユーザー・アカウントが Shared Services からアクセス可能であることを確認します。これは、Shared Services Console からユーザーを検索することで実行できます。Oracle Enterprise Performance Management System User Security Administration Guide のユーザー、グループ、役割および委任リストの検索に関する項を参照してください。

ターゲット・ユーザー・ディレクトリを外部ユーザー・ディレクトリとして構成する際、「ログイン属性」プロパティが、ソース・ユーザー・ディレクトリで元々そのユーザー名の属性値として使用されていた属性を指していることを確認します。149 ページの「前提条件」を参照してください。

ターゲット・ユーザー・ディレクトリの検索順の変更

注： ターゲット・ユーザー・ディレクトリ名がソース・ディレクトリ名と同じ場合、ソース・ユーザー・ディレクトリを EPM System 構成から削除する必要があります。

Shared Services では、新たに追加されたユーザー・ディレクトリに、既存のディレクトリに割り当てられている検索順より低い順序が割り当てられます。ターゲット・ユーザー・ディレクトリの検索順がソース・ユーザー・ディレクトリよりも上になるよう検索順を変更します。これによって、Oracle Hyperion Shared Services がソースを検索する前にターゲット・ユーザー・ディレクトリでユーザーを検出できるようになります。Oracle Enterprise Performance Management System User Security Administration Guide のユーザー・ディレクトリの検索順の管理に関する項を参照してください。

EPM System の再起動

Oracle Hyperion Foundation Services とその他の EPM System コンポーネントを再起動し、変更を反映します。

インポート・ファイルの編集

注： EPM System 構成のターゲット・ユーザー・ディレクトリ名がソース・ユーザー・ディレクトリ名と同じ場合、この手順は必要ありません。

ネイティブ・ディレクトリでのデータの再作成には、ライフサイクル管理で作成されたエクスポート・ファイルをソースとして使用します。エクスポート・ファイルは、ネイティブ・ディレクトリからのエクスポート時に、指定されたディレクトリに生成されます。150 ページの「ネイティブ・ディレクトリ・データのエクスポート」を参照してください。

各エクスポート・ファイルで、ソース・ユーザー・ディレクトリへの参照をすべてターゲット・ユーザー・ディレクトリへの参照に置き換えます。通常、割り当てられている役割のエクスポート・ファイルを編集します。オプションで次のファイルも編集します。

- ソース・ユーザー・ディレクトリのユーザーが、ネイティブ・ディレクトリ・グループのメンバーの場合、`Groups.csv`。
- ソース・ユーザー・ディレクトリのユーザーが委任リストに割り当てられている場合、`Delegated Lists.csv`。

インポート・ファイルは、`EPM_ORACLE_INSTANCE/import_export/USER_NAME/EXPORT_DIR/resource/Native Directory`(`USER_NAME` は、`admin@Native Directory` などのエクスポート操作を行うユーザーの ID で、`EXPORT_DIR` は、エクスポート・ディレクトリの名前)にあります。

➤ インポート・ファイルを編集するには:

1 テキスト・エディタを使用して、インポート・ファイルを開きます。

2 ソース・ユーザー・ディレクトリの名前を、「定義済ユーザー・ディレクトリ」画面の「ディレクトリ名」列に表示されているターゲット・ユーザー・ディレクトリの名前に置き換えます。

3 インポート・ファイルを保存して閉じます。

更新されたデータのインポート

作成/更新オプションを使用してライフサイクル管理を実行し、ネイティブ・ディレクトリからエクスポートしてあったデータをインポートします。150 ページの「ネイティブ・ディレクトリ・データのエクスポート」を参照してください。

注： Oracle Hyperion Enterprise Performance Management System ライフサイクル管理を使用したデータのインポートの詳細な手順については、Oracle Enterprise Performance Management System Lifecycle Management Guide を参照してください。

➤ 更新されたプロビジョニング・データをネイティブ・ディレクトリにインポートするには:

1 Oracle Hyperion Shared Services Console のビュー・ペインで、「ファイル・システム」を展開します。

2 インポート・ファイルのファイル・システムの場所を選択します。

3 プロビジョニング情報をインポートするアーティファクトのタイプを選択します。

4 「移行の定義」をクリックします。

5 「ソース・オプション」で「次へ」をクリックします。

6 「宛先」で「次へ」をクリックします。

7 「宛先オプション」で、「インポート操作タイプ」が「作成/更新」に設定されていることを確認します。

8 「次へ」をクリックします。

9 「移行の実行」をクリックします。

ネイティブ・ディレクトリの更新ユーティリティの実行

ネイティブ・ディレクトリの更新ユーティリティを実行して、陳腐化したデータをネイティブ・ディレクトリから消去します。付録 C「ネイティブ・ディレクトリの更新ユーティリティの使用方法」を参照してください。

個々の製品の更新

注意 個々の製品を更新する前に、Oracle Enterprise Performance Management System コンポーネントによって使用されているリポジトリのユーザーとグループのデータをバックアップすることをお薦めします。ローカル製品リポジトリの情報を更新した後は、バックアップからのみ、元のローカル製品リポジトリのユーザーとグループのデータに戻すことができます。

Planning

Planning は、Planning リポジトリでプロビジョニングされたユーザーとグループに関する情報を保管します。ユーザーとグループをユーザー・ディレクトリ間で移行した結果、ネイティブ・ディレクトリ内のユーザー ID が変更された場合、「ユーザーの移行」/「グループの移行」を選択して、Planning リポジトリの情報と、ネイティブ・ディレクトリの情報を同期化する必要があります。このボタンは、データ・フォーム、メンバーまたはタスク・リストへのアクセスを割り当てる場合に、Oracle Hyperion Planning で使用可能です。

Financial Management

Financial Management は、ローカル Financial Management リポジトリ内のオブジェクトにアクセスするようプロビジョニングされたユーザーとグループに関する情報を記録します。ユーザーとグループをユーザー・ディレクトリ間で移行した結果、ネイティブ・ディレクトリ内のユーザーとグループの情報が変更された場合、Oracle Hyperion Financial Management リポジトリの情報と、ネイティブ・ディレクトリの情報を同期化する必要があります。

Reporting and Analysis

Reporting and Analysis では、syncCSSId ユーティリティを使用して、ネイティブ・ディレクトリ内の ID が反映されるようリレーショナル・データベースに保管されているユーザーとグループ ID を同期化します。ユーザーが Oracle Hyperion Reporting and Analysis にアクセスできるようにするには、ネイティブ・ディレクトリのプロビジョニング・データの移行後、このユーティリティを実行する必要があります。syncCSSId ユーティリティは、`EPM_ORACLE_INSTANCE/bin/ReportingAnalysis/syncCSSId` ディレクトリにインストールされています(`C:/Oracle/Middleware/user_projects/epmsystem1/bin/ReportingAnalysis/syncCSSId` など)。

syncCSSId ユーティリティの実行の詳細な手順については、`EPM_ORACLE_INSTANCE/bin/ReportingAnalysis/syncCSSId/ReadmeSyncCSSId_BI.txt` を参照してください。

用語集

ID 外部認証におけるユーザーまたはグループの一意の識別です。

Shared Services レジストリ ほとんどの EPM System 製品の EPM System 配置情報(インストール・ディレクトリ、データベース設定、コンピュータ名、ポート、サーバー、URL、依存サービス・データなど)を管理する Shared Services リポジトリの一部です。

アクセス権 リソースに対してユーザーが実行できる一連の操作です。

アップグレード ソフトウェアの新しいリリースを配置し、以前の配置から新規配置へデータとプロビジョニング情報を移行するプロセスです。

アプリケーション 1)特定のタスクまたはタスクのグループを実行するために設計されたソフトウェア・プログラムです(スプレッドシート・プログラム、データベース管理システムなど)。2)必要とされる特定の分析のセットまたはレポートのセット、あるいはその両方に対応するために使用される、次元および次元メンバーの関連するセットです。

アプリケーション移行ユーティリティ アプリケーションとアーティファクトの移行に使用されるコマンド・ライン・ユーティリティです。

アーティファクト 個別のアプリケーションまたはリポジトリ・アイテムです(スクリプト、フォーム、ルール・ファイル、Interactive Reporting ドキュメント、財務レポートなど)。オブジェクトとも呼ばれます。

移行 アプリケーション、アーティファクト、またはユーザーを、別の環境またはコンピュータにコピーするプロセスです。たとえば、テスト環境から本番環境にコピーします。

移行監査レポート 移行ログから生成されるレポートです。アプリケーションの移行に関する追跡情報を提供します。

移行スナップショット アプリケーションの移行のスナップショットです。移行ログに取込まれます。

移行定義ファイル(.mdf) アプリケーションの移行に使用される移行パラメータを含むファイルです。これによりバッチ・スクリプトを処理できます。

移行ログ アプリケーションの移行のすべてのアクションとメッセージを取込むログ・ファイルです。

管理対象サーバー 内蔵された Java 仮想マシン(Java Virtual Machine (JVM))で実行されるアプリケーション・サーバー・プロセスです。

外部認証 Oracle EPM System 製品にアプリケーション外に格納されているユーザー情報でログオンします。ユーザー・アカウントは、EPM System によって維持されますが、パスワード管理およびユーザー認証は、Oracle Internet Directory (OID)または Microsoft Active Directory (MSAD)などの企業ディレクトリを使用して、外部サービスによって実行されます。

グループ 複数のユーザーに同様のアクセス権を割り当てるためのコンテナです。

コンテキスト変数 タスクフロー・インスタンスのコンテキストを特定するために、特定のタスクフローに定義される変数です。

集約役割 Hyperion 製品内の複数の事前定義された役割を集約するカスタム役割です。

手動ステージ ユーザーの操作が必要なステージです。

シングル・サインオン(SSO) 一度ログオンすると、再度認証を要求されることなしに複数のアプリケーションにアクセスできる機能です。

自動ステージ ユーザーの操作を必要としないステージです(データ・ロードなど)。

ステージ 1)通常は個別のユーザーにより実行される、タスクフロー内の 1 つの論理ステップを形成するタスクの説明です。ステージには手動と自動の 2 つのタイプがあります。2)Profitability では、組織内での割当てプロセスのステップを表す、モデル内の論理区分です。

ステージ・アクション 自動ステージで、ステージを実行するために呼び出されたアクションです。

製品 Shared Services における、Planning や Performance Scorecard などのアプリケーション・タイプです。

セキュリティ・エージェント Web アクセス管理プロバイダ(Oracle Access Manager、Oracle Single Sign-On、CA SiteMinder など)です。企業の Web リソースを保護します。

セキュリティ・プラットフォーム Oracle EPM System 製品で外部認証とシングル・サインオン機能を使用するためのフレームワークです。

タスクフロー ビジネス・プロセスの自動化を指します。手続きのルールに従って、あるタスクフロー参加者から別の参加者にタスクが渡されます。

タスクフロー・インスタンス タスクフローの状態と関連データが含まれる、タスクフローの単一のインスタンスです。

タスクフロー管理システム タスクフローを定義および作成し、その実行を管理するシステムです。定義付け、ユーザーまたはアプリケーションのやりとり、およびアプリケーションの実行可能ファイルが含まれます。

タスクフロー参加者 手動ステージおよび自動ステージの両方について、タスクフローのステージのインスタンスに関連付けられているタスクを実行するリソースです。

タスクフロー定義 ステージとステージ間の関係のネットワーク、タスクフローの開始と終了を示す基準、および個別のステージに関する情報(参加者、関連アプリケーション、関連アクティビティなど)から構成される、タスクフロー管理システムのビジネス・プロセスです。

タスク・リスト 特定のユーザーについて、タスクの詳細ステータスを示すリストです。

統合 Shared Services を使用して Oracle Hyperion アプリケーションでデータを移動するために実行されるプロセスです。データ統合の定義によりソース・アプリケーションと宛先アプリケーションの間でのデータの移動が指定され、データの動きのグループ化、順序付けおよびスケジュールが決定されます。

トークン 外部認証システム上の 1 つの有効なユーザーまたはグループの暗号化された識別です。

同期 Shared Services とアプリケーション・モデルの同期です。

同期済 モデルの最新バージョンがアプリケーションと Shared Services の両方に存在する状態を指します。「モデル」も参照してください。

認証 安全対策としての ID の確認です。一般に、認証はユーザー名およびパスワードに基づきます。パスワードおよびデジタル・シグネチャは認証のフォームです。

バックアップ アプリケーション・インスタンスの重複コピーです。

ビジネス・プロセス 集合的にビジネス上の目標を達成するための一連のアクティビティです。

フィルタ データ・セットで、特定の基準に従って値を制限する制約です。たとえば、特定のテーブル、メタデータ、または値を除外したり、アクセスをコントロールしたりする場合に使用されます。

プロジェクト 実装でグループ化された Oracle Hyperion 製品のインスタンスです。たとえば、Planning プロジェクトには Planning アプリケーション、Essbase キューブ、Financial Reporting サーバー・インスタンスが含まれることがあります。

プロビジョニング ユーザーおよびグループに対して、リソースへのアクセス権を付与するプロセスです。

変換 1)アプリケーションの移行後も宛先環境で正しく機能するように、アーティファクトを変換するプロセスです。2)データ・マイニングで、キューブおよびアルゴリズムのセルの間で(双方向に)流れるデータを変更することです。

モデル 1)データ・マイニングで、アルゴリズムにより検査されたデータに関する情報の集合です。より広範なデータ・セットにモデルを適用することにより、データに関する有用な情報を生成できます。2)アプリケーション固有の方法で示したデータが含まれるファイルまたはコンテンツの文字列です。モデルは Shared Services により管理される基本データであり、次元と非次元のアプリケーション・オブジェクトという 2 つの主要なタイプがあります。3) Business Modeling で、検査対象の領域からの業務および財務上のフローを示し、また計算するために接続されたマシン・ネットワークです。

役割 リソースへのアクセス権をユーザーおよびグループに付与する際に使用される手段です。

ユーザー・ディレクトリ ユーザーおよびグループの情報を集中管理する場所で、リポジトリまたはプロバイダとも呼ばれます。最も普及したユーザー・ディレクトリとして、Oracle Internet Directory (OID)、Microsoft Active Directory (MSAD)、Sun Java System Directory Server などがあります。

ライフサイクル管理 製品環境間でのアプリケーション、リポジトリ、または個別のアーティファクトの移行プロセスです。

リポジトリ ビューおよびクエリーに使用するためのメタデータ、フォーマットおよび注釈の情報のストレージの場所です。

リンク (1)リポジトリ・オブジェクトへの参照です。リンクは、フォルダ、ファイル、ショートカットおよび他のリンクを参照できます。(2)タスクフローで、あるステージのアクティビティが終了して次のアクティビティが開始するポイントです。

リンク条件 タスクフローのステージを順序付けるためにタスクフロー・エンジンにより評価される論理式です。

ロード・バランシング 複数のサーバーに要求を分散すること。これによって、エンド・ユーザーのパフォーマンスが最適化されます。

索引

A - Z

あ行

か行

さ行

た行

な行

は行

や行

ら行