Oracle® Enterprise Performance Management System
보안 구성 가이드

릴리스 11.2
F28627-22
2023년 12월

ORACLE

Oracle Enterprise Performance Management System 보안 구성 가이드NOT_SUPPORTED릴리스 11.2

F28627-22

주요 작성자: EPM Information Development Team

목차

B 사용자정의 로그인 클래스 구현

C 사용자 디렉토리 간 사용자 및 그룹 마이그레이션

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info를 참조하거나, 청각 장애가 있는 경우 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs를 방문하십시오.

설명서 피드백

이 문서에 대한 피드백을 제공하려면 모든 Oracle 도움말 센터 항목의 페이지 맨아래에 있는 [피드백] 버튼을 누릅니다. epmdoc_ww@oracle.com으로 전자메일을 보낼 수도 있습니다.

1

EPM System 보안 정보

참조:

- EPM System 정보
- 사전 지식
- 보안 인프라 구성요소
- 사용자 인증
- 프로비저닝(역할 기반 권한부여)
- Shared Services Console 실행

EPM System 정보

Oracle Enterprise Performance Management System 제품은 보고 및 분석을 위한 가장 포괄적인 비즈니스 인텔리전스 기능과 모듈식 재무 관리 및 계획 애플리케이션 제품군을 통합하는 포괄적인 엔터프라이즈급 시스템을 구성합니다. EPM System 제품의 주요 구성요소는 다음과 같습니다.

- Oracle Hyperion Foundation Services
- Oracle Essbase
- Oracle Hyperion Financial Management
- Oracle Hyperion Planning

이러한 제품군 각각의 제품과 구성요소에 대한 자세한 내용은 *Oracle Enterprise Performance Management System 설치 시작 페이지*를 참조하십시오.

사전 지식

이 가이드는 Oracle Enterprise Performance Management System 구성요소를 구성하고, 보안하고, 관리하는 시스템 관리자를 위한 것입니다. 이 가이드에서는 관리자에게 다음과 같은 지식이 있다고 가정합니다.

- 다음을 비롯한 조직의 보안 인프라를 확실하게 이해하고 있습니다.
  - 디렉토리 서버(예: Oracle Internet Directory, Sun Java System Directory Server 및 Microsoft Active Directory)
  - SSL(Secure Socket Layer)을 사용하여 통신 채널 보호
  - 액세스 관리 시스템(예: Oracle Access Manager 및 SiteMinder)
  - SSO(싱글 사인온) 인프라(예: Kerberos)
- 조직과 관련된 EPM System 보안 개념에 대한 지식이 있습니다.

보안 인프라 구성요소

Oracle Enterprise Performance Management System은 여러 보안 구성요소를 통합하여 강력한 애플리케이션 보안을 보장합니다. 보안 인프라에 통합되면 EPM System에서 데이터 및 액세스 보안을 보장하는 매우 안전한 애플리케이션 제품군을 제공합니다. EPM System을 보호하는 데 사용할 수 있는 인프라 구성요소는 다음과 같습니다.

- 선택적 액세스 관리 시스템(예: EPM System 구성요소에 SSO 액세스를 제공하는 Oracle Access Manager)
- 통합 SSO 인프라(예: Kerberos) 사용

  액세스 관리 시스템(SiteMinder)에서 Kerberos 인증을 사용하여 Windows 사용자가 SiteMinder 및 EPM System 구성요소에 투명하게 로그인하도록 보장할 수 있습니다.
- EPM System 구성요소와 클라이언트에서 SSL(Secure Socket Layer)을 사용하여 통신 채널 보호

사용자 인증

사용자 인증을 통해 Oracle Enterprise Performance Management System 구성요소에서 SSO(싱글 사인온) 기능이 사용으로 설정되면 각 사용자의 로그인 정보를 검증하여 인증된 사용자를 확인할 수 있습니다. 사용자 인증은 구성요소별 권한부여와 함께 EPM System 구성요소에 대한 액세스 권한을 사용자에게 부여합니다. 권한을 부여하는 프로세스를 프로비저닝이라고 합니다.

인증 구성요소

다음 절에서는 SSO를 지원하는 구성요소에 대해 설명합니다.

- Native Directory
- 외부 사용자 디렉토리

Native Directory

Native Directory는 Oracle Hyperion Shared Services에서 프로비저닝을 지원하고 기본 사용자 계정과 같은 시드 데이터를 저장하는 데 사용하는 관계형 데이터베이스입니다.

Native Directory 기능:

- 기본 EPM System 사용자 계정을 유지관리하고 관리합니다.
- 모든 EPM System 프로비저닝 정보(사용자, 그룹 및 역할 간 관계) 저장

Native Directory는 Oracle Hyperion Shared Services Console을 사용하여 액세스하고 관리합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "Native Directory 관리"를 참조하십시오.

외부 사용자 디렉토리

사용자 디렉토리는 EPM System 구성요소와 호환되는 기업 사용자 및 ID 관리 시스템입니다.

EPM System 구성요소는 Oracle Internet Directory, Sun Java System Directory Server(이전의 SunONE Directory Server) 및 Microsoft Active Directory와 같은 LDAP 기반 사용자 디렉토리를 포함하여 여러 사용자 디렉토리에서 지원됩니다. 관계형

데이터베이스도 사용자 디렉토리로 지원됩니다. 이 문서에서는 Native Directory를 제외한 사용자 디렉토리를 외부 사용자 디렉토리라고 합니다.

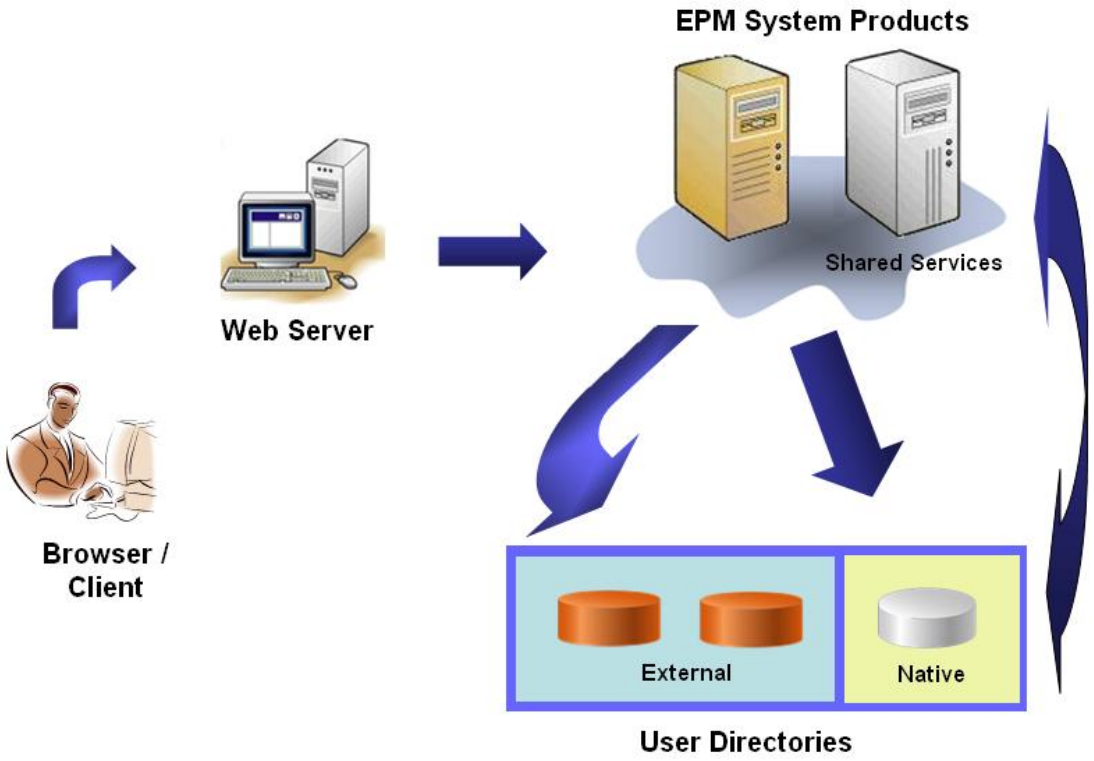
지원되는 사용자 디렉토리 목록은 OTN(Oracle Technology Network)의 Oracle Fusion Middleware Supported System Configurations 페이지에 게시된 *Oracle Enterprise Performance Management System Certification Matrix*를 참조하십시오.

Shared Services Console에서는 여러 외부 사용자 디렉토리를 EPM System 사용자 및 그룹의 소스로 구성할 수 있습니다. 각 EPM System 사용자는 구성된 사용자 디렉토리에 고유한 계정이 있어야 합니다. 일반적으로 EPM System 사용자는 원활한 프로비저닝을 위해 그룹에 지정됩니다.

기본 EPM System 싱글 사인온

EPM System은 애플리케이션의 인증된 사용자가 인증서를 다시 입력하지 않고도 원활하게 다른 애플리케이션을 탐색할 수 있도록 하여 EPM System 웹 애플리케이션에서 SSO를 지원합니다. EPM System 구성요소에서 사용자 인증 및 프로비저닝(역할 기반 권한부여)을 처리하는 공통 보안 환경을 통합하여 SSO를 구현합니다.

다음 그림에는 기본 SSO 프로세스가 설명되어 있습니다.



1. 사용자는 브라우저를 통해 EPM System 구성요소 로그인 화면에 액세스하여 사용자 이름과 비밀번호를 입력합니다.

   EPM System 구성요소는 구성된 사용자 디렉토리(Native Directory 포함)를 쿼리하여 사용자 인증서를 확인합니다. 사용자 디렉토리에서 일치하는 사용자 계정을 찾으면 검색이 종료되고 사용자 정보가 EPM System 구성요소로 반환됩니다.

   구성된 사용자 디렉토리에서 사용자 계정을 찾지 못하면 액세스가 거부됩니다.

2. EPM System 구성요소는 검색된 사용자 정보로 Native Directory를 쿼리하여 사용자에 대한 프로비저닝 세부정보를 얻습니다.

3. EPM System 구성요소는 구성요소의 ACL(액세스 제어 목록)을 확인하여 사용자가 액세스할 수 있는 애플리케이션 아티팩트를 결정합니다.

Native Directory에서 프로비저닝 정보를 받으면 사용자는 EPM System 구성요소를 사용할 수 있습니다. 이때 SSO는 사용자가 프로비저닝된 모든 EPM System 구성요소에 대해 사용으로 설정됩니다.

액세스 관리 시스템의 싱글 사인온

EPM System 구성요소 보안을 강화하기 위해 Oracle Access Manager 또는 SiteMinder와 같은 지원되는 액세스 관리 시스템을 구현하여 EPM System 구성요소에 인증된 사용자 인증서를 제공하고 사전 정의된 액세스 권한을 기반으로 액세스를 제어할 수 있습니다.

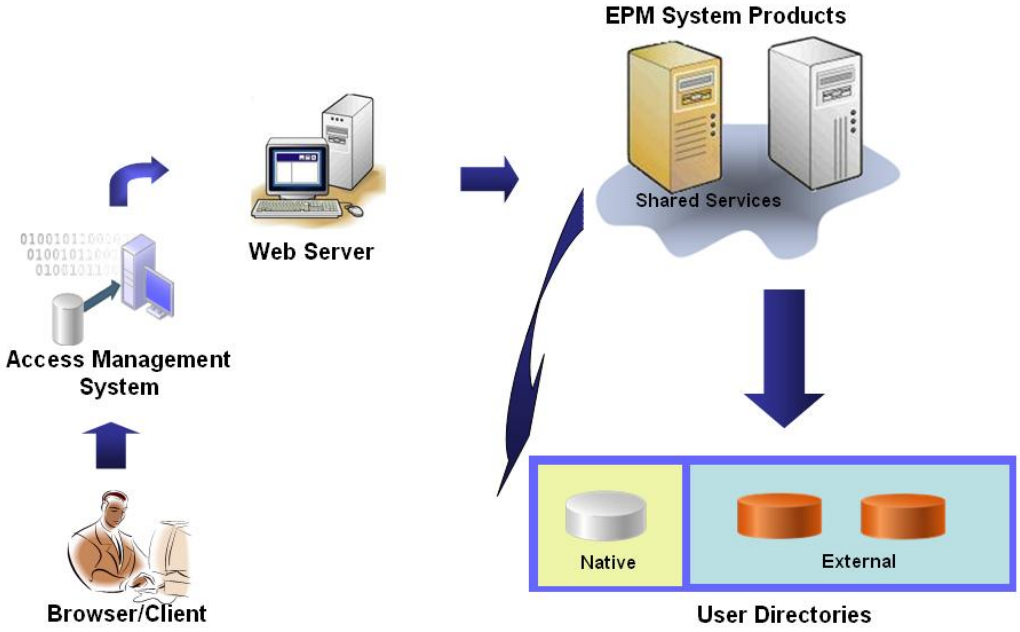
보안 에이전트의 SSO는 EPM System 웹 애플리케이션에만 사용할 수 있습니다. 이 시나리오에서 EPM System 구성요소는 보안 에이전트가 제공하는 사용자 정보를 사용하여 사용자의 액세스 권한을 확인합니다. 보안을 강화하려면 모든 요청을 SSO 포털을 통해 라우팅하도록 서버에 대한 직접 액세스를 방화벽으로 차단하는 것이 좋습니다.

허용되는 SSO 메커니즘을 통해 인증된 사용자 인증서를 수락하는 방법으로 액세스 관리 시스템의 SSO를 지원합니다. 지원되는 SSO 방법를 참조하십시오. 액세스 관리 시스템은 사용자를 인증하고 로그인 이름을 EPM System에 전달합니다. EPM System은 구성된 사용자 디렉토리에 대해 로그인 이름을 확인합니다.

다음 항목을 참조하십시오.

- Oracle Access Manager의 싱글 사인온
- OracleAS 싱글 사인온
- SiteMinder SSO
- Kerberos 싱글 사인온

그림으로 된 개념은 다음과 같습니다.



1. 사용자는 브라우저를 사용하여 Oracle Access Manager 또는 SiteMinder와 같은 액세스 관리 시스템에서 보호하는 리소스에 대한 액세스를 요청합니다.

> **주:**
>
> EPM System 구성요소는 액세스 관리 시스템에서 보호하는 리소스로 정의됩니다.

액세스 관리 시스템은 요청을 가로채서 로그인 화면을 표시합니다. 사용자가 사용자 이름과 비밀번호를 입력하면 액세스 관리 시스템에 구성된 사용자 디렉토리에 대해 검증하여 사용자 인증을 확인합니다. EPM System 구성요소도 이러한 사용자 디렉토리를 사용하도록 구성됩니다.

인증된 사용자 관련 정보는 해당 정보를 유효한 정보로 수락하는 EPM System 구성요소로 전달됩니다.

액세스 관리 시스템은 허용되는 SSO 메커니즘을 사용하여 사용자의 로그인 이름(`Login Attribute` 값)을 EPM System 구성요소에 전달합니다. 지원되는 SSO 방법를 참조하십시오.

2. 사용자 인증서를 확인하기 위해 EPM System 구성요소는 사용자 디렉토리에서 사용자를 찾습니다. 일치하는 사용자 계정을 찾은 경우 사용자 정보가 EPM System 구성요소로 반환됩니다. EPM System 보안은 EPM System 구성요소에서 SSO 토큰을 사용으로 설정하는 SSO 토큰을 설정합니다.
3. EPM System 구성요소는 검색된 사용자 정보로 Native Directory를 쿼리하여 사용자에 대한 프로비저닝 세부정보를 얻습니다.

   사용자 프로비저닝 정보를 받으면 사용자는 EPM System 구성요소를 사용할 수 있습니다. SSO는 사용자가 프로비저닝된 모든 EPM System 구성요소에 대해 사용으로 설정됩니다.

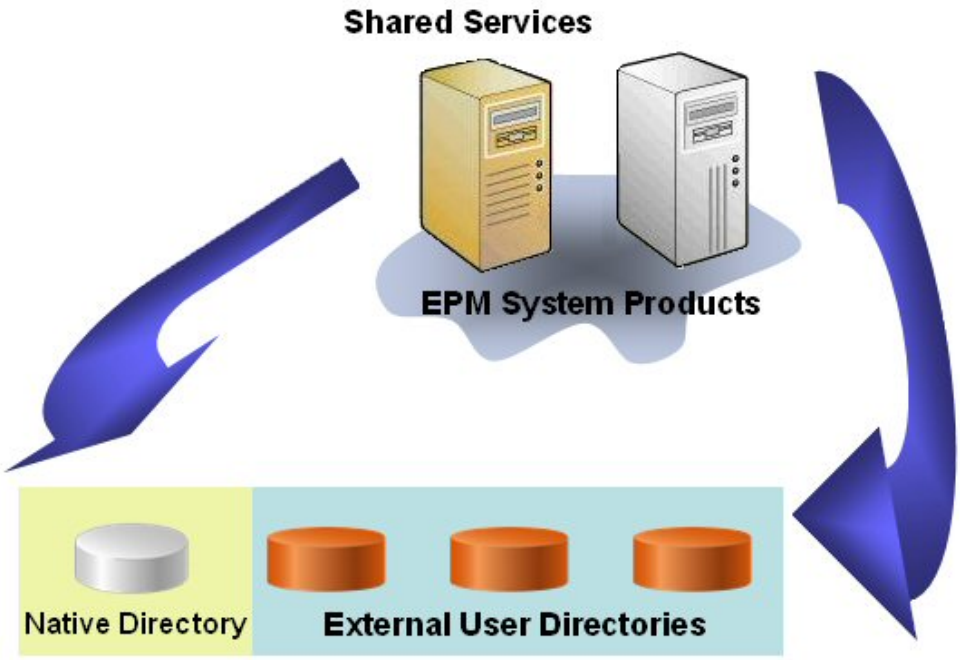
프로비저닝(역할 기반 권한부여)

Oracle Enterprise Performance Management System 보안에서는 역할 개념을 사용하여 애플리케이션에 대한 사용자 액세스 권한을 결정합니다. 역할은 애플리케이션 기능에 대한 사용자 액세스 권한을 결정하는 권한입니다. 일부 EPM System 구성요소는 보고서 및 멤버와 같은 아티팩트에 대한 사용자 액세스 권한을 더욱 구체적으로 지정하기 위해 객체 레벨 ACL을 적용합니다.

각 EPM System 구성요소는 다양한 비즈니스 요구에 맞게 수정된 여러 기본 역할을 제공합니다. EPM System 구성요소에 속하는 각 애플리케이션은 해당 역할을 상속합니다. Oracle Hyperion Shared Services에 등록된 애플리케이션의 사전 정의된 역할은 Oracle Hyperion Shared Services Console에서 사용할 수 있습니다. 특정 요구사항에 맞게 기본 역할을 종합한 추가 역할을 생성할 수도 있습니다. 이러한 역할은 프로비저닝 설정에 사용됩니다. EPM System 애플리케이션 및 해당 리소스에 속한 특정 역할을 사용자 및 그룹에 부여하는 프로세스를 *프로비저닝*이라고 합니다.

Native Directory 및 구성된 사용자 디렉토리는 프로비저닝 프로세스에 사용되는 사용자 및 그룹 정보 소스입니다. Shared Services Console에 구성된 모든 사용자 디렉토리에서 사용자 및 그룹을 찾아 프로비전닝할 수 있습니다. 프로비저닝 프로세스에서는 Native Directory에 생성된 애플리케이션별 역할 집합도 사용할 수 있습니다.

권한부여 프로세스에 대한 그림 개요:



1. 사용자가 인증되면 EPM System 구성요소는 사용자 디렉토리를 쿼리하여 사용자 그룹을 확인합니다.
2. EPM System 구성요소는 그룹 및 사용자 정보를 사용하여 Shared Services에서 사용자의 프로비저닝 데이터를 검색합니다. 이 구성요소는 해당 데이터를 사용하여 사용자가 액세스할 수 있는 리소스를 확인합니다.

   제품별 액세스 제어 설정 등 제품별 프로비저닝 태스크는 각 제품에 대해 수행됩니다. 이 데이터는 프로비저닝 데이터와 결합하여 사용자의 제품 액세스를 결정합니다.

EPM System 제품의 역할 기반 프로비저닝에서는 다음과 같은 개념을 사용합니다.

역할

역할은 EPM System 리소스에 대한 기능을 수행할 수 있도록 사용자 및 그룹에 부여된 액세스 권한을 정의하는 구성자입니다(액세스 제어 목록과 유사). 역할은 리소스 또는 리소스 유형(예: 사용자가 액세스할 수 있는 특정 보고서)과 사용자가 리소스에 대해 수행할 수 있는 작업(예: 보기 및 편집)의 조합입니다.

EPM System 애플리케이션 리소스에 대한 액세스는 제한되어 있습니다. 액세스 권한을 제공하는 역할이 사용자에게 지정되거나 사용자가 속한 그룹에 지정된 후에야 사용자는 리소스에 액세스할 수 있습니다. 역할에 따른 액세스 제한으로 관리자는 애플리케이션 액세스를 제어하고 관리할 수 있습니다.

글로벌 역할

여러 제품에 적용되는 Shared Services 역할인 글로벌 역할을 통해 사용자는 여러 EPM System 제품에서 특정 태스크를 수행할 수 있습니다. 예를 들어 Shared Services 관리자는 모든 EPM System 애플리케이션에 대해 사용자를 프로비저닝할 수 있습니다.

사전 정의된 역할

사전 정의된 역할은 EPM System 제품의 기본 제공 역할로, 삭제할 수 없습니다. EPM System 제품에 속한 각 애플리케이션 인스턴스는 제품의 사전 정의된 역할을 상속합니다. 각 애플리케이션의 이러한 역할은 애플리케이션 생성 중에 Shared Services에 등록됩니다.

역할 집합

사용자정의 역할이라고도 하는 역할 집합은 애플리케이션에 속한 여러 사전 정의된 역할을 종합합니다. 역할 집합은 다른 역할 집합에 포함될 수 있습니다. 예를 들어 Shared Services 관리자 또는 프로비저닝 관리자는 Oracle Hyperion Planning 애플리케이션의 플래너 및 보기 사용자 역할을 결합하는 역할 집합을 생성할 수 있습니다. 역할 집합은 역할이 여러 개로 세분화된 애플리케이션의 관리를 단순화할 수 있습니다. 글로벌 Shared Services 역할은 역할 집합에 포함될 수 있지만 여러 애플리케이션 또는 제품에 적용되는 역할 집합은 생성할 수 없습니다.

사용자

사용자 디렉토리에는 EPM System 제품에 액세스할 수 있는 사용자에 대한 정보가 저장됩니다. 인증 및 권한 부여 프로세스에는 사용자 정보가 사용됩니다. Shared Services Console에서만 Native Directory 사용자를 생성하고 관리할 수 있습니다.

Shared Services Console에서는 구성된 모든 사용자 디렉토리의 사용자를 볼 수 있습니다. 이러한 사용자는 Shared Services에 등록된 EPM System 제품에 대한 액세스 권한을 부여하기 위해 개별적으로 프로비전닝할 수 있지만, 개별 사용자를 프로비저닝하는 것은 권장되지 않습니다.

기본 EPM System 관리자

배포 프로세스 중에 기본 이름이 `admin`인 관리자 계정이 Native Directory에 생성됩니다. 이 계정은 가장 강력한 EPM System 계정이며, EPM System 보안 및 환경을 관리하는 정보 기술 전문가인 시스템 관리자를 설정할 때만 사용해야 합니다.

EPM System 관리자의 사용자 이름 및 비밀번호는 Oracle Hyperion Foundation Services 배포 중에 설정됩니다. 이 계정에는 회사 계정 비밀번호 정책을 적용할 수 없으므로 시스템 관리자 계정을 생성한 후에는 이 계정을 비활성화하는 것이 좋습니다.

일반적으로 기본 EPM System 관리자 계정은 다음 태스크를 수행하는 데 사용됩니다.

- 회사 디렉토리를 외부 사용자 디렉토리로 구성합니다. 사용자 디렉토리 구성을 참조하십시오.
- Shared Services 관리자 역할로 회사 정보 기술 전문가를 프로비저닝하여 시스템 관리자 계정을 생성합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자 및 그룹 프로비저닝"을 참조하십시오.

시스템 관리자

시스템 관리자는 일반적으로 EPM System 배포에 관련된 모든 서버에 대한 읽기, 쓰기, 실행 액세스 권한이 있는 기업 정보 기술 전문가입니다.

일반적으로 시스템 관리자는 다음 태스크를 수행합니다.

- 기본 EPM System 관리자 계정을 사용 안함으로 설정합니다.
- 기능 관리자를 하나 이상 생성합니다.
- Shared Services Console을 사용하여 EPM System에 대한 보안 구성을 설정합니다.
- 선택적으로 사용자 디렉토리를 외부 사용자 디렉토리로 구성합니다.
- 로그 분석 툴을 정기적으로 실행하여 EPM System을 모니터합니다.

  기능 관리자가 수행하는 태스크는 이 가이드에 설명되어 있습니다.

기능 관리자를 생성하는 절차:

- 회사 디렉토리를 외부 사용자 디렉토리로 구성합니다. 사용자 디렉토리 구성을 참조하십시오.

- 사용자 또는 그룹을 필요한 역할로 프로비저닝하여 기능 관리자를 생성합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자 및 그룹 프로비저닝"을 참조하십시오.

  기능 관리자를 다음 역할로 프로비저닝해야 합니다.

  - Shared Services의 LCM 관리자 역할
  - 배포된 각 EPM System 구성요소의 관리자 및 프로비저닝 관리자 역할

기능 관리자

기능 관리자는 EPM System 전문가인 기업 사용자입니다. 일반적으로 이 사용자는 Shared Services에 외부 사용자 디렉토리로 구성되어 있는 기업 디렉토리에서 정의됩니다.

기능 관리자는 다른 기능 관리자 생성, 위임된 관리 설정, 애플리케이션과 아티팩트 생성 및 프로비저닝, EPM System 감사 설정 등 EPM System 관리 태스크를 수행합니다. 기능 관리자가 수행하는 태스크는 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*에 설명되어 있습니다.

그룹

그룹은 사용자 또는 다른 그룹의 컨테이너입니다. Shared Services Console에서 Native Directory 그룹을 생성하고 관리할 수 있습니다. 구성된 모든 사용자 디렉토리의 그룹이 Shared Services Console에 표시됩니다. Shared Services에 등록된 EPM System 제품에 대한 권한을 부여하기 위해 이러한 그룹을 프로비전닝할 수 있습니다.

Shared Services Console 실행

Oracle Hyperion Enterprise Performance Management Workspace의 메뉴 옵션을 사용하여 Oracle Hyperion Shared Services Console에 액세스합니다.

Shared Services Console을 실행하려면 다음을 수행합니다.

1. 다음 위치로 이동합니다.

   `http://web_server_name:port_number/workspace`

   이 URL에서 *web_server_name*은 Oracle Hyperion Foundation Services에서 사용하는 웹 서버가 실행 중인 컴퓨터의 이름을 나타내고, *port_number*는 웹 서버 포트(예: `http://myWebserver:19000/workspace`)를 나타냅니다.

   > **주:**
   >
   > 보안 환경에서 EPM Workspace에 액세스하는 경우 프로토콜로 `https`(`http` 아님)를 사용하고 보안 웹 서버 포트 번호를 사용합니다. 예를 들어 `https://myserver:19043/workspace`와 같은 URL을 사용합니다.

2. **애플리케이션 실행**을 누릅니다.

   > **주:**
   >
   > 팝업 차단기로 인해 EPM Workspace가 열리지 않을 수 있습니다.

3. **로그온**에 사용자 이름과 비밀번호를 입력합니다.

   처음에 Shared Services Console에 액세스할 수 있는 사용자는 배포 프로세스 중에 사용자 이름과 비밀번호가 지정된 Oracle Enterprise Performance Management System 관리자뿐입니다.

4. **로그온**을 누릅니다.

5. **탐색**, **관리**, **Shared Services Console** 순으로 선택합니다.

2

EPM System 구성요소에서 SSL을 사용으로 설정

참조:

가정

- 사용자가 배포 토폴로지를 결정하고 SSL을 사용하여 보호할 통신 링크를 확인했습니다.
- 사용자가 잘 알려져 있는 인증 기관(CA) 또는 자체 CA에서 필요한 인증서를 가져오거나 자체 서명된 인증서를 생성했습니다. 필요한 인증서을 참조하십시오.
- 인증서 임포트와 같은 SSL 개념 및 절차에 대해 잘 알고 있습니다.

  참조 문서 목록은 정보 소스를 참조하십시오.

정보 소스

SSL을 사용하는 Oracle Enterprise Performance Management System에서는 애플리케이션 서버, 웹 서버, 데이터베이스, 사용자 디렉토리 등의 구성요소가 SSL을 사용하여 통신하도록 준비해야 합니다. 이 문서에서는 사용자가 이러한 구성요소의 SSL을 사용으로 설정하는 태스크에 대해 잘 알고 있다고 가정합니다.

- **Oracle WebLogic Server**: *WebLogic Server 보호 가이드*의 "SSL 구성"을 참조하십시오.
- **Oracle HTTP Server:** *Oracle HTTP Server Administrator's Guide*의 다음 항목을 참조하십시오.
  - 보안 관리
  - Oracle HTTP Server용 SSL 사용으로 설정
- **사용자 디렉토리**: 사용자 디렉토리 공급업체의 설명서를 참조하십시오. 유용한 링크는 다음과 같습니다.

  - **Oracle Internet Directory:** Oracle Internet Directory Administrator's Guide 참조
  - **Sun Java System Directory Server:** *Sun Java System Directory Server 관리 가이드*의 "Directory Server 보안" 참조
  - **Active Directory:** Microsoft 설명서 참조
- **데이터베이스**: 데이터베이스 공급업체의 설명서를 참조하십시오.

위치 참조

이 문서에서는 다음 설치 및 배포 위치를 참조합니다.

- *MIDDLEWARE_HOME*은 Oracle WebLogic Server 그리고 선택적으로 하나 이상의 *EPM_ORACLE_HOME* 같은 미들웨어 구성요소의 위치입니다. *MIDDLEWARE_HOME*은 Oracle Enterprise Performance Management System 제품 설치 중에 정의됩니다. 기본 *MIDDLEWARE_HOME* 디렉토리는 `Oracle/Middleware`입니다.
- *EPM_ORACLE_HOME*은 EPM System 제품을 지원하는 데 필요한 파일이 포함된 설치 디렉토리입니다. *EPM_ORACLE_HOME*은 *MIDDLEWARE_HOME* 내에 있습니다. 기본 *EPM_ORACLE_HOME*은 *MIDDLEWARE_HOME*`/EPMSystem11R1`, 예를 들어 `Oracle/Middleware/EPMSystem11R1`입니다.

  EPM System 제품은 *EPM_ORACLE_HOME*`/products` 디렉토리, 예를 들어 `Oracle/Middleware/EPMSystem11R1/products`에 설치됩니다.

  그뿐 아니라 일부 제품은 EPM System 제품 구성 중에 구성요소를 *MIDDLEWARE_HOME*`/user_projects/epmsystem1`(예: `Oracle/Middleware/user_projects/epmsystem1`)에 배포합니다.
- *EPM_ORACLE_INSTANCE*는 일부 제품이 구성요소를 배포하는 구성 프로세스 중에 정의된 위치를 나타냅니다. *EPM_ORACLE_INSTANCE*의 기본 위치는 *MIDDLEWARE_HOME*`/user_projects/epmsystem1`, 예를 들어 `Oracle/Middleware/user_projects/epmsystem1`입니다.

EPM System 제품에서 SSL을 사용으로 설정하는 작업에 대한 정보

Oracle Enterprise Performance Management System 배포 프로세스가 SSL 모드와 비SSL 모드로 Oracle EPM System 제품을 자동 배포합니다.

> **주:**
>
> - EPM System은 SSL over HTTP 및 JDBC만 지원합니다. Thrift 및 ODBC와 같은 보안 통신을 위한 다른 표준은 지원하지 않습니다.
> - SSLv3 프로토콜에 대한 공격인 Poodle(Padding Oracle On Downgraded Legacy Encryption) 취약점으로부터 보호하려면 EPM System 구성요소에 액세스하는 데 사용되는 서버 및 브라우저에서 SSLv3 지원을 사용 안함으로 설정해야 합니다. SSLv3 지원을 사용 안함으로 설정하려면 서버 및 브라우저 설명서를 참조하십시오.
> - SSL을 구성한 후 비SSL 모드를 사용 안함으로 설정하면 EPM System 서버가 시작되지 않을 수 있습니다.
> 비SSL 모드가 사용 안함으로 설정된 경우 도메인의 모든 EPM System 서버에 대해 보안 복제를 사용으로 설정하여 서버가 시작되도록 하십시오.

EPM System의 공통 설정을 지정하는 동안 배포의 모든 서버 간 통신에서 SSL을 사용으로 설정할 것인지 여부를 지정합니다.

배포 프로세스 중 SSL 설정을 선택해도 SSL에 대한 환경이 자동으로 구성되지 않습니다. Oracle Hyperion Shared Services Registry를 사용하는 모든 EPM System 구성요소에서 서버 간 통신에 보안 프로토콜(HTTPS)을 사용해야 한다는 것을 나타내는 플래그만 Shared Services Registry에서 설정하는 것입니다. 환경에서 SSL을 사용으로 설정하려면 추가 절차를 완료해야 합니다. 이 문서에서 이러한 절차를 다룹니다.

> **주:**
>
> 애플리케이션을 재배포하면 SSL을 사용하도록 지정한 사용자정의 애플리케이션 서버 및 웹 서버 설정이 지워집니다.

> **주:**
>
> Enterprise Performance Management System 릴리스 11.2.x에서는 RCU(Repository Creation Utility)에서 MS SQL Server용 SSL(Secure Sockets Layer)이 지원되지 않습니다.

지원되는 SSL 시나리오

다음 SSL 시나리오가 지원됩니다.

- SSL 오프로더의 SSL 종료. SSL 오프로더에서 SSL 종료을 참조하십시오.
- 전체 SSL 배포. EPM System의 전체 SSL 배포을 참조하십시오.

필요한 인증서

SSL 통신에서는 인증서를 사용하여 구성요소 간에 인증을 설정합니다. 프로덕션 환경에서는 잘 알려진 타사 CA의 인증서를 사용하여 Oracle Enterprise Performance Management System의 SSL을 사용으로 설정하는 것이 좋습니다.

> **주:**
>
> EPM System은 하나의 SSL 인증서로 여러 하위 도메인을 보호할 수 있는 와일드카드 인증서 사용을 지원합니다. 와일드카드 인증서를 사용하면 관리 시간 및 비용을 줄일 수 있습니다.
>
> 와일드카드 인증서를 사용하여 통신을 암호화하는 경우 Oracle WebLogic Server에서 호스트 이름 확인을 사용 안함으로 설정해야 합니다.

EPM System 구성요소를 호스트하는 각 서버에 대해 다음 인증서가 있어야 합니다.

- 루트 CA 인증서

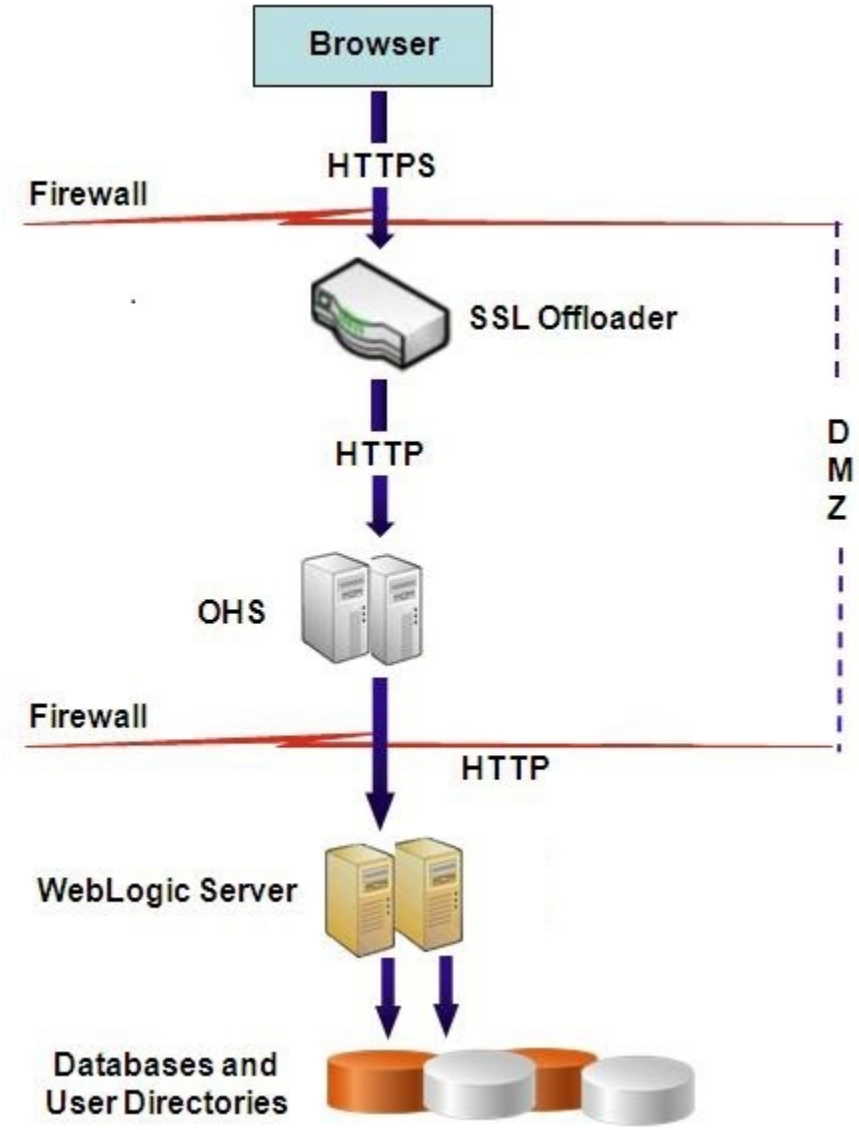
  > **주:**
  >
  > 해당 루트 인증서가 Java 키 저장소에 이미 설치되어 있는 잘 알려진 타사 CA의 인증서를 사용하는 경우 루트 CA 인증서를 Java 키 저장소에 설치할 필요가 없습니다.
  >
  > Firefox 및 Internet Explorer는 잘 알려진 타사 CA의 인증서를 사용하여 사전 로드됩니다. 자체 CA 역할을 하고 있는 경우 해당 브라우저에서 액세스되는 클라이언트가 사용하는 키 저장소로 CA 루트 인증서를 임포트해야 합니다. 자체 CA 역할을 하고 있는 경우 클라이언트에 액세스하는 브라우저에서 CA 루트 인증을 사용할 수 없는 경우 웹 클라이언트가 서버와의 SSL 핸드셰이크를 설정할 수 없습니다.

- 배포에 있는 각 Oracle HTTP Server의 서명된 인증서
- WebLogic Server 호스트 머신의 서명된 인증서. 이 머신의 관리 서버도 이 인증서를 사용할 수 있습니다.
- SSL 오프로더/로드 밸런서의 두 인증서. 이러한 인증서 중 하나는 외부 통신용이며 다른 하나는 내부 통신용입니다.

SSL 오프로더에서 SSL 종료

배포 아키텍처

이 시나리오에서는 SSL을 사용하여 Oracle Enterprise Performance Management System 클라이언트(예: 브라우저)와 SSL 오프로더 간의 통신 링크를 보호합니다. 그림으로 된 개념은 다음과 같습니다.



가정

SSL 오프로더 및 로드 밸런서

로드 밸런서를 사용하는 전체 구성된 SSL 오프로더가 배포 환경에 있어야 합니다.

로드 밸런서는 가상 호스트에서 수신하는 모든 요청을 Oracle HTTP Server로 전달하도록 구성되어야 합니다.

SSL이 OHS(Oracle HTTP Server) 또는 로드 밸런서에서 종료되는 경우 다음을 수행해야 합니다.

- 모든 논리 웹 애플리케이션을 로드 밸런서 또는 Oracle HTTP Server의 비SSL 가상 호스트로 설정해야 합니다(예: `empinternal.myCompany.com:80`, 여기서 80은 비SSL 포트임). 구성 화면을 열고 다음 단계를 완료합니다.
  1. **Hyperion Foundation** 구성 태스크를 확장합니다.
  2. **웹 애플리케이션의 논리 주소 구성**을 선택합니다.
  3. *호스트 이름*, 비SSL 포트 번호 및 SSL 포트 번호를 지정합니다.

- 로드 밸런서 또는 Oracle HTTP Server의 SSL 지원 가상 호스트에 대한 외부 URL을 설정합니다(예: `empexternal.myCompany.com:443`, 여기서 443은 SSL 포트임). 구성 화면을 열고 다음 단계를 완료합니다.
  1. **Hyperion Foundation** 구성 태스크를 확장합니다.
  2. **공통 설정 구성**을 선택합니다.
  3. 외부 URL 세부정보에서 **SSL 오프로딩 사용**을 선택합니다.
  4. *외부 URL 호스트* 및 *외부 URL 포트*를 지정합니다.

> **주:**
>
> **configtool**을 사용하여 웹 애플리케이션을 재배포하거나 웹 서버를 재구성하면 논리 웹 애플리케이션 및 외부 URL에 대한 설정이 대체됩니다.

가상 호스트

SSL 오프로더에서 종료되는 SSL 구성에서는 두 개의 서버 별칭을 사용합니다. 즉, SSL 오프로더/로드 밸런서의 `epm.myCompany.com` 및 `empinternal.myCompany.com`와 같이 하나는 오프로더와 브라우저 간의 외부 통신용이며 다른 하나는 EPM System 서버의 내부 통신용입니다. 서버 별칭은 머신의 IP 주소를 가리키고 DNS를 통해 확인할 수 있어야 합니다.

오프로더 및 브라우저 간 외부 통신을 지원하는(`epm.myCompany.com` 사용) 서명된 인증서는 오프로더/로드 밸런서에 설치되어야 합니다.

EPM System 구성

EPM System 구성요소의 기본 배포는 SSL 오프로더에서 SSL 종료를 지원합니다. 추가 작업이 필요하지 않습니다.

EPM System을 구성할 때 웹 애플리케이션의 논리 주소가 내부 통신용으로 생성된 별칭(예: `empinternal.myCompany.com`)을 가리키는지 확인하십시오. EPM System을 설치 및 구성하려면 다음 정보 소스를 참조하십시오.

- *Oracle Enterprise Performance Management System 설치 및 구성 가이드*
- *Oracle Enterprise Performance Management System 설치 시작 페이지*
- *Oracle Enterprise Performance Management System 설치 및 구성 문제 해결 가이드*

배포 테스트

배포 프로세스를 완료한 후 보안 Oracle Hyperion Enterprise Performance Management Workspace URL에 연결하여 모든 요소가 작동하는지 확인합니다.

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

`https://epm.myCompany.com:443/workspace/index.jsp`를 예로 들 수 있습니다. 여기서, 443은 SSL 포트입니다.

EPM System의 전체 SSL 배포

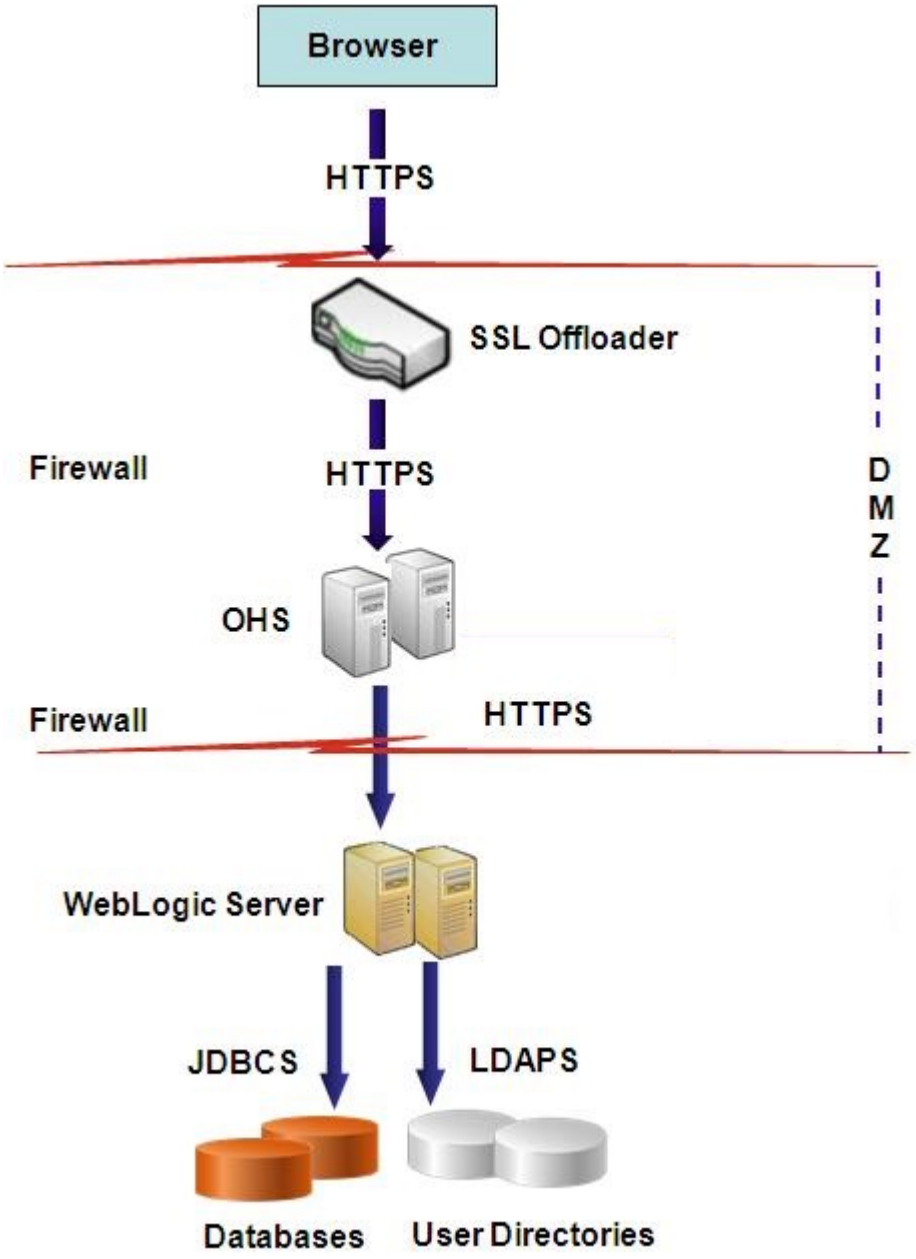
참조:

- 배포 아키텍처
- 가정
- 전체 SSL에 대해 EPM System 구성

배포 아키텍처

전체 SSL 모드에서는 모든 보안 가능 채널의 통신이 SSL을 통해 보호됩니다. 이 Oracle Enterprise Performance Management System 배포 시나리오가 가장 안전합니다.

그림으로 된 개념은 다음과 같습니다.



가정

데이터베이스

데이터베이스 서버 및 클라이언트의 SSL이 사용으로 설정되어 있습니다. 데이터베이스 서버 및 클라이언트의 SSL을 사용으로 설정하는 작업에 대한 정보는 데이터베이스 설명서를 참조하십시오.

EPM System

Oracle WebLogic Server 및 Oracle HTTP Server를 비롯한 Oracle Enterprise Performance Management System 구성요소가 설치 및 배포되어 있습니다. 또한, EPM

System 환경을 테스트하여 비SSL 모드에서도 모든 요소가 작동하는 것을 확인했습니다. 다음 정보 소스를 참조하십시오.

- *Oracle Enterprise Performance Management System 설치 및 구성 가이드*
- *Oracle Enterprise Performance Management System 설치 시작 페이지*
- *Oracle Enterprise Performance Management System 설치 및 구성 문제 해결 가이드*

구성 프로세스 중에 데이터베이스 연결에서 SSL을 사용으로 설정하려는 경우 각 데이터베이스 구성 화면에서 **고급 옵션** 링크를 선택하고 다음을 포함한 필수 설정을 지정해야 합니다.

- **데이터베이스에 보안 연결 사용(SSL)**을 선택하고 보안 데이터베이스 URL(예: `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=`*`myDBhost`*`)(PORT=1529)(CONNECT_DATA=(SERVICE NAME=`*`myDBhost.myCompany.com`*`)))`) 입력
- **인증된 키 저장소**
- **인증된 키 저장소 비밀번호**

자세한 내용은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

SSL 오프로더 및 로드 밸런서

로드 밸런서를 사용하는 전체 구성된 SSL 오프로더가 배포 환경에 있어야 합니다.

전체 SSL 구성에서는 `epm.myCompany.com` 및 `empinternal.myCompany.com`과 같은 두 개의 서버 별칭을 SSL 오프로더에서 사용합니다. 하나는 오프로더와 브라우저 간 외부 통신을 위한 것이며, 다른 하나는 EPM System 서버 간 내부 통신을 위한 것입니다. 서버 별칭은 머신의 IP 주소를 가리키고 DNS를 통해 확인할 수 있어야 합니다.

로드 밸런서는 가상 호스트에서 수신하는 모든 요청을 Oracle HTTP Server로 전달하도록 구성되어야 합니다.

오프로더/로드 밸런서에는 두 개의 서명된 인증서가 설치되어야 합니다. 하나는 오프로더 및 브라우저 간 외부 통신을 지원하는 인증서이고(`epm.myCompany.com` 사용) 다른 하나는 애플리케이션 간 내부 통신을 지원하는 인증서입니다(`empinternal.myCompany.com` 사용). 이러한 인증서는 서버 이름의 노출을 방지하고 보안을 강화하기 위해 서버 별칭에 연결하는 것이 좋습니다.

전체 SSL에 대해 EPM System 구성

참조:

- EPM System 공통 설정 재구성
- 선택사항: WebLogic Server에 대해 루트 CA 인증서 설치
- WebLogic Server에 인증서 설치
- WebLogic Server 구성
- SSL 지원 Oracle 데이터베이스에서 HFM 서버 연결 사용
- Oracle HTTP Server 절차
- WebLogic Server에 배포된 EPM System 웹 구성요소 구성
- 도메인 구성 업데이트
- 서버 및 EPM System 재시작
- 배포 테스트

- SSL 사용 외부 사용자 디렉토리 구성

EPM System 공통 설정 재구성

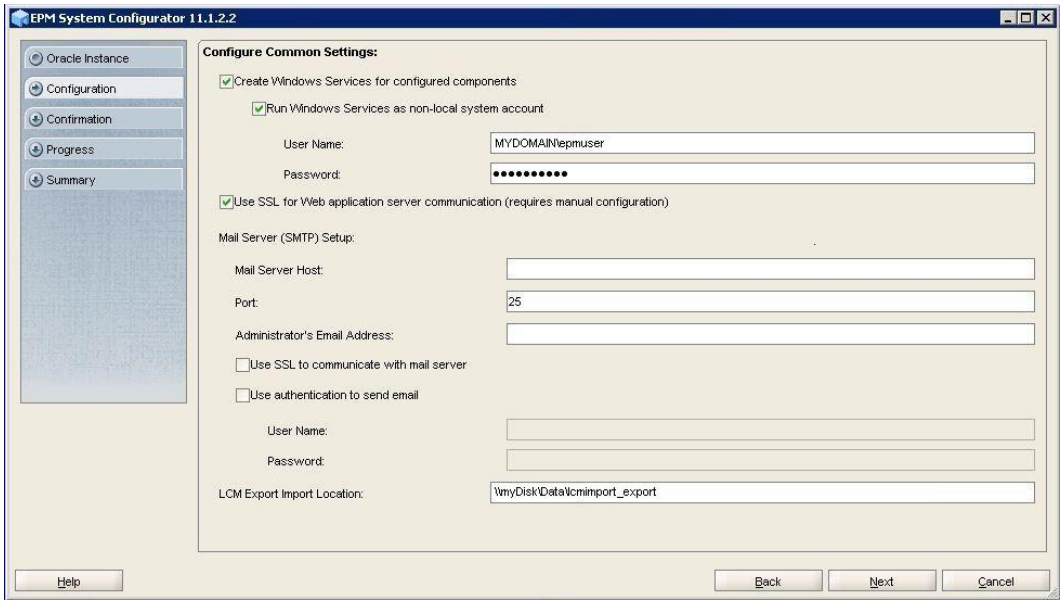
이 프로세스를 진행하는 중에 Oracle Enterprise Performance Management System 구성요소에서 SSL 통신을 사용하도록 하는 설정을 선택합니다.

> **주:**
>
> **Oracle Hyperion Financial Management 웹 서버의 SSL을 사용으로 설정하는 경우:** Financial Management를 구성하기 전에 `weblogic.xml`에서 HFM WebApp의 세션 기술자를 편집하여 쿠키 보안을 설정해야 합니다.
>
> 1. 7 Zip과 같은 툴을 사용하여 Financial Management 웹 아카이브를 확장합니다. 아카이브에서 `weblogic.xml`의 위치는 *EPM_ORACLE_HOME*`\products\FinancialManagement\AppServer\InstallableApps\HFMWebAppli`입니다.
> 2. `weblogic.xml`에서 HFM WebApp의 세션 기술자에 다음 지시어를 포함합니다.
>
>    ```
>    <cookie-secure>true</cookie-secure>
>    ```
>
> 3. `weblogic.xml`을 저장합니다.
> 4. 7 Zip에서 아카이브를 업데이트할 것인지 쿼리하면 **예**를 누릅니다.

SSL에 대해 EPM System을 재구성하려면 다음을 수행합니다.

1. EPM System Configurator를 실행합니다.
2. **구성을 적용할 EPM Oracle 인스턴스 선택**에서 다음 단계를 완료합니다.
   a. **EPM Oracle 인스턴스 이름**에는 처음에 EPM System 구성요소를 구성할 때 사용한 인스턴스 이름을 입력합니다.
   b. **다음**을 누릅니다.
3. 구성 화면에서 다음 단계를 완료합니다.
   a. **모두 선택 취소**를 지웁니다.
   b. **Hyperion Foundation** 구성 태스크를 확장하고 **공통 설정 구성**을 선택합니다.
   c. **다음**을 누릅니다.
4. **공통 설정 구성**에서 다음 단계를 완료합니다.

> **주의:**
>
> SSL을 사용하여 전자메일 서버와 통신하는 설정을 선택하기 전에 전자메일 서버가 SSL에 대해 구성되어 있는지 확인하십시오.



a. **Java 웹 애플리케이션 서버 통신에 SSL 사용(수동 구성 필요)**을 선택하여 EPM System이 통신에 SSL을 사용하도록 지정합니다.

b. **선택사항: 메일 서버 호스트** 및 **포트**에 정보를 입력합니다. SSL 통신을 지원하려면 SMTP 메일 서버에서 사용하는 보안 포트를 지정해야 합니다.

c. **선택사항:** SMTP 메일 서버와의 SSL 통신을 지원하려면 **메일 서버와 통신에 SSL 사용**을 선택합니다.

d. 나머지 필드의 설정을 선택하거나 입력합니다.

e. **다음**을 누릅니다.

5. 이어지는 EPM System Configurator 화면에서 **다음**을 누릅니다.
6. 배포 프로세스가 완료되면 요약 화면이 표시됩니다. **완료**를 누릅니다.

선택사항: WebLogic Server에 대해 루트 CA 인증서 설치

잘 알려진 타사 CA의 루트 인증서가 이미 JVM 키 저장소에 설치되어 있습니다. 잘 알려진 타사 CA의 인증서를 사용하지 않는 경우(권장되지 않음) 이 절의 절차를 완료하십시오. 기본 JVM 키 저장소 위치는 *MIDDLEWARE_HOME*/jdk/jre/lib/security/cacerts입니다.

> **주:**
>
> 각 Oracle Enterprise Performance Management System 서버에서 이 절차를 수행하십시오.

루트 CA 인증서를 설치하려면 다음을 수행합니다.

1. 루트 CA 인증서를 Oracle WebLogic Server가 설치된 머신의 로컬 디렉토리에 복사합니다.
2. 콘솔에서 디렉토리를 *MIDDLEWARE_HOME*/jdk/jre/bin으로 변경합니다.

3. 다음과 같은 keytool 명령을 실행하여 루트 CA 인증서를 JVM 키 저장소에 설치합니다.

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

예를 들어 다음 명령을 사용하면 현재 디렉토리에 저장된 `CAcert.crt` 인증서를 키 저장소의 인증서 별칭으로 `Blister`를 사용하여 JVM 키 저장소에 추가할 수 있습니다. 저장소 비밀번호는 `example_pwd`라고 가정합니다.

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

> **주:**
>
> 앞의 명령 및 예에서는 keytool을 사용하여 인증서를 임포트하는 구문 중 일부가 사용되었습니다. 임포트 구문 전체 목록은 keytool 설명서를 참조하십시오.

WebLogic Server에 인증서 설치

기본 Oracle WebLogic Server 설치에서는 데모 인증서를 사용하여 SSL을 지원합니다. 잘 알려진 타사 인증서를 설치하여 환경 보안을 강화하는 것이 좋습니다.

WebLogic Server를 호스트하는 각 머신에서 툴(예: keytool)을 사용하여 WebLogic Server 및 Oracle Enterprise Performance Management System 웹 구성요소의 서명된 인증서를 저장할 사용자정의 키 저장소를 생성하십시오.

사용자정의 키 저장소를 생성하고 인증서를 임포트하려면 다음을 수행합니다.

1. 콘솔에서 디렉토리를 *MIDDLEWARE_HOME*`/jdk/jre/bin`으로 변경합니다.
2. 다음과 같은 keytool 명령을 실행하여 기존 디렉토리에 사용자정의 키 저장소(명령에서 `-keystore` 지시어로 확인됨)를 생성합니다.

```
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
```

> **주:**
>
> 설정된 cn(공용 이름)은 서버 이름과 일치해야 합니다. FQDN(전체 도메인 이름)을 cn으로 사용하는 경우 웹 구성요소를 배포할 때 FQDN을 사용해야 합니다.

3. 인증서 요청을 생성합니다.

   ```
   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 머신의 서명된 인증서를 가져옵니다.
5. 다음과 같이 서명된 인증서를 키 저장소로 임포트합니다.

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

WebLogic Server 구성

Oracle Enterprise Performance Management System 웹 구성요소를 배포한 후에는 SSL 통신에 대해 구성해야 합니다.

웹 구성요소를 SSL에 대해 구성하려면 다음을 수행합니다.

1. *MIDDLEWARE_HOME*/user_projects/domains/EPMSystem/bin/startWebLogic.cmd를 실행하여 Oracle WebLogic Server를 시작합니다.
2. 다음 URL에 액세스하여 WebLogic Server 관리 콘솔을 실행합니다.

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

   예를 들어 myServer의 기본 포트에 배포된 WebLogic Server 콘솔에 액세스하려면 http://myServer:7001/console을 사용해야 합니다.
3. 시작 화면에서 EPM System Configurator에서 지정한 WebLogic Server 관리자 사용자 이름 및 비밀번호를 입력합니다.
4. **변경 센터**에서 **잠금 및 편집**을 누릅니다.
5. 콘솔의 왼쪽 창에서 **환경**을 확장하고 **서버**를 선택합니다.
6. 서버 요약 화면에서 SSL을 사용으로 설정할 서버의 이름을 누릅니다.

   예를 들어 Oracle Hyperion Foundation Services 구성요소에 대해 SSL을 사용으로 설정하려면 EPMServer0 서버로 작업합니다.
7. **수신 포트 사용**을 지워 HTTP 수신 포트를 사용 안함으로 설정합니다.
8. **SSL 수신 포트 사용**이 선택되어 있는지 확인합니다.
9. **SSL 수신 포트**에서 이 서버가 요청을 수신해야 하는 SSL 수신 포트를 입력합니다.
10. 사용할 ID 및 인증 키 저장소를 지정하려면 **키 저장소**를 선택하여 [키 저장소] 탭을 엽니다.
11. **변경**을 누릅니다.
12. 옵션을 선택합니다.
    - **사용자정의 ID 및 사용자정의 인증** - 잘 알려진 타사 CA의 서버 인증서를 사용하고 있지 않은 경우
    - **사용자정의 ID 및 Java Standard Trust** - 잘 알려진 타사 CA의 서버 인증서를 사용하고 있는 경우

13. **저장**을 누릅니다.
14. **사용자정의 ID 키 저장소**에서 서명된 WebLogic Server 인증서가 설치된 키 저장소의 경로를 입력합니다.
15. **사용자정의 ID 키 저장소 유형**에서 `jks`를 입력합니다.
16. **사용자정의 ID 키 저장소 문장암호** 및 **사용자정의 ID 키 저장소 문장암호 확인**에 키 저장소 비밀번호를 입력합니다.
17. **키 저장소**에서 **사용자정의 ID 및 사용자정의 인증**을 선택한 경우 다음을 수행합니다.
    - **사용자정의 인증 키 저장소**에서 서버 인증서에 서명한 CA의 루트 인증서를 사용할 수 있는 사용자정의 키 저장소 경로를 입력합니다.
    - **사용자정의 인증 키 저장소 유형**에서 `jks`를 입력합니다.
    - **사용자정의 트러스트 키 저장소 문장암호** 및 **사용자정의 트러스트 키 저장소 문장암호 확인**에 키 저장소 비밀번호를 입력합니다.
18. **저장**을 누릅니다.
19. SSL 설정을 지정합니다.
    - **SSL**을 선택합니다.
    - **개인 키 별칭**에서 서명된 WebLogic Server 인증서를 임포트하는 중 지정한 별칭을 입력합니다.
    - **개인 키 문장암호** 및 **개인 키 문장암호 확인**에서 개인 키를 검색하는 데 사용할 비밀번호를 입력합니다.
    - **저장**을 누릅니다.

    > **주:**
    >
    > SHA-2 인증서를 사용하는 경우 EPM System을 지원하는 데 사용되는 모든 관리 서버에 대해 **JSSE SSL 사용** 설정을 선택해야 합니다. 이 설정은 SSL 페이지의 [고급] 탭에서 사용할 수 있습니다. 이 변경사항을 활성화하려면 WebLogic Server를 재시작해야 합니다.

20. 서버에 대해 보안 복제를 사용으로 설정합니다.
    a. 콘솔의 왼쪽 창에서 **환경**을 확장하고 **클러스터**를 누릅니다.
    b. 클러스터 요약에서 보안 복제를 사용으로 설정할 서버 이름(예: `Foundation Services`)을 누릅니다.

       선택한 서버의 설정 화면 [구성] 탭이 표시됩니다.
    c. **복제**를 눌러 [복제] 탭을 엽니다.
    d. **보안 복제 사용**을 선택합니다. 이 옵션을 선택하려면 먼저 **잠금 및 편집**을 눌러야 합니다.
    e. **저장**을 누릅니다.
21. 이 호스트에 속하는 각 관리 서버에 대해 6단계부터 20단계까지 완료합니다.
22. 보안 복제를 사용으로 설정하여 클러스터에 복제 호출을 위한 채널을 제공합니다.

    자세한 내용은 Oracle MetaLink 문서 1319381.1을 참조하십시오.
    - 관리 콘솔에서 **환경**을 확장하고 **클러스터**를 선택합니다.

- **복제**를 선택합니다.
- **복제**에서 **보안 복제 사용**을 선택합니다.
- **저장**을 누릅니다.

23. **변경 센터**에서 **변경 활성화**를 누릅니다.

SSL 지원 Oracle 데이터베이스에서 HFM 서버 연결 사용

SSL을 사용하여 HFM 데이터소스와 Oracle 데이터베이스 간 네트워크 연결을 암호화할 수 있습니다. 이 작업을 수행하려면 Oracle 설명서에 요약된 대로 Oracle Wallet을 구성해야 합니다. SSL로 암호화된 연결에 사용되는 새 포트에서 수신하도록 TNS 리스너도 구성해야 합니다. 마지막으로 HFM 데이터소스를 호스트하는 서버의 키 저장소 및 트러스트 저장소로 적절한 인증서를 로드해야 합니다. 아래 지침은 Oracle 데이터베이스 설명서에서 참조되었습니다.

사전 필수 조건

아래 단계를 진행하기 전에 다음 사전 필수 조건이 충족되었는지 확인하십시오.

- 작동되는 데이터베이스 서버.
- SSL 지원 TNS 리스너가 실행되고 있는 포트의 서버와 통신하는 것을 차단하는 로컬 또는 네트워크 방화벽이 없는지 확인하십시오.

아래 예에서는 MS Windows Server 2016에서 실행 중인 Oracle 12c(12.1.0.2) 버전이 사용되었습니다. 이 지침은 지정된 경로가 전자 지갑 파일용이며 Linux 파일 시스템 경로 및 환경 변수 대체가 데이터베이스 서버에서 사용되는 쉘에 적절하게 변경된 경우 Linux 설치에서도 동일하게 적용됩니다. 동일한 지침이 19c 개발 및 지원 인스턴스에서도 성공적으로 사용되었습니다.

이 문서의 예에서는 자체 서명 인증서를 사용하지만 원하는 경우 적절한 인증 기관 인증서를 사용할 수도 있습니다. 인증 기관에서 발행한 인증서를 설치할 때 수행할 정확한 단계는 Oracle 데이터베이스 설명서를 참조하십시오.

Oracle 데이터베이스 구성

Oracle 데이터베이스를 구성하려면 아래 단계를 따르십시오.

1. 데이터베이스 서버에 새 자동 로그인 전자 지갑을 생성합니다.

> **주:**
>
> 이러한 단계는 이전에 Oracle Wallet을 생성하지 않은 경우에만 필요합니다. 데이터베이스 서버에서 GUI Oracle Wallet 툴이 사용되는 경우 다음 단계가 필요하지 않습니다.

```
C:\> cd %ORACLE_HOME%
C:\oracledb\12.1.0\home> mkdir wallet
C:\oracledb\12.1.0\home> orapki wallet create -wallet wallet -pwd
password1 -auto_login
```

`orapki` 명령행에서 `-auto_login_local`을 사용하라는 메시지는 무시할 수 있습니다. SSL 인증 실패 오류가 발생하는 경우 Doc ID 2238096.1을 참조하여 이슈를 해결하십시오.

또한, 전자 지갑 디렉토리 아래에 있는 cwallet.sso 파일의 보안 권한을 확인하여 Oracle 리스너 서비스 사용자에게 이 파일에 대한 읽기 권한이 있는지 확인하십시오. 읽기 권한이 없는 경우 나중에 SSL 핸드셰이크가 실패합니다. Oracle 데이터베이스가 로그온할 수 없는 제안된 Oracle 사용자로 설치된 경우 이러한 상황이 발생합니다. Oracle 데이터베이스가 Oracle 사용자로 설치된 경우 TNS 리스너는 다른 사용자로 실행되어야 합니다.

2. 자체 서명 인증서를 생성하여 전자 지갑에 로드합니다.

```
C:\oracledb\12.1.0\home> orapki wallet add -wallet wallet -pwd
password1 -dn "CN={FQDN of db server}" -
keysize 1024 -self_signed -validity 3650
```

위 예의 password1 비밀번호는 *1단계*에서 지정된 비밀번호와 일치해야 합니다.

3. 새로 생성된 자체 서명 인증서를 익스포트합니다.

```
C:\oracledb\12.1.0\home> orapki wallet export -wallet wallet -pwd
password1 -dn "CN={FQDN of db server}"
-cert %COMPUTERNAME%-certificate.crt
```

4. 익스포트한 Base64 인증서 파일을 HFM 서버에 복사합니다.
5. SQL*NET 및 TNS 리스너를 구성합니다.
   a. 데이터베이스 서버에서 사용되지 않는 포트를 확인합니다. 아래 예에서는 1522 포트에 새 리스너를 생성합니다. SSL 연결에 일반적으로 사용되는 포트는 2484이며 모든 사용가능한 포트를 사용할 수 있습니다. 계속 진행하기 전에 사용하려는 포트가 데이터베이스 서버에서 사용가능한지 확인하고 필요에 따라 조정해야 합니다.
   b. SQLNET.ORA를 업데이트합니다. WALLET_LOCATION 선언의 DIRECTORY 요소는 *1단계*에서 생성된 전자 지갑을 가리켜야 합니다.

```
SQLNET.AUTHENTICATION_SERVICES= (TCPS, NTS, BEQ)
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
WALLET_LOCATION=
(SOURCE =
(METHOD = FILE)
(METHOD_DATA =
(DIRECTORY = C:\oracledb\12.1.0\home\wallet)
)
)
SSL_CLIENT_AUTHENTICATION = FALSE
```

   c. LISTENER.ORA를 업데이트하여 새 리스너를 정의합니다. 위의 *5a 단계*에서 확인된 포트를 사용합니다.

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = CLRExtProc)
(ORACLE_HOME = C:\oracledb\12.1.0\home)
(PROGRAM = extproc)
(ENVS =
"EXTPROC_DLLS=ONLY:C:\oracledb\12.1.0\home\bin\oraclr12.dll")
```

```
)
)
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION=
(SOURCE =
(METHOD = FILE)
(METHOD_DATA =
(DIRECTORY = C:\oracledb\12.1.0\home\wallet)
)
)
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = myServer)(PORT = 1521))
)
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCPS)(HOST = myServer)(PORT = 1522))
)
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
)
)
ADR_BASE_LISTENER = C:\oracledb
```

d. 새 포트의 TNSNAMES.ORA에 새 항목을 생성합니다.

```
ORCL_SSL =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCPS)(HOST = myServer)(PORT = 1522))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = myServer_service)
)
)
```

위의 *5a 단계*에서 확인되고 *5c 단계*에서 사용된 포트와 동일한 포트를 지정해야 합니다.

e. TNS 리스너를 재시작합니다.

```
C:\oracledb\12.1.0\home>lsnrctl stop
C:\oracledb\12.1.0\home>lsnrctl start
```

f. 새 TNS 리스너가 작동하는지 확인합니다.

```
C:\oracledb\12.1.0\home>tnsping orcl_ssl
TNS Ping Utility for 64-bit Windows: Version 12.1.0.2.0 - Production
on 10-SEP-2019 15:43:22
Copyright (c) 1997, 2014, Oracle. All rights reserved.
Used parameter files:
C:\oracledb\12.1.0\home\network\admin\sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)
(HOST = myServer)
(PORT = 1522)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
```

```
myServer_service)))
OK (130 msec)
```

SSL 데이터베이스 연결을 사용하도록 HFM 서버 구성

HFM 서버의 트러스트 저장소에 데이터베이스 인증서 추가

HFM 데이터소스가 실행되는 모든 EPM 서버마다 다음 단계를 수행해야 합니다. 아래에서 사용된 `%MW_HOME%` 환경 변수는 Oracle Middleware가 설치되는 위치입니다. 이 환경 변수는 EPM 설치 중 기본적으로 생성되지 않으며 아래 단계에서는 EPM 설치의 상위 디렉토리를 표시하는 데 사용됩니다.

EPM이 설치되는 위치는 `EMP_ORACLE_HOME` 환경 변수로 지정됩니다. 아래 예에서는 키 저장소 및 트러스트 저장소를 EPM이 설치된 동일한 위치의 디렉토리에 배치합니다. 키 저장소 및 트러스트 저장소 파일은 HFM 서버의 파일 시스템 어느 곳이든 있을 수 있습니다.

1. `%MW_HOME%` 아래에 새 디렉토리를 생성하여 Java 키 저장소와 PKCS12 트러스트 저장소를 저장합니다.
   a. cd `%MW_HOME%`
   b. mkdir `certs`
2. JDK에서 Java 키 저장소 파일 cacerts를 복사합니다.
   a. cd `%MW_HOME%\certs`
   b. copy `%MW_HOME%\jdk1.8.0_181\jre\lib\security\cacerts testing_cacerts`
      JDK 키 저장소를 복사하여 JDK 기본 키 저장소 대신 사용하는 이유는 JDK가 업그레이드되고 이전 JDK가 삭제되는 경우 기본 키 저장소에 삽입된 키 및 인증서가 손실되기 때문입니다.
3. Base 64 인증서를 `%MW_HOME%\certs`에 복사합니다.
4. 인증서를 Java 키 저장소 파일 `testing_cacerts`로 임포트합니다.
   a. `keytool -importcert -file bur00cbb-certificate.crt -keystore testing_cacerts -alias "myserver"`를 예로 들 수 있습니다.
      i. 키 저장소의 비밀번호를 지정해야 합니다.
      ii. "myserver"를 데이터베이스 서버의 정규화된 도메인 이름으로 바꿔야 합니다.
   b. 인증서를 신뢰하는지 확인하는 질문이 표시되면 **y**를 지정합니다.
5. JDK Java 키 저장소 파일에서 PKCS12 형식으로 트러스트 저장소를 생성합니다. 예를 들면 다음과 같습니다.

```
keytool -importkeystore -srckeystore testing_cacerts -srcstoretype
JKS -deststoretype PKCS12 -destkeystore testing_cacerts.pfx
```

SSL을 사용하도록 HFM JDBC 연결 업데이트

1. SSL을 사용하도록 HFM 데이터베이스 JDBC 연결을 재구성합니다.
   a. EPM 구성 툴을 실행합니다.
      i. **Financial Management** 노드 아래에서 **데이터베이스 구성** 및 **애플리케이션 서버에 배포** 노드를 선택합니다.

ii. **다음**을 누릅니다.

iii. HFM JDBC 연결에 대해 이러한 단계를 각각 수행합니다.

i. 포트, 서비스 이름, 사용자 이름, 비밀번호 열에 해당 연결에 대한 SSL 포트, 서비스 이름, 사용자 이름, 비밀번호를 입력합니다.

ii. **(+)**를 눌러 **고급 데이터베이스 옵션**을 엽니다.

iii. **보안 연결 사용** 확인란을 선택합니다.

iv. *2단계*에서 생성된 Java 키 저장소의 위치를 입력합니다.

v. **적용**을 누릅니다.

vi. **(+)**를 눌러 **고급 데이터베이스 옵션**을 엽니다.

vii. **수정된 JDBC URL 편집 및 사용**을 누릅니다. 표시된 JDBC URL을 변경하지 않아야 합니다.

viii. **적용**을 누릅니다.

ix. **다음**을 누릅니다.

b. EPM 설명서에 설명된 대로 나머지 단계를 진행하여 HFM 애플리케이션을 배포합니다.

2. 명령 창 또는 쉘을 열어 데이터소스에서 사용하는 ODBC 연결에 SSL이 지원될 수 있도록 EPM 레지스트리를 수동으로 업데이트합니다.
아래 나열된 각 명령을 실행합니다.

```
epmsys_registry.bat addproperty FINANCIAL_MANAGEMENT_PRODUCT/
DATABASE_CONN/@ODBC_TRUSTSTORE "C:
\Oracle\Middleware\certs\testing_cacerts.pfx"
epmsys_registry.bat addencryptedproperty FINANCIAL_MANAGEMENT_PRODUCT/
DATABASE_CONN
/@ODBC_TRUSTSTOREPASSWORD <truststorepassword>
epmsys_registry.bat addproperty FINANCIAL_MANAGEMENT_PRODUCT/DATABASE_CONN
/@ODBC_VALIDATESERVERCERTIFICATE false
```

위의 예에서 `C:\Oracle\Middleware` 경로는 1, 2, 3단계의 `%MW_HOME%` 값입니다.

자체 서명 인증서가 사용되는 경우 `FINANCIAL_MANAGEMENT_PRODUCT/DATABASE_CONN/@ODBC_VALIDATESERVERCERTIFICATE` 등록정보는 false로만 설정되어야 합니다. `FINANCIAL_MANAGEMENT_PRODUCT/DATABASE_CONN/@ODBC_TRUSTSTOREPASSWORD` 값은 *2단계*에서 복사된 원래 Java 키 저장소의 비밀번호여야 합니다.

HFM에서 사용되는 TNS 이름 항목 업데이트

`TNSNAMES.ORA`를 편집하여 새 항목을 생성하고 이전 항목의 이름을 바꿉니다. 다음 예에서는 필요한 변경사항이 적용된 HFM 서버의 업데이트된 `TNSNAMES.ORA` 파일을 보여 줍니다. 이러한 변경은 HFM에서 `HFMTNS`라는 TNS 이름 항목을 찾아 사용하기 때문에 필요합니다. 이 항목이 제대로 작동하려면 `XFMDataSource`에 맞게 변경된 프로토콜 및 포트가 포함되어 있어야 합니다.

```
HFMTNS_UNENC =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = myserver)(PORT = 1521))
)
```

```
(CONNECT_DATA =
(SERVICE_NAME = myserver_service)
(SERVER = DEDICATED)
)
)
HFMTNS =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCPS)(HOST = myserver)(PORT = 1522))
)
(CONNECT_DATA =
(SERVICE_NAME = myserver_service)
(SERVER = DEDICATED)
)
)
```

원래 HFMTNS 항목의 이름이 HFMTNS_UNENC로 바뀌었습니다. HFMTNS_UNENC 항목을 복사하고 HFMTNS로 이름을 바꿔 새 HFMTNS가 생성되었습니다. 그런 다음, 프로토콜이 TCPS로 업데이트되고 포트가 1522로 변경되었습니다. 지정된 포트는 TNS LISTENER.ORA 파일에 지정된 포트와 동일해야 합니다.

Oracle HTTP Server 절차

Oracle HTTP Server의 전자 지갑 생성 및 인증서 설치

기본 전자 지갑은 Oracle HTTP Server와 함께 자동으로 설치됩니다. 배포의 각 Oracle HTTP Server에 대해 실제 전자 지갑을 구성해야 합니다.

참고: 11.2.x부터 Oracle Wallet Manager는 Oracle HTTP Server와 함께 설치되지 않습니다. Oracle 데이터베이스 클라이언트를 설치하는 경우에만 Oracle Wallet Manager가 설치됩니다. 데이터베이스 클라이언트와 함께 제공되는 전자 지갑 관리자를 사용하여 전자 지갑을 생성하고 인증서를 임포트해야 합니다. SSL용 Oracle HTTP Server를 구성하는 경우 항상 EPM System 제품 설치의 일부로 Oracle 데이터베이스 클라이언트 64비트를 설치해야 합니다.

Oracle HTTP Server 인증서를 생성 및 설치하려면 다음을 수행합니다.

1. Oracle HTTP Server를 호스트하는 각 머신에서 Wallet Manager를 실행합니다.

   시작, **모든 프로그램**, **Oracle-OHxxxxxx**, **통합된 관리 툴**, **Wallet Manager** 순으로 선택합니다.

   xxxxxx는 Oracle HTTP Server 인스턴스 번호입니다.

2. 비어 있는 새 전자 지갑을 생성합니다.
   a. Oracle Wallet Manager에서 **전자 지갑**, **새로 작성** 순으로 선택합니다.
   b. **예**를 눌러 기본 전자 지갑 디렉토리를 생성하거나 **아니요**를 눌러 선택한 위치에 전자 지갑 파일을 생성합니다.
   c. 새 전자 지갑 화면의 **전자 지갑 비밀번호** 및 **비밀번호 확인**에 사용할 비밀번호를 입력합니다.
   d. **확인**을 누릅니다.
   e. 확인 대화상자에서 **아니요**를 누릅니다.

3. **선택사항:** Oracle HTTP Server에 알려진 CA를 사용하지 않는 경우 루트 CA 인증서를 전자 지갑으로 임포트합니다.
   a. Oracle Wallet Manager에서 **인증된 인증서**를 마우스 오른쪽 버튼으로 누르고 **인증된 인증서 임포트**를 선택합니다.
   b. 루트 CA 인증서를 찾아보고 선택합니다.
   c. **열기**를 선택합니다.
4. 인증서 요청을 생성합니다.
   a. Oracle Wallet Manager에서 **인증서: [비어 있음]**을 마우스 오른쪽 버튼으로 누르고 **인증서 요청 추가**를 선택합니다.
   b. 인증서 요청 생성에서 필수 정보를 입력합니다.

      공용 이름의 경우 시스템의 `hosts` 파일에 제공된 전체 서버 별칭(예: `epm.myCompany.com` 또는 `epminternal.myCompany.com`)을 입력합니다.
   c. **확인**을 누릅니다.
   d. 확인 대화상자에서 **확인**을 누릅니다.
   e. 생성한 인증서 요청을 마우스 오른쪽 버튼으로 누르고 **인증서 요청 익스포트**를 선택합니다.
   f. 인증서 요청 파일의 이름을 지정합니다.
5. 인증서 요청 파일을 사용하여 CA에서 서명된 인증서를 가져옵니다.
6. 서명된 인증서를 임포트합니다.
   a. Oracle Wallet Manager에서 서명된 인증서를 가져오는 데 사용한 인증서 요청을 마우스 오른쪽 버튼으로 누르고 **사용자 인증서 임포트**를 선택합니다.
   b. 인증서 임포트에서 **확인**을 눌러 파일에서 인증서를 임포트합니다.
   c. 인증서 임포트에서 인증서 파일을 선택하고 **열기**를 누릅니다.
7. 전자 지갑을 편리한 위치(예: *EPM_ORACLE_INSTANCE*`/httpConfig/ohs/config/OHS/ohs_component/keystores/epmsystem`)에 저장합니다.
8. **전자 지갑**, **자동 로그인** 순으로 선택하여 자동 로그인을 활성화합니다.

ORAPKI를 사용한 Oracle Wallet 설정(Linux)

ORAPKI 명령행을 사용하여 Oracle Wallet을 설정하려면 다음 단계를 완료하십시오.

1. 전자 지갑에서 사용할 폴더 생성:

   ```
   $ mkdir /MIDDLEWARE_HOME/oracle_common/wallet
   ```

2. orapki 유틸리티의 위치를 경로에 추가:

   ```
   $ export PATH=$PATH:$MIDDLEWARE_HOME/oracle_common/bin
   ```

3. 인증서를 보관할 전자 지갑 생성:

   ```
   >$ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet create -wallet
   [wallet_location] -auto_login
   ```

이 명령은 명령행에 비밀번호가 지정되지 않은 경우 전자 지갑 비밀번호를 입력한 후 다시 입력하라는 메시지를 표시합니다. -전자 지갑용으로 지정된 위치에 전자 지갑을 생성합니다.

4. 인증서 서명 요청(CSR)을 생성하고 전자 지갑에 추가:

```
$ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet add -wallet
[wallet_location] -dn 'CN=<CommonName>,OU=<OrganizationUnit>,
O=<Company>, L=<Location>, ST=<State>, C=<Country>' -keysize 512|
1024|2048|4096 -pwd [Wallet_Password]
```

5. 신뢰할 수 있는 키 저장소에 루트 및 임시 인증서 추가

```
$ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet add -wallet
[wallet_location] -trusted_cert -cert [certificate_location] [-pwd]
```

6. CA(인증 기관)를 사용하여 CSR(인증서 서명 요청)에 서명합니다. Oracle Wallet에서 인증서 요청을 익스포트하려면:

```
$ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet export -wallet
[wallet_location] -dn 'CN=<CommonName>,OU=<OrganizationUnit>,
O=<Company>, L=<Location>, ST=<State>, C=<Country>' -request
[certificate_request_filename] [-pwd]
```

7. 서명된 CSR을 전자 지갑으로 임포트:

```
    $ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet add -wallet
[wallet_location] -user_cert -cert [certificate_location] [-pwd]
```

8. 전자 지갑의 내용을 표시하려면:

```
$ MIDDLEWARE_HOME/oracle_common/bin/orapki wallet display -wallet
[wallet_location] [-pwd]
```

Oracle HTTP Server의 SSL을 사용으로 설정

Oracle HTTP Server를 호스트하는 각 머신에서 웹 서버를 재구성한 후 기본 전자 지갑의 위치를 생성한 전자 지갑의 위치로 바꾸어 Oracle HTTP Server 구성 파일을 업데이트하십시오.

SSL에 대해 Oracle HTTP Server를 구성하려면 다음을 수행합니다.

1. 배포에 있는 각 Oracle HTTP Server 호스트 머신에서 웹 서버를 재구성합니다.
2. 인스턴스의 EPM System Configurator를 시작합니다.
3. 구성 태스크 선택 화면에서 다음 단계를 완료하고 **다음**을 누릅니다.
   a. **모두 선택 취소**의 선택 항목을 지웁니다.
   b. **Hyperion Foundation** 태스크 그룹을 확장하고 **웹 서버 구성**을 선택합니다.
4. **웹 서버 구성**에서 **다음**을 누릅니다.
5. **확인**에서 **다음**을 누릅니다.
6. **요약**에서 **완료**를 누릅니다.

7. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*/httpConfig/ohs/config/fmwconfig/components/OHS/ohs_component/ssl.conf를 엽니다.
8. 사용 중인 SSL 포트가 다음과 유사하게 OHS Listen port 아래에 나열되어 있는지 확인합니다.

   19443을 SSL 통신 포트로 사용하고 있는 경우 입력이 다음과 같아야 합니다.

   ```
   Listen 19443
   ```

9. SSLSessionCache 매개변수 값을 none으로 설정합니다.
10. 배포에 있는 각 Oracle HTTP Server의 구성 설정을 업데이트합니다.
    a. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*/httpConfig//ohs/config/fmwconfig/components/OHS/ohs_component/ssl.conf를 엽니다.
    b. SSLWallet 지시어를 찾아 값이 인증서를 설치한 전자 지갑을 가리키도록 해당 값을 변경합니다. *EPM_ORACLE_INSTANCE*httpConfig/ohs/config/OHS/ohs_component/keystores/epmsystem에서 전자 지갑을 생성한 경우 SSLWallet 지시어는 다음과 같을 수 있습니다.

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

    c. ssl.conf를 저장한 후 닫습니다.
11. 배포에 있는 각 Oracle HTTP Server에서 mod_wl_ohs.conf를 업데이트합니다.
    a. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*/httpConfig//ohs/config/fmwconfig/components/OHS/ohs_component/mod_wl_ohs.conf를 엽니다.
    b. WLSSLWallet 지시어가 SSL 인증서가 저장된 Oracle Wallet을 가리키는지 확인합니다.

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

       예: C:/Oracle/Middleware/ohs/bin/wallets/myWallet
    c. SecureProxy 지시어의 값을 ON으로 설정합니다.

       ```
       SecureProxy ON
       ```

    d. 배포된 Oracle Enterprise Performance Management System 구성요소의 LocationMatch 정의가 다음 Oracle Hyperion Shared Services 예와 유사한지 확인합니다. 이 예에서는 하나의 Oracle WebLogic Server 클러스터가 있다고 가정합니다(myserver1 및 myserver2에서 SSL 포트 28443 사용).

       ```
       <LocationMatch /interop/>
           SetHandler weblogic-handler
           pathTrim /
           WeblogicCluster myServer1:28443,myServer2:28443
           WLProxySSL ON
       </LocationMatch>
       ```

    e. mod_wl_ohs.conf를 저장하고 닫습니다.

WebLogic Server에 배포된 EPM System 웹 구성요소 구성

Oracle Enterprise Performance Management System 웹 구성요소를 배포한 후에는 SSL 통신에 대해 구성해야 합니다.

웹 구성요소를 SSL에 대해 구성하려면 다음을 수행합니다.

1. *EPM_ORACLE_INSTANCE*/domains/EPMSystem/bin/startWebLogic.cmd에 저장된 파일을 실행하여 Oracle WebLogic Server를 시작합니다.
2. 다음 URL에 액세스하여 WebLogic Server 관리 콘솔을 실행합니다.

   http://*SERVER_NAME:Port*/console

   예를 들어 myServer의 기본 포트에 배포된 WebLogic Server 콘솔에 액세스하려면 http://myServer:7001/console을 사용해야 합니다.
3. 시작 화면에서 사용자 이름과 비밀번호를 입력하여 EPMSystem에 액세스합니다. 사용자 이름과 비밀번호는 구성 프로세스 중 EPM System Configurator에서 지정됩니다.
4. **변경 센터**에서 **잠금 및 편집**을 누릅니다.
5. 콘솔의 왼쪽 창에서 **환경**을 확장하고 **서버**를 선택합니다.
6. 서버 요약 화면에서 SSL을 사용으로 설정할 서버의 이름을 누릅니다.

   예를 들어 Oracle Hyperion Foundation Services 구성요소를 모두 설치한 경우 다음과 같은 서버에서 SSL을 사용으로 설정할 수 있습니다.

   - CalcManager
   - FoundationServices
7. **수신 포트 사용**을 지워 HTTP 수신 포트를 사용 안함으로 설정합니다.
8. **SSL 수신 포트 사용**이 선택되어 있는지 확인합니다.
9. **SSL 수신 포트**에서 WebLogic Server SSL 수신 포트를 입력합니다.
10. 사용할 ID 및 인증 키 저장소를 지정합니다.
    - **키 저장소**를 선택하여 [키 저장소] 탭을 엽니다.
    - **키 저장소**에서 다음과 같이 옵션을 선택합니다.

    a. **키 저장소**를 선택하여 [키 저장소] 탭을 엽니다.

    b. **키 저장소**에서 다음과 같이 옵션을 선택합니다.
       - **사용자정의 ID 및 사용자정의 인증** - 잘 알려진 타사 CA의 서버 인증서를 사용하고 있지 않은 경우
       - **사용자정의 ID 및 Java Standard Trust** - 잘 알려진 타사 CA의 서버 인증서를 사용하고 있는 경우

    c. **사용자정의 ID 키 저장소**에서 서명된 WebLogic Server 인증서가 설치된 키 저장소의 경로를 입력합니다.

    d. **사용자정의 ID 키 저장소 유형**에서 jks를 입력합니다.

    e. **사용자정의 ID 키 저장소 문장암호** 및 **사용자정의 ID 키 저장소 문장암호 확인**에 키 저장소 비밀번호를 입력합니다.

f. **키 저장소**에서 **사용자정의 ID 및 사용자정의 인증**을 선택한 경우 다음을 수행합니다.
   - **사용자정의 인증 키 저장소**에서 서버 인증서에 서명한 CA의 루트 인증서를 사용할 수 있는 사용자정의 키 저장소 경로를 입력합니다.
   - **사용자정의 인증 키 저장소 유형**에서 `jks`를 입력합니다.
   - **사용자정의 트러스트 키 저장소 문장암호** 및 **사용자정의 트러스트 키 저장소 문장암호 확인**에 키 저장소 비밀번호를 입력합니다.

g. **저장**을 누릅니다.

11. SSL 설정을 지정합니다.
    - **SSL**을 선택합니다.
    - **개인 키 별칭**에서 서명된 WebLogic Server 인증서를 임포트하는 중 지정한 별칭을 입력합니다.
    - **개인 키 문장암호** 및 **개인 키 문장암호 확인**에서 개인 키를 검색하는 데 사용할 비밀번호를 입력합니다.
    - **Oracle Hyperion Provider Services 웹 애플리케이션에만 해당:** 와일드카드 인증서를 사용하여 WebLogic Server 및 다른 EPM System 서버 구성요소 간에 통신을 암호화하는 경우 Provider Services 웹 애플리케이션의 호스트 이름 확인을 사용 안함으로 설정합니다.
      – **고급**을 선택합니다.
      – **호스트 이름 확인**에서 **없음**을 선택합니다.
    - **저장**을 누릅니다.
12. **변경 센터**에서 **변경 활성화**를 누릅니다.

도메인 구성 업데이트

이 프로세스는 도메인 구성을 업데이트합니다. 이 절차를 시작하기 전에 배포의 전체 백업을 생성합니다. 프로덕션 배포를 변경하기 전에 테스트 배포에서 이 절차를 테스트하는 것이 좋습니다.

도메인 구성을 업데이트하려면:

1. `MIDDLEWARE_HOME/oracle_common/bin directory` 디렉토리로 이동합니다.
   `cd MIDDLEWARE_HOME/oracle_common/bin`
2. *ORACLE_HOME*, *WL_HOME*, *JAVA_HOME*을 설정합니다.
   `set ORACLE_HOME= /Oracle/Middleware`

   `set WL_HOME= /Oracle/Middleware/wlserver`

   `set JAVA_HOME= /Oracle/Middleware/jdk`
3. Web Logic 콘솔에서 관리 서버의 HTTP 포트를 사용으로 설정합니다.
4. 다음과 같은 명령을 사용하여 키 저장소를 생성합니다.
   `libovdconfig.bat -host HOSTNAME -port 7001 -userName USERNAME -domainPath %MWH%\user_projects\domains\EPMSystem -createKeystore`

   이 명령에서 *HOSTNAME* 및 *USERNAME*을 각각 Web Logic 서버의 호스트 이름 및 관리자의 사용자 이름으로 바꿉니다. 출력에 OVD 키 저장소의 성공적인 생성이 보고되는지 확인합니다.
5. AdminServer에서 SSL 인증서를 익스포트합니다.

> **Note:**
>
> 이 단계는 삽입된 LDAP(기본 인증자)에만 적용할 수 있습니다. 기타 LDAP의 경우 적절한 LDAP 특정 명령을 사용하여 인증서를 익스포트해야 합니다. 인증서 파일 형식은 **Base 64 인코딩 x.509**여야 합니다.

a. Internet Explorer를 통해 `https://HOSTNAME:7002/console`에 연결하여 Web Logic 관리 콘솔에 액세스합니다.

b. **인증서 보기**, **세부정보** 순으로 누르고 **파일로 복사**를 선택하여 SSL 인증서를 익스포트합니다.

c. 인증서를 **Base 64 인코딩 x.509** 인증서 파일로 로컬 디렉토리에 저장합니다(예: `C:\certificate\slc17rby.cer`).

d. 인증서를 서버로 이동합니다.

6. keytool을 사용하여 인증서를 4단계에서 생성한 키 저장소로 임포트합니다. 다음과 같은 명령을 사용합니다(*JAVA_HOME* 및 keytool 실행 파일이 경로에 있다고 가정).

```
export PATH=$JAVA_HOME/bin:$PATH
```

```
keytool -importcert -keystore
DOMAIN_HOME\config\fmwconfig\ovd\default\keystores/adapters.jks -
storepass PASSWORD -alias wcp_ssl -file CERTIFICATE_PATH -noprompt
```

, 예:

```
keytool -importcert -keystore %MWH%
\user_projects\domains\EPMSystem\config\fmwconfig\ovd\default\keystores/
adapters.jks -storepass examplePWD -alias wcp_ssl -file
C:\certificate\slc17rby.cer -noprompt
```

> **Note:**
>
> - 이 명령에 사용되는 비밀번호는 4단계에서 키 저장소를 생성하는 동안 사용된 비밀번호와 일치해야 합니다.
> - *CERTIFICATE_PATH*는 인증서의 위치 및 이름입니다.
> - alias는 원하는 별칭으로 선택할 수 있습니다.

인증서를 성공적으로 임포트하면 keytool에 `Certificate was added to keystore`(인증서가 키 저장소에 추가됨) 메시지가 표시됩니다.

7. Web Logic 콘솔에서 HTTP 포트 외에 관리 서버의 SSL 포트를 사용으로 설정합니다.
8. Weblogic 관리 서버(Admin Server) 및 관리 서버(Managed Server)를 재시작합니다.
9. 보안 연결을 통해 Oracle Hyperion Enterprise Performance Management Workspace에 로그인하여 모든 것이 작동하는지 확인합니다.

서버 및 EPM System 재시작

배포의 모든 서버를 재시작한 후 각 서버의 Oracle Enterprise Performance Management System을 시작하십시오.

배포 테스트

SSL 배포를 완료한 후 모든 요소가 작동하는지 확인하십시오.

배포를 테스트하려면 다음을 수행합니다.

1. 브라우저를 사용하여 다음과 같이 보안 Oracle Hyperion Enterprise Performance Management Workspace URL에 액세스합니다.

   `epm.myCompany.com`을 외부 통신을 위한 서버 별칭으로 사용하고 `4443`을 SSL 포트로 사용하는 경우 EPM Workspace URL은 다음과 같습니다.

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

2. [로그온] 화면에서 사용자 이름과 비밀번호를 입력합니다.
3. **로그온**을 누릅니다.
4. 배포된 Oracle Enterprise Performance Management System 구성요소에 안전하게 액세스할 수 있는지 확인합니다.

SSL 사용 외부 사용자 디렉토리 구성

가정

- Oracle Hyperion Shared Services Console에서 구성하려는 외부 사용자 디렉토리는 SSL이 사용으로 설정되어 있습니다.
- 잘 알려진 타사 CA의 인증서를 사용하여 사용자 디렉토리에서 SSL을 사용으로 설정하지 않은 경우에는 서버 인증서에 서명한 CA의 루트 인증서 사본이 있습니다.

루트 CA 인증서 임포트

잘 알려진 타사 CA의 인증서를 사용하여 사용자 디렉토리에서 SSL을 사용으로 설정하지 않은 경우에는 서버 인증서에 서명한 CA의 루트 인증서를 다음 키 저장소로 임포트해야 합니다.

> **주:**
>
> WebLogic은 애플리케이션 배포 중 `setDomainEnv.sh` 또는 `setDomainEnv.cmd`의 `DemoTrust.jks`를 가리키는 `-Djavax.net.ssl.trustStore` 지시어를 추가합니다. 기본 WebLogic 인증서를 사용하지 않는 경우 `setDomainEnv.sh` 또는 `setDomainEnv.cmd`에서 `-Djavax.net.ssl.trustStore`를 제거하십시오.

keytool과 같은 툴을 사용하여 루트 CA 인증서를 임포트하십시오.

- 모든 Oracle Enterprise Performance Management System 서버.

  JVM 키 저장소: *`MIDDLEWARE_HOME`*`/jdk/jre/lib/security/cacerts`

- 각 EPM System 구성요소 호스트 머신의 JVM에서 사용하는 키 저장소. 기본적으로 EPM System 구성요소는 다음 키 저장소를 사용합니다.

  `MIDDLEWARE_HOME``/jdk/jre/lib/security/cacerts`

외부 사용자 디렉토리 구성

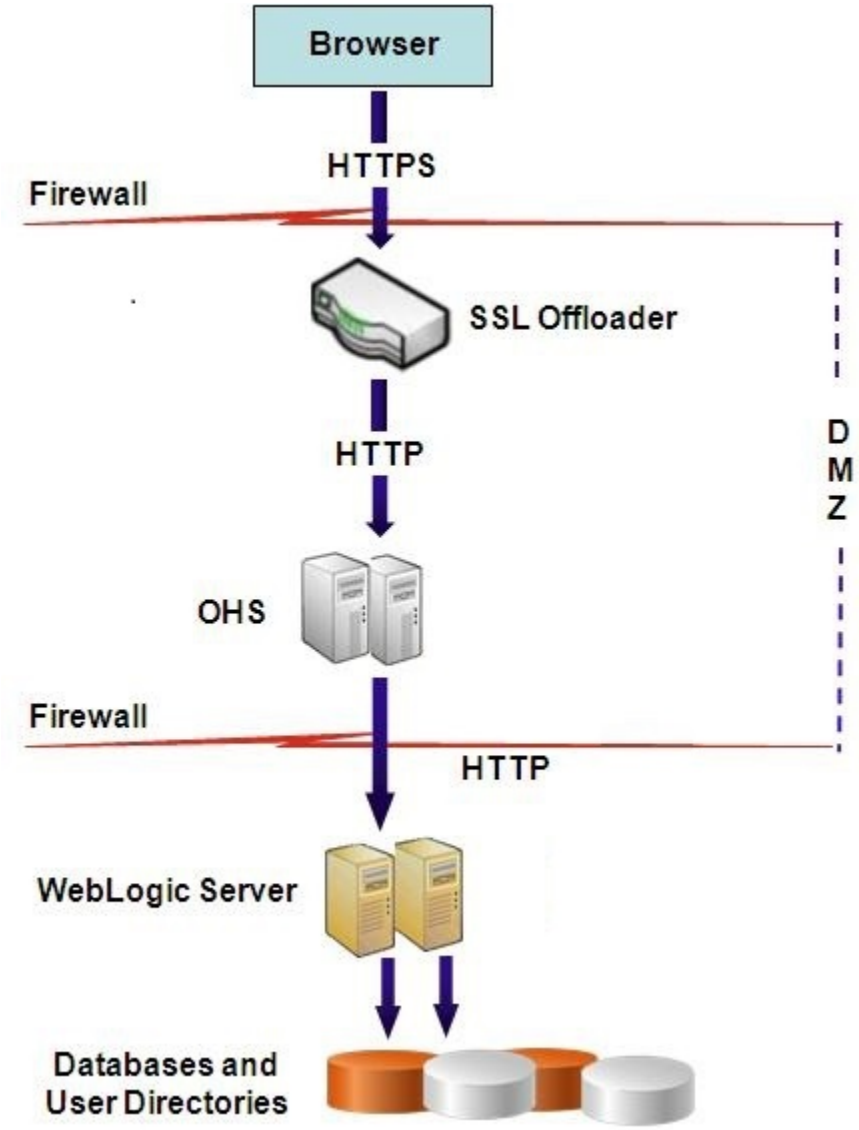
Shared Services Console을 사용하여 사용자 디렉토리를 구성합니다. 사용자 디렉토리를 구성하는 동안 EPM System 보안에서 보안 프로토콜을 사용하여 사용자 디렉토리와 통신하도록 지시하는 `SSL Enabled` 옵션을 선택해야 합니다. EPM System 보안과 LDAP 사용 사용자 디렉토리 간 연결에 대해 SSL을 사용으로 설정할 수 있습니다(예: Oracle Internet Directory 및 Microsoft Active Directory).

*Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자 디렉토리 구성"을 참조하십시오.

웹 서버에서 SSL 종료

배포 아키텍처

이 시나리오에서는 SSL을 사용하여 Oracle Enterprise Performance Management System 클라이언트(예: 브라우저)와 Oracle HTTP Server 간의 통신 링크를 보호합니다. 그림으로 된 개념은 다음과 같습니다.



가정

이 구성은 웹 서버에서 `epm.myCompany.com` 및 `empinternal.myCompany.com`와 같은 두 개의 서버 별칭을 사용합니다. 하나는 웹 서버와 브라우저 간의 외부 통신용이며 다른 하나는 EPM System 서버의 내부 통신용입니다. 서버 별칭은 머신의 IP 주소를 가리키고 DNS를 통해 확인할 수 있어야 합니다.

브라우저와의 외부 통신을 지원하는(예를 들어 `epm.myCompany.com` 사용) 서명된 인증서는 보안 외부 통신을 지원하는 가상 호스트가 정의된 웹 서버에 설치되어야 합니다. 이 가상 호스트는 SSL을 종료하고 HTTP 요청을 Oracle HTTP Server에 전달해야 합니다.

SSL이 OHS(Oracle HTTP Server) 또는 로드 밸런서에서 종료되는 경우 다음을 수행해야 합니다.

- 모든 논리 웹 애플리케이션을 로드 밸런서 또는 Oracle HTTP Server의 비SSL 가상 호스트로 설정해야 합니다(예: `empinternal.myCompany.com:80`, 여기서 80은 비SSL 포트임). 구성 화면을 열고 다음 단계를 완료합니다.
  1. **Hyperion Foundation** 구성 태스크를 확장합니다.

2. **웹 애플리케이션의 논리 주소 구성**을 선택합니다.
3. *호스트 이름*, 비SSL 포트 번호 및 SSL 포트 번호를 지정합니다.

- 로드 밸런서 또는 Oracle HTTP Server의 SSL 지원 가상 호스트에 대한 외부 URL을 설정합니다(예: `empexternal.myCompany.com:443`, 여기서 443은 SSL 포트임). 구성 화면을 열고 다음 단계를 완료합니다.
  1. **Hyperion Foundation** 구성 태스크를 확장합니다.
  2. **공통 설정 구성**을 선택합니다.
  3. 외부 URL 세부정보에서 **SSL 오프로딩 사용**을 선택합니다.
  4. *외부 URL 호스트* 및 *외부 URL 포트*를 지정합니다.

> **주:**
>
> **configtool**을 사용하여 웹 애플리케이션을 재배포하거나 웹 서버를 재구성하면 논리 웹 애플리케이션 및 외부 URL에 대한 설정이 대체됩니다.

EPM System 구성

EPM System 구성요소의 기본 배포는 웹 서버에서 SSL 종료를 지원합니다. 추가 작업이 필요하지 않습니다.

EPM System을 구성할 때 논리적 웹 애플리케이션이 내부 통신용으로 생성된 가상 호스트(예: `empinternal.myCompany.com`)를 가리키는지 확인하십시오. EPM System을 설치 및 구성하려면 다음 정보 소스를 참조하십시오.

- *Oracle Enterprise Performance Management System 설치 및 구성 가이드*
- *Oracle Enterprise Performance Management System 설치 시작 페이지*

배포 테스트

배포 프로세스를 완료한 후 보안 Oracle Hyperion Enterprise Performance Management Workspace URL에 연결하여 모든 요소가 작동하는지 확인합니다.

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

`https://epm.myCompany.com:443/workspace/index.jsp`를 예로 들 수 있습니다. 여기서, 443은 SSL 포트입니다.

Essbase 11.1.2.4에 대한 SSL

개요

이 섹션에는 MaxL, Oracle Essbase Administration Services 서버, Oracle Essbase Studio 서버, Oracle Hyperion Provider Services, Oracle Hyperion Foundation Services, Oracle Hyperion Planning, Oracle Hyperion Financial Management, Oracle Hyperion Shared Services Registry와 같은 Oracle Essbase 인스턴스 및 구성요소 간에 통신을 보호하는 데 사용되는 기본 인증서를 바꾸는 절차가 설명되어 있습니다.

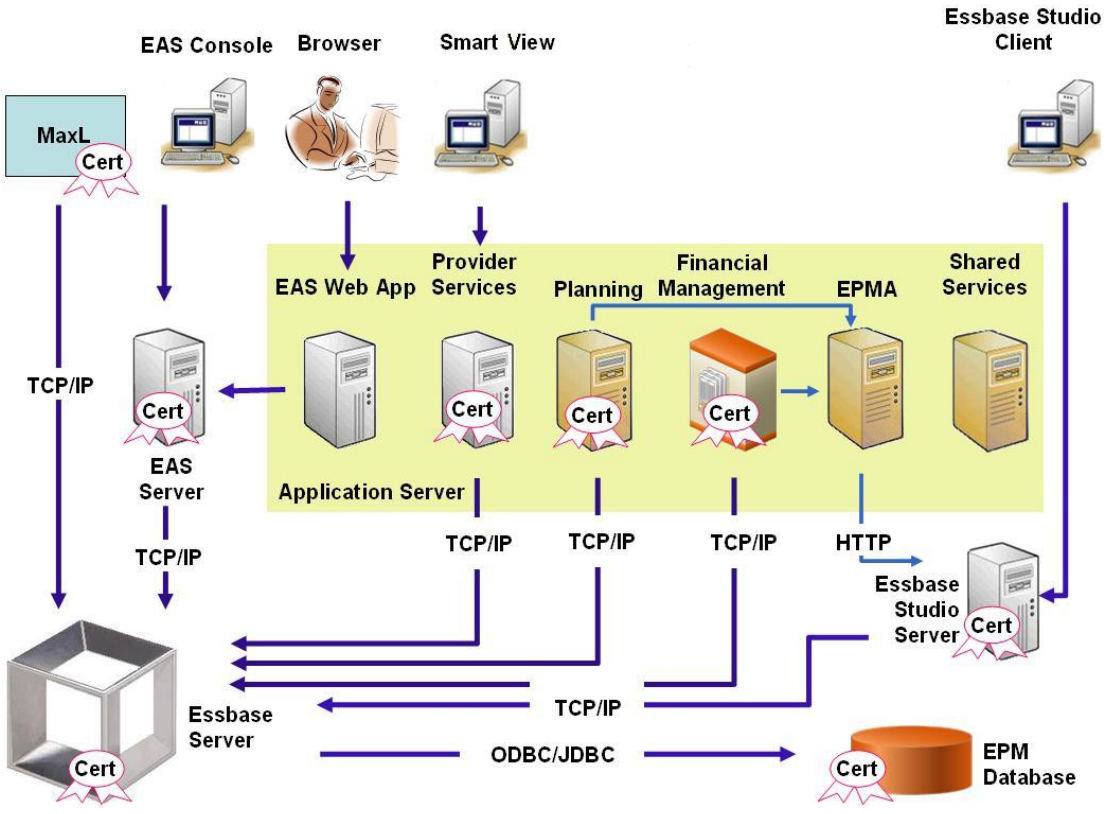
기본 배포

Essbase는 SSL 및 비SSL 모드로 사용하도록 배포할 수 있습니다. Essbase 에이전트는 비보안 포트에서 수신합니다. 보안 포트에서 수신하도록 구성할 수도 있습니다. 보안 포트에 액세스하는 모든 연결은 SSL 연결로 처리됩니다. 클라이언트가 비SSL 포트의 Essbase 에이전트에 연결하는 경우 연결은 비SSL 연결로 처리됩니다. 구성요소는 하나의 Essbase 에이전트에 대해 비SSL 연결 및 SSL 연결을 동시에 설정할 수 있습니다.

로그인할 때 보안 프로토콜 및 포트를 지정하여 세션별로 SSL을 제어할 수 있습니다. 세션당 SSL 연결 설정을 참조하십시오.

SSL이 사용으로 설정된 경우 데이터 보안이 보장되도록 Essbase 인스턴스 내의 모든 통신이 암호화됩니다.

보안 모드의 기본 Essbase 구성요소 배포에서는 자체 서명 인증서를 사용하여 주로 테스트용으로 SSL 통신을 사용으로 설정합니다. 프로덕션 환경에서는 잘 알려진 타사 CA의 인증서를 사용하여 Essbase의 SSL을 사용으로 설정하는 것이 좋습니다.



일반적으로 Oracle Wallet은 인증서를 저장하여 Essbase RTC를 사용하는 클라이언트와의 SSL 통신을 사용으로 설정하고, Java 키 저장소는 인증서를 저장하여 통신에 JAPI를 활용하는 구성요소와의 SSL 통신을 사용으로 설정합니다. SSL 통신을 설정하기 위해 Essbase 클라이언트 및 툴은 Essbase 서버 및 에이전트 인증서에 서명한 CA의 루트 인증서를 저장합니다. 필요한 인증서 및 해당 위치를 참조하십시오.

필요한 인증서 및 해당 위치

프로덕션 환경에서는 잘 알려진 타사 CA의 인증서를 사용하여 Essbase의 SSL을 사용으로 설정하는 것이 좋습니다. 테스트용으로 기본 자체 서명 인증서를 사용할 수 있습니다.

> **주:**
>
> Essbase는 하나의 SSL 인증서로 여러 하위 도메인을 보호할 수 있는 와일드카드 인증서 사용을 지원합니다. 와일드카드 인증서를 사용하면 관리 시간 및 비용을 줄일 수 있습니다.
>
> 호스트 이름 확인이 사용으로 설정된 경우 와일드카드 인증서를 사용할 수 없습니다.

다음 인증서가 필요합니다.

- 루트 CA 인증서
  Essbase RTC를 사용하여 Essbase 연결을 설정하는 구성요소에서는 루트 CA 인증서가 Oracle Wallet에 저장되어야 합니다. JAPI를 사용하여 연결을 설정하는 구성요소에서는 루트 CA 인증서가 Java 키 저장소에 저장되어야 합니다. 필요한 인증서 및 해당 위치는 다음 테이블에 표시되어 있습니다.

  > **주:**
  >
  > 해당 루트 인증서가 Oracle Wallet에 이미 설치되어 있는 잘 알려진 타사 CA의 인증서를 사용하는 경우 루트 CA 인증서를 설치할 필요가 없습니다.

- Essbase 서버 및 Essbase 에이전트의 서명된 인증서

표 2-1 필요한 인증서 및 해당 위치

구성요소[1]	키 저장소	인증서[2]
MaxL	Oracle Wallet	루트 CA 인증서
Administration Services 서버	Oracle Wallet	루트 CA 인증서
Provider Services	Oracle Wallet	루트 CA 인증서
Oracle Enterprise Performance Management System 데이터베이스	Oracle Wallet	루트 CA 인증서
Essbase Studio 서버	Java 키 저장소	루트 CA 인증서
Planning	• Oracle Wallet • Java 키 저장소	루트 CA 인증서
Financial Management	Java 키 저장소	루트 CA 인증서
Essbase(서버 및 에이전트)[3]	• Oracle Wallet • Java 키 저장소	• 루트 CA 인증서 • Essbase 서버 및 에이전트의 서명된 인증서
Oracle Hyperion Shared Services 저장소		

[1] 유사한 키 저장소를 사용하는 여러 구성요소를 지원하는 데는 하나의 키 저장소 인스턴스만 있으면 됩니다.

[2] 여러 구성요소가 하나의 키 저장소에 설치된 하나의 루트 인증서를 사용할 수 있습니다.

[3] 인증서는 기본 Oracle Wallet 및 Java 키 저장소에 설치되어야 합니다.

Essbase 구성요소 설치 및 배포

구성 프로세스를 사용하면 보안 에이전트 포트(기본값은 6423)를 선택할 수 있습니다. 이 포트는 Oracle Essbase를 구성할 때 변경할 수 있습니다. 기본적으로 배포 프로세스는 필수 자체 서명 인증서를 설치하여 테스트를 위한 기능 보안 배포를 생성합니다.

EPM System Installer는 Oracle HTTP Server가 설치된 경우 Essbase 인스턴스를 호스트하는 머신의 *ARBOR_PATH* 내에 Oracle Wallet 및 자체 서명 인증서를 설치합니다. 단일 호스트 배포에서는 모든 Essbase 구성요소가 이 인증서를 공유합니다.

Essbase에 인증된 타사 CA 인증서 사용

인증서 요청 생성 및 인증서 가져오기

인증서 요청을 생성하여 Oracle Essbase 서버 및 Essbase 에이전트를 호스트하는 서버에 대한 인증서를 가져오십시오. 인증서 요청에는 DN(고유 이름)과 관련된 암호화된 정보가 포함되어 있습니다. 인증서 요청을 서명 기관에 제출하여 SSL 인증서를 가져옵니다.

keytool 또는 Oracle Wallet Manager와 같은 툴을 사용하여 인증서 요청을 생성합니다. 인증서 요청 생성에 대한 자세한 내용은 사용 중인 툴의 설명서를 참조하십시오.

keytool을 사용하는 경우 다음과 같은 명령을 사용하여 인증서 요청을 생성하십시오.

```
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
```

루트 CA 인증서 가져오기 및 설치

루트 CA 인증서는 SSL을 지원하는 데 사용되는 인증서의 유효성을 확인합니다. 인증서에 서명하는 데 사용된 개인 키와 일치시켜 인증을 확인하는 공개 키가 포함되어 있습니다. SSL 인증서에 서명한 인증 기관의 루트 CA 인증서를 가져올 수 있습니다.

Essbase 서버 또는 에이전트와 연결된 클라이언트에 Essbase 서버 인증서에 서명한 CA의 루트 인증서를 설치합니다. 클라이언트에 적합한 키 저장소에 루트 인증서가 설치되어 있는지 확인하십시오. 필요한 인증서 및 해당 위치를 참조하십시오.

> **주:**
>
> 여러 구성요소가 하나의 서버 머신에 설치된 하나의 루트 CA 인증서를 사용할 수 있습니다.

Oracle Wallet

Oracle Wallet에 CA 루트 인증서가 있어야 하는 구성요소의 목록은 필요한 인증서 및 해당 위치을 참조하십시오. 전자 지갑을 생성하거나 기본 자체 서명 인증서가 설치된 데모 전자 지갑에 인증서를 설치할 수 있습니다.

전자 지갑을 생성하고 루트 CA 인증서를 임포트하는 자세한 절차는 Oracle Wallet Manager 설명서를 참조하십시오.

Java 키 저장소

Java 키 저장소에 루트 CA 인증서가 있어야 하는 구성요소의 목록은 필요한 인증서 및 해당 위치을 참조하십시오. 기본 자체 서명 인증서가 설치된 키 저장소에 인증서를 추가하거나 인증서를 저장할 키 저장소를 생성할 수 있습니다.

> **주:**
>
> 많은 잘 알려진 타사 CA의 루트 CA 인증서가 이미 Java 키 저장소에 설치되어 있습니다.

자세한 지침은 사용 중인 툴의 설명서를 참조하십시오. keytool을 사용하는 경우 다음과 같은 명령을 사용하여 루트 인증서를 임포트하십시오.

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

서명된 인증서 설치

Essbase 서버 및 Essbase 에이전트를 호스트하는 서버에 서명된 SSL 인증서를 설치합니다. Essbase RTC(C API)를 사용하여 Essbase 서버 또는 에이전트 연결을 설정하는 구성요소에서는 루트 CA 인증서가 있는 Oracle Wallet에 인증서가 저장되어야 합니다. JAPI를 사용하여 Essbase 서버 또는 에이전트 연결을 설정하는 구성요소에서는 루트 CA 인증서와 서명된 SSL 인증서가 Java 키 저장소에 저장되어야 합니다. 자세한 절차는 다음 정보 소스를 참조하십시오.

- Oracle Wallet Manager 설명서
- 인증서를 임포트하는 데 사용하는 툴(예: keytool)의 설명서 또는 온라인 도움말

keytool을 사용하는 경우 다음과 같은 명령을 사용하여 인증서를 임포트하십시오.

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

Essbase 서버 레지스트리 값 업데이트

Windows

1. 명령 프롬프트에서 디렉토리를 *EPM_ORACLE_INSTANCE*/epmsystem1/bin으로 변경합니다.
2. 다음 명령을 실행하여 Windows 레지스트리를 업데이트합니다.

   ```
   epmsys_registry.bat updateproperty "#<Object ID>/@EnableSecureMode"
   true

   epmsys_registry.bat updateproperty "#<Object ID>/@EnableClearMode"
   false
   ```

<Object ID>를 Essbase 서버 구성 프로세스가 완료된 후 생성되는 레지스트리 보고서에서 사용할 수 있는 Essbase 서버 구성요소 ID로 바꿔야 합니다.

Linux

1. 콘솔에서 디렉토리를 *EPM_ORACLE_INSTANCE*/epmsystem1/bin으로 변경합니다.
2. 다음 명령을 실행하여 레지스트리를 업데이트합니다.
   ```
   epmsys_registry.sh updateproperty "#<Object ID>/@EnableSecureMode" true

   epmsys_registry.sh updateproperty "#<Object ID>/@EnableClearMode" false
   ```
   <Object ID>를 Essbase 서버 구성 프로세스가 완료된 후 생성되는 레지스트리 보고서에서 사용할 수 있는 Essbase 서버 구성요소 ID로 바꿔야 합니다.

Essbase SSL 설정 업데이트

essbase.cfg에 다음 값을 지정하여 Essbase 서버 및 클라이언트의 SSL 설정을 사용자정의합니다.

- 보안 모드를 사용으로 지정하는 설정
- 지우기 모드를 사용으로 지정하는 설정
- 클라이언트와 통신하는 기본 모드(클라이언트만 사용)
- 보안 포트
- 암호 모음
- Oracle Wallet 경로

> **주:**
>
> essbase.cfg에서 누락된 필수 매개변수(특히, EnableSecureMode 및 AgentSecurePort)를 추가하고 해당 값을 설정해야 합니다.

essbase.cfg를 업데이트하려면 다음을 수행합니다.

1. Essbase 서버 인증서가 있는 Oracle Wallet을 *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/wallet으로 복사합니다.
   Essbase 서버에 허용되는 유일한 Oracle Wallet 위치입니다.
2. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/essbase.cfg를 엽니다.
3. 필요에 따라 설정을 입력합니다. 기본 Essbase 설정은 내재되어 있습니다. 기본 동작을 변경해야 하는 경우 essbase.cfg에서 사용자정의 동작 설정을 추가합니다. 예를 들어 EnableClearMode는 기본적으로 적용되며, 이 설정에 따라 Essbase 서버는 암호화되지 않은 채널을 통해 통신할 수 있습니다. Essbase 서버에서 암호화되지 않은 채널을 통해 통신하는 기능을 끄려면 essbase.cfg에 EnableClearMode FALSE를 지정해야 합니다. 다음 테이블을 참조하십시오.

표 2-2 Essbase SSL 설정

설정	설명 [1]
EnableClearMode[2]	Essbase 애플리케이션과 Essbase 에이전트 간에 암호화되지 않은 통신이 사용됩니다. 이 등록정보가 FALSE로 설정되면 Essbase에서 비SSL 요청을 처리하지 않습니다. **기본값:** EnableClearMode TRUE **예:** EnableClearMode FALSE
EnableSecureMode	Essbase 클라이언트와 Essbase 에이전트 간에 SSL 암호화 통신이 사용됩니다. 이 등록정보를 TRUE로 설정해야 SSL이 지원됩니다. **기본값:** FALSE **예:** EnableSecureMode TRUE
SSLCipherSuites	SSL 통신에 사용할 암호 모음이 환경설정 순서대로 표시된 목록입니다. Essbase 에이전트는 이러한 암호 모음 중 하나를 SSL 통신에 사용합니다. 에이전트가 암호 모음을 선택하는 경우 목록의 첫번째 암호 모음에 가장 높은 우선순위가 부여됩니다. **기본값:** SSL_RSA_WITH_RC4_128_MD5 **예:** SSLCipherSuites SSL_RSA_WITH_AES_256_CBC_SHA256,SSL_RSA_WITH_AES_256_GCM_SHA384
APSRESOLVER	Oracle Hyperion Provider Services URL입니다. 여러 Provider Services 서버를 사용 중인 경우 각 URL을 세미콜론을 사용하여 구분하십시오. **예:** APSRESOLVER https://exampleAPShost1:*PORT*/aps;https://exampleAPShost2:*PORT*/aps
AgentSecurePort	에이전트가 수신하는 보안 포트입니다. **기본값:** 6423 **예:** AgentSecurePort 16001
WalletPath	루트 CA 인증서 및 서명된 인증서를 저장하는 Oracle Wallet(1,024자 미만)의 위치입니다. **기본값:** *ARBORPATH*/bin/wallet **예:** WalletPath/usr/local/wallet
ClientPreferredMode[3]	클라이언트 세션의 모드(보안 또는 지우기)입니다. 이 등록정보를 보안으로 설정하면 SSL 모드가 모든 세션에 사용됩니다. 이 등록정보를 지우기로 설정하면 클라이언트 로그인 요청에 보안 전송 키워드가 포함되어 있는지 여부에 따라 전송이 선택됩니다. 세션당 SSL 연결 설정을 참조하십시오. **기본값:** CLEAR **예:** ClientPreferredMode SECURE

[1] essbase.cfg에서 해당 등록정보를 사용할 수 없는 경우 기본값이 적용됩니다.

[2] EnableClearMode 및 EnableSecureMode 둘 다 FALSE로 설정되는 경우 Essbase가 작동하지 않게 됩니다.

[3] 클라이언트는 이 설정을 사용하여 Essbase와 보안 연결을 설정할 것인지 비보안 연결을 설정할 것인지 결정합니다.

4. essbase.cfg를 저장한 후 닫습니다.

SSL용 분산 Essbase 노드 업데이트

> **주:**
>
> 이 섹션은 Essbase 분산 배포에만 적용됩니다.

루트 CA 인증서 및 서명된 인증서가 포함된 Wallet 폴더(예: `WalletPath/usr/local/wallet`)가 각 분산 노드의 필요한 위치에 있는지 확인하십시오.

1. Wallet 폴더를 각 분산 노드의 다음 위치에 복사합니다.
   - *EPM_ORACLE_HOME*/common/EssbaseRTC/11.1.2.0/bin
   - *EPM_ORACLE_HOME*/common/EssbaseRTC-64/11.1.2.0/bin
2. Wallet 폴더를 각 분산 노드의 다음 위치에 복사합니다(있는 경우).
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseServer/bin
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseServer-32/bin
   - *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin
3. *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/essbase.cfg를 각 분산 노드의 다음 위치에 복사합니다.
   - *EPM_ORACLE_HOME*/common/EssbaseRTC/11.1.2.0/bin
   - *EPM_ORACLE_HOME*/common/EssbaseRTC-64/11.1.2.0/bin
4. *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/essbase.cfg를 각 분산 노드의 다음 위치에 복사합니다(있는 경우).
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseServer/bin
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseServer-32/bin
   - *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin
5. Wallet 폴더를 각 분산 노드의 다음 Essbase 클라이언트 설치 위치에 복사합니다.
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseClient/bin
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseClient-32/bin
6. *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/essbase.cfg를 각 분산 노드의 다음 Essbase 클라이언트 설치 위치에 복사합니다.
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseClient/bin
   - *EPM_ORACLE_HOME*/products/Essbase/EssbaseClient-32/bin
7. 다음 등록정보를 `essbase.properties` 파일에 추가합니다.
   - `essbase.ssleverywhere=true`
   - `olap.server.ssl.alwaysSecure=true`
   - `APSRESOLVER=http[s]://host:httpsPort/aps`
     이 값을 적절한 URL로 바꿔야 합니다.

   각 분산 노드의 다음 위치에 있는 `essbase.properties` 파일을 업데이트해야 합니다(있는 경우).

- *EPM_ORACLE_HOME*/common/EssbaseJavaAPI/11.2.0/bin/essbase.properties
- *EPM_ORACLE_HOME*/products/Essbase/aps/bin/essbase.properties
- *EPM_ORACLE_INSTANCE*/aps/bin/essbase.properties

8. *EPM_ORACLE_HOME*/products/Essbase/aps/bin/essbase.properties를 각 분산 노드의 *EPM_ORACLE_HOME*/products/Essbase/eas 디렉토리에 복사합니다 (사용가능한 경우).
9. **Oracle Hyperion Planning만 해당:** 다음 세 개의 등록정보를 essbase.properties 파일에 추가합니다.
   - essbase.ssleverywhere=true
   - olap.server.ssl.alwaysSecure=true
   - APSRESOLVER=*APS_URL*
     *APS_URL*을 Provider Services URL로 바꾸십시오. 여러 Provider Services 서버를 사용 중인 경우 각 URL을 세미콜론을 사용하여 구분하십시오. 예를 들어 https://exampleAPShost1:*PORT*/aps;https://exampleAPShost2:*PORT*/aps와 같이 구분합니다.

     각 분산 노드의 다음 위치에 있는 essbase.properties 파일을 업데이트해야 합니다.
     - *EPM_ORACLE_HOME*/products/Planning/config/essbase.properties
     - *EPM_ORACLE_HOME*/products/Planning/lib/essbase.properties
10. **Oracle Hyperion Financial Reporting만 해당:** 다음 세 개의 등록정보를 *EPM_ORACLE_HOME*/products/financialreporting/bin/EssbaseJAPI/bin/essbase.properties 파일에 추가합니다.
    - essbase.ssleverywhere=true
    - olap.server.ssl.alwaysSecure=true
    - APSRESOLVER=*APS_URL*
      *APS_URL*을 Provider Services URL로 바꾸십시오. 여러 Provider Services 서버를 사용 중인 경우 각 URL을 세미콜론을 사용하여 구분하십시오. 예를 들어 https://exampleAPShost1:*PORT*/aps;https://exampleAPShost2:*PORT*/aps와 같이 구분합니다.

> **주:**
>
> 전체 SSL 환경에서 Financial Reporting에 연결하려면 Essbase 클러스터 이름이 필요합니다. 호스트 이름을 사용하여 연결하면 연결이 실패하게 됩니다.

11. a. 다음과 같이 환경 변수를 설정합니다.
    - **Windows:** 새 시스템 변수 API_DISABLE_PEER_VERIFICATION을 생성하고 값을 1로 설정합니다.
    - **Linux:** setCustomParamsPlanning.sh에서 API_DISABLE_PEER_VERIFICATION=1 지시어를 추가합니다.

    b. API_DISABLE_PEER_VERIFICATION=1 in *EPM_ORACLE_INSTANCE*/EssbaseServer/essbaseserver1/bin/setEssbaseenv.bat 또는

EPM_ORACLE_INSTANCE/EssbaseServer/essbaseserver1/bin/setEssbaseenv.sh 지시어를 추가합니다.

환경 변수를 설정합니다.

JAPI 클라이언트의 SSL 등록정보 사용자정의

JAPI를 사용하는 Essbase 구성요소의 경우 여러 기본 등록정보가 사전 정의되어 있습니다. essbase.properties에 등록정보를 포함하여 기본 등록정보를 대체할 수 있습니다.

> **주:**
>
> 다음 테이블에서 표시된 SSL 등록정보 중 일부만 essbase.properties에서 외부화됩니다. 외부화되지 않는 등록정보를 추가해야 합니다.

JAPI 클라이언트의 SSL 등록정보를 업데이트하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *EPM_ORACLE_HOME*/common/EssbaseJavaAPI/11.1.2.0/bin/essbase.properties를 엽니다.
2. 필요에 따라 등록정보를 업데이트합니다. 사용자정의 가능한 JAPI 클라이언트 등록정보에 대한 설명은 다음 테이블을 참조하십시오.
   원하는 등록정보가 essbase.properties에 포함되어 있지 않으면 해당 등록정보를 추가하십시오.

표 2-3 JAPI 클라이언트의 기본 SSL 등록정보

등록정보	설명
olap.server.ssl.alwaysSecure	클라이언트가 모든 Essbase 인스턴스에 대해 사용해야 하는 모드를 설정합니다. SSL 모드를 적용하려면 이 등록정보 값을 true로 변경하십시오. **기본값:** false
olap.server.ssl.securityHandler	프로토콜을 처리하기 위한 패키지 이름입니다. 다른 처리기를 표시하도록 이 값을 변경할 수 있습니다. **기본값:** java.protocol.handler.pkgs
olap.server.ssl.securityProvider	Oracle에서는 Sun SSL 프로토콜 구현을 사용합니다. 다른 제공자를 표시하도록 이 값을 변경할 수 있습니다. **기본값:** com.sun.net.ssl.internal.www.protocol
olap.server.ssl.supportedCiphers	보안 통신을 위해 사용으로 설정할 추가 암호의 쉼표로 구분된 목록입니다. Essbase가 지원하는 암호만 지정해야 합니다. **예:** SSL_RSA_WITH_AES_256_CBC_SHA256,SSL_RSA_WITH_AES_2

표 2-3 (계속) JAPI 클라이언트의 기본 SSL 등록정보

등록정보	설명
olap.server.ssl.trustManagerClass	서명을 확인하고 인증서 만기 날짜를 점검하여 SSL 인증서를 검증하는 데 사용하는 TrustManager 클래스입니다. 기본적으로 이 등록정보는 모든 확인 검사를 실행하도록 설정되지 않습니다. 확인 검사를 실행하지 않으려면 모든 검증 검사가 성공할 수 있는 기본 TrustManager 클래스인 com.essbase.services.olap.security.EssDefaultTrust로 이 매개변수의 값을 설정하십시오. 사용자정의 TrustManager를 구현하려면 javax.net.ssl.X509TrustManager 인터페이스를 구현하는 TrustManager 클래스의 전체 클래스 이름을 지정하십시오. **예:** com.essbase.services.olap.security.EssDefaultTrust

3. essbase.properties를 저장한 후 닫습니다.
4. 모든 Essbase 구성요소를 재시작합니다.

세션당 SSL 연결 설정

Oracle Essbase 구성요소(예: MaxL)는 secure를 전송 키워드로 사용하여 Essbase 에이전트에 연결함으로써 세션 레벨에서 SSL을 제어할 수 있습니다. 예를 들어 MaxL 콘솔에서 다음 명령 중 하나를 실행하여 MaxL과 Essbase 에이전트 간의 보안 연결을 설정할 수 있습니다.

```
login admin example_password on hostA:PORT:secure
```

```
login admin example_password on hostA:secure
```

essbase.cfg에 지정된 구성 설정보다 세션당 제어에 우선순위가 있습니다. 전송 키워드가 지정되지 않은 경우 Essbase 클라이언트는 ClientPreferredMode에 대해 설정된 값을 사용하여 Essbase와 보안 연결을 시작할 것인지 결정합니다. ClientPreferredMode 설정이 보안으로 설정되지 않은 경우 비보안 채널을 통해 통신이 수행됩니다.

Essbase 21c에 대한 SSL

개요

이 섹션에는 MaxL, Oracle Essbase Administration Services 서버, Oracle Hyperion Provider Services, Oracle Hyperion Foundation Services, Oracle Hyperion Planning, Oracle Hyperion Financial Management, Oracle Hyperion Shared Services Registry와 같은 Oracle Essbase 인스턴스 및 구성요소 간에 통신을 보호하는 데 사용되는 기본 인증서를 바꾸는 절차가 설명되어 있습니다.

> **주:**
>
> EAS(Essbase Administration Services) Lite는 EPM Configurator를 사용하여 구성된 HTTP Server SSL 포트(예: 443)를 사용하지 않습니다. `easconsole.jnlp` 파일의 보안 URL은 기본적으로 비SSL 포트(80)로 설정됩니다.
>
> **해결 방법:** `easconsole.jnlp`에서 확인된 보안 URL의 기본 비SSL 포트를 업데이트된 보안 URL로 바꿉니다.
>
> 기본 보안 URL: `https://myserver:SECURE_PORT/easconsole/console.html`. 예: `https://myserver:80/easconsole/console.html`
>
> 업데이트된 보안 URL: `https://myserver:SECURE_PORT/easconsole/console.html`. 예: `https://myserver:443/easconsole/console.html`
>
> 자세한 내용은 MOS(My Oracle Support) 문서 - 문서 ID 1926558.1 - EAS 웹 콘솔의 easconsole.jnlp에 포함되지 않은 SSL 포트를 참조하십시오.

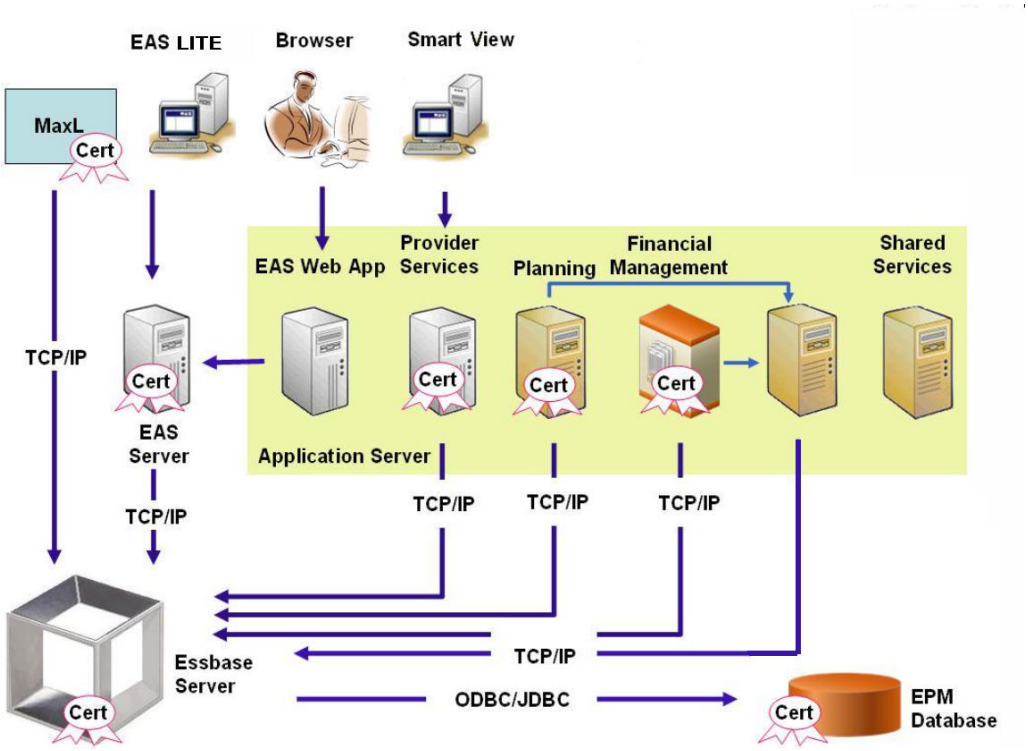
기본 배포

Essbase는 SSL 및 비SSL 모드로 사용하도록 배포할 수 있습니다. Essbase 에이전트는 비보안 포트에서 수신합니다. 보안 포트에서 수신하도록 구성할 수도 있습니다. 보안 포트에 액세스하는 모든 연결은 SSL 연결로 처리됩니다. 클라이언트가 비SSL 포트의 Essbase 에이전트에 연결하는 경우 연결은 비SSL 연결로 처리됩니다. 구성요소는 하나의 Essbase 에이전트에 대해 비SSL 연결 및 SSL 연결을 동시에 설정할 수 있습니다.

로그인할 때 보안 프로토콜 및 포트를 지정하여 세션별로 SSL을 제어할 수 있습니다. 세션당 SSL 연결 설정을 참조하십시오.

SSL이 사용으로 설정된 경우 데이터 보안이 보장되도록 Essbase 인스턴스 내의 모든 통신이 암호화됩니다.

보안 모드의 기본 Essbase 구성요소 배포에서는 자체 서명 인증서를 사용하여 주로 테스트용으로 SSL 통신을 사용으로 설정합니다. 프로덕션 환경에서는 잘 알려진 타사 CA의 인증서를 사용하여 Essbase의 SSL을 사용으로 설정하는 것이 좋습니다.



일반적으로 Oracle Wallet은 인증서를 저장하여 Essbase RTC를 사용하는 클라이언트와의 SSL 통신을 사용으로 설정하고, Java 키 저장소는 인증서를 저장하여 통신에 JAPI를 활용하는 구성요소와의 SSL 통신을 사용으로 설정합니다. SSL 통신을 설정하기 위해 Essbase 클라이언트 및 툴은 Essbase 서버 및 에이전트 인증서에 서명한 CA의 루트 인증서를 저장합니다.

필요한 인증서 및 해당 위치

프로덕션 환경에서는 잘 알려진 타사 CA의 인증서를 사용하여 Essbase의 SSL을 사용으로 설정하는 것이 좋습니다. 테스트용으로 기본 자체 서명 인증서를 사용할 수 있습니다.

> **주:**
>
> Essbase는 하나의 SSL 인증서로 여러 하위 도메인을 보호할 수 있는 와일드카드 인증서 사용을 지원합니다. 와일드카드 인증서를 사용하면 관리 시간 및 비용을 줄일 수 있습니다.
>
> 호스트 이름 확인이 사용으로 설정된 경우 와일드카드 인증서를 사용할 수 없습니다.

다음 인증서가 필요합니다.

- 루트 CA 인증서
  Essbase RTC를 사용하여 Essbase 연결을 설정하는 구성요소에서는 루트 CA 인증서가 Oracle Wallet에 저장되어야 합니다. JAPI를 사용하여 연결을 설정하는 구성요소에서는 루트 CA 인증서가 Java 키 저장소에 저장되어야 합니다. 필요한 인증서 및 해당 위치는 다음 테이블에 표시되어 있습니다.

> **주:**
>
> 해당 루트 인증서가 Oracle Wallet에 이미 설치되어 있는 잘 알려진 타사 CA의 인증서를 사용하는 경우 루트 CA 인증서를 설치할 필요가 없습니다.

- Essbase 서버 및 Essbase 에이전트의 서명된 인증서

표 2-4 필요한 인증서 및 해당 위치

구성요소[1]	키 저장소	인증서 [2]
MaxL	Oracle Wallet	루트 CA 인증서
Administration Services 서버	Oracle Wallet	루트 CA 인증서
Provider Services	Oracle Wallet	루트 CA 인증서
Oracle Enterprise Performance Management System 데이터베이스	Oracle Wallet	루트 CA 인증서
Planning	• Oracle Wallet • Java 키 저장소	루트 CA 인증서
Financial Management	Java 키 저장소	루트 CA 인증서
Essbase(서버 및 에이전트) [3]	• Oracle Wallet • Java 키 저장소	• 루트 CA 인증서 • Essbase 서버 및 에이전트의 서명된 인증서
Oracle Hyperion Shared Services 저장소		

[1] 유사한 키 저장소를 사용하는 여러 구성요소를 지원하는 데는 하나의 키 저장소 인스턴스만 있으면 됩니다.

[2] 여러 구성요소가 하나의 키 저장소에 설치된 하나의 루트 인증서를 사용할 수 있습니다.

[3] 인증서는 기본 Oracle Wallet 및 Java 키 저장소에 설치되어야 합니다.

Essbase 구성요소 설치 및 배포

구성 프로세스를 사용하면 보안 에이전트 포트(기본값은 6423)를 선택할 수 있습니다. 이 포트는 Oracle Essbase를 구성할 때 변경할 수 있습니다. 기본적으로 배포 프로세스는 필수 자체 서명 인증서를 설치하여 테스트를 위한 기능 보안 배포를 생성합니다.

EPM System Installer는 Oracle HTTP Server가 설치된 경우 Essbase 인스턴스를 호스트하는 머신의 *ARBOR_PATH* 내에 Oracle Wallet 및 자체 서명 인증서를 설치합니다. 단일 호스트 배포에서는 모든 Essbase 구성요소가 이 인증서를 공유합니다.

Essbase에 인증된 타사 CA 인증서 사용

인증서 요청 생성 및 인증서 가져오기

인증서 요청을 생성하여 Oracle Essbase 서버 및 Essbase 에이전트를 호스트하는 서버에 대한 인증서를 가져오십시오. 인증서 요청에는 서버의 CN(일반 이름)에 특정한 암호화된 정보가 포함되어 있습니다. 인증서 요청을 서명 기관에 제출하여 SSL 인증서를 가져옵니다.

keytool 또는 Oracle Wallet Manager와 같은 툴을 사용하여 인증서 요청을 생성합니다. 인증서 요청 생성에 대한 자세한 내용은 사용 중인 툴의 설명서를 참조하십시오.

keytool 사용 예:

JKS(Java Keystore)를 생성하고 전용 키를 생성합니다.

```
keytool.exe -genkey -dname "cn=myserver, ou=EPM, o=Oracle, c=US"
-alias essbase_ssl -keypass password -keystore
C:\oracle\Middleware\EPMSystem11R1\ssl\EPM.JKS -storepass password
-validity 365 -keyalg RSA -keysize 2048 -sigalg SHA256withRSA -noprompt
```

인증서 요청을 생성합니다:

```
keytool -certreq -alias essbase_ssl -file
C:\oracle\Middleware\EPMSystem11R1\ssl\essabse_server.csr -keypass
password
-keystore C:\oracle\Middleware\EPMSystem11R1\ssl\EPM.JKS -storepass
password
```

전용 키를 익스포트합니다(이러한 단계를 수행하려면 `openssl` 유틸리티가 필요함).

1. `openssl.exe pkcs12 -in C:\oracle\Middleware\EPMSystem11R1\ssl\EPM.JKS -passin pass:password -legacy -nocerts -out c:\Apache24\ssl\Apache24.key -passout pass:password`
2. CA(인증 기관)를 사용하여 새로 생성된 인증서 요청에 서명하고 다음 파일에 붙여넣습니다.
   `C:\oracle\Middleware\EPMSystem11R1\ssl\essbase.cer`.

루트 CA 인증서 가져오기 및 설치

루트 CA 인증서는 SSL을 지원하는 데 사용되는 인증서의 유효성을 확인합니다. 인증서에 서명하는 데 사용된 개인 키와 일치시켜 인증을 확인하는 공개 키가 포함되어 있습니다. SSL 인증서에 서명한 인증 기관의 루트 CA 인증서를 가져올 수 있습니다.

Essbase 서버 또는 에이전트와 연결된 클라이언트에 Essbase 서버 인증서에 서명한 CA의 루트 인증서를 설치합니다. 클라이언트에 적합한 키 저장소에 루트 인증서가 설치되어 있는지 확인하십시오. 필요한 인증서 및 해당 위치를 참조하십시오.

> **주:**
>
> 여러 구성요소가 하나의 서버 머신에 설치된 하나의 루트 CA 인증서를 사용할 수 있습니다.

CA 서명 인증서 설치

CA 서명 인증서 설치에 관해서는 다음 링크를 참조하십시오.

- Essbase에 대한 Weblogic TLS 연결 설정
- TLS 인증서 업데이트

다음 아래 `tls.properties` 파일 업데이트

```
%EPM_HOME%\essbase\bin\tls_tools.properties:
certCA=c:\\ssl\\ca.crt;c:\\ssl\\intermediate.crt;c:\\ssl\
\essbase.key;c:\\ssl\\essbase.cer;
```

위치:

```
C:\ssl\ca.crt - root CA certificate.
C:\ssl\intermediate.crt - intermediate CA certificate.
C:\ssl\essbase.key - your private key generated in the previous step.
C:\ssl\essbase.cer - your server's signed certificate issued by your CA.
```

다음을 실행하여 Essbase 서버를 새 인증서로 업데이트합니다.

```
set ORACLE_HOME=c:\OracleSSL
set EPM_HOME=%ORACLE_HOME%
set WL_HOME=%ORACLE_HOME%\wlserver
set JAVA_HOME=%ORACLE_HOME%\jdk
set DOMAIN_HOME=%ORACLE_HOME%\user_projects\domains\essbase_domain
%EPM_HOME%\essbase\bin\tls_tools.properties:
%ORACLE_HOME%\\jdk\bin\java.exe -Xmx256m -jar %ORACLE_HOME%
\essbase\lib\tlsTools.jar %EPM_HOME%\essbase\bin\tls_tools.properties
```

Essbase SSL 설정 업데이트

`essbase.cfg`에 다음 값을 지정하여 Essbase 서버 및 클라이언트의 SSL 설정을 사용자정의합니다.

- 보안 모드를 사용으로 지정하는 설정
- 지우기 모드를 사용으로 지정하는 설정
- 클라이언트와 통신하는 기본 모드(클라이언트만 사용)
- 보안 포트
- 암호 모음
- Oracle Wallet 경로

> **주:**
>
> `essbase.cfg`에서 누락된 필수 매개변수(특히, `EnableSecureMode` 및 `AgentSecurePort`)를 추가하고 해당 값을 설정해야 합니다.

다음에 있는 `essbase.cfg`를 업데이트하려면:

```
ESSBASE_DOMAIN_HOME\config\fmwconfig\essconfig\essbase
```

1. 필요에 따라 설정을 입력합니다. 기본 Essbase 설정은 내재되어 있습니다. 기본 동작을 변경해야 하는 경우 `essbase.cfg`에서 사용자정의 동작 설정을 추가합니다. 예를 들어 `EnableClearMode`는 기본적으로 적용되며, 이 설정에 따라 Essbase 서버는 암호화되지 않은 채널을 통해 통신할 수 있습니다. Essbase 서버에서 암호화되지 않은 채널을 통해 통신하는 기능을 끄려면 `essbase.cfg`에 `EnableClearMode FALSE`를 지정해야 합니다. 다음 테이블을 참조하십시오:

표 2-5 Essbase SSL 설정

설정	설명 [1]
EnableClearMode[2]	Essbase 애플리케이션과 Essbase 에이전트 간에 암호화되지 않은 통신이 사용됩니다. 이 등록정보가 FALSE로 설정되면 Essbase에서 비SSL 요청을 처리하지 않습니다. **기본값:** EnableClearMode TRUE **예:** EnableClearMode FALSE
EnableSecureMode	Essbase 클라이언트와 Essbase 에이전트 간에 SSL 암호화 통신이 사용됩니다. 이 등록정보를 TRUE로 설정해야 SSL이 지원됩니다. **기본값:** FALSE **예:** EnableSecureMode TRUE
SSLCipherSuites	SSL 통신에 사용할 암호 모음이 환경설정 순서대로 표시된 목록입니다. Essbase 에이전트는 이러한 암호 모음 중 하나를 SSL 통신에 사용합니다. 에이전트가 암호 모음을 선택하는 경우 목록의 첫번째 암호 모음에 가장 높은 우선순위가 부여됩니다. **기본값:** SSL_RSA_WITH_RC4_128_MD5 **예:** SSLCipherSuites SSL_RSA_WITH_AES_256_CBC_SHA256,SSL_RSA_WITH_AES_256_GCM_SHA384
APSRESOLVER	Oracle Hyperion Provider Services URL입니다. 여러 Provider Services 서버를 사용 중인 경우 각 URL을 세미콜론을 사용하여 구분하십시오. **예:** https://exampleAPShost1:*PORT*/essbase;https://exampleAPShost2:*PORT*/essbase
AgentSecurePort	에이전트가 수신하는 보안 포트입니다. **기본값:** 6423 **예:** AgentSecurePort 16001
WalletPath	루트 CA 인증서 및 서명된 인증서를 저장하는 Oracle Wallet(1,024자 미만)의 위치입니다. **기본값:** *ARBORPATH*/bin/wallet **예:** WalletPath/usr/local/wallet
ClientPreferredMode [3]	클라이언트 세션의 모드(보안 또는 지우기)입니다. 이 등록정보를 보안으로 설정하면 SSL 모드가 모든 세션에 사용됩니다. 이 등록정보를 지우기로 설정하면 클라이언트 로그인 요청에 보안 전송 키워드가 포함되어 있는지 여부에 따라 전송이 선택됩니다. 세션당 SSL 연결 설정을 참조하십시오. **기본값:** CLEAR **예:** ClientPreferredMode SECURE

- [1] essbase.cfg에서 해당 등록정보를 사용할 수 없는 경우 기본값이 적용됩니다.
- [2] EnableClearMode 및 EnableSecureMode 둘 다 FALSE로 설정되는 경우 Essbase가 작동하지 않게 됩니다.
- [3] 클라이언트는 이 설정을 사용하여 Essbase와 보안 연결을 설정할 것인지 비보안 연결을 설정할 것인지 결정합니다.

2. essbase.cfg를 저장한 후 닫습니다.

SSL용 분산 Essbase 노드 업데이트

> **주:**
>
> 이 섹션은 Essbase 분산 배포에만 적용됩니다.

루트 CA 인증서 및 서명된 인증서가 포함된 Wallet 폴더(예: `WalletPath/usr/local/wallet`)가 각 분산 노드의 필요한 위치에 있는지 확인하십시오.

1. TLS 도구를 사용하여 모든 새 CA 인증서를 임포트합니다.

   자세한 내용은 다음 링크를 참조하십시오.

   - Essbase에 대한 Weblogic TLS 연결 설정
   - TLS 인증서 업데이트

2. 소스 위치 `ESSBASE_DOMAIN_HOME\config\fmwconfig\essconfig\essbase`로 이동하고 `essbase.properties` 파일에서 다음 등록정보를 수정합니다.

   - `essbase.ssleverywhere=true`
   - `olap.server.ssl.alwaysSecure=true`
   - `APSRESOLVER=`*`APS_URL`*
     *`APS_URL`*을 `Provider Services URL`로 바꿉니다. 여러 `Provider Services` 서버를 사용 중인 경우 가 `URL`을 세미콜론을 사용하여 구분합니다.

     `https://exampleAPShost1:`*`PORT`*`/essbase;https://exampleAPShost2:`*`PORT`*`/essbase`.

3. `Wallet` 폴더, `Walletssl` 폴더, `essbase.cfg` 파일 및 `essbase.properties` 파일을 다음 대상 경로에 복사합니다.

표 2-6 대상 경로

대상 경로	Wallet	Walletssl	essbase.cfg	essbase.properties
`EPM_ORACLE_HOME\common\EssbaseRTC-21C\11.1.2.0\bin`	예	예	예	예
`EPM_ORACLE_HOME\common\EssbaseJavaAPI-21C\11.1.2.0\bin`	예	예	예	예
`ESSBASE_DOMAIN_HOME\config\fmwconfig\essconfig\aps`	예	예	예	예
`ESSBASE_DOMAIN_HOME\config\fmwconfig\essconfig\essbase`	예	예	예	예
`MIDDLEWARE_HOME\essbase\products\Essbase\template_files\essbase`	예	예	예	예
`MIDDLEWARE_HOME\essbase\products\Essbase\EssbaseServer\bin`	예	예	예	예
`MIDDLEWARE_HOME\essbase\products\Essbase\aps\bin`	예	예	예	예
`MIDDLEWARE_HOME\essbase\products\Essbase\eas`	예	예	예	예
`MIDDLEWARE_HOME\essbase\common\EssbaseJavaAPI\bin`	예	예	예	예

표 2-6 (계속) 대상 경로

대상 경로	Wallet	Walletss	essbase.ssl	essbase.properties
Oracle Hyperion Financial Reporting만 해당 `EPM_ORACLE_HOME/products/financialreporting/bin/EssbaseJAPI/bin/` **참고:** 전체 SSL 환경에서 Financial Reporting에 연결하려면 Essbase 클러스터 이름이 필요합니다. 호스트 이름을 사용하여 연결하면 연결이 실패하게 됩니다.	예	예	예	예

Oracle Hyperion Planning만 해당
`EPM_ORACLE_HOME/products/Planning/config/`
`EPM_ORACLE_HOME/products/Planning/lib/`
	예	예	예	예

4. 다음과 같이 환경 변수를 설정합니다.
   - **Windows:** 새 시스템 변수 `API_DISABLE_PEER_VERIFICATION`을 생성하고 값을 `1`로 설정합니다.
   - **Linux:** `setCustomParamsPlanning.sh`에서 `API_DISABLE_PEER_VERIFICATION=1` 지시어를 추가합니다.

JAPI 클라이언트의 SSL 등록정보 사용자정의

JAPI를 사용하는 Essbase 구성요소의 경우 여러 기본 등록정보가 사전 정의되어 있습니다. `essbase.properties`에 등록정보를 포함하여 기본 등록정보를 대체할 수 있습니다.

> **주:**
>
> 다음 테이블에서 표시된 SSL 등록정보 중 일부만 `essbase.properties`에서 외부화됩니다. 외부화되지 않는 등록정보를 추가해야 합니다.

JAPI 클라이언트의 SSL 등록정보를 업데이트하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 `EPM_ORACLE_HOME/common/EssbaseJavaAPI-21C/11.2.0/bin/essbase.properties`를 엽니다.
2. 필요에 따라 등록정보를 업데이트합니다. 사용자정의 가능한 JAPI 클라이언트 등록정보에 대한 설명은 다음 테이블을 참조하십시오.
   원하는 등록정보가 `essbase.properties`에 포함되어 있지 않으면 해당 등록정보를 추가하십시오.

표 2-7 JAPI 클라이언트의 기본 SSL 등록정보

등록정보	설명
`olap.server.ssl.alwaysSecure`	클라이언트가 모든 Essbase 인스턴스에 대해 사용해야 하는 모드를 설정합니다. SSL 모드를 적용하려면 이 등록정보 값을 `true`로 변경하십시오. **기본값:** `false`

표 2-7 (계속) JAPI 클라이언트의 기본 SSL 등록정보

등록정보	설명
olap.server.ssl.securityHandlerPkgs	프로토콜을 처리하기 위한 패키지 이름입니다. 다른 처리기를 표시하도록 이 값을 변경할 수 있습니다. **기본값:** java.protocol.handler.pkgs
olap.server.ssl.securityProvider	Oracle에서는 Sun SSL 프로토콜 구현을 사용합니다. 다른 제공자를 표시하도록 이 값을 변경할 수 있습니다. **기본값:** com.sun.net.ssl.internal.www.protocol
olap.server.ssl.supportedCipherSuites	보안 통신을 위해 사용으로 설정할 추가 암호의 쉼표로 구분된 목록입니다. Essbase가 지원하는 암호만 지정해야 합니다. **예:** SSL_RSA_WITH_AES_256_CBC_SHA256,SSL_RSA_WITH_AES_256_GCM_SHA384
olap.server.ssl.trustManagerClassName	서명을 확인하고 인증서 만기 날짜를 점검하여 SSL 인증서를 검증하는 데 사용하는 TrustManager 클래스입니다. 기본적으로 이 등록정보는 모든 확인 검사를 실행하도록 설정되지 않습니다. 확인 검사를 실행하지 않으려면 모든 검증 검사가 성공할 수 있는 기본 TrustManager 클래스인 com.essbase.services.olap.security.EssDefaultTrustManager로 이 매개변수의 값을 설정하십시오. 사용자정의 TrustManager를 구현하려면 javax.net.ssl.X509TrustManager 인터페이스를 구현하는 TrustManager 클래스의 전체 클래스 이름을 지정하십시오. **예:** com.essbase.services.olap.security.EssDefaultTrustManager

3. essbase.properties를 저장한 후 닫습니다.
4. 모든 Essbase 구성요소를 재시작합니다.

세션당 SSL 연결 설정

Oracle Essbase 구성요소(예: MaxL)는 secure를 전송 키워드로 사용하여 Essbase 에이전트에 연결함으로써 세션 레벨에서 SSL을 제어할 수 있습니다. 예를 들어 MaxL 콘솔에서 다음 명령 중 하나를 실행하여 MaxL과 Essbase 에이전트 간의 보안 연결을 설정할 수 있습니다.

```
login admin example_password on hostA:PORT:secure
```

```
login admin example_password on hostA:secure
```

essbase.cfg에 지정된 구성 설정보다 세션당 제어에 우선순위가 있습니다. 전송 키워드가 지정되지 않은 경우 Essbase 클라이언트는 ClientPreferredMode에 대해 설정된 값을 사용하여 Essbase와 보안 연결을 시작할 것인지 결정합니다. ClientPreferredMode 설정이 보안으로 설정되지 않은 경우 비보안 채널을 통해 통신이 수행됩니다.

3

보안 에이전트로 SSO 활성화

참조:

- 지원되는 SSO 방법
- Oracle Access Manager의 싱글 사인온
- OracleAS 싱글 사인온
- SSO를 위한 EPM System 제품 보호
- ID 관리 제품을 사용하는 머리글 기반 SSO
- Oracle Identity Cloud Services를 통해 EPM System에서 머리글 기반 SSO를 지원하도록 구성
- SiteMinder SSO
- Kerberos 싱글 사인온
- SSO에 대해 EPM System 구성
- Smart View의 싱글 사인온 옵션

지원되는 SSO 방법

SSO를 사용하려면 웹 ID 관리 솔루션이 인증된 사용자의 로그인 이름을 Oracle Enterprise Performance Management System 제품에 전달해야 합니다. 다음 표준 EPM System 메소드를 사용하여 EPM System을 상용 및 사용자정의 웹 기반 SSO 솔루션과 통합할 수 있습니다.

- HTTP 머리글
- 사용자정의 로그인 클래스
- HTTP 권한부여 머리글
- HTTP 요청에서 원격 사용자 가져오기
- ID 관리 제품을 사용하는 머리글 기반 인증

> **주의:**
>
> 조직에서 ID 전달을 위해 머리글의 사용자 ID를 전달하는 메소드를 사용하는 경우 보안 측정항목으로 웹 서버와 애플리케이션 서버 간에 클라이언트 인증서 인증(양방향 SSL)을 구현하는 것이 좋습니다.

HTTP 머리글

OSSO(Oracle Single Sign-on), SiteMinder 또는 Oracle Access Manager를 웹 ID 관리 솔루션으로 사용 중인 경우 사용자정의 HTTP 머리글이 인증된 사용자의 로그인 이름을 EPM System 구성요소로 전달하도록 EPM System 보안에서 자동으로 선택합니다.

EPM System 제품 사용자의 로그인 이름은 Oracle Hyperion Shared Services에서 사용자 디렉토리를 구성하는 중에 지정된 `Login Attribute`에 의해 결정됩니다. `Login Attribute`에 관한 간단한 설명은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.

HTTP 머리글에는 로그인 속성으로 설정된 속성 값이 있어야 합니다. 예를 들어 `uid`가 `Login Attribute` 값인 경우 HTTP 머리글은 `uid` 속성 값을 전달해야 합니다.

사용자정의 HTTP 머리글의 정의 및 실행에 대한 자세한 내용은 웹 ID 관리 솔루션 설명서를 참조하십시오.

EPM System 보안은 HTTP 머리글을 구문분석하고 전달할 로그인 이름을 Shared Services에 구성된 사용자 디렉토리에 대해 검증합니다.

사용자정의 로그인 클래스

사용자 로그인 시 웹 ID 관리 솔루션은 디렉토리 서버에 대해 사용자를 인증하고, 인증된 사용자의 인증서를 SSO 메커니즘으로 캡슐화하여 다운스트림 시스템에서 SSO를 사용할 수 있도록 합니다. 웹 ID 관리 솔루션이 EPM System 제품에서 지원하지 않는 메커니즘을 사용하거나 SSO 메커니즘에서 `Login Attribute` 값을 사용할 수 없는 경우 사용자정의 로그인 클래스를 사용하여 `Login Attribute` 값을 파생시켜 EPM System 제품에 전달합니다.

사용자정의 로그인 클래스를 사용하면 EPM System에서 X509 인증서 기반 인증을 사용하는 보안 에이전트와 통합할 수 있습니다. 이 인증 메커니즘을 사용하려면 표준 공유 서비스 API를 구현하여 EPM System 구성요소와 웹 ID 관리 솔루션 간에 SSO 인터페이스를 정의해야 합니다. 사용자정의 로그인 클래스는 로그인 속성 값을 EPM System 제품에 전달해야 합니다. `Login Attribute`에 관한 간단한 설명은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오. 샘플 코드 및 구현 단계는 사용자정의 로그인 클래스 구현을 참조하십시오.

사용자정의 로그인 클래스(기본 이름은 `com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl`)를 사용하려면 `com.hyperion.css.CSSSecurityAgentIF` 인터페이스 구현을 클래스 경로에서 사용할 수 있어야 합니다. `CSSSecurityAgentIF`는 사용자 이름과 비밀번호(선택사항)를 검색하기 위한 getter 메소드를 정의합니다. 인터페이스에서 Null 비밀번호를 반환하는 경우 보안 인증에서 제공자를 신뢰할 수 있는 제공자로 처리하고 구성된 제공자의 사용자 존재를 확인합니다. 인터페이스에서 비밀번호에 대해 Null이 아닌 값을 반환하는 경우 EPM System은 이 구현에서 반환한 사용자 이름과 비밀번호를 사용하여 요청을 인증하려고 시도합니다.

`CSSSecurityAgentIF`는 2가지 메서드 즉, `getUserName`과 `getPassword`로 구성됩니다.

getUserName 메소드

이 메서드는 인증에 대해 사용자 이름을 반환합니다.

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

`req` 매개변수는 사용자 이름을 확인하는 데 사용되는 정보를 전달하는 HTTP 요청을 식별합니다. `res` 매개변수(역호환성을 위해 미리 설정됨)는 사용되지 않습니다.

getPassword 메소드

이 메서드는 인증에 대해 일반 텍스트 비밀번호를 반환합니다. 비밀번호 검색은 선택 사항입니다.

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

`req` 매개변수는 사용자 이름을 확인하는 데 사용되는 정보를 전달하는 HTTP 요청을 식별합니다. `res` 매개변수(역호환성을 위해 미리 설정됨)는 사용되지 않습니다.

HTTP 권한부여 머리글

EPM System 보안은 HTTP 권한부여 머리글을 사용하여 웹 ID 관리 솔루션에서 EPM System 제품으로 `Login Attribute`의 값을 전달하도록 지원합니다. EPM System 제품은 권한부여 머리글을 구문분석하여 사용자의 로그인 이름을 검색합니다.

HTTP 요청에서 원격 사용자 가져오기

EPM System 보안은 HTTP 요청을 사용하여 웹 ID 관리 솔루션에서 EPM System 제품으로 `Login Attribute`의 값을 전달하도록 지원합니다. 웹 ID 관리 솔루션이 `setRemoteUser` 함수를 사용하여 설정된 `Login Attribute`의 값을 포함하는 HTTP 요청을 전달하는 경우 이 SSO 메소드를 사용하십시오.

ID 관리 제품을 사용하는 머리글 기반 인증

EPM System에서는 Oracle Identity Cloud Services, Microsoft Azure AD, Okta와 같이 머리글 기반 인증을 지원하는 모든 ID 관리 제품을 지원합니다. 개념 워크플로우는 다음과 같습니다.

- 역방향 프록시 역할을 하는 게이트웨이 애플리케이션은 인증되지 않은 네트워크 액세스를 제한하여 EPM System 구성요소를 보호합니다.
- 게이트웨이 애플리케이션은 EPM System 구성요소에 대한 HTTP(S) 요청을 가로채고 ID 관리 제품에서 사용자를 인증한 후 EPM System 구성요소에 요청을 전달하도록 합니다.
- 요청을 EPM System 구성요소에 전달하는 동안 게이트웨이 애플리케이션은 인증된 사용자의 ID를 HTTP 머리글 요청을 통해 EPM System 구성요소로 전파합니다.

이 인증 시나리오를 지원하려면 EPM System이 HTTP(S) 머리글 요청을 통해 전파되는 인증된 사용자 ID로 작동하도록 구성되어야 합니다.

Oracle Access Manager의 싱글 사인온

Oracle Enterprise Performance Management System은 로그인 속성 값이 포함된 사용자정의 HTTP 머리글(기본 이름 `HYPLOGIN`)을 수락하여 Oracle Access Manager와 통합됩니다. Oracle Hyperion Shared Services에서 외부 사용자 디렉토리를 구성할 때 로그인 속성이 설정됩니다. `Login Attribute`에 관한 간단한 설명은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.

EPM System에 로그인 속성 값을 제공하는 모든 머리글 이름을 사용할 수 있습니다. 머리글 이름은 Oracle Access Manager에서 SSO에 대해 Shared Services를 구성할 때 사용합니다.

EPM System은 로그인 속성 값을 사용하여 구성된 사용자 디렉토리(이 경우 Oracle Access Manager가 사용자를 인증하는 사용자 디렉토리)에 대해 사용자를 인증하고 EPM System에서 SSO를 사용으로 설정할 수 있는 EPM SSO 토큰을 생성합니다. Native Directory에서 사용자에 대한 프로비저닝 정보를 확인하여 EPM System 리소스에 대한 권한을 사용자에게 부여합니다.

> **주:**
>
> Thick Client인 Oracle Essbase Administration Services 콘솔은 Oracle Access Manager의 SSO를 지원하지 않습니다.

Oracle Access Manager 구성과 HTTP 머리글 및 정책 도메인 설정 같은 태스크 수행에 대한 정보는 Oracle Access Manager 설명서에 제공되어 있습니다. 이 가이드에서는 다음 태스크를 완료한 Oracle Access Manager 배포가 작동 중이라고 가정합니다.

- EPM System 구성요소에 필요한 정책 도메인 설정
- 로그인 속성 값을 EPM System에 전달하는 HTTP 머리글 구성
- 보호할 리소스에 나열된 EPM System 리소스 보호. 보호된 리소스에 대한 액세스 요청은 Oracle Access Manager에서 수행됩니다.
- 보호 해제할 리소스에 나열된 EPM System 리소스 보호 해제. 보호 해제된 리소스에 대한 액세스 요청은 Oracle Access Manager에서 수행되지 않습니다.

Oracle Access Manager에서 SSO에 대해 EPM System을 구성하려면 다음을 수행합니다.

1. Oracle Access Manager가 사용자를 인증하는 데 사용하는 사용자 디렉토리를 EPM System의 외부 사용자 디렉토리로 추가합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.

   > **주:**
   >
   > 연결 정보 화면의 **인증** 확인란이 선택되어 사용자 디렉토리가 인증된 SSO 소스임을 나타내야 합니다.

2. SSO에 대해 EPM System을 구성합니다. SSO에 대해 EPM System 구성을 참조하십시오.

   SSO 제공자 또는 에이전트 목록에서 Oracle Access Manager를 선택합니다. Oracle Access Manager의 HTTP 머리글이 `HYPLOGIN` 이외의 이름을 사용하는 경우 **SSO 메커니즘** 목록 옆의 텍스트 상자에 사용자정의 머리글 이름을 입력합니다.

3. Oracle Data Relationship Management만 해당:

   a. Shared Services 인증에 대해 Data Relationship Management를 구성합니다.

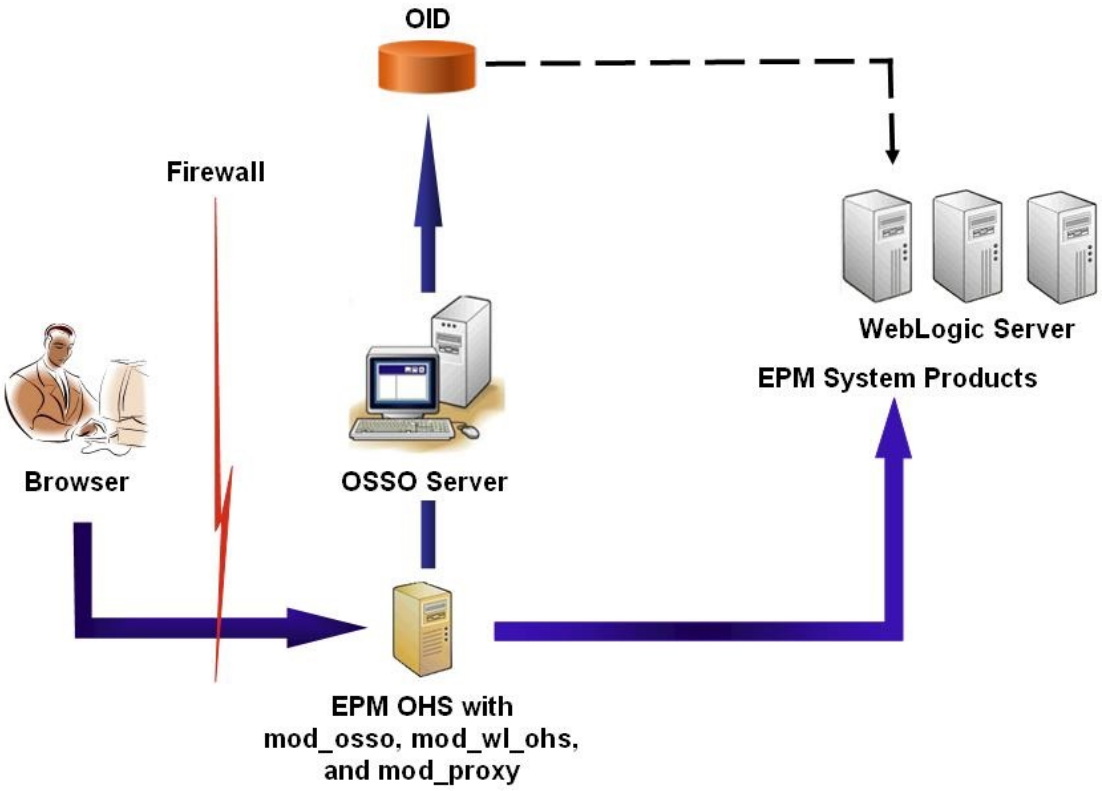
   b. Data Relationship Management 콘솔에서 SSO를 사용으로 설정합니다.

      자세한 내용은 Data Relationship Management 설명서를 참조하십시오.

OracleAS 싱글 사인온

OSSO(OracleAS Single Sign-on) 솔루션은 OID(Oracle Internet Directory)를 사용자 디렉토리로 사용하는 웹 애플리케이션에 대한 SSO 액세스를 제공합니다. 사용자는 OID에 정의된 사용자 이름과 비밀번호를 사용하여 Oracle Enterprise Performance Management System 제품에 로그인합니다.

프로세스 플로우



OSSO 프로세스는 다음과 같습니다.

1. 사용자는 EPM System URL(예: `http://OSSO_OHS_Server_NAME:OSSO_OHS_Server_PORT/interop/index.jsp`)을 사용하여 OSSO 보호 애플리케이션으로 정의된 EPM System 구성요소에 액세스합니다.
2. URL에 OSSO 보호가 적용되므로 Oracle HTTP Server에 배포된 `mod_osso`가 요청을 가로챕니다. Oracle HTTP Server는 `mod_osso`를 사용하여 적합한 쿠키를 확인합니다. 요청에서 적합한 쿠키를 사용할 수 없는 경우 Oracle HTTP Server는 사용자를 OSSO 서버로 리디렉션합니다. 이 서버는 사용자에게 인증서를 요청하며 OID에 대해 인증서를 인증합니다.
3. OSSO 서버는 obSSOCookie를 생성하고 Oracle HTTP Server의 `mod_osso` 모듈에 제어를 반환합니다. Oracle HTTP Server는 브라우저에서 obSSOCookie를 설정합니다. 또한 `mod_wl_ohs`(Oracle WebLogic Server)를 통해 요청을 EPM System 리소스로 리디렉션합니다. 요청을 EPM System 리소스로 전달하기 전에 Oracle HTTP Server에서는 EPM System 보안에서 SSO를 사용으로 설정하는 데 사용하는 `Proxy-Remote-User` 머리글을 설정합니다.

4. EPM System 구성요소는 해당 ID가 `Proxy-Remote-User`에서 검색되는 사용자가 OID에 있는지 확인합니다. 이 프로세스가 작동하려면 OSSO 서버로 구성된 OID를 Oracle Hyperion Shared Services의 외부 사용자 디렉토리로 구성해야 합니다.

사전 필수 조건

1. 완전한 기능을 갖춘 Oracle Application Server Infrastructure

   Oracle Application Server Infrastructure를 설정하려면 Oracle Identity Management Infrastructure 10.1.4를 설치하고 구성합니다. OSSO가 사용으로 설정되어 있는지 확인하십시오. Oracle Identity Management Infrastructure 10.1.4 설치에는 OSSO를 지원하는 다음 구성요소가 포함되어 있습니다.

   - Oracle 10*g* OSSO 서버
   - OSSO 서버가 인증서를 검증하는 데 사용하는 OID. 다음 가이드를 참조하십시오.
     - *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*
     - *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
   - OSSO 서버의 프런트엔드인 Oracle HTTP Server. 이 설치에는 `mod_osso`가 포함되어 있으므로 OSSO에 대한 파트너 애플리케이션을 정의할 수 있습니다.

     > **주:**
     >
     > 이 Oracle HTTP Server 인스턴스는 OSSO 인프라의 일부입니다. EPM System 구성요소에 대해 OSSO를 구성하는 데 직접 사용되지 않습니다.

     설치 프로세스 중에 `mod_osso`가 OSSO 서버에 파트너 애플리케이션으로 등록되었는지 확인하십시오.

2. 완전한 기능을 갖춘 EPM System 배포
   EPM System 구성요소에 대해 웹 서버를 구성하는 경우 EPM System Configurator가 Oracle HTTP Server에서 `mod_wl_ohs.conf`를 구성하여 요청을 WebLogic Server로 프록시합니다.

배포 테스트

SSL 배포를 완료한 후 모든 요소가 작동하는지 확인하십시오.

배포를 테스트하려면 다음을 수행합니다.

1. 브라우저를 사용하여 다음과 같이 보안 Oracle Hyperion Enterprise Performance Management Workspace URL에 액세스합니다.

   `epm.myCompany.com`을 외부 통신을 위한 서버 별칭으로 사용하고 `4443`을 SSL 포트로 사용하는 경우 EPM Workspace URL은 다음과 같습니다.

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

2. [로그온] 화면에서 사용자 이름과 비밀번호를 입력합니다.
3. **로그온**을 누릅니다.

4. 배포된 Oracle Enterprise Performance Management System 구성요소에 안전하게 액세스할 수 있는지 확인합니다.

EPM System에 OSSO 사용

이 절에서는 완전하게 구성된 OSSO 인프라가 있다고 가정합니다. *Oracle Application Server Administrator's Guide*를 참조하십시오.

EPM System 웹 서버를 파트너 애플리케이션으로 등록

Oracle Identity Manager SSO 등록 툴(`ssoreg.sh` 또는 `ssoreg.bat`)을 사용하여 Oracle Enterprise Performance Management System 웹 서버를 OSSO 서버의 프런트엔드에 사용되는 Oracle HTTP Server의 파트너 애플리케이션으로 등록합니다.

이 절차는 OSSO 서버의 프런트엔드에 사용되는 Oracle HTTP Server를 호스트하는 서버에서 수행하십시오. 이 프로세스에서는 쉽게 알아낼 수 없는 `osso.conf`를 생성하여 사용자가 선택한 위치에 저장합니다.

EPM System 웹 서버를 파트너 애플리케이션으로 등록하려면 다음을 수행합니다.

1. OSSO 서버의 프런트엔드로 사용되는 Oracle HTTP Server를 호스트하는 서버에서 콘솔을 열고 Oracle HTTP Server의 *`ORACLE_HOME`*`/sso/bin` 디렉토리, 예를 들어 `C:/OraHome_1/sso/bin`(Windows)으로 이동합니다.
2. 다음과 유사한 명령을 `-remote_midtier` 옵션을 사용하여 실행합니다.

```
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
```

아래에서는 이 명령에 사용되는 매개변수를 설명합니다. 이 설명에서 파트너 애플리케이션은 EPM System 웹 서버로 사용되는 Oracle HTTP Server입니다.

- `-site_name`은 파트너 애플리케이션의 웹 사이트(예: `epm.myCompany.com`)를 나타냅니다.
- `-mod_osso_url`은 *`PROTOCOL://HOST_NAME:PORT`* 형식의 파트너 애플리케이션 URL을 나타냅니다. EPM System 웹 서버가 들어오는 클라이언트 요청을 수락하는 URL입니다(예: `http://epm.myCompany.com:19000`).
- `-config_mod_osso`는 파트너 애플리케이션이 `mod_osso`를 사용하는지 확인합니다. `osso.conf`를 생성하려면 `config_mod_osso`가 매개변수를 포함해야 합니다.
- `-update_mode`는 업데이트 모드를 나타냅니다. 기본값인 `CREATE`를 사용하여 새 레코드를 생성하십시오.
- `-remote_midtier`는 `mod_osso` 파트너 애플리케이션이 원격 중간 계층에 있도록 지정합니다. 파트너 애플리케이션이 OSSO 서버 이외의 *`ORACLE_HOME`*에 있는 경우 이 옵션을 사용합니다.
- `-virtualhost`는 파트너 애플리케이션 URL이 가상 호스트임을 나타냅니다. 가상 호스트를 사용하지 않는 경우 이 매개변수를 사용하지 마십시오.
  가상 호스트에 연결된 파트너 애플리케이션 URL을 등록하는 경우 `httpd.conf`에서 가상 호스트를 정의해야 합니다. 선택사항: 가상 호스트 정의을 참조하십시오.
- `-config_file`은 `osso.conf` 파일이 생성될 경로를 나타냅니다.

선택사항: 가상 호스트 정의

파트너 애플리케이션을 등록할 때 가상 호스트 URL을 사용한 경우 EPM System 웹 서버로 사용되는 Oracle HTTP Server에서 `httpd.conf`를 업데이트하여 가상 호스트를 정의해야 합니다.

가상 호스트를 정의하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*`/httpConfig/ohs/config/OHS/ohs_component/httpd.conf`를 엽니다.
2. 다음과 유사한 정의를 추가합니다. 이 정의에서는 웹 서버가 `epm.myCompany.com:19400` 포트의 가상 서버 `epm.myCompany.com`에서 실행 중이라고 가정합니다. 요구사항에 맞게 설정을 수정합니다.

```
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` 생성

EPM System 웹 서버의 프런트엔드로 사용되는 Oracle HTTP Server에 `mod_osso.conf`를 생성하십시오.

`mod_osso.conf`를 생성하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 파일을 생성합니다.
2. 다음 콘텐츠를 파일에 복사하여 환경에 맞게 수정합니다.

```
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` 정의 내에서 다음과 유사한 위치 정의를 포함하여 OSSO를 통해 보호하려는 각 리소스를 확인합니다.

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

4. 파일을 `mod_osso.conf`로 저장합니다.

`osso.conf` 재배치

EPM System 웹 서버를 파트너 애플리케이션으로 등록(EPM System 웹 서버를 파트너 애플리케이션으로 등록 참조)하는 프로세스는 쉽게 알아낼 수 없는 `osso.conf` 파일을 `-config_file` 지시어로 확인된 위치에 생성합니다.

`osso.conf`를 재배치하려면 다음을 수행합니다.

1. EPM System 웹 서버를 파트너 애플리케이션으로 등록할 때 생성된 `osso.conf`를 찾습니다(EPM System 웹 서버를 파트너 애플리케이션으로 등록 참조).
2. `mod_osso.conf`(mod_osso.conf 생성 참조)에 정의되어 있는 `OssoConifgFile` 등록정보를 통해 확인된 디렉토리(OSSO 서버의 프런트엔드로 사용되는 Oracle HTTP Server의 디렉토리)에 `osso.conf`를 복사합니다.

OSSO에 대해 EPM System 구성

OSSO 솔루션과 통합된 OID를 EPM System의 외부 사용자 디렉토리로 구성하고 SSO를 사용으로 설정하십시오.

OSSO에 대해 EPM System을 구성하려면 다음을 수행합니다.

1. OSSO 솔루션에서 외부 사용자 디렉토리로 사용하는 OID를 구성합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.
2. EPM System에서 SSO를 사용으로 설정합니다. SSO에 대해 EPM System 구성

> **주:**
>
> OSSO를 ID 관리 솔루션으로 구성하려면 **SSO 제공자 또는 에이전트**에서 `Other`, **SSO 메커니즘**에서 `Custom HTTP Header`를 선택하고 사용자정의 HTTP 머리글 이름으로 `Proxy-Remote-User`를 입력해야 합니다.

3. 하나 이상의 OID 사용자를 Oracle Hyperion Shared Services 관리자로 프로비저닝합니다.
4. Shared Services 보안 API를 사용하는 EPM System 제품과 사용자정의 애플리케이션을 다시 시작합니다.

> **주:**
>
> Shared Services와 함께 구성된 OID가 실행 중인지 확인한 후 EPM System 제품을 시작합니다.

선택사항: OSSO 서버에서 디버깅 메시지 사용으로 설정

OSSO 서버에 디버깅 메시지를 기록하려면 `policy.properties`를 수정하십시오. 디버깅 메시지는 *`ORACLE_HOME/sso/log/ssoServer.log`*에 기록됩니다.

디버그 메시지를 기록하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *`ORACLE_HOME/sso/conf/policy.properties`*를 엽니다(예: OSSO 서버의 `C:\OraHome_1\sso\conf\policy.properties`).

2. `debugLevel` 등록정보의 값을 DEBUG로 설정합니다.

   ```
   debugLevel = DEBUG
   ```

3. `policy.properties`를 저장한 후 닫습니다.

선택사항: 보호된 리소스에 대한 디버깅 메시지 사용

`mod_osso.conf`를 통해 보호되는 리소스의 OSSO 디버깅 메시지를 기록하려면 EPM System 웹 서버의 `httpd.conf`를 수정합니다. 디버깅 메시지는 *EPM_ORACLE_INSTANCE*`/httpConfig/ohs/diagnostics/logs/OHS/ohs_component/ohs_component.log`에 기록됩니다.

보호되는 리소스에 대한 디버깅 메시지를 기록하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*`/httpConfig/ohs/config/OHS/ohs_component/httpd.conf`를 엽니다.
2. `OraLogSeverity` 등록정보의 값을 TRACE로 설정합니다.

   ```
   OraLogSeverity TRACE:32
   ```

3. `httpd.conf`를 저장한 후 닫습니다.

SSO를 위한 EPM System 제품 보호

사용자의 SSO 요청이 보안 에이전트(OAM, OSSO 또는 SiteMinder)로 리디렉션되도록 Oracle Enterprise Performance Management System 리소스를 보호해야 합니다.

Oracle HTTP Server는 `mod_osso`를 사용하여 사용자를 OSSO 서버로 리디렉션합니다. 사용자는 자신이 요청한 URL이 `mod_osso`에서 보호되도록 구성된 경우에만 리디렉션됩니다. *Oracle HTTP Server Administrator's Guide*의 보안 관리를 참조하십시오.

SiteMinder SSO용 리소스를 보호하는 방법에 대한 자세한 내용은 SiteMinder 설명서를 참조하십시오.

OAM만 해당: 기본 머리글이 응답에 추가되지 않도록 합니다.

기본적으로 OAM은 보호되는 URL에 Pragma: no-cache 및 Cache-Control: no-cache 두 개의 헤더를 추가합니다. 이러한 헤더는 EPM System 및 웹 애플리케이션에서 추가한 유사한 캐시 지시어와 충돌하므로 브라우저가 성능 저하의 원인이 되는 보호되는 URL의 콘텐츠를 캐시하지 못할 수 있습니다.

이러한 OAM 헤더가 응답에 추가되지 않도록 하는 방법에 대한 자세한 내용은 *Oracle Security Token Service가 포함된 Oracle Access Manager용 Fusion Middleware 관리자 가이드*의 "Oracle Access Management 성능 조정" 섹션에서 "*OAM 에이전트 조정*"을 참조하십시오.

보호할 리소스

다음 테이블에는 보호 대상 컨텍스트가 나열되어 있습니다. `interop` 등을 사용하여 OSSO를 위해 리소스를 보호하기 위한 구문은 다음과 같습니다.

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

```
order deny,allow
deny from all
allow from myServer.myCompany.com
satisfy any
</Location>
```

`allow from` 매개변수는 컨텍스트 보호를 무시할 수 있는 서버를 지정합니다.

Oracle Hyperion Enterprise Performance Management Workspace 및 Oracle Hyperion Financial Reporting의 경우 다음 예에 표시된 매개변수만 설정해야 합니다.

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

표 3-1 보호할 EPM System 리소스

EPM System 제품	보호할 컨텍스트
Oracle Hyperion Shared Services	• /interop • /interop/.../*
EPM Workspace	• /workspace • /workspace/.../*
Financial Reporting	• /hr • /hr/.../*
Oracle Hyperion Planning	• /HyperionPlanning • /HyperionPlanning/.../*
Oracle Integrated Operational Planning	• /interlace • /interlace/.../*
Oracle Hyperion Financial Management	• /hfmadf • /hfmadfe/.../* • /hfmofficeprovider • /hfmofficeprovider/.../* • /hfmsmartviewprovider • /hfmsmartviewprovider/.../*
Oracle Hyperion Financial Reporting Web Studio	/frdesigner/**
Oracle Data Relationship Management	• /drm-web-client • /drm-web-client/.../*
Oracle Essbase Administration Services	• /hbrlauncher • /hbrlauncher/.../*
Oracle Hyperion Financial Data Quality Management	• /HyperionFDM • /HyperionFDM/.../*
Oracle Hyperion Calculation Manager	• /calcmgr • /calcmgr/.../*
Oracle Hyperion Provider Services	• /aps • /aps/.../*
Oracle Hyperion Profitability and Cost Management	• /profitability • /profitability/.../*

표 3-1 (계속) 보호할 EPM System 리소스

EPM System 제품	보호할 컨텍스트
Account Reconciliation Manager	• /arm • /arm/.../*
Oracle Hyperion Financial Close Management	• /fcc • /fcc/.../*
Oracle Hyperion Financial Data Quality Management, Enterprise Edition	• /aif • /aif/.../*
Oracle Hyperion Tax Governance	/tss
Tax Operations	/taxop
Oracle Hyperion Tax Provision	/taxprov
Supplemental Data Manager	• /sdm* • /sdm/** • /sdm/../** • /SDM-Datamodel-context-root/**
Oracle Essbase	• /essbase/.../* • /essbase/** • /essbase*

보호 해제할 리소스

다음 테이블에는 보호 대상이 아닌 컨텍스트가 나열되어 있습니다. `/interop/framework(.*)` 등을 사용하여 OSSO용 리소스 보호를 해제하기 위한 구문은 다음과 같습니다.

```
<LocationMatch /interop/framework(.*)>
   Require valid-user
   AuthType Basic
   allow from all
   satisfy any
</LocationMatch>
```

표 3-2 EPM System 리소스 보호 해제

EPM System 제품	보호 해제할 컨텍스트
Shared Services	• /interop/framework • /interop/framework* • /interop/framework.* • /interop/framework/.../* • /interop/Audit • /interop/Audit* • /interop/Audit.* • /interop/Audit/.../* • /interop/taskflow • /interop/taskflow* • /interop/taskflow/.../* • /interop/WorkflowEngine • /interop/WorkflowEngine/* • /interop/WorkflowEngine/.../* • /interop/TaskReceiver • /framework/lcm/HSSMigration
EPM Workspace	• /epmstatic/.../* • /workspace/bpmstatic/.../* • /workspace/static/.../* • /workspace/cache/.../*
Planning	• /HyperionPlanning/Smartview • /HyperionPlanning/faces/PlanningCentral • /HyperionPlanning/servlet/HspDataTransfer • /HyperionPlanning/servlet/HspLCMServlet • /HyperionPlanning/servlet/HspADMServlet/.../* • /HyperionPlanning/servlet/HspADMServlet/** • /HyperionPlanning/servlet/HspADMServlet* • /HyperionPlanning/servlet/HspAppManagerServlet/.../* • /HyperionPlanning/servlet/HspAppManagerServlet/** • /HyperionPlanning/servlet/HspAppManagerServlet*
Financial Reporting	• /hr/common/HRLogon.jsp • /hr/services • /hr/services/* • /hr/services/.../* • /hr/modules/com/hyperion/reporting/web/reportViewer/HRStaticReport.jsp • /hr/modules/com/hyperion/reporting/web/repository/HRObjectListXML.jsp • /hr/modules/com/hyperion/reporting/web/reportViewer/HRHtmlReport.jsp • /hr/modules/com/hyperion/reporting/web/bookViewer/HRBookTOCFns.jsp • /hr/modules/com/hyperion/reporting/web/bookViewer/HRBookPdf.jsp
Data Relationship Management	/drm-migration-client

표 3-2 (계속) EPM System 리소스 보호 해제

EPM System 제품	보호 해제할 컨텍스트
Calculation Manager	• /calcmgr/importexport.postExport.do • /calcmgr/common.performAction.do • /calcmgr/lcm.performAction.do* • /calcmgr/lcm.performAction.do/*
Administration Services	• /eas • /easconsole • /easdocs
Financial Management	• /hfm/EIE/EIEListener.asp • /hfmapplicationservice • /oracle-epm-fm-webservices • /hfmlcmservice
Financial Close Management	• /FCC-DataModel-context-root • /oracle-epm-erpi-webservices/* • /ARM-DataModel-context-root • /oracle-epm-erpi-webservices/** • /arm/batch/armbatchexecutionservlet • /ARM-DataModel-context-root
Integrated Operational Planning	• /interlace/services/ • /interlace/services/* • /interlace/services/.* • /interlace/services/.../* • /interlace/anteros • /interlace/anteros/* • /interlace/anteros/.* • /interlace/anteros/.../* • /interlace/interlace • /interlace/interlace/* • /interlace/interlace/.* • /interlace/interlace/.../* • /interlace/WebHelp • /interlace/WebHelp/* • /interlace/WebHelp/.* • /interlace/WebHelp/.../* • /interlace/html • /interlace/html/* • /interlace/html/.* • /interlace/html/.../* • /interlace/email-book • /interlace/email-book/* • /interlace/email-book/.* • /interlace/email-book/.../*
Profitability and Cost Management	• /profitability/cesagent • /profitability/lcm • /profitability/control • /profitability/ApplicationListener • /profitability/HPMApplicationListener

표 3-2 (계속) EPM System 리소스 보호 해제

EPM System 제품	보호 해제할 컨텍스트
Oracle Essbase	• /essbase/agent/.../* • /essbase/jet/logout.html • /essbase/jet/.+\.(js|css|gif|jpe?g|png)$
FDMEE	• /aif/services/FDMRuleService • /aif/services/RuleService • /aif/LCMServlet

ID 관리 제품을 사용하는 머리글 기반 SSO

사전 필수 조건

- 완전히 구성된 Oracle Enterprise Performance Management System. 사용자에게 권한을 부여하려면 EPM System에서 ID 관리 제품의 디렉토리 서버를 사용자 디렉토리로 구성해야 합니다.
- 머리글 기반 인증이 지원되는 완전히 구성된 ID 관리 제품(Microsoft Azure AD, Okta 등).

다음 일반 프로세스는 호환되는 ID 관리 제품을 통해 EPM System에서 머리글 기반 SSO를 지원하도록 구성하는 작업과 관련이 있습니다. 관련된 특정 단계는 사용 중인 제품에 따라 다르므로 자세한 단계는 ID 관리 제품 설명서를 참조하십시오.

Oracle Identity Cloud Services를 통해 머리글 기반 인증을 구성하는 자세한 단계는 Oracle Identity Cloud Services를 통해 EPM System에서 머리글 기반 SSO를 지원하도록 구성을 참조하십시오.

1. ID 관리 제품에서 EPM System을 엔터프라이즈 애플리케이션으로 등록합니다. 이 단계에서는 ID 관리 관리자가 다단계 인증과 같은 지원되는 기능을 포함하여 엔터프라이즈 애플리케이션의 인증을 구성할 수 있습니다.
   `workspace/index.jsp`가 추가된 게이트웨이의 FQDN(정규화된 도메인 이름)(예: `https://gateway.server.example.com:443/workspace/index.jsp`)을 EPM System의 엔터프라이즈 애플리케이션 URL로 사용합니다.

   HTTP 머리글을 전파하도록 EPM System 엔터프라이즈 애플리케이션을 구성합니다. 임의의 예약되지 않은 머리글 이름을 HTTP 머리글 이름으로 선택할 수 있습니다. 머리글 값은 EPM System 사용자를 고유하게 확인하는 등록정보여야 합니다.
2. 엔터프라이즈 애플리케이션에서 인증된 요청만 EPM System으로 전달하도록 애플리케이션 게이트웨이를 설치하고, 구성하고, 등록합니다.
   다음 구성 설정을 사용합니다.
   - 게이트웨이 서버의 FQDN(예: `gateway.server.example.com:443`)을 액세스 지점으로 사용합니다.
   - EPM System의 FQDN(예: `epm.server.example.com:443`)을 인증된 HTTP(S) 요청을 전달해야 하는 리소스로 사용합니다.
3. EPM System에서 SSO를 사용으로 설정하여 애플리케이션 게이트웨이에서 전파한 HTTP(S) 머리글을 사용합니다. 자세한 내용은 보안 옵션 설정를 참조하십시오.
   SSO를 사용으로 설정하려면 다음을 수행합니다.
   a. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행를 참조하십시오.

b. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.

c. **보안 옵션**을 누릅니다.

d. **싱글 사인온 구성** 섹션에서 다음을 수행합니다.

   i. **SSO 사용** 확인란을 선택합니다.

   ii. **SSO 제공자 또는 보안 에이전트** 드롭다운 목록에서 **기타**를 선택합니다.

   iii. **SSO 메커니즘** 드롭다운 목록에서 **사용자정의 HTTP 머리글**을 선택하고 보안 에이전트가 EPM System으로 전달하는 머리글의 이름을 지정합니다.

e. **확인**을 누릅니다.

4. Oracle Hyperion Enterprise Performance Management Workspace의 로그오프 이후 URL 설정을 EPM System에서 로그아웃할 때 사용자에게 표시할 웹 페이지의 URL로 업데이트합니다.
   EPM Workspace의 로그오프 이후 URL 설정을 업데이트하려면 다음을 수행합니다.

   a. EPM Workspace에 시스템 관리자로 액세스합니다. EPM Workspace 액세스을 참조하십시오.

   b. **탐색**, **Workspace 설정**, **서버 설정** 순으로 선택합니다.

   c. **Workspace 서버 설정**에서 **로그오프 이후 URL**을 EPM System에서 로그아웃할 때 사용자에게 표시할 웹 페이지의 URL로 변경합니다.

   d. **확인**을 누릅니다.

5. Oracle Hyperion Foundation Services 및 모든 EPM System 관리 서버를 재시작합니다.

Oracle Identity Cloud Services를 통해 EPM System에서 머리글 기반 SSO를 지원하도록 구성

이 시나리오에서는 Oracle Identity Cloud Services에서 Oracle Enterprise Performance Management System 사용자를 인증하고 필요한 HTTP 머리글을 전파하여 SSO를 사용으로 설정합니다.

이 섹션에서는 Oracle Identity Cloud Services를 통해 SSO를 지원하도록 EPM System을 설정하고 구성하는 단계에 대해 설명합니다. 다음 단계를 도출하여 머리글 기반 인증이 지원되는 ID 관리 시스템(예: Azure AD) 또는 IaaS(Infrastructure as a Service) 제공자로 EPM System의 머리글 기반 인증을 지원할 수 있습니다.

개념 워크플로우는 다음과 같습니다.

- 역방향 프록시 역할을 하는 게이트웨이 애플리케이션은 인증되지 않은 네트워크 액세스를 제한하여 EPM System 구성요소를 보호합니다.
- 게이트웨이 애플리케이션은 EPM System 구성요소에 대한 HTTP(S) 요청을 가로채고 ID 관리 제품에서 사용자를 인증한 후 EPM System 구성요소에 요청을 전달하도록 합니다.
- 요청을 EPM System 구성요소에 전달하는 동안 게이트웨이 애플리케이션은 인증된 사용자의 ID를 HTTP 머리글 요청을 통해 EPM System 구성요소로 전파합니다.

사전 필수 조건 및 샘플 URL

Oracle Identity Cloud Services를 통해 머리글 기반 SSO를 설정하려면 다음이 필요합니다.

- 완전히 구성된 Oracle Enterprise Performance Management System.
- 인증되지 않은 액세스를 제한하여 EPM System을 보호하는 역방향 프록시 역할을 하는 완전히 구성된 Oracle App Gateway가 있는 호스트 또는 컨테이너.
  Oracle App Gateway는 EPM System 구성요소에 대한 HTTP 요청을 가로채고 Oracle Identity Cloud Services에서 사용자를 인증한 후 EPM System에 요청을 전달하도록 구성되어야 합니다. 요청을 EPM System 구성요소에 전달하는 동안 Oracle App Gateway는 인증된 사용자의 ID를 HTTP 머리글 요청을 통해 전파해야 합니다.
- Oracle Identity Cloud Services에 대한 도메인 관리자 액세스.

이 설명에서는 다음 샘플 URL이 사용됩니다.

- Oracle Identity Cloud Services 서버(ID 제공자)의 FQDN(정규화된 도메인 이름) 기본 URL:
  `https://identity.server.example.com:443/`
- Oracle App Gateway 서버(게이트웨이 애플리케이션을 호스트하는 서버)의 FQDN:
  `https://gateway.server.example.com:443/`
- EPM System의 엔터프라이즈 애플리케이션 URL. `workspace/index.jsp`가 추가된 Oracle App Gateway 서버의 FQDN입니다.
  `https://gateway.server.example.com:443/workspace/index.jsp`

> **Note:**
>
> Oracle Identity Cloud Services 및 Oracle App Gateway는 HTTPS 지원을 포함하도록 구성됩니다. EPM System의 경우 HTTPS 지원은 선택사항입니다.
> 이 설명에서는 EPM System이 HTTPS 지원을 포함하도록 구성되었다고 가정합니다.

EPM System에 머리글 기반 인증 사용

Oracle Enterprise Performance Management System에 머리글 기반 인증을 사용하도록 설정하는 단계는 다음과 같습니다.

- Oracle Identity Cloud Services에 EPM System 애플리케이션 및 게이트웨이 추가
- App Gateway 구성
- 권한부여를 위한 사용자 디렉토리 구성
- EPM System에서 SSO 사용
- EPM Workspace 설정 업데이트

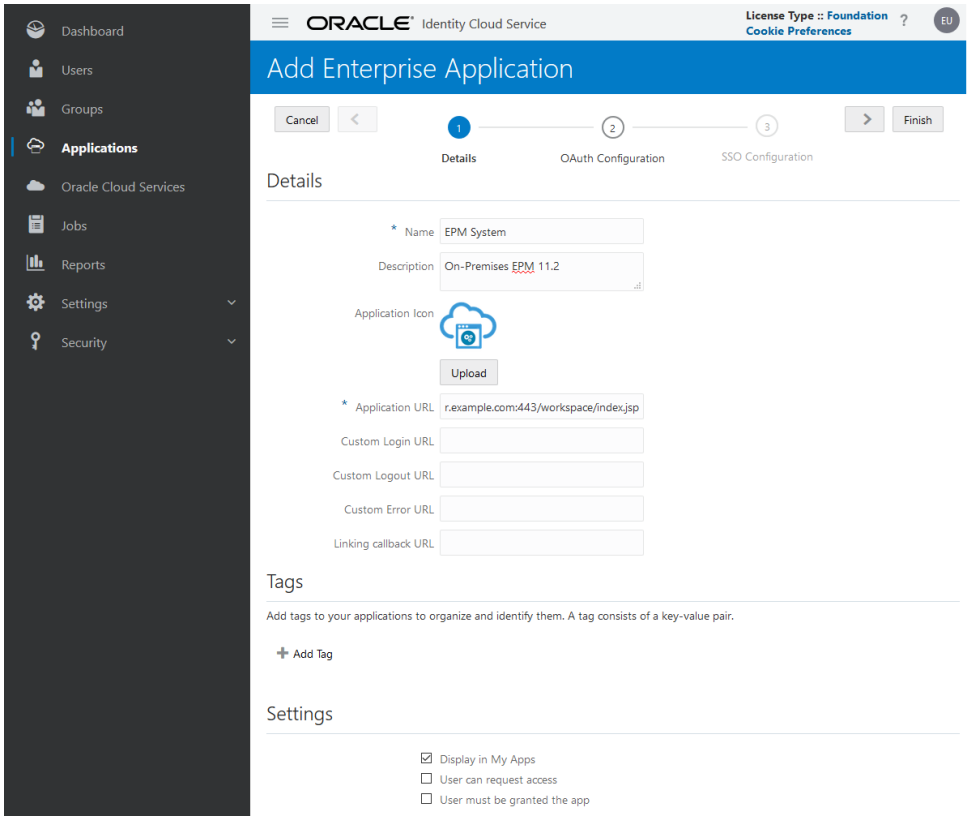
Oracle Identity Cloud Services에 EPM System 애플리케이션 및 게이트웨이 추가

머리글 기반 인증을 설정하려면 Oracle Enterprise Performance Management System을 엔터프라이즈 애플리케이션으로 생성해야 합니다.

Oracle Cloud Identity Console에서 EPM System을 엔터프라이즈 애플리케이션으로 추가

EPM System을 엔터프라이즈 애플리케이션으로 추가하려면 다음을 수행합니다.

1. Oracle Cloud Identity Console에 도메인 관리자로 액세스합니다.
   a. 브라우저를 사용하여 `https://www.oracle.com/cloud/sign-in.html`로 이동합니다.
   b. Oracle Fusion Cloud EPM 계정 이름을 입력합니다.
   c. Oracle Fusion Cloud EPM 계정 로그인 페이지에서 사용자 이름 및 비밀번호를 입력하고 **로그인**을 누릅니다.
   d. **네비게이션 전환기**에서 **사용자**, **ID(기본)** 순으로 누릅니다.
   e. **ID 콘솔**을 누릅니다.
2. EPM System을 엔터프라이즈 애플리케이션으로 추가합니다.
   a. 네비게이션 전환기에서 **애플리케이션**을 누릅니다.
   b. **추가**, **엔터프라이즈 애플리케이션** 순으로 누릅니다.



3. 애플리케이션 세부정보를 추가합니다.
   a. **이름**에 EPM System 엔터프라이즈 애플리케이션을 확인할 수 있는 고유한 이름을 입력합니다.
   b. 선택적 설명을 입력합니다.

c. 선택적으로, EPM System 애플리케이션 아이콘을 업로드합니다. 아이콘을 선택하여 업로드하려면 **업로드**를 누르십시오.

d. **애플리케이션 URL**에는 게이트웨이에서 사용자를 리디렉션할 실행 URL을 입력합니다. 이 URL은 EPM System 애플리케이션 컨텍스트인 `workspace/index.jsp`가 추가된 Oracle App Gateway의 FQDN입니다.

e. **설정**에서 **내 애플리케이션에 표시**를 선택하여 Oracle Cloud Identity Console의 **내 애플리케이션** 페이지 **SSO 구성** 탭에 EPM System 엔터프라이즈 애플리케이션을 표시합니다.

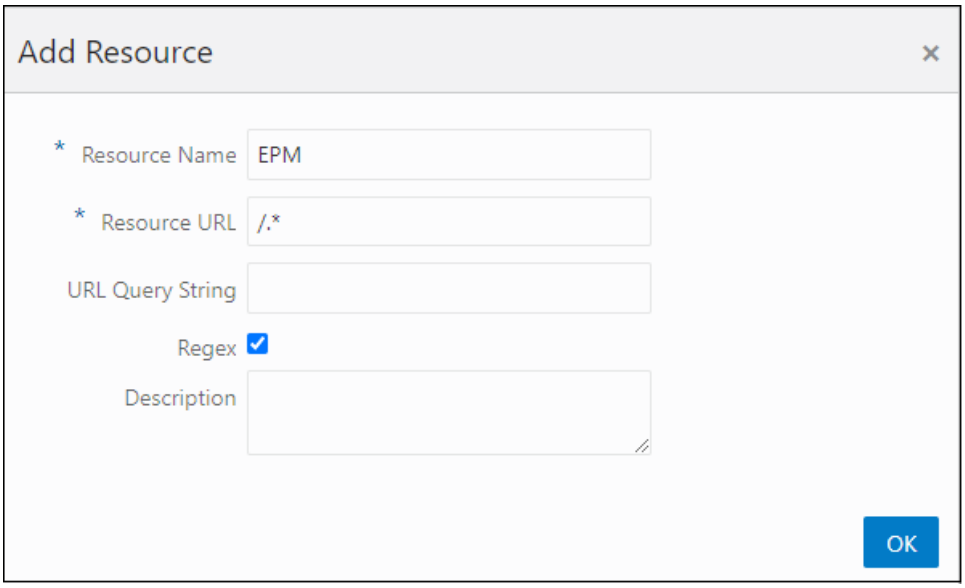
f. **다음**을 누릅니다.

4. SSO 구성 세부정보를 지정합니다.

a. **SSO 구성**을 누릅니다.

b. 엔터프라이즈 애플리케이션 리소스를 추가합니다.
SSO 구성에서 **리소스**를 확장합니다.

i. **추가**를 누릅니다.



ii. 고유한 리소스 이름을 지정합니다.

iii. **리소스 URL**에 `/.*`를 입력합니다.

iv. **Regex** 확인란을 선택합니다.

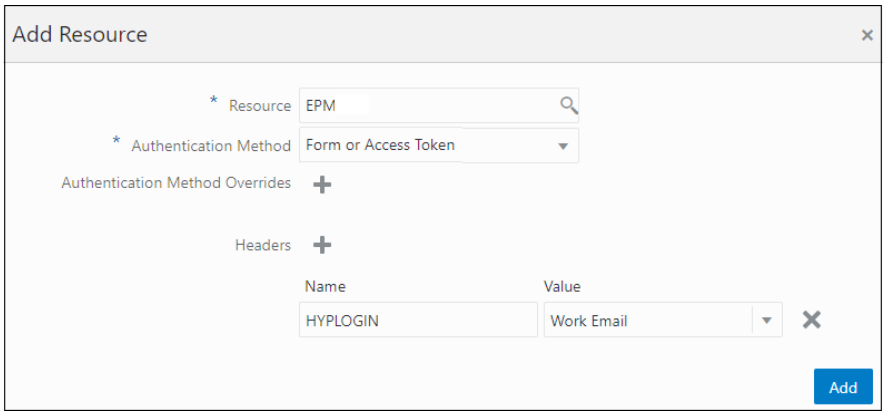
v. **확인**을 누릅니다.

vi. **SSO 구성**에서 **리소스**를 확장합니다.

c. 인증 정책을 추가합니다.
SSO 구성에서 **인증 정책**을 확장합니다.

i. **CORS 허용** 및 **보안 쿠키 필요** 확인란을 선택합니다.

ii. **관리 리소스**에서 **추가**를 누르고 **양식 또는 액세스 토큰**을 SSO 리소스 인증 방법으로 정의합니다.



iii. **리소스**에서는 이전 단계에서 추가한 SSO 리소스를 선택합니다.

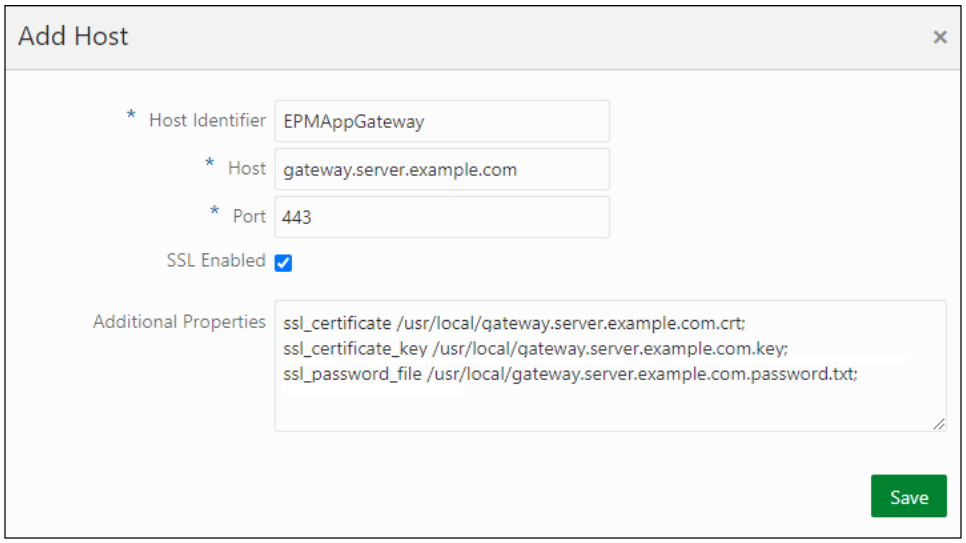
iv. **머리글**을 확장합니다.

v. EPM System으로 전파될 HTTP 머리글 이름을 입력합니다.
기본 인증 머리글 이름은 `HYPLOGIN`입니다. 원하는 대로 선택한 이름을 사용할 수 있습니다.

vi. **값**에서 EPM System 사용자를 고유하게 확인하는 등록정보를 선택합니다.
이 필드의 값은 EPM System의 사용자 ID와 일치해야 합니다. 예를 들어 EPM System의 사용자 ID가 전자메일 ID인 경우 직장 전자메일을 값으로 선택합니다.

vii. **저장**을 누릅니다.

5. **완료**를 눌러 엔터프라이즈 애플리케이션을 생성합니다.
6. **활성화**를 눌러 애플리케이션을 사용으로 설정합니다.
7. App Gateway를 등록하고 EPM System의 호스트 및 애플리케이션을 설정합니다.
   a. **네비게이션 전환기**에서 **보안**, **애플리케이션 게이트웨이** 순으로 누릅니다.
   b. **추가**를 누릅니다.
   c. **세부정보**에 게이트웨이의 고유한 이름 및 설명(선택사항)을 입력합니다.
   d. **다음**을 눌러 호스트 화면을 엽니다.
   e. EPM System의 App Gateway 호스트를 추가합니다.
      i. 호스트 화면에서 **추가**를 누릅니다.



ii. **호스트 식별자**에는 `EPMAppGateway`를 입력합니다.

iii. **호스트**에는 App Gateway 서버를 호스트하는 컴퓨터의 정규화된 도메인 이름(예: `gateway.server.example.com`)을 입력합니다.

iv. **포트**에는 App Gateway 서버가 HTTPS 요청에 응답하는 포트를 입력합니다.

v. **SSL 사용** 확인란을 선택합니다.

vi. **추가 등록정보**에 다음을 입력합니다.

- SSL 인증서 위치
- SSL 인증서 키
- SSL 비밀번호 파일(필요한 경우)

자세한 내용은 *Oracle Identity Cloud Service 관리*의 "App Gateway 설정"에 있는 "App Gateway 등록"을 참조하십시오.

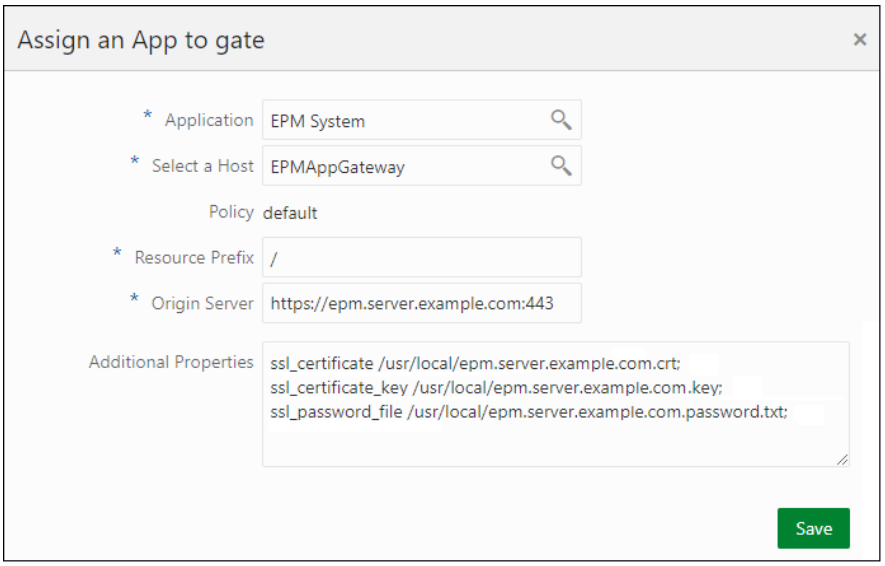
vii. **저장**을 누릅니다.

viii. **다음**을 눌러 애플리케이션 화면을 엽니다.

f. EPM System 엔터프라이즈 애플리케이션을 App Gateway에 추가합니다.

i. **애플리케이션**에서 **추가**를 누릅니다.

ii. **애플리케이션**에서 이전에 Oracle Cloud Identity Console에 추가한 EPM System 엔터프라이즈 애플리케이션을 선택합니다.



iii. **호스트 선택**에서 `EPMAppGateway`(App Gateway에 추가한 EPM System 호스트)를 선택합니다.

iv. **리소스 접두어**에서 `/`를 입력하여 EPM System 호스트로 모든 요청을 전달합니다.

v. **원래 서버**에서 Oracle Hyperion Enterprise Performance Management Workspace를 호스트하는 컴퓨터의 정규화된 도메인 이름과 EPM Workspace에서 사용하는 포트 번호를 입력합니다.

vi. **저장**을 누릅니다.

8. App Gateway의 클라이언트 ID 및 클라이언트 비밀을 기록합니다. 이러한 값은 App Gateway를 설정하는 데 필수입니다.

   a. **네비게이션 전환기**에서 **보안**, **애플리케이션 게이트웨이** 순으로 누릅니다.

   b. EPM System 엔터프라이즈 애플리케이션용으로 추가한 게이트웨이의 이름을 누릅니다.

   c. 클라이언트 ID(영숫자 문자열)를 텍스트 편집기로 복사합니다.

   d. **비밀 표시**를 눌러 클라이언트 비밀 코드를 표시합니다.

   e. 클라이언트 비밀(영숫자 문자열)을 텍스트 편집기로 복사합니다.

   f. 텍스트 파일을 저장합니다.

> **Note:**
>
> Oracle Identity Cloud Services에 구성을 업데이트할 때마다 앱 게이트웨이 서버를 다시 시작해야 합니다. 앱 게이트웨이 서버를 시작 및 중지하려면 앱 게이트웨이 시작 및 중지를 참조하십시오.

App Gateway 구성

자세한 내용은 *Oracle Identity Cloud Service 관리*의 "App Gateway 설정"을 참조하십시오.

App Gateway 서버 구성에는 이전 섹션에서 기록한 클라이언트 ID 및 클라이언트 비밀이 필요합니다.

권한부여를 위한 사용자 디렉토리 구성

일부 ID 관리 제품(예: Oracle Identity Cloud Services 및 Microsoft Azure)은 Oracle Enterprise Performance Management System에서 직접 사용자 디렉토리로 구성할 수 없습니다. 해당 제품을 Oracle Unified Directory 또는 Oracle Virtual Directory로 구성한 후 나중에 EPM System에서 사용자 디렉토리로 구성할 수 있습니다. 사용자 디렉토리 구성에 대한 자세한 단계는 사용자 디렉토리 구성을 참조하십시오.

EPM System에서 SSO 사용

SSO를 사용하려면 Oracle Enterprise Performance Management System에서 보안 옵션을 구성합니다. 자세한 지침은 보안 옵션 설정을 참조하십시오.

SSO를 사용으로 설정하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. **보안 옵션**을 누릅니다.
4. **싱글 사인온 구성** 섹션에서 다음을 수행합니다.
   a. **SSO 사용** 확인란을 선택합니다.
   b. **SSO 제공자 또는 보안 에이전트** 드롭다운 목록에서 **기타**를 선택합니다.
   c. **SSO 메커니즘** 드롭다운 목록에서 **사용자정의 HTTP 머리글**을 선택하고 보안 에이전트가 EPM System으로 전달하는 머리글의 이름(`HYPLOGIN` 또는 Oracle Cloud Identity Console에서 엔터프라이즈 애플리케이션 리소스를 추가할 때 지정한 사용자정의 이름)을 지정합니다.
5. **확인**을 누릅니다.

> **Note:**
>
> SSO 구성 변경 후 모든 EPM System 서비스를 다시 시작해야 합니다.

EPM Workspace 설정 업데이트

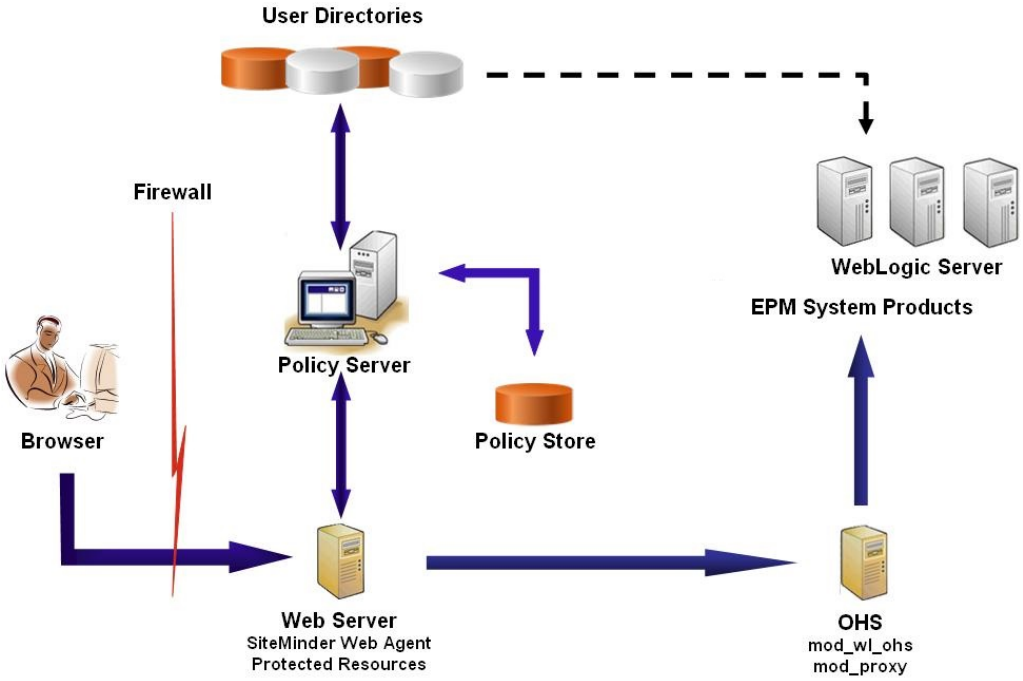
1. Oracle Hyperion Enterprise Performance Management Workspace에 시스템 관리자로 액세스합니다. EPM Workspace 액세스을 참조하십시오.
2. **탐색**, **Workspace 설정**, **서버 설정** 순으로 선택합니다.
3. **Workspace 서버 설정**에서 **로그오프 이후 URL**을 Oracle Enterprise Performance Management System에서 로그아웃할 때 사용자에게 표시할 웹 페이지의 URL로 변경합니다.
4. **확인**을 누릅니다.
5. Oracle Hyperion Foundation Services 및 모든 EPM System 구성요소를 재시작합니다.

SiteMinder SSO

SiteMinder는 웹 전용 솔루션입니다. 데스크탑 애플리케이션과 그 추가 기능(예: Microsoft Excel, Report Designer)은 SiteMinder를 통해서 인증을 사용할 수 없습니다. 하지만 Oracle Smart View for Office는 SiteMinder 인증을 사용할 수 있습니다.

프로세스 플로우

SiteMinder 사용 SSO에 대한 그림 개요:



SiteMinder SSO 프로세스:

1. 사용자가 SiteMinder 보호 Oracle Enterprise Performance Management System 리소스에 액세스하려고 시도합니다. SiteMinder 정책 서버의 프런트엔드에 사용되는 웹 서버에 연결하는 URL(예: `http://WebAgent_Web_Server_Name:WebAgent_Web_ServerPort/interop/index.jsp`)을 사용합니다.
2. 웹 서버는 사용자에게 인증서를 요청하는 정책 서버로 사용자를 리디렉션합니다. 구성된 사용자 디렉토리에 대해 인증서를 확인한 후에는 정책 서버가 SiteMinder 웹 에이전트를 호스트하는 웹 서버로 해당 인증서를 전달합니다.
3. SiteMinder 웹 에이전트를 호스트하는 웹 서버는 EPM System의 프런트엔드로 사용되는 Oracle HTTP Server로 이 요청을 리디렉션합니다. Oracle HTTP Server는 Oracle WebLogic Server에 배포되어 있는 요청된 애플리케이션으로 사용자를 리디렉션합니다.
4. EPM System 구성요소는 프로비저닝 정보를 확인하고 콘텐츠를 제공합니다. 이 프로세스가 작동하려면 SiteMinder가 사용자를 인증하는 데 사용하는 사용자 디렉토리를 EPM System의 외부 사용자 디렉토리로 구성해야 합니다. 이러한 디렉토리는 인증된 디렉토리로 구성되어야 합니다.

특수 고려 사항

SiteMinder는 웹 전용 솔루션입니다. 데스크탑 애플리케이션과 그 추가 기능(예: Microsoft Excel, Report Designer)은 SiteMinder를 통해서 인증을 사용할 수 없습니다. 하지만 Smart View는 SiteMinder 인증을 사용할 수 있습니다.

사전 필수 조건

1. 다음 구성요소로 이루어진 완전한 기능을 갖춘 SiteMinder 설치.
   - 정책 및 에이전트 객체가 정의된 SiteMinder 정책 서버
   - SiteMinder 정책 서버의 프런트엔드로 사용되는 웹 서버에 설치된 SiteMinder 웹 에이전트
2. 완전한 기능을 갖춘 EPM System 배포
   EPM System 구성요소에 대해 웹 서버를 구성하는 경우 EPM System Configurator는 WebLogic Server에 대한 요청을 프록시하도록 `mod_wl_ohs.conf`를 구성합니다.

SiteMinder 웹 에이전트 사용

EPM System 리소스에 대한 요청을 가로채는 웹 서버에는 웹 에이전트가 설치되어 있습니다. 인증되지 않은 사용자가 보호되는 EPM System 리소스에 액세스하려고 시도하면 웹 에이전트가 사용자에게 SSO 인증서를 요청합니다. 사용자가 인증되면 정책 서버에서 머리글이 가져오는 인증된 사용자의 로그인 이름을 추가합니다. 따라서 요청을 리디렉션하는 EPM System 웹 서버로 HTTP 요청이 전달됩니다. EPM System 구성요소는 머리글에서 인증된 사용자 인증서를 추출합니다.

SiteMinder는 이기종 웹 서버 플랫폼에서 실행되는 EPM System 제품 전체에서 SSO를 지원합니다. EPM System 제품이 여러 웹 서버를 사용하는 경우 동일한 도메인 내의 웹 서버 간에 SiteMinder 쿠키가 전달될 수 있도록 해야 합니다. 해당 EPM System 애플리케이션 도메인을 각 웹 서버의 `WebAgent.conf` 파일에 있는 `Cookiedomain` 등록정보 값으로 지정하여 이 작업을 수행합니다.

*Netegrity SiteMinder 에이전트 가이드*의 "웹 에이전트 구성"을 참조하십시오.

> **주:**
>
> Oracle Hyperion Shared Services에서 기본 인증을 사용하여 콘텐츠를 보호하기 때문에 Shared Services에 대한 요청을 가로채는 웹 서버는 SiteMinder의 SSO를 지원하도록 기본 인증을 사용으로 설정해야 합니다.

웹 에이전트는 SiteMinder 웹 에이전트 구성 마법사를 실행하여 구성합니다(*`WEBAGENT_HOME`*`/install_config_info/nete-wa-config`, 예를 들어 Windows의 경우 `C:\netegrity\webagent\install_config_info\nete-wa-config.exe`를 실행함). 구성 프로세스는 SiteMinder 웹 서버에 대해 `WebAgent.conf`를 생성합니다.

SiteMinder 웹 에이전트를 사용으로 설정하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 `WebAgent.conf`를 엽니다. 이 파일의 위치는 사용하는 웹 서버에 따라 다릅니다.
2. `enableWebAgent` 등록정보의 값을 `Yes`로 설정합니다.

   ```
   enableWebAgent="YES"
   ```

3. 웹 에이전트 구성 파일을 저장하고 닫습니다.

예 3-1 SiteMinder Policy Server 구성

SiteMinder 관리자는 정책 서버를 구성하여 EPM System 제품에 SSO를 활성화해야 합니다.

구성 프로세스는 다음과 같습니다.

- SiteMinder 웹 에이전트를 생성하고 SiteMinder 웹 서버에 적합한 구성 객체를 추가합니다.
- 보호되어야 하는 각 EPM System 리소스에 대한 영역을 생성하고 해당 영역에 웹 에이전트를 추가합니다. 보호할 리소스를 참조하십시오.
- 보호되는 EPM System 리소스에 대해 생성된 영역 내에 보호 해제되는 리소스에 대한 영역을 생성합니다. 보호 해제할 리소스를 참조하십시오.
- HTTP 머리글 참조를 생성합니다. 머리글은 `Login Attribute` 값을 EPM System 애플리케이션에 제공해야 합니다. `Login Attribute`에 관한 간단한 설명은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.
- Get, Post 및 Put을 사용하여 영역 내에 웹 에이전트 작업으로 규칙을 생성합니다.
- 값이 `hyplogin=<%userattr="SM_USERLOGINNAME"%>`인 응답 속성을 생성합니다.
- 정책을 생성하고, 사용자 디렉토리 액세스 권한을 지정하고, EPM System에 대해 생성한 규칙을 현재 멤버 목록에 추가합니다.
- EPM System 구성요소에 대해 생성한 규칙의 응답을 설정합니다.

예 3-2 SiteMinder 웹 서버가 EPM System 웹 서버로 요청을 전달하도록 구성

인증된 사용자(사용자를 확인할 수 있는 머리글 포함)의 요청을 EPM System 웹 서버로 전달하는 SiteMinder 웹 에이전트가 호스트된 웹 서버를 구성하십시오.

Apache 기반 웹 서버의 경우 다음과 유사한 지시어를 사용하여 인증된 요청을 전달합니다.

```
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
```

이 지시어에서 *EPM_WEB_SERVER* 및 *EPM_WEB_SERVER_PORT*를 사용자 환경의 실제 값으로 바꾸십시오.

예 3-3 EPM System의 SiteMinder 활성화

SiteMinder 통합에서는 EPM System 제품에 대해 SiteMinder 인증을 활성화해야 합니다. SSO에 대해 EPM System 구성을 참조하십시오.

Kerberos 싱글 사인온

개요

EPM System 제품의 호스트인 애플리케이션 서버가 Kerberos 인증에 대해 설정된 경우 Oracle Enterprise Performance Management System 제품에서 Kerberos SSO를 지원합니다.

Kerberos는 각 Kerberos 클라이언트가 다른 Kerberos 클라이언트(사용자, 네트워크 서비스 등)의 ID를 유효한 것으로 신뢰하는 인증된 인증 서비스입니다.

사용자가 EPM System 제품에 액세스하는 경우 다음을 수행합니다.

1. Windows 컴퓨터에서는 사용자가 Kerberos 영역이기도 한 Windows 도메인에 로그인합니다.
2. Windows 통합 인증을 사용하도록 구성된 브라우저를 사용하여 사용자는 애플리케이션 서버에서 실행 중인 EPM System 제품에 로그인을 시도합니다.
3. 애플리케이션 서버(Negotiate Identity Asserter)는 요청을 가로채고 브라우저의 권한부여 머리글에서 Kerberos 티켓이 포함된 SPNEGO(Simple and Protected Generic Security Services API(GSSAPI) Negotiation Mechanism) 토큰을 가져옵니다.
4. Asserter는 해당 ID 저장소에 대해 토큰에 포함된 사용자 ID를 검증하여 사용자에 대한 정보를 EPM System 제품으로 전달합니다. EPM System 제품은 Active Directory에 대해 사용자 이름을 검증합니다. EPM System 제품은 모든 EPM System 제품에서 SSO를 지원하는 SSO 토큰을 발행합니다.

지원 제한

Kerberos SSO는 모든 EPM System 제품에서 지원되나, 다음은 예외입니다:

- Oracle Smart View for Office 이외의 Thick Client에 대해서는 Kerberos SSO가 지원되지 않습니다.
- Smart View는 Oracle Essbase, Oracle Hyperion Planning 및 Oracle Hyperion Financial Management 제공자에 대해서만 Kerberos 통합을 지원합니다.

가정

이 문서는 애플리케이션 레벨 Kerberos 구성 단계를 포함하고 있으며, 사용자가 시스템 레벨의 Kerberos 구성에 대한 지식이 있다고 가정합니다. 이러한 절차를 시작하기 전에 해당 태스크에 대한 사전 필수 조건이 충족되었는지 확인하십시오.

이 문서에서는 Windows 클라이언트 머신이 Kerberos 인증에 대해 구성되어 있는, 완전한 기능을 갖춘 Kerberos 사용 네트워크 환경에서 작업하고 있다고 가정합니다.

- 회사 Active Directory가 Kerberos 인증에 대해 구성되어 있습니다. Microsoft Windows Server 설명서을 참조하십시오.
- EPM System 제품에 액세스하는 데 사용되는 브라우저는 Kerberos 티켓을 사용하여 조정하도록 구성되어 있습니다.
- KDC와 클라이언트 머신 간에 존재하는 5분 이하의 시간 차이는 동기화됩니다. http://technet.microsoft.com/en-us/library/cc780011(WS.10).aspx에서 "Authentication Errors are Caused by Unsynchronized Clocks"(동기화되지 않은 시계로 인해 인증 오류가 발생함)를 참조하십시오.

Kerberos SSO와 WebLogic Server

Oracle WebLogic Server Kerberos SSO는 Negotiate Identity Asserter를 사용하여 SPNEGO 토큰을 조정하고 디코딩한 후 Microsoft 클라이언트에서 SSO를 사용할 수 있도록 설정합니다. WebLogic Server는 SPNEGO 토큰을 디코딩하여 Kerberos 티켓을 얻고 이 티켓을 검증하여 WebLogic Server 사용자에게 매핑합니다. WebLogic Server의 Active Directory 인증자를 Negotiate Identity Asserter와 함께 사용하여 Active Directory를 WebLogic Server 사용자의 사용자 디렉토리로 구성할 수 있습니다.

브라우저가 EPM System 제품에 대한 액세스를 요청하면 KDC는 브라우저에 Kerberos 티켓을 발급하고, 이것으로 지원되는 GSS 토큰 유형이 들어 있는 SPNEGO 토큰이 생성됩니다. Negotiate Identity Asserter는 SPNEGO 토큰을 디코딩하고 GSSAPI를 사용하여 보안

컨텍스트를 수락합니다. 요청을 시작한 사용자의 ID는 사용자 이름에 매핑되어 WebLogic Server로 다시 전달됩니다. 또한, WebLogic Server는 사용자가 속하는 그룹을 결정합니다. 이 단계에서 요청된 EPM System 제품이 사용자에게 사용 가능한 상태로 됩니다.

> **주:**
>
> 사용자는 SPNEGO(예: Internet Explorer 또는 Firefox)를 지원하는 브라우저를 사용하여 WebLogic Server에서 실행되는 EPM System 제품에 액세스해야 합니다.

EPM System 제품 인증 프로세스는 인증 프로세스에서 파생된 사용자 ID를 사용하여 프로비저닝 데이터를 확인합니다. EPM System 제품에 대한 액세스는 프로비저닝 데이터에 따라 제한됩니다.

WebLogic Server에서 Kerberos 인증을 지원하기 위한 절차

Kerberos 인증을 지원하려면 관리자가 다음 태스크를 완료해야 합니다.

- EPM System의 WebLogic 도메인을 생성합니다. EPM System의 WebLogic 도메인 생성를 참조하십시오.
- 인증 제공자를 생성합니다. WebLogic Server에서 LDAP 인증 제공자 생성을 참조하십시오.
- Negotiate Identity Asserter를 생성합니다. Negotiate Identity Asserter 생성을 참조하십시오.
- Kerberos 식별을 생성합니다. WebLogic Server에 대한 Kerberos 식별 생성를 참조하십시오.
- Kerberos의 JVM 옵션을 업데이트합니다. Kerberos의 JVM 옵션 업데이트를 참조하십시오.
- 권한부여 정책을 구성합니다. 권한부여 정책 구성를 참조하십시오.
- SSODiag를 배포하고 이를 사용하여 WebLogic Server에서 EPM System에 대해 Kerberos SSO를 지원할 준비가 되었는지 확인합니다. SSODiag를 사용하여 Kerberos 환경 테스트를 참조하십시오.

EPM System의 WebLogic 도메인 생성

일반적으로 EPM System 구성요소는 `EPMSystem` WebLogic 도메인(기본 위치는 *`MIDDLEWARE_HOME`*`/user_projects/domains/EPMSystem`임)에 배포됩니다.

Kerberos 인증에 대해 EPM System WebLogic 도메인을 구성하려면 다음을 수행합니다.

1. EPM System 구성요소를 설치합니다.
2. Oracle Hyperion Foundation Services만 배포합니다.
   Foundation Services 배포에서는 기본 EPM System WebLogic 도메인을 생성합니다.
3. Oracle Hyperion Shared Services Console에 로그인하여 Foundation Services가 성공적으로 배포되었는지 확인합니다. Shared Services Console 실행을 참조하십시오.

WebLogic Server에서 LDAP 인증 제공자 생성

WebLogic Server 관리자는 외부 LDAP 서버에 사용자 및 그룹 정보를 저장하는 LDAP 인증 제공자를 생성합니다. LDAP v2 또는 v3 규격 LDAP 서버는 WebLogic Server와 호환됩니다. 다음 항목을 참조하십시오.

- *Oracle Fusion Middleware Securing Oracle WebLogic Server* 가이드의 LDAP 인증 제공자 구성
- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*의 인증 및 Identity Assertion 제공자 구성

Negotiate Identity Asserter 생성

Negotiate Identity Assertion 제공자는 Microsoft 클라이언트에서 SSO를 사용으로 설정합니다. SPNEGO 토큰을 디코딩하여 Kerberos 토큰을 가져오고, Kerberos 토큰을 검증하고, WebLogic 사용자에게 토큰을 매핑합니다. WebLogic 보안 프레임워크에 정의된 SSPI(Security Service Provider Interface)의 구현인 Negotiate Identity Assertion 제공자는 클라이언트의 SPNEGO 토큰을 기반으로 클라이언트를 인증하는 데 필요한 논리를 제공합니다.

- *Oracle Fusion Middleware Securing Oracle WebLogic Server* 가이드의 Negotiate Identity Assertion 제공자 구성
- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*의 인증 및 Identity Assertion 제공자 구성

Negotiate Identity Assertion 제공자를 생성하는 경우 모든 인증에 대해 [JAAS 제어 플래그] 옵션을 `SUFFICIENT`로 설정합니다. Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help의 "JAAS 제어 플래그 설정"을 참조하십시오.

WebLogic Server에 대한 Kerberos 식별 생성

Active Directory 도메인 컨트롤러 머신에서 WebLogic Server 및 EPM System 웹 서버를 나타내는 사용자 객체를 생성하고 Kerberos 영역에서 WebLogic Server 및 웹 서버를 나타내는 SPN(서비스 사용자 이름)에 매핑합니다. 클라이언트는 SPN이 없는 서비스를 찾을 수 없습니다. SPN은 WebLogic Server 도메인으로 복사되는 keytab 파일에 저장되어 로그인 프로세스에 사용됩니다.

자세한 절차는 *Oracle Fusion Middleware Securing Oracle WebLogic Server* 가이드의 WebLogic Server에 대한 식별 생성을 참조하십시오.

WebLogic Server에 대한 Kerberos 식별을 생성하려면 다음을 수행합니다.

1. Active Directory 도메인 컨트롤러 시스템에서 사용자 계정을 생성합니다(예: WebLogic Server 도메인을 호스트하는 컴퓨터의 경우 `epmHost`).

> **주:**
>
> 머신이 아닌 사용자 객체로 ID를 생성합니다.
> 컴퓨터에 간단한 이름을 사용합니다. 예를 들어 호스트 이름이 `epmHost.example.com`으로 지정된 경우 `epmHost`를 사용합니다.
>
> 사용자 객체를 생성하는 경우 사용하는 비밀번호를 기록합니다. SPN을 생성하는 데 필요합니다.
>
> 비밀번호 옵션, 특히 `User must change password at next logon` 옵션을 선택하지 마십시오.

2. Kerberos 프로토콜을 준수하도록 사용자 객체를 수정합니다. 계정은 Kerberos 사전 인증을 요구해야 합니다.
   - **계정** 탭에서 사용할 암호화를 선택합니다.
   - 다른 계정 옵션(특히 `Do not require Kerberos pre-authentication`)은 선택되지 않아야 합니다.
   - 암호화 유형을 설정하면 객체의 비밀번호가 손상될 수 있으므로 객체를 생성할 때 설정하는 비밀번호로 해당 비밀번호를 재설정합니다.
3. Active Directory 도메인 컨트롤러를 호스트하는 컴퓨터에서 명령 프롬프트 창을 열고 Active Directory 지원 툴이 설치된 디렉토리로 이동합니다.
4. 필요한 SPN을 생성하고 구성합니다.
   a. 다음과 유사한 명령을 사용하여 이 절차의 1단계에서 생성한 사용자 객체(`epmHost`)와 SPN이 연계되어 있는지 확인합니다.

      ```
      setspn -L epmHost
      ```

   b. 다음과 같은 명령을 사용하여 AD DS(Active Directory 도메인 서비스)의 WebLogic Server에 대해 SPN을 구성하고 공유 보안 키가 포함된 keytab 파일을 생성합니다.

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

5. WebLogic Server를 호스트하는 컴퓨터에서 keytab 파일을 생성합니다.
   a. 명령 프롬프트를 엽니다.
   b. *MIDDLEWARE_HOME*`/jdk/bin`으로 이동합니다.
   c. 다음과 같은 명령을 실행합니다.

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

   d. 비밀번호를 입력하라는 메시지가 표시되면 이 절차의 1단계에서 사용자를 생성할 때 설정한 비밀번호를 입력합니다.
6. keytab 파일을 WebLogic 도메인 내 시작 디렉토리(예: `C:\Oracle\Middleware\user_projects\domains\EPMSystem`)에 복사합니다.

7. Kerberos 인증이 올바르게 작동하는지 확인합니다.

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

이 명령에서 `account-name`은 Kerberos 사용자(예: `HTTP/epmHost.example.com@EXAMPLE.COM`)를 나타냅니다. 이 명령의 출력은 다음과 유사해야 합니다.

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

Kerberos의 JVM 옵션 업데이트

*Oracle Fusion Middleware Securing Oracle WebLogic Server 11g Release 1 (10.3.1)*의 WebLogic Server에서 Kerberos 인증에 시작 인수 사용 및 JAAS 로그인 파일 생성을 참조하십시오.

EPM System 관리 서버가 Windows 서비스로 실행되는 경우 Windows 레지스트리를 업데이트하여 JVM 시작 옵션을 설정하십시오.

Windows 레지스트리에서 JVM 시작 옵션을 업데이트하려면 다음을 수행합니다.

1. Windows 레지스트리 편집기를 엽니다.
2. **내 컴퓨터**, **HKEY_LOCAL_MACHINE**, **Software**, **Hyperion Solutions**, **Foundationservices0**, **HyS9EPMServer_epmsystem1** 순으로 선택합니다.
3. 다음 문자열 값을 생성합니다.

> **주:**
>
> 다음 테이블에 나열된 이름은 예시입니다.

표 3-3 Kerberos 인증의 JVM 시작 옵션

이름	유형	데이터
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. 추가된 JVMOptions(현재 10진수 값에 4 추가)가 반영되도록 JVMOptionCount DWord의 값을 업데이트합니다.

권한부여 정책 구성

EPM System에 액세스하는 Active Directory 사용자에 대한 권한부여 정책 구성 정보는 *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server* 가이드의 웹 애플리케이션 및 EJB 리소스 보안 옵션을 참조하십시오.

샘플 정책 구성 단계는 SSODiag에 대한 정책 생성을 참조하십시오.

SSODiag를 사용하여 Kerberos 환경 테스트

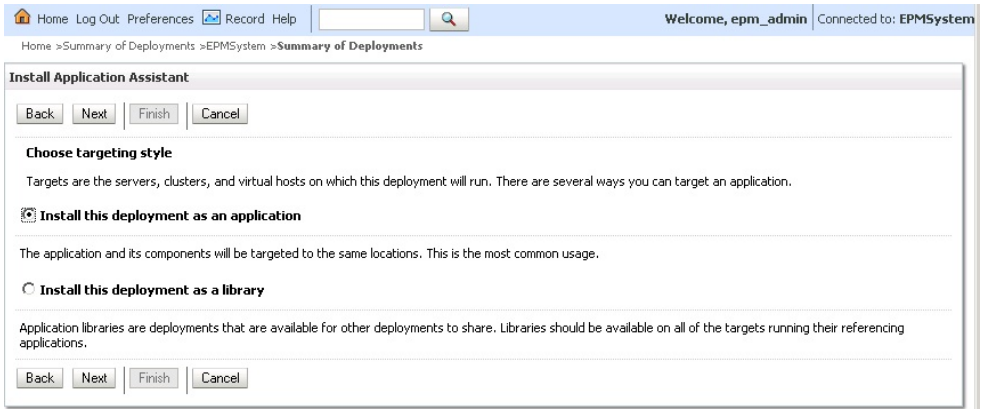
SSODiag는 Kerberos 환경의 WebLogic Server가 EPM System을 지원할 준비가 되어 있는지 테스트하는 진단 웹 애플리케이션입니다.

SSODiag 배포

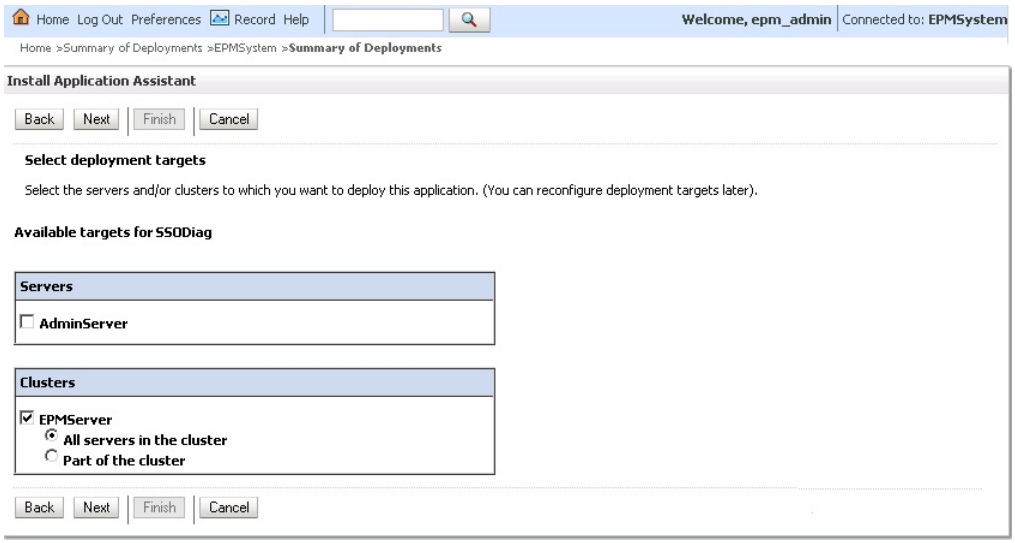
SSODiag를 배포하려면 Foundation Services 배포 중 지정한 WebLogic Server 관리자 인증서(기본 사용자 이름은 `epm_admin`)를 사용합니다.

SSOdiag를 배포하고 구성하려면 다음을 수행합니다.

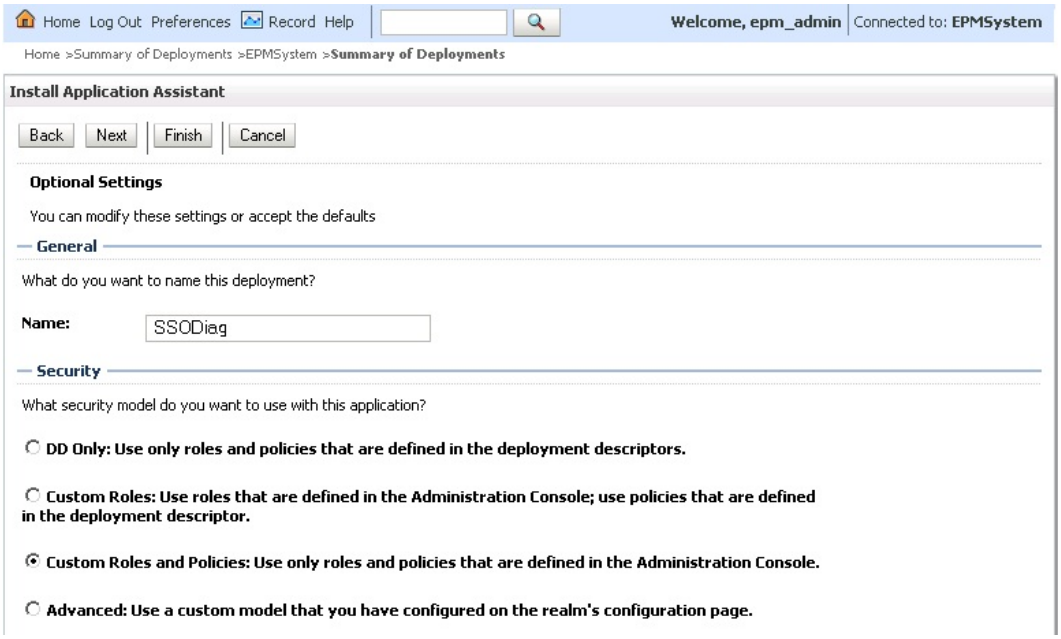
1. EPM System 도메인의 WebLogic Server 관리 콘솔에 로그온합니다.
2. 변경 센터에서 **잠금 및 편집**을 선택합니다.
3. **도메인 구조**의 **EPMSystem**에서 **배포**를 누릅니다.
4. **배포 요약**에서 **설치**를 누릅니다.
5. **경로**에서 *EPM_ORACLE_HOME*`/products/Foundation/AppServer/InstallableApps/common/SSODiag.war`를 선택합니다.
6. **다음**을 누릅니다.
7. **타겟 지정 스타일 선택**에서 **이 배포를 애플리케이션으로 설치**가 선택되어 있는지 확인한 후 **다음**을 누릅니다.



8. **배포 타겟 선택**에서 다음 항목을 선택하고 **다음**을 누릅니다.
   - **EPMServer**
   - **클러스터의 모든 서버**



9. **선택적 설정**에서 **사용자정의 역할 및 정책: 관리 콘솔에 정의된 역할 및 정책만 사용**을 보안 모델로 선택합니다.



10. **다음**을 누릅니다.
11. 검토 화면에서 **아니요, 나중에 구성을 검토하겠습니다.**를 선택합니다.
12. **완료**를 누릅니다.
13. 변경 센터에서 **변경 활성화**를 선택합니다.

SSODiag에 대해 Oracle HTTP Server 구성

`mod_wl_ohs.conf`를 업데이트하여 Oracle HTTP Server에서 SSODiag URL 요청을 WebLogic Server에 전달하도록 구성합니다.

Oracle HTTP Server에서 URL 전달을 구성하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *EPM_ORACLE_INSTANCE*/httpConfig/ohs/config/fmwconfig/components/OHS/ohs_component/mod_wl_ohs.conf를 엽니다.
2. SSODiag에 대한 LocationMatch 정의를 추가합니다.

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

   앞의 샘플에서 myServer는 Foundation Services 호스트 머신을 나타내고 28080은 Oracle Hyperion Shared Services가 요청을 수신 대기하는 포트를 나타냅니다.
3. mod_wl_ohs.conf를 저장하고 닫습니다.
4. Oracle HTTP Server를 재시작합니다.

SSODiag에 대한 정책 생성

WebLogic Server 관리 콘솔에서 다음 SSODiag URL을 보호하는 정책을 생성하십시오.

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

이 샘플에서 *OHS_HOST_NAME*은 Oracle HTTP Server를 호스트하는 서버의 이름을 나타내고 *PORT*는 Oracle HTTP Server가 요청을 수신 대기하는 포트를 나타냅니다.

SSODiag를 보호하는 정책을 생성하려면 다음을 수행합니다.

1. EPM System 도메인의 WebLogic Server 관리 콘솔에 있는 변경 센터에서 **잠금 및 편집**을 선택합니다.
2. **배포**, **SSODiag**, **보안**, **URLPatterns**, **정책** 순으로 선택합니다.
3. 다음 URL 패턴을 생성합니다.
   - /
   - /index.jsp
4. 생성한 각 URL 패턴을 수정합니다.
   a. **독립형 웹 애플리케이션 URL 패턴**의 URL 패턴 목록에서 생성한 패턴(/)을 눌러 해당 패턴을 엽니다.
   b. **조건 추가**를 선택합니다.
   c. **술어 목록**에서 **사용자**를 선택합니다.
   d. **다음**을 선택합니다.
   e. **사용자 인수 이름**에는 Kerberos 인증에 대해 구성된 클라이언트 데스크탑에 액세스하는 데 계정이 사용되는 Active Directory 사용자를 입력하고(예: krbuser1) **추가**를 선택합니다. krbuser1은 Active Directory 또는 Windows 데스크탑 사용자입니다.
   f. **완료**를 선택합니다.
5. **저장**을 선택합니다.

SSODiag를 사용하여 Kerberos 인증에 대한 WebLogic Server 구성 테스트

Kerberos 인증에 대한 WebLogic Server 구성이 올바르게 작동하는 경우 *Oracle Hyperion Kerberos SSO 진단 유틸리티 V 1.0* 페이지에 다음 메시지가 표시됩니다.

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

> **주의:**
>
> SSODiag에서 Kerberos 사용자 이름을 검색할 수 없는 경우 Kerberos 인증에 대한 EPM System 구성요소를 구성하지 마십시오.

Kerberos 인증에 대한 WebLogic Server 구성을 테스트하려면 다음을 수행합니다.

1. Foundation Services 및 Oracle HTTP Server를 시작합니다.
2. WebLogic Server 관리 콘솔을 사용하여 모든 요청을 처리할 SSODiag 웹 애플리케이션을 시작합니다.
3. 적합한 Active Directory 인증서를 사용하여 Kerberos 인증에 대해 구성된 클라이언트 머신에 로그온합니다.
4. 브라우저를 사용하여 다음 SSODiag URL에 연결합니다.

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

   이 샘플에서 *OHS_HOST_NAME*은 Oracle HTTP Server를 호스트하는 서버의 이름을 나타내고 *PORT*는 Oracle HTTP Server가 요청을 수신 대기하는 포트를 나타냅니다.

   Kerberos 인증이 제대로 작동하는 경우 SSODiag에 다음 정보가 표시됩니다.

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

   Kerberos 인증이 제대로 작동하지 않는 경우 SSODiag에 다음 정보가 표시됩니다.

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

보안 모델 변경

보안 영역으로 보호되는 웹 애플리케이션의 기본 보안 모델은 `DDonly`입니다. 보안 모델을 `CustomRolesAndPolicies`로 변경해야 합니다.

보안 모델을 변경하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *MIDDLEWARE_HOME*`/user_projects/domains/EPMSystem/config/config.xml`을 엽니다.
2. 각 Foundation Services 구성요소의 애플리케이션 배포 기술자에서 다음 요소를 찾습니다.

   ```
   <security-dd-model>DDOnly</security-dd-model>
   ```

3. 각 구성요소의 보안 모델을 다음과 같이 변경합니다.

```
<security-dd-model>CustomRolesAndPolicies</security-dd-model>
```

4. `config.xml`을 저장한 후 닫습니다.

EPM System 보안 구성 업데이트

EPM System 보안 구성을 변경하여 Kerberos SSO를 사용으로 설정합니다.

Kerberos 인증에 대해 EPM System을 구성하려면 다음을 수행합니다.

1. Shared Services Console에 관리자로 로그온합니다.
2. Kerberos 인증에 대해 구성된 Active Directory 도메인을 Shared Services의 외부 사용자 디렉토리로 추가합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "OID 및 기타 LDAP 기반 사용자 디렉토리 구성"을 참조하십시오.
3. SSO를 사용으로 설정합니다. OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성을 참조하십시오.
   보안 옵션에서 다음 테이블의 설정을 선택하여 Kerberos SSO를 활성화합니다.

표 3-4 Kerberos SSO 사용 설정

필드	필수 설정
SSO 사용	선택됨
SSO 제공자 또는 에이전트	기타
SSO 메커니즘	HTTP 요청에서 원격 사용자 가져오기

4. Foundation Services를 재시작합니다.

Kerberos SSO 테스트

Foundation Services에 로그인하여 Kerberos SSO가 제대로 작동하는지 확인하십시오.

Kerberos SSO를 테스트하려면 다음을 수행합니다.

1. Foundation Services 및 Oracle HTTP Server가 실행 중인지 확인합니다.
2. 적합한 Active Directory 인증서를 사용하여 Kerberos 인증에 대해 구성된 클라이언트 머신에 로그온합니다.
3. 브라우저를 사용하여 Foundation Services URL에 연결합니다.

EPM System 구성요소 구성

EPM System Configurator를 사용하여 다른 EPM System 구성요소를 구성하고 Foundation Services가 배포된 WebLogic 도메인에 배포합니다.

Kerberos 인증에 대해 EPM System 관리 서버 구성

Microsoft Windows 환경에서는 EPM System 관리 서버가 Windows 서비스로 실행됩니다. 각 WebLogic 관리 서버의 시작 JVM 옵션을 수정해야 합니다. 포괄적인 비압축 배포 모드의 관리 서버 목록은 다음과 같습니다.

- AnalyticProviderServices0
- CalcMgr0

- ErpIntegrator0
- EssbaseAdminServer0
- FinancialReporting0
- HFMWeb0
- FoundationServices0
- HpsAlerter0
- HpsWebReports0
- Planning0
- Profitability0

EPM System 웹 애플리케이션이 압축 배포 모드로 배포된 경우 `EPMSystem0` 관리 서버의 시작 JVM 옵션만 업데이트해야 합니다. 압축 관리 서버가 여러 개인 경우 모든 관리 서버에 대해 시작 JVM 옵션을 업데이트해야 합니다.

Oracle Fusion Middleware Securing Oracle WebLogic Server 가이드의 WebLogic Server에서 Kerberos 인증에 시작 인수 사용을 참조하십시오.

> **주:**
>
> 다음 절차에서는 FoundationServices 관리 서버에 대해 시작 JVM 옵션을 설정하는 방법에 대해 설명합니다. 배포의 각 WebLogic 관리 서버에 대해 이 태스크를 수행해야 합니다.

WebLogic Server 시작 스크립트에서 JVM 옵션을 구성하는 자세한 절차는 Kerberos의 JVM 옵션 업데이트를 참조하십시오.

WebLogic Server 시작 스크립트에서 JVM 옵션을 구성하려면

권한부여 정책 구성

Foundation Services 이외의 EPM System 구성요소에 액세스할 Active Directory 사용자에 대한 권한부여 정책을 구성합니다. WebLogic 관리 콘솔에서 보안 정책을 구성하는 방법에 대한 정보는 권한부여 정책 구성을 참조하십시오.

EPM System 구성요소의 기본 보안 모델 변경

EPM System 구성 파일을 편집하여 기본 보안 모델을 변경합니다. 비압축 EPM System 배포의 경우 `config.xml`에 기록된 각 EPM System 웹 애플리케이션의 기본 보안 모델을 변경해야 합니다. EPM System 웹 애플리케이션 목록은 다음과 같습니다.

- AIF
- APS
- CALC
- EAS
- FINANCIALREPORTING
- PLANNING
- PROFITABILITY

- SHAREDSERVICES
- WORKSPACE

보안 모델을 변경하려면 다음을 수행합니다.

1. 텍스트 편집기를 사용하여 *MIDDLEWARE_HOME*/user_projects/domains/EPMSystem/config/config.xml을 엽니다.
2. 다음 예에 표시된 대로 각 EPM System 구성요소의 app-deployment 정의에서 <security-dd-model>의 값을 CustomRolesAndPolicies로 설정합니다.

```
<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.xml을 저장한 후 닫습니다.
4. WebLogic Server를 재시작합니다.

EPM System 구성요소에 대한 URL 보호 정책 생성

WebLogic Server 관리 콘솔에서 각 EPM System 구성요소 URL을 보호하는 URL 보호 정책을 생성합니다. 자세한 내용은 *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server* 가이드의 웹 애플리케이션 및 EJB 리소스 보안 옵션을 참조하십시오.

URL 보호 정책을 생성하려면 다음을 수행합니다.

1. EPM System 도메인의 WebLogic Server 관리 콘솔에 있는 변경 센터에서 **잠금 및 편집**을 누릅니다.
2. **배포**를 누릅니다.
3. 배포에서 EPM System 엔터프라이즈 애플리케이션(예: PLANNING)을 확장하고 해당 웹 애플리케이션(예: HyperionPlanning)을 누릅니다. EPM System 구성요소 목록은 EPM System 구성요소의 기본 보안 모델 변경을 참조하십시오.

> **주:**
>
> 일부 엔터프라이즈 애플리케이션(예: Oracle Essbase Administration Services)은 URL 패턴을 정의해야 하는 여러 웹 애플리케이션을 구성합니다.

4. 웹 애플리케이션에 대한 URL 패턴 범위 정책을 생성합니다.
   - AIF
   - APS
   - CALC
   - EAS

- FINANCIALREPORTING
- PLANNING
- PROFITABILITY
- SHAREDSERVICES
- WORKSPACE

a. **보안**, **정책**, **새로 작성** 순으로 누릅니다.

b. **URL 패턴**에 EPM System 제품의 보호되는 URL 및 보호되지 않은 URL을 입력하십시오. 자세한 내용은 EPM System 리소스 보호 및 보호 해제를 참조하십시오.

c. **확인**을 누릅니다.

d. 생성한 URL 패턴을 누릅니다.

e. **조건 추가**를 누릅니다.

f. **술어 목록**에서 정책 조건을 선택하고 **다음**을 누릅니다.
이 보안 정책을 지정된 그룹의 모든 멤버에게 부여하는 `Group` 조건을 사용하는 것이 좋습니다.

g. 선택한 술어와 관련된 인수를 지정합니다. 예를 들어 이전 단계에서 `Group`을 선택한 경우 다음 단계를 완료해야 합니다.

h. **그룹 인수 이름**에서 웹 애플리케이션에 액세스할 수 있어야 하는 사용자가 포함된 그룹의 이름을 입력합니다. 입력하는 이름은 Active Directory 그룹 이름과 정확하게 일치해야 합니다.

- **추가**를 누릅니다.
- 그룹을 더 추가하려면 이전 단계를 반복합니다.

i. **완료**를 누릅니다.
Active Directory에서 그룹을 찾을 수 없는 경우 WebLogic Server에서 오류 메시지를 표시합니다. 계속 진행하려면 먼저 이 오류를 해결해야 합니다.

j. **저장**을 선택합니다.

5. 배포의 다른 EPM System 구성요소에 대해 이 절차의 3단계와 4단계를 반복합니다.
6. 변경 센터에서 **구성 해제**를 누릅니다.
7. WebLogic Server를 재시작합니다.

웹 애플리케이션에서 클라이언트 인증서 기반 인증 사용

EPM_ORACLE_HOME`/products/`에 있는 다음 애플리케이션 아카이브의 구성 파일에 `login-config` 정의를 삽입합니다.

- `Essbase/eas/server/AppServer/InstallableApps/Common/eas.ear`
- `FinancialDataQuality/AppServer/InstallableApps/aif.ear`
- `financialreporting/InstallableApps/HReports.ear`
- `Profitability/AppServer/InstallableApps/common/profitability.ear`

클라이언트 인증서 기반 인증을 사용으로 설정하려면 다음을 수행합니다.

1. EPM System 구성요소 및 프로세스를 중지합니다.

2. 7 Zip을 사용하여 엔터프라이즈 아카이브 내에 포함된 웹 아카이브를 확장합니다(예: *EPM_ORACLE_HOME*/products/Essbase/eas/server/AppServer/InstallableApps/Common/eas.ear/eas.war).
3. WEB-INF로 이동합니다.
4. </webapp> 요소 바로 앞에 다음 login_config 정의를 추가하여 web.xml을 수정합니다.

```
<login-config>
     <auth-method>CLIENT-CERT</auth-method>
</login-config>
```

5. web.xml을 저장합니다.
6. 7 Zip에서 아카이브를 업데이트할 것인지 쿼리하면 **예**를 누릅니다.

EPM System 보안 구성 업데이트

SSO를 적용하도록 EPM System 보안을 구성합니다. SSO에 대해 EPM System 구성을 참조하십시오.

SSO에 대해 EPM System 구성

Oracle Enterprise Performance Management System 제품은 SSO에 대해 보안 에이전트를 지원하도록 구성되어야 합니다. Oracle Hyperion Shared Services에 지정된 구성은 모든 EPM System 제품에 대해 다음을 확인합니다.

- 보안 에이전트에서 SSO를 수락하는지 여부
- SSO를 수락하는 인증 메커니즘

SSO 사용 환경에서 사용자가 처음 액세스한 EPM System 제품은 SSO 메커니즘 구문을 분석하여 그 안에 포함된 인증된 사용자 ID를 검색합니다. EPM System 제품은 Shared Services에 구성된 사용자 디렉토리를 기준으로 사용자 ID를 확인하여 사용자를 적합한 EPM System 사용자라고 판단합니다. 또한 EPM System 제품에서 SSO를 활성화하는 토큰을 발급합니다.

Shared Services에 지정된 구성으로 SSO를 활성화하고 모든 EPM System 제품에 대해 SSO를 허용할 인증 메커니즘을 결정할 수 있습니다.

웹 ID 관리 솔루션에서 SSO를 사용으로 설정하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console을 Shared Services 관리자로 실행합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 웹 ID 관리 솔루션에 사용되는 사용자 디렉토리가 Shared Services에서 외부 사용자 디렉토리로 구성되었는지 확인합니다.

   예를 들어 Kerberos SSO를 사용으로 설정하려면 Kerberos 인증을 위해 구성된 Active Directory를 외부 사용자 디렉토리로 구성해야 합니다.

   지침은 사용자 디렉토리 구성을 참조하십시오.
4. **보안 옵션**을 선택합니다.
5. **고급 옵션 표시**를 선택합니다.
6. [정의된 사용자 디렉토리] 화면의 **단일 사인온 구성**에서 다음 단계를 수행합니다:

a. **SSO 사용**을 선택합니다.

b. **SSO 제공자 또는 에이전트**에서 웹 ID 관리 솔루션을 선택합니다. Kerberos를 사용하여 SSO를 구성하는 경우 **기타**를 선택합니다.

   권장 SSO 메커니즘이 자동으로 선택됩니다. 다음 테이블을 참조하십시오. 또한, 지원되는 SSO 방법을 참조하십시오.

> **주:**
>
> 권장 SSO 메커니즘을 사용하지 않는 경우에는 **SSO 제공자 또는 에이전트**에서 기타를 선택해야 합니다. 예를 들어 SiteMinder에 대해 HTTP 머리글 이외의 메커니즘을 사용하려면 **SSO 제공자 또는 에이전트**에서 `Other`를 선택하고 **SSO 메커니즘**에서 사용할 SSO 메커니즘을 선택합니다.

표 3-5 웹 ID 관리 솔루션용 기본 SSO 메커니즘

웹 ID 관리 솔루션	권장 SSO 메커니즘
Oracle Access Manager	`Custom HTTP Header` [1]
OSSO	`Custom HTTP Header`
SiteMinder	`Custom HTTP Header`
Kerberos	`HTTP` 요청에서 원격 사용자 가져오기

[1] 기본 HTTP 머리글 이름은 `HYPLOGIN`입니다. 사용자정의 HTTP 머리글을 사용하는 경우 해당 이름을 바꿉니다.

7. **확인**을 누릅니다.

Smart View의 싱글 사인온 옵션

Oracle Smart View for Office가 브라우저가 아닌 Thick Client라도 HTTP를 사용하여 서버 구성요소에 연결하고 시스템 관점에서 브라우저와 매우 비슷하게 작동합니다. Smart View는 브라우저 인터페이스가 지원하는 모든 표준 웹 기반 통합 메소드를 지원합니다. 하지만 몇 가지 제한 사항이 있습니다.

- Smart View가 Oracle Enterprise Performance Management System 구성요소에 연결된 기존 브라우저 세션에서 실행되는 경우 사용자는 기존 세션에서 쿠키를 공유하지 않으므로 Smart View에 다시 로그인해야 합니다.
- 기본 Oracle Access Manager 로그인 양식이 아닌 사용자정의 HTML 기반 로그인 양식을 사용하는 경우 사용자정의 양식의 소스에 `loginform` 문자열이 포함되어 있는지 확인합니다. Smart View에서 Oracle Access Manager와의 통합이 작동하도록 하려면 이 문자열이 필요합니다.

4

사용자 디렉토리 구성

참조:

- 사용자 디렉토리 및 EPM System 보안
- 사용자 디렉토리 구성 관련 작업
- Oracle Identity Manager 및 EPM System
- Active Directory 정보
- OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성
- 사용자 디렉토리로 관계형 데이터베이스 구성
- 사용자 디렉토리 연결 테스트
- 사용자 디렉토리 설정 편집
- 사용자 디렉토리 구성 삭제
- 사용자 디렉토리 검색 순서 관리
- 보안 옵션 설정
- 암호화 키 다시 생성
- 특수 문자 사용

사용자 디렉토리 및 EPM System 보안

Oracle Enterprise Performance Management System 제품은 여러 사용자 및 ID 관리 시스템에서 지원됩니다. 이러한 관리 시스템을 총체적으로 사용자 디렉토리라고 합니다. 여기에는 Sun Java System Directory Server(이전의 SunONE Directory Server) 및 Active Directory와 같은 LDAP(Lightweight Directory Access Protocol) 사용 사용자 디렉토리가 포함됩니다. EPM System은 관계형 데이터베이스도 외부 사용자 디렉토리로 지원합니다.

일반적으로 EPM System 제품은 프로비저닝에서 Native Directory 및 외부 사용자 디렉토리를 사용합니다. 지원되는 사용자 디렉토리 목록은 Oracle Enterprise Performance Management System Certification Matrix를 참조하십시오.

EPM System 제품을 사용하려면 제품에 액세스하는 각 사용자에 대한 사용자 디렉토리 계정이 있어야 합니다. 프로비저닝이 용이하도록 이러한 사용자를 그룹에 지정할 수도 있습니다. EPM System 역할과 객체 ACL을 사용하여 사용자와 그룹을 프로비전닝할 수 있습니다. 관리 오버헤드 때문에 개별 사용자 프로비저닝은 권장되지 않습니다. 구성된 모든 사용자 디렉토리의 사용자 및 그룹은 Oracle Hyperion Shared Services Console에서 볼 수 있습니다.

기본적으로 EPM System Configurator는 Shared Services 저장소를 Native Directory로 구성하여 EPM System 제품을 지원합니다. 디렉토리 관리자는 Shared Services Console을 사용하여 Native Directory에 액세스하고 관리합니다.

사용자 디렉토리 구성 관련 작업

SSO와 권한을 지원하려면 시스템 관리자가 외부 사용자 디렉토리를 구성해야 합니다. Oracle Hyperion Shared Services Console에서 시스템 관리자는 사용자 디렉토리 구성 및 관리와 관련된 여러 태스크를 수행할 수 있습니다. 다음 항목에 지침이 있습니다.

- 사용자 디렉토리 구성
  - OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성
  - 사용자 디렉토리로 관계형 데이터베이스 구성
- 사용자 디렉토리 연결 테스트
- 사용자 디렉토리 설정 편집
- 사용자 디렉토리 구성 삭제
- 사용자 디렉토리 검색 순서 관리
- 보안 옵션 설정

Oracle Identity Manager 및 EPM System

Oracle Identity Manager는 엔터프라이즈 리소스에 걸쳐 사용자 계정 및 속성 레벨 자격을 둘 다 추가, 업데이트 및 삭제하는 프로세스를 자동화하는 역할 및 사용자 관리 솔루션입니다. Oracle Identity Manager는 독립형 제품으로 사용하거나 Oracle Identity and Access Management Suite Plus의 일부로 사용할 수 있습니다.

Oracle Enterprise Performance Management System은 LDAP 그룹인 엔터프라이즈 역할을 사용하여 Oracle Identity Manager와 통합됩니다. EPM System 구성요소의 역할은 엔터프라이즈 역할에 지정될 수 있습니다. Oracle Identity Manager 엔터프라이즈 역할에 추가된 사용자 또는 그룹은 지정된 EPM System 역할을 자동으로 상속합니다.

예를 들어 Budget Planning이라는 *Oracle Hyperion Planning* 애플리케이션이 있다고 가정합니다. 이 애플리케이션을 지원하기 위해 Oracle Identity Manager에서 Budget Planning 대화식 사용자, Budget Planning 일반 사용자 및 Budget Planning 관리자의 세 가지 엔터프라이즈 역할을 생성할 수 있습니다. EPM System 역할을 프로비저닝하는 동안 프로비저닝 관리자가 *Budget Planning* 및 Shared Services를 비롯한 다른 EPM System 구성요소의 필수 역할로 Oracle Identity Manager의 엔터프라이즈 역할을 프로비저닝하는지 확인합니다. EPM System 구성 요소의 필수 역할을 사용하여 Oracle Identity Manager의 엔터프라이즈 역할을 규정했는지 확인합니다. Oracle Identity Manager 배포 및 관리에 대한 자세한 내용은 Oracle Identity Manager 설명서를 참조하십시오.

Oracle Identity Manager를 EPM System과 통합하려면 관리자가 다음 단계를 수행해야 합니다.

- EPM System 프로비저닝에 사용할 Oracle Identity Manager 엔터프라이즈 역할의 멤버(사용자 및 그룹)가 LDAP 사용 사용자 디렉토리(예: OID 또는 Active Directory)에 정의되어 있는지 확인합니다.
- 엔터프라이즈 역할의 멤버가 정의된 LDAP 지원 사용자 디렉토리를 EPM System의 외부 사용자 디렉토리로 구성합니다. OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성을 참조하십시오.

Active Directory 정보

이 절에서는 이 문서에 사용된 Microsoft Active Directory 개념을 설명합니다.

DNS 룩업 및 호스트 이름 룩업

시스템 관리자는 Oracle Hyperion Shared Services가 정적 호스트 이름 룩업 또는 DNS 룩업을 수행하여 Active Directory를 식별할 수 있도록 Active Directory를 구성할 수 있습니다. 정적 호스트 이름 룩업은 Active Directory 장애 조치를 지원하지 않습니다.

고가용성을 확보하기 위해 여러 도메인 컨트롤러에 Active Directory를 구성한 시나리오의 경우 DNS 룩업을 사용하면 Active Directory의 고가용성이 보장됩니다. DNS 룩업을 수행하도록 구성하면 Shared Services가 DNS 서버에 쿼리하여 등록된 도메인 컨트롤러를 식별하고 가중치가 가장 높은 도메인 컨트롤러에 연결합니다. Shared Services와 연결된 도메인 컨트롤러가 작동하지 않으면 Shared Services는 가중치가 가장 높은 도메인 컨트롤러 중 사용가능한 다음 도메인 컨트롤러로 동적으로 전환합니다.

> **주:**
>
> DNS 룩업은 장애 조치를 지원하는 중복 Active Directory 설정이 사용가능한 경우에만 구성할 수 있습니다. 자세한 내용은 Microsoft 설명서를 참조하십시오.

글로벌 카탈로그

글로벌 카탈로그는 포리스트의 모든 Active Directory 객체 복사본을 저장하는 도메인 컨트롤러입니다. 글로벌 카탈로그는 호스트 도메인의 디렉토리에 있는 모든 객체의 전체 사본과, 포리스트의 다른 모든 도메인에 있는 모든 객체의 부분 사본을 저장합니다. 이러한 사본이 일반적인 사용자 검색 작업에 사용됩니다. 글로벌 카탈로그 설정에 관한 정보는 Microsoft 설명서를 참조하십시오.

조직에서 글로벌 카탈로그를 사용하는 경우 다음 방법 중 하나를 통해 Active Directory를 구성합니다.

- 외부 사용자 디렉토리로 글로벌 카탈로그 서버 구성(권장)
- 각 Active Directory 도메인을 별도의 외부 사용자 디렉토리로 구성

개별 Active Directory 도메인 대신 글로벌 카탈로그를 구성하면 Oracle Enterprise Performance Management System 제품에서 포리스트 내의 로컬 그룹과 범용 그룹에 액세스할 수 있습니다.

OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성

시스템 관리자는 이 절에 설명된 절차를 사용하여 OID, Sun Java System Directory Server, Oracle Virtual Directory, Active Directory, IBM Tivoli Directory Server 또는 구성 화면에 표시되지 않는 LDAP 기반 사용자 디렉토리와 같은 LDAP 기반 기업 사용자 디렉토리를 구성합니다.

OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리를 구성하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.

2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.

   [제공자 구성] 탭이 열립니다. Native Directory를 포함하여 구성된 모든 사용자 디렉토리가 표시됩니다.

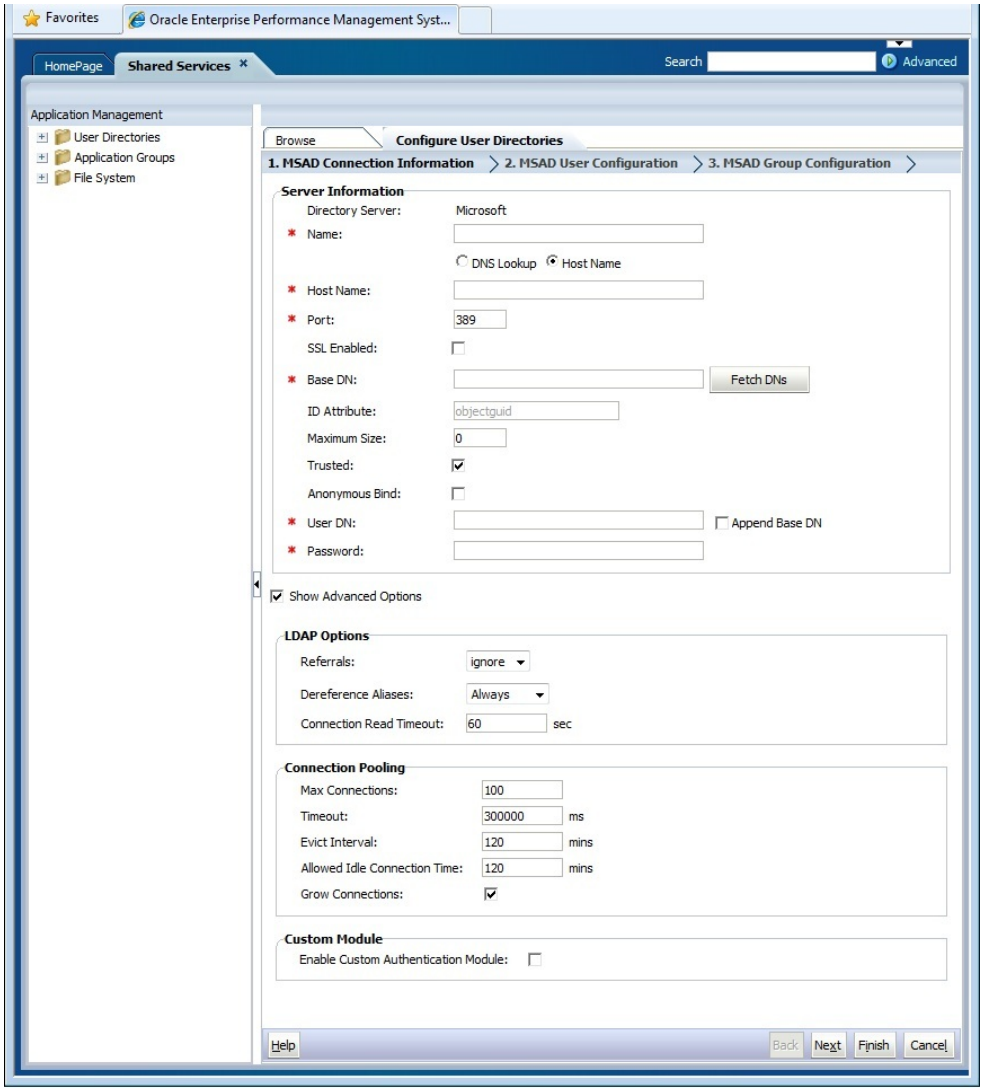
3. **새로 작성**을 누릅니다.

4. **디렉토리 유형**에서 다음 옵션 중 하나를 선택합니다.

   - **LDAP(Lightweight Directory Access Protocol)** - Active Directory 이외의 LDAP 기반 사용자 디렉토리를 구성합니다. Oracle Virtual Directory를 구성하려면 이 옵션을 선택합니다.
   - **MSAD(Microsoft Active Directory)** - Active Directory를 구성합니다.

     Active Directory 및 ADAM(Active Directory Application Mode)에만 해당: 사용자정의 ID 속성(`sAMAccountName` 같은 `ObjectGUID` 이외의 속성)을 Active Directory 또는 ADAM과 함께 사용하려는 경우 **LDAP(Lightweight Directory Access Protocol)**를 선택하고 디렉토리 유형 `Other`로 구성합니다.

5. **다음**을 누릅니다.



6. 필수 매개변수를 입력합니다.

표 4-1 연결 정보 화면

레이블	설명
디렉토리 서버	사용자 디렉토리를 선택합니다. **ID 속성** 값은 선택한 제품에 대해 권장되는 일정한 고유 ID 속성으로 변경됩니다. 4단계에서 Active Directory를 선택한 경우에는 이 등록정보가 자동으로 선택됩니다. 다음과 같은 시나리오에서 `Other`를 선택합니다. • Oracle Virtual Directory와 같이 목록에 없는 사용자 디렉토리 유형을 구성합니다. • 목록에 있는 LDAP 사용 사용자 디렉토리(예: OID)를 구성하지만 사용자정의 ID 속성을 사용하려고 합니다. • 사용자정의 ID 속성을 사용하도록 Active Directory 또는 ADAM을 구성합니다. **주:** Oracle Virtual Directory에서는 LDAP 디렉토리 및 RDMBS 데이터 저장소의 가상화된 추상화를 단일 디렉토리 뷰로 제공하므로 Oracle Enterprise Performance Management System에서는 이를 Oracle Virtual Directory가 지원하는 사용자 디렉토리 수 및 유형에 관계없이 단일 외부 사용자 디렉토리로 간주합니다. **예:** `Oracle Internet Directory`
이름	사용자 디렉토리의 설명 이름으로, 사용자 디렉토리가 여러 개 구성된 경우 특정 사용자 디렉토리를 식별하는 데 사용됩니다. 이름에는 공백과 밑줄 이외의 특수 문자를 사용할 수 없습니다. **예:** `Corporate_OID`

표 4-1 (계속) 연결 정보 화면

레이블	설명
DNS 룩업	**Active Directory에만 해당:** DNS 룩업을 사용하도록 설정하려면 이 옵션을 선택합니다. DNS 룩업 및 호스트 이름 룩업을 참조하십시오. 연결 실패를 방지하려면 DNS 룩업을 프로덕션 환경에서 Active Directory에 연결하는 방법으로 구성하는 것이 좋습니다. **주:** 글로벌 카탈로그를 구성하는 경우에는 이 옵션을 선택하지 마십시오. 이 옵션을 선택하면 다음 필드가 표시됩니다. • **도메인**: Active Directory 포리스트의 도메인 이름입니다. **예:** `example.com` 또는 `us.example.com` • **AD 사이트**: Active Directory 사이트 이름으로서, 일반적으로 Active Directory 구성 컨테이너에 저장된 사이트 객체의 상대적인 고유 이름입니다. 일반적으로 AD 사이트는 도시, 주, 지역 또는 국가 등의 지리적 위치를 식별합니다. **예:** `Santa Clara` 또는 `US_West_region` • **DNS 서버**: 도메인 컨트롤러에 대한 DNS 서버 룩업을 지원하는 서버의 DNS 이름입니다.
호스트 이름	**Active Directory에만 해당:** 정적 호스트 이름 룩업을 사용하도록 설정하려면 이 옵션을 선택합니다. DNS 룩업 및 호스트 이름 룩업을 참조하십시오. **주:** Active Directory 글로벌 카탈로그를 구성하는 경우 이 옵션을 선택합니다.
호스트 이름	사용자 디렉토리 서버의 DNS 이름입니다. SiteMinder에서 SSO를 지원하기 위해 사용자 디렉토리를 사용하는 경우에는 정규화된 도메인 이름을 사용합니다. 테스트용으로만 Active Directory 연결을 설정하려면 호스트 이름을 사용하는 것이 좋습니다. **주:** Active Directory 글로벌 카탈로그를 구성하는 경우 글로벌 카탈로그 서버 호스트 이름을 지정합니다. 글로벌 카탈로그를 참조하십시오. **예:** `MyServer`

표 4-1 (계속) 연결 정보 화면

레이블	설명
포트	사용자 디렉토리가 실행되는 포트 번호입니다. **주:** Active Directory 글로벌 카탈로그를 구성하는 경우 글로벌 카탈로그 서버에 사용되는 포트(기본값 3268)를 지정합니다. 글로벌 카탈로그를 참조하십시오. **예:** `389`
SSL 사용	이 사용자 디렉토리와의 보안 통신을 활성화하는 확인란입니다. 보안 통신이 가능하도록 사용자 디렉토리를 구성해야 합니다.
기준 DN	사용자 및 그룹 검색을 시작하는 노드의 DN(고유 이름)입니다. 또한 **DN 가져오기** 버튼을 사용하여 사용 가능한 기준 DN을 나열한 다음 목록에서 해당 기준 DN을 선택할 수도 있습니다. **주:** 글로벌 카탈로그를 구성하는 경우 포리스트의 기준 DN을 지정합니다. 특수 문자 사용에 대한 제한 사항은 특수 문자 사용을 참조하십시오. 모든 EPM System 제품 사용자와 그룹이 들어 있는 최하위 DN을 선택하는 것이 좋습니다 **예:** `dc=example,dc=com`
ID 속성	이 속성 값은 **디렉토리 유형**에서 기타를 선택한 경우에만 수정할 수 있습니다. 이 속성은 디렉토리 서버의 사용자 및 그룹 객체에 있는 일반 속성이어야 합니다. OID `orclguid`, SunONE(`nsuniqueid`), IBM Directory Server(`Ibm-entryUuid`), Novell eDirectory(`GUID`) 및 Active Directory(`ObjectGUID`)에 대해 이 속성의 권장 값이 자동으로 설정됩니다. **예:** `orclguid` 예를 들어 Oracle Virtual Directory를 구성하기 위해 **디렉토리 서버**에서 기타를 선택한 후 수동으로 설정할 경우 ID 속성 값은 다음과 같아야 합니다. • 고유한 속성을 가리켜야 합니다. • 특정 위치가 되면 안 됩니다. • 시간에 따라 변경되면 안 됩니다.

표 4-1 (계속) 연결 정보 화면

레이블	설명
최대 크기	검색 시 반환될 수 있는 최대 결과 수입니다. 이 값이 사용자 디렉토리 설정에서 지원하는 값보다 큰 경우 사용자 디렉토리 값이 이 값을 재정의합니다. Active Directory 이외의 사용자 디렉토리에 대해 검색 기준과 일치하는 사용자 및 그룹을 모두 검색하려면 이 필드를 비워 둡니다. Active Directory의 경우 검색 기준과 일치하는 사용자 및 그룹을 모두 검색하려면 이 값을 `0`으로 설정합니다. 위임된 관리 모드에서 Oracle Hyperion Shared Services를 구성하는 경우 이 값을 0으로 설정합니다.
인증	제공자가 신뢰할 수 있는 SSO 소스임을 나타내는 확인란입니다. 신뢰할 수 있는 소스에서 가져온 SSO 토큰에는 사용자 비밀번호가 포함되어 있지 않습니다.
익명 바인딩	사용자와 그룹을 검색하기 위해 Shared Services를 사용자 디렉토리에 익명으로 바인딩할 수 있음을 나타내는 확인란입니다. 사용자 디렉토리에서 익명 바인딩이 허용되는 경우에만 사용할 수 있습니다. 이 옵션을 선택하지 않으면 사용자 정보가 저장된 디렉토리를 검색할 수 있는 액세스 권한이 있는 계정을 사용자 DN에 지정해야 합니다. 익명 바인딩은 사용하지 않는 것이 좋습니다. **주:** OID의 경우 익명 바인딩이 지원되지 않습니다.
사용자 DN	**익명 바인딩**을 선택한 경우 이 옵션은 비활성화됩니다. Shared Services가 사용자 디렉토리에 바인딩하는 데 사용하는 사용자의 고유 이름입니다. 이 사용자에게 DN 내의 RDN 속성에 대한 검색 권한이 있어야 합니다. 예를 들어 dn: `cn=John Doe, ou=people, dc=myCompany, dc=com`에서 바인딩 사용자에게는 `cn` 속성에 대한 검색 액세스 권한이 있어야 합니다. [사용자 DN]에서 특수 문자는 이스케이프 문자를 사용하여 지정해야 합니다. 제한 사항에 대해서는 특수 문자 사용을 참조하십시오. **예:** `cn=admin,dc=myCompany,dc=com`
기준 DN 추가	기준 DN을 사용자 DN에 추가하는 확인란입니다. 사용자 DN으로 디렉토리 관리자 계정을 사용하는 경우에는 기준 DN을 추가하지 마십시오. 익명 바인딩 옵션을 선택한 경우 이 확인란은 비활성화됩니다.
비밀번호	사용자 DN 비밀번호입니다. 익명 바인딩 옵션을 선택한 경우 이 상자는 비활성화됩니다. **예:** `UserDNpassword`
고급 옵션 표시	고급 옵션을 표시하는 확인란입니다.
참조	**Active Directory에만 해당:** Active Directory가 참조를 따르도록 구성된 경우 LDAP 참조를 자동으로 따르려면 `follow`를 선택합니다. 참조를 사용하지 않으려면 무시를 선택합니다.

표 4-1 (계속) 연결 정보 화면

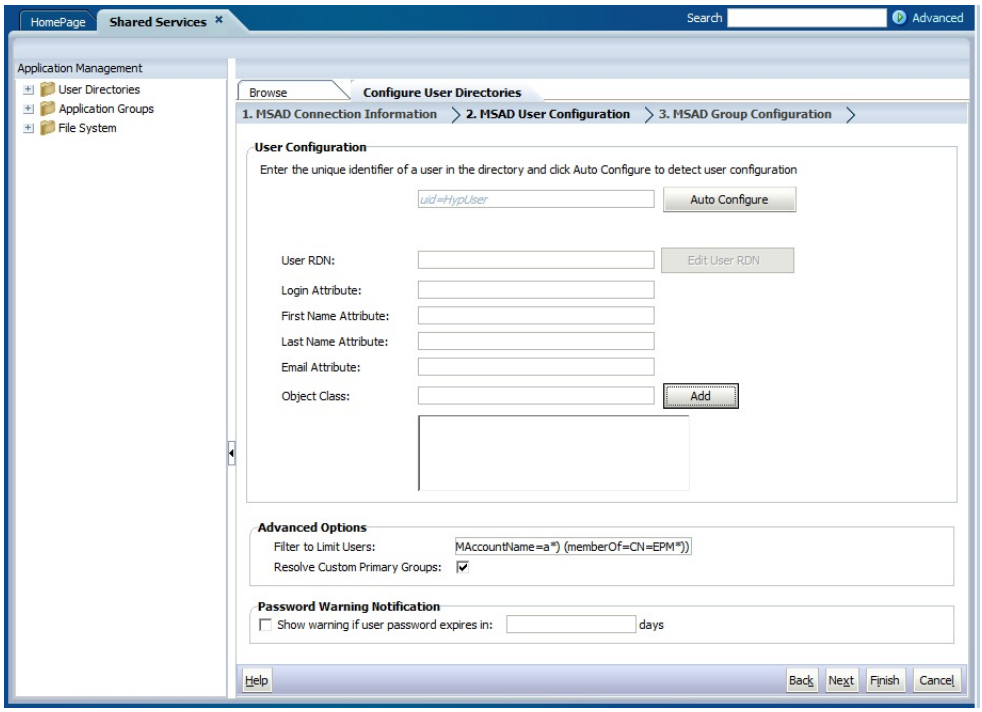
레이블	설명
별칭 참조 해제	별칭의 DN이 가리키는 객체를 검색할 수 있도록 Shared Services 검색 시 사용자 디렉토리에서 별칭 참조 해제를 위해 사용하는 방법을 선택합니다. 다음 중에서 선택합니다. • **항상**: 항상 별칭을 참조 해제합니다. • **사용 안 함**: 별칭 참조를 해제하지 않습니다. • **찾기**: 이름 확인 중에만 별칭을 참조 해제합니다. • **검색**: 이름 확인 후에만 별칭을 참조 해제합니다.
연결 읽기 시간 초과	LDAP 제공자가 응답을 얻지 못하는 경우 LDAP 읽기 시도를 중단하기까지 경과되는 간격(초)입니다. **기본값:** 60초
최대 연결	연결 풀의 최대 연결 수입니다. 기본값은 Active Directory를 포함하여 LDAP 기반 디렉토리의 경우 100입니다. **기본값:** 100
시간 초과	풀에서 연결 가져오기 제한 시간입니다. 이 기간이 지나면 예외가 발생합니다. **기본값:** 300000밀리초(5분)
제거 간격	**선택 사항:** 풀을 정리하기 위한 제거 프로세스를 실행하는 간격입니다. 제거 프로세스는 허용되는 유휴 연결 시간을 초과하는 유휴 연결을 제거합니다. **기본값:** 120분
허용되는 유휴 연결 시간	**선택 사항:** 제거 프로세스에 의해 풀의 유휴 연결이 제거되기까지 경과하는 시간입니다. **기본값:** 120분
연결 늘리기	이 옵션은 연결 풀이 최대 연결을 초과하여 늘어날 수 있는지 여부를 나타냅니다. 기본적으로 선택되어 있습니다. 연결 풀이 늘어나는 것을 허용하지 않으면 시간 초과로 설정된 시간 이내에 연결을 사용할 수 없는 경우 오류가 발생합니다.
사용자정의 인증 모듈 사용	사용자정의 인증 모듈을 사용하여 이 사용자 디렉토리에 정의된 사용자를 인증할 수 있도록 하는 확인란입니다. [보안 옵션] 화면에서 인증 모듈의 정규화된 Java 클래스 이름도 입력해야 합니다. 보안 옵션 설정을 참조하십시오. 사용자정의 인증 모듈 인증은 씬 클라이언트와 씩 클라이언트에 투명하며 클라이언트 배포 변경이 필요하지 않습니다. *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "사용자정의 인증 모듈 사용"을 참조하십시오.

7. **다음**을 누릅니다.

Shared Services는 [사용자 구성] 화면에 설정된 등록정보를 사용하여 사용자 검색이 시작되는 노드를 확인하는 데 사용되는 사용자 URL을 생성합니다. 이 URL을 사용하면 검색 속도가 향상됩니다.

> **주의:**
>
> 사용자 URL은 별칭을 가리키면 안 됩니다. EPM System 보안에서는 사용자 URL이 실제 사용자를 가리켜야 합니다.

화면의 [자동 구성] 영역을 사용하여 필요한 정보를 검색하는 것이 좋습니다.



> **주:**
>
> 사용자 구성에서 사용할 수 있는 특수 문자의 목록은 특수 문자 사용을 참조하십시오.

8. **자동 구성**에 *attribute=identifier* 형식(예: `uid=jdoe`)으로 고유 사용자 식별자를 입력합니다.

   사용자 속성은 [사용자 구성] 영역에 표시됩니다.

   OID의 루트 DSE는 이름 지정 컨텍스트 속성의 항목을 포함하지 않으므로 OID를 구성하는 경우 사용자 필터를 자동으로 구성할 수 없습니다. *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*의 이름 지정 컨텍스트 관리를 참조하십시오.

> **주:**
>
> [사용자 구성] 영역의 텍스트 상자에 필수 사용자 속성을 수동으로 입력할 수 있습니다.

표 4-2 사용자 구성 화면

레이블	설명[1]
사용자 RDN	사용자의 상대 DN입니다. DN의 각 구성요소는 RDN이라고 하며 디렉토리 트리의 각 분기를 나타냅니다. 사용자 RDN은 일반적으로 `uid` 또는 `cn`과 동일합니다. 제한 사항에 대해서는 특수 문자 사용을 참조하십시오. **예:** `ou=People`
로그인 속성	사용자의 로그인 이름을 저장하는 고유한 속성(사용자정의 속성일 수 있음)입니다. 사용자는 EPM System 제품에 로그인할 때 이 속성 값을 사용자 이름으로 사용합니다. 사용자 ID(로그인 속성 값)는 모든 사용자 디렉토리에서 고유해야 합니다. 예를 들어 `uid` 및 `sAMAccountName`을 각각 SunONE 및 Active Directory 구성에 대한 로그인 속성으로 사용할 수 있습니다. 이러한 속성 값은 Native Directory를 비롯한 모든 사용자 디렉토리에서 고유해야 합니다. **주:** 사용자 ID는 대소문자를 구분하지 않습니다. **주:** Kerberos 환경에서 Oracle Application Server에 배포된 EPM System 제품의 외부 사용자 디렉토리로 OID를 구성하는 경우 이 등록정보를 `userPrincipalName`으로 설정해야 합니다. **기본값** • **Active Directory:** `cn` • **Active Directory 이외의 LDAP 디렉토리:** `uid`
이름 속성	사용자의 이름을 저장하는 속성입니다. **기본값:** `givenName`
성 속성	사용자의 성을 저장하는 속성입니다. **기본값:** `sn`
전자메일 속성	**선택 사항**: 사용자의 전자메일 주소를 저장하는 속성입니다. **기본값:** `mail`

표 4-2 (계속) 사용자 구성 화면

레이블	설명[1]
객체 클래스	사용자의 객체 클래스입니다(사용자와 연관될 수 있는 필수 및 선택적 속성). Shared Services는 검색 필터에서 이 화면에 나열된 객체 클래스를 사용합니다. Shared Services는 이 객체 클래스를 사용하여 프로비전닝되어야 할 모든 사용자를 찾습니다. **주:** 사용자정의 ID 속성을 사용하도록 Active Directory 또는 ADAM을 사용자 디렉토리 유형 `Other`로 구성하는 경우 이 값을 `user`로 설정해야 합니다. 필요한 경우 다른 객체 클래스를 수동으로 추가할 수 있습니다. 객체 클래스를 추가하려면 **객체 클래스** 상자에 객체 클래스 이름을 입력하고 **추가**를 누릅니다. 객체 클래스를 삭제하려면 객체 클래스를 선택하고 **제거**를 누릅니다. **기본값** • **Active Directory:** `user` • **Active Directory 이외의 LDAP 디렉토리:** `person, organizationalPerson, inetorgperson`
사용자를 제한하는 필터링	EPM System 제품 역할로 프로비전닝될 사용자만 검색하는 LDAP 쿼리입니다. 예를 들어 LDAP 쿼리 `(uid=Hyp*)`는 이름이 `Hyp`로 시작하는 사용자만 검색합니다. [사용자 구성] 화면에서는 사용자 RDN을 검증하므로 필요한 경우 사용자 필터를 사용하는 것이 좋습니다. 사용자 필터는 쿼리 중에 반환되는 사용자 수를 제한합니다. 이는 특히 사용자 RDN으로 식별되는 노드에 프로비전닝할 필요가 없는 다수의 사용자가 포함되어 있는 경우에 중요합니다. 사용자 필터는 프로비전닝하지 않을 사용자는 제외하도록 설계되었으므로 성능을 향상시킬 수 있습니다.
다중 속성 RDN에 대한 사용자 검색 속성	**Active Directory 이외의 LDAP 지원 사용자 디렉토리에만 해당:** 디렉토리 서버가 다중 속성 RDN을 사용하도록 구성된 경우에만 이 값을 설정합니다. 설정하는 값은 RDN 속성 중 하나여야 합니다. 지정하는 속성 값은 고유해야 하고 속성이 검색 가능해야 합니다. 예를 들어 SunONE 디렉토리 서버가 cn(`cn=John Doe`) 및 uid(`uid=jDoe12345`) 속성을 결합하여 다음과 비슷한 다중 속성 RDN을 생성하도록 구성되어 있다고 가정합니다. `cn=John Doe+uid=jDoe12345, ou=people, dc=myCompany, dc=com` 이 경우 이 속성이 다음 조건을 충족하면 `cn` 또는 `uid`를 사용할 수 있습니다. • [연결 정보] 탭의 [사용자 DN] 필드에서 식별되는 사용자가 속성을 검색할 수 있습니다. • 이 속성의 경우 사용자 디렉토리에서 고유한 값을 설정해야 합니다.

표 4-2 (계속) 사용자 구성 화면

레이블	설명[1]
사용자정의 기본 그룹 확인	**Active Directory에만 해당:** 효율적인 역할을 결정하기 위해 기본 사용자 그룹을 확인할지 여부를 나타내는 확인란입니다. 이 확인란은 기본적으로 선택되어 있습니다. 이 설정을 변경하지 않는 것이 좋습니다.
사용자 비밀번호가 만료되는 경우 경고를 표시합니다.	**Active Directory에만 해당:** Active Directory 사용자 비밀번호가 지정된 일수 내에 만료되는 경우 경고 메시지를 표시할지 여부를 나타내는 확인란입니다.

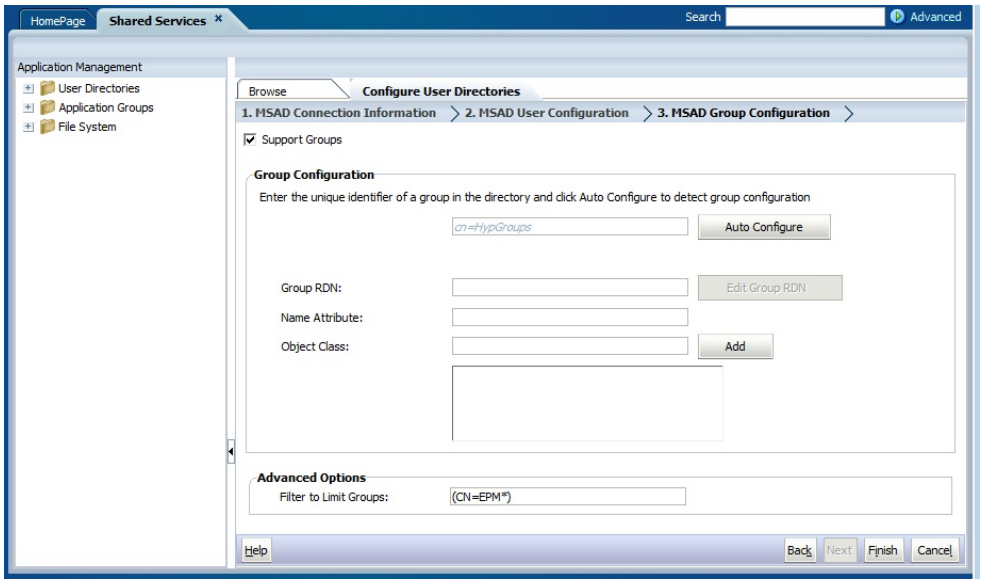
[1] EPM System 보안에서는 구성 값이 선택 사항인 일부 필드에 기본값을 사용할 수 있습니다. 이러한 필드에 값을 입력하지 않으면 런타임 기간 동안 기본값이 사용됩니다.

9. **다음**을 누릅니다.

[그룹 구성] 화면이 열립니다. Shared Services는 이 화면에 설정된 등록정보를 사용하여 그룹 검색을 시작하는 노드를 확인하는 데 사용되는 그룹 URL을 생성합니다. 이 URL을 사용하면 검색 속도가 향상됩니다.

> **주의:**
>
> 그룹 URL이 별칭을 가리켜서는 안 됩니다. EPM System 보안에서는 그룹 URL이 실제 그룹을 가리켜야 합니다. 그룹 별칭을 사용하는 Novell eDirectory를 구성하는 경우 그룹 URL 내에서 그룹 별칭과 그룹 계정을 사용할 수 있어야 합니다.

> **주:**
>
> [그룹 구성] 화면의 데이터 입력은 선택 사항입니다. 그룹 URL 설정을 입력하지 않으면 Shared Services는 기준 DN 내에서 검색하여 그룹을 찾습니다. 그러므로, 특히 사용자 디렉토리에 여러 그룹이 포함되어 있는 경우 성능에 부정적인 영향을 줄 수 있습니다.



10. 조직에서 그룹을 프로비전닝하지 않으려는 경우 또는 사용자가 사용자 디렉토리의 그룹으로 분류되지 않은 경우에는 **그룹 지원**의 선택을 취소합니다. 이 옵션의 선택을 취소하면 이 화면의 필드가 비활성화됩니다.

    그룹을 지원하는 경우 자동 구성 기능을 사용하여 필요한 정보를 검색하는 것이 좋습니다.

    OID를 사용자 디렉토리로 구성하는 경우 OID의 루트 DSE는 이름 지정 컨텍스트 속성의 항목을 포함하지 않으므로 자동 구성 기능을 사용할 수 없습니다. *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*의 이름 지정 컨텍스트 관리를 참조하십시오.

11. **자동 구성** 텍스트 상자에 고유한 그룹 식별자를 입력하고 **이동**을 누릅니다.

    그룹 식별자는 *attribute=identifier* 형식(예: `cn=western_region`)으로 표현해야 합니다.

    그룹 속성은 [그룹 구성] 영역에 표시됩니다.

> **주:**
>
> [그룹 구성] 텍스트 상자에 필요한 그룹 속성을 입력할 수 있습니다.

> **주의:**
>
> 노드 이름에 /(슬래시) 또는 \(백슬래시)가 포함되는 사용자 디렉토리의 그룹 URL이 설정되어 있지 않으면 사용자 및 그룹 검색에 실패합니다. 예를 들어 `OU=child\ou,OU=parent/ou` 또는 `OU=child/ou,OU=parent \ ou`와 같은 노드에 사용자와 그룹이 있는 사용자 디렉토리에 대해 그룹 URL을 지정하지 않으면 사용자 또는 그룹을 나열하는 작업은 실패합니다.

표 4-3 그룹 구성 화면

레이블	설명[1]
그룹 RDN	그룹의 상대 DN입니다. 기준 DN에 상대적인 경로인 이 값은 그룹 URL로 사용됩니다. 프로비전닝하려는 모든 그룹이 있는 최하위 사용자 디렉토리 노드를 식별하는 그룹 RDN을 지정합니다. 프로비저닝에 Active Directory 기본 그룹을 사용하는 경우 기본 그룹이 그룹 RDN에 속하는지 확인합니다. Shared Services는 기본 그룹이 그룹 URL의 범위를 벗어나는 경우 기본 그룹을 검색하지 않습니다. 그룹 RDN은 로그인 및 검색 성능에 커다란 영향을 미칩니다. 그룹 RDN은 모든 그룹 검색의 시작점이므로 EPM System 제품에 대한 모든 그룹을 사용할 수 있는 최하위 노드를 식별해야 합니다. 성능을 최적화하려면 그룹 RDN에 있는 그룹 수가 10,000개를 초과하지 않아야 합니다. 그룹 수가 이를 초과하는 경우에는 그룹 필터를 사용하여 프로비전닝하려는 그룹만 검색합니다. **주:** 그룹 URL에서 사용 가능한 그룹 수가 10,000개를 초과하면 Shared Services에 경고가 표시됩니다. 제한 사항에 대해서는 특수 문자 사용을 참조하십시오. **예:** `ou=Groups`
이름 속성	그룹 이름을 저장하는 속성입니다. **기본값** • **Active Directory를 비롯한 LDAP 디렉토리:** `cn` • **Native Directory:** `cssDisplayNameDefault`

표 4-3 (계속) 그룹 구성 화면

레이블	설명[1]
객체 클래스	그룹의 객체 클래스입니다. Shared Services는 검색 필터에서 이 화면에 나열된 객체 클래스를 사용합니다. Shared Services는 이러한 객체 클래스를 사용하여 사용자와 연관된 모든 그룹을 찾습니다. **주:** 사용자정의 ID 속성을 사용하도록 Active Directory 또는 ADAM을 사용자 디렉토리 유형 `Other`로 구성하는 경우 이 값을 `group?member`로 설정해야 합니다. 필요한 경우 다른 객체 클래스를 수동으로 추가할 수 있습니다. 객체 클래스를 추가하려면 객체 클래스 상자에 객체 클래스 이름을 입력하고 **추가**를 누릅니다. 객체 클래스를 삭제하려면 객체 클래스를 선택하고 **제거**를 누릅니다. **기본값** • **Active Directory:** `group?member` • **Active Directory 이외의 LDAP 디렉토리:** `groupofuniquenames?uniquemember, groupOfNames?member` • **Native Directory:** `groupofuniquenames?uniquemember, cssGroupExtend?cssIsActive`
그룹을 제한하는 필터링	EPM System 제품 역할만으로 프로비전닝될 그룹을 검색하는 LDAP 쿼리입니다. 예를 들어 LDAP 쿼리 `(|(cn=Hyp*)(cn=Admin*))`는 이름이 `Hyp` 또는 `Admin`으로 시작하는 그룹만 검색합니다. 그룹 필터는 쿼리 중에 반환되는 그룹 수를 제한하는 데 사용됩니다. 이는 특히 그룹 RDN으로 식별되는 노드에 프로비전닝할 필요가 없는 다수의 그룹이 포함되어 있는 경우에 중요합니다. 그룹 필터는 프로비전닝하지 않을 그룹은 제외하도록 설계되었으므로 성능을 향상시킬 수 있습니다. 프로비저닝에 Active Directory 기본 그룹을 사용하는 경우 설정하는 그룹 필터가 그룹 URL의 범위 내에 포함된 기본 그룹을 검색할 수 있는지 확인합니다. 예를 들어 `(|(cn=Hyp*)(cn=Domain Users))` 필터는 이름이 `Hyp`로 시작하는 그룹과 이름이 `Domain Users`인 기본 그룹을 검색합니다.

[1] EPM System 보안에서는 구성 값이 선택 사항인 일부 필드에 기본값을 사용할 수 있습니다. 이러한 필드에 값을 입력하지 않으면 런타임 기간 동안 기본값이 사용됩니다.

12. **완료**를 누릅니다.

    Shared Services는 구성을 저장하고 [정의된 사용자 디렉토리] 화면으로 돌아갑니다. 이 화면에는 이제 사용자가 구성한 사용자 디렉토리가 나열됩니다.

13. 구성을 테스트합니다. 사용자 디렉토리 연결 테스트를 참조하십시오.
14. 필요한 경우 검색 순서 지정을 변경합니다. 자세한 내용은 사용자 디렉토리 검색 순서 관리를 참조하십시오.
15. 필요한 경우 보안 옵션을 지정합니다. 자세한 내용은 보안 옵션 설정을 참조하십시오.

16. Oracle Hyperion Foundation Services 및 다른 EPM System 구성요소를 재시작합니다.

사용자 디렉토리로 관계형 데이터베이스 구성

Oracle, SQL Server, IBM DB2 관계형 데이터베이스에 있는 시스템 테이블의 사용자 및 그룹 정보를 사용하여 프로비저닝을 지원할 수 있습니다. 데이터베이스의 시스템 스키마에서 그룹 정보를 파생시킬 수 없는 경우 Oracle Hyperion Shared Services는 해당 데이터베이스 제공자의 그룹 프로비저닝을 지원하지 않습니다. 예를 들어 데이터베이스에서 운영 체제에 정의된 그룹을 사용하기 때문에 Shared Services는 이전 버전의 IBM DB2에서 그룹 정보를 추출할 수 없습니다. 그러나 프로비저닝 관리자는 이러한 사용자를 Native Directory의 그룹에 추가하고 그룹을 프로비저닝할 수 있습니다. 지원되는 플랫폼 정보는 OTN(Oracle Technology Network)의 Oracle Fusion Middleware Supported System Configurations 페이지에 게시된 *Oracle Enterprise Performance Management System Certification Matrix*를 참조하십시오.

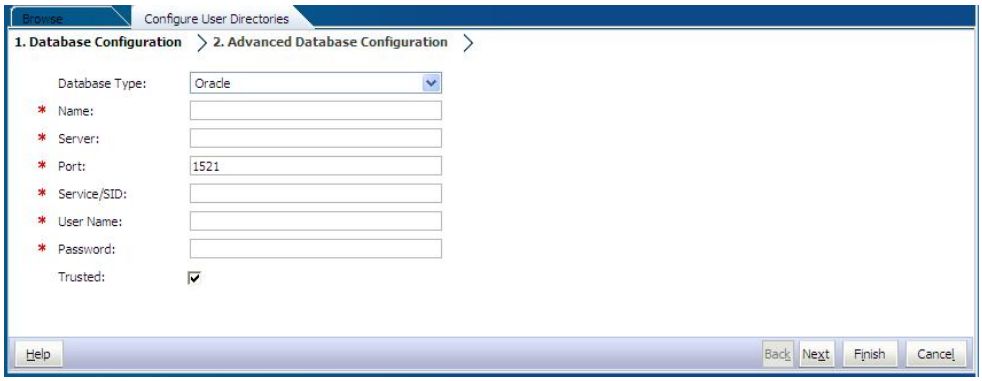
> **주:**
>
> DB2 데이터베이스를 사용하는 경우 사용자 이름이 8자 이상이어야 합니다. 사용자 이름은 256자(Oracle 및 SQL Serve 데이터베이스) 또는 1000자(DB2) 이하여야 합니다.

사용자와 그룹의 목록을 검색하려면 데이터베이스 관리자(예: Oracle `SYSTEM` 사용자)로 데이터베이스에 연결하도록 Shared Services를 구성합니다.

> **주:**
>
> Shared Services는 프로비저닝에 대해 활성 데이터베이스 사용자만 검색합니다. 비활성이고 잠긴 데이터베이스 사용자 계정은 무시됩니다.

데이터베이스 제공자를 구성하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행를 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. **새로 작성**을 누릅니다.
4. **디렉토리 유형** 화면에서 **관계형 데이터베이스(Oracle, DB2, SQL Server)**를 선택합니다.
5. **다음**을 누릅니다.



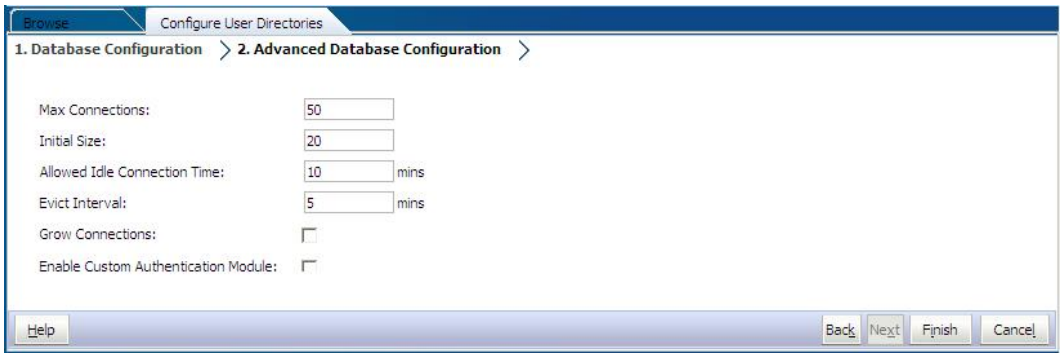
6. [데이터베이스 구성] 탭에서 구성 매개변수를 입력합니다.

표 4-4 데이터베이스 구성 탭

레이블	설명
데이터베이스 유형	관계형 데이터베이스 제공자입니다. Shared Services는 데이터베이스 제공자로 Oracle 및 SQL Server 데이터베이스만 지원합니다. **예:** `Oracle`
이름	데이터 제공자의 고유한 구성 이름입니다. **예:** `Oracle_DB_FINANCE`
서버	데이터베이스 서버가 실행 중인 컴퓨터의 DNS 이름입니다. **예:** `myserver`
포트	데이터베이스 서버 포트 번호입니다. **예:** `1521`
서비스/SID(Oracle만 해당)	시스템 식별자입니다(기본값은 `orcl`). **예:** `orcl`
데이터베이스(SQL Server와 DB2만 해당)	Shared Services가 연결해야 하는 데이터베이스입니다. **예:** `master`
사용자 이름	Shared Services가 데이터베이스에 액세스할 때 사용할 사용자 이름입니다. 이 데이터베이스 사용자는 데이터베이스 시스템 테이블에 대한 액세스 권한을 가지고 있어야 합니다. Oracle에서는 Oracle 데이터베이스에 `system` 계정을, SQL Server 데이터베이스에 데이터베이스 관리자의 사용자 이름을 각각 사용하도록 권장합니다. **예:** `SYSTEM`
비밀번호	**사용자 이름**에서 식별된 사용자의 비밀번호입니다. **예:** `system_password`
인증	이 제공자가 신뢰할 수 있는 SSO 소스임을 지정하는 확인란입니다. 신뢰할 수 있는 소스에서 가져온 SSO 토큰에는 사용자 비밀번호가 포함되어 있지 않습니다.

7. **선택 사항: 다음**을 눌러서 연결 풀을 구성합니다.

[고급 데이터베이스 구성] 탭이 열립니다.



8. [고급 데이터베이스 구성]에서 연결 풀 매개변수를 입력합니다.

표 4-5 고급 데이터베이스 구성 탭

레이블	설명
최대 연결	풀의 최대 연결 수입니다. 기본값은 50입니다.
초기 크기	풀을 초기화했을 때 사용 가능한 연결 수입니다. 기본값은 20입니다.
허용되는 유휴 연결 시간	**선택 사항:** 제거 프로세스에 의해 풀의 유휴 연결이 제거되기까지 경과하는 시간입니다. 기본값은 10분입니다.
제거 간격	**선택 사항:** 제거 프로세스가 풀을 정리하는 실행 간격입니다. 제거는 허용되는 유휴 연결 시간을 초과한 유휴 연결을 제거합니다. 기본값은 5분입니다.
연결 늘리기	연결 풀이 최대 연결을 초과하여 늘어날 수 있는지 여부를 나타냅니다. 기본적으로 이 옵션은 선택 취소되어 있어 풀이 늘어날 수 없습니다. 연결 풀이 늘어나는 것을 허용하지 않으면 시간 초과로 설정된 시간 이내에 연결을 사용할 수 없는 경우 오류가 발생합니다.
사용자정의 인증 모듈 사용	사용자정의 인증 모듈을 사용하여 이 사용자 디렉토리에 정의된 사용자를 인증할 수 있도록 하는 확인란입니다. [보안 옵션] 화면에서 인증 모듈의 정규화된 Java 클래스 이름도 입력해야 합니다. 보안 옵션 설정를 참조하십시오. 사용자정의 인증 모듈 인증은 Thin Client 및 Thick Client에 투명합니다. *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "사용자정의 인증 모듈 사용"을 참조하십시오.

9. **완료**를 누릅니다.
10. **확인**을 눌러서 [정의된 사용자 디렉토리] 화면으로 돌아갑니다.
11. 데이터베이스 제공자 구성을 테스트합니다. 사용자 디렉토리 연결 테스트를 참조하십시오.
12. 필요한 경우 검색 순서 지정을 변경합니다. 자세한 내용은 사용자 디렉토리 검색 순서 관리를 참조하십시오.
13. 필요한 경우 보안 설정을 지정합니다. 보안 옵션 설정을 참조하십시오.
14. Oracle Hyperion Foundation Services 및 다른 Oracle Enterprise Performance Management System 구성요소를 다시 시작합니다.

사용자 디렉토리 연결 테스트

사용자 디렉토리를 구성한 후 연결을 테스트하여 현재 설정으로 Oracle Hyperion Shared Services를 사용자 디렉토리에 연결할 수 있는지 확인합니다.

사용자 디렉토리 연결을 테스트하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 사용자 디렉토리 목록에서 테스트할 외부 사용자 디렉토리를 선택합니다.
4. **테스트**, **확인** 순으로 누릅니다.

사용자 디렉토리 설정 편집

관리자는 이름을 제외하고 사용자 디렉토리 구성의 모든 매개변수를 수정할 수 있습니다. 프로비저닝에 사용된 사용자 디렉토리의 구성 데이터는 편집하지 않는 것이 좋습니다.

> **주의:**
>
> 사용자 디렉토리 구성에서 `ID` 속성과 같은 일부 설정을 편집하면 프로비저닝 데이터가 무효화됩니다. 프로비전닝된 사용자 디렉토리의 설정을 수정할 때는 특별히 주의하십시오.

사용자 디렉토리 구성을 편집하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 편집할 사용자 디렉토리를 선택합니다.
4. **편집**을 누릅니다.
5. 구성 설정을 수정합니다.

   > **주:**
   >
   > 구성 이름은 수정할 수 없습니다. LDAP 사용자 디렉토리 구성을 수정하는 경우 디렉토리 서버 목록에서 다른 디렉토리 서버 또는기타(사용자정의 LDAP 디렉토리의 경우)를 선택합니다. Native Directory 매개변수는 편집할 수 없습니다.

   편집할 수 있는 매개변수에 대한 설명은 다음 테이블을 참조하십시오.

   - Active Directory 및 기타 LDAP 기반 사용자 디렉토리, OID, Active Directory 및 기타 LDAP 기반 사용자 디렉토리 구성의 테이블을 참조하십시오.
   - 데이터베이스: 사용자 디렉토리로 관계형 데이터베이스 구성의 테이블을 참조하십시오.
6. **확인**을 눌러 변경사항을 저장합니다.

사용자 디렉토리 구성 삭제

시스템 관리자는 외부 사용자 디렉토리 구성을 언제든지 삭제할 수 있습니다. 구성을 삭제하면 사용자 디렉토리에서 파생된 사용자와 그룹의 모든 프로비저닝 정보가 무효화되고 검색 순서에서 해당 디렉토리가 제거됩니다.

> **팁:**
>
> 프로비저닝에 사용한 구성된 사용자 디렉토리를 사용하지 않으려면 사용자와 그룹이 검색되지 않도록 검색 순서에서 해당 사용자 디렉토리를 제거합니다. 이렇게 하면 프로비저닝 정보의 무결성이 유지되고 나중에 사용자 디렉토리를 사용할 수 있습니다.

사용자 디렉토리 구성을 삭제하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 디렉토리를 선택합니다.
4. **삭제**를 누릅니다.
5. **확인**을 누릅니다.
6. **확인**을 다시 누릅니다.
7. Oracle Hyperion Foundation Services 및 다른 Oracle Enterprise Performance Management System 구성요소를 다시 시작합니다.

사용자 디렉토리 검색 순서 관리

시스템 관리자가 외부 사용자 디렉토리를 구성하면 Oracle Hyperion Shared Services에서 해당 사용자 디렉토리를 검색 순서에 자동으로 추가하고 Native Directory 앞에 오는 사용가능한 다음 검색 순서를 지정합니다. 검색 순서는 Oracle Enterprise Performance Management System이 사용자와 그룹을 검색할 때 구성된 사용자 디렉토리를 순환하는 데 사용됩니다.

시스템 관리자는 검색 순서에서 사용자 디렉토리를 제거할 수 있으며, 이 경우 Shared Services에서 나머지 디렉토리의 검색 순서를 자동으로 재지정합니다. 검색 순서에 포함되지 않은 사용자 디렉토리는 인증 및 프로비저닝 지원에 사용되지 않습니다.

> **주:**
>
> Shared Services는 지정된 계정이 나타나면 해당 사용자 또는 그룹 검색을 종료합니다. 대부분의 EPM System 사용자가 포함된 기업 디렉토리를 검색 순서의 맨위에 배치하는 것이 좋습니다.

기본적으로 Native Directory는 검색 순서의 마지막 디렉토리로 설정됩니다. 관리자는 다음 태스크를 수행하여 검색 순서를 관리할 수 있습니다.

- 검색 순서에 사용자 디렉토리 추가

- 검색 순서 변경
- 검색 순서 지정 제거

검색 순서에 사용자 디렉토리 추가

새로 구성된 사용자 디렉토리는 자동으로 검색 순서에 추가됩니다. 검색 순서에서 디렉토리를 제거한 경우 검색 순서 끝에 디렉토리를 추가할 수 있습니다.

검색 순서에 사용자 디렉토리를 추가하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 검색 순서에 추가할 비활성 사용자 디렉토리를 선택합니다.
4. **포함**을 누릅니다.

   이 버튼은 검색 순서에 없는 사용자 디렉토리를 선택한 경우에만 사용할 수 있습니다.
5. **확인**을 눌러서 [정의된 사용자 디렉토리] 화면으로 돌아갑니다.
6. Oracle Hyperion Foundation Services 및 다른 EPM System 구성요소를 재시작합니다.

검색 순서 지정 제거

검색 순서에서 사용자 디렉토리를 제거해도 디렉토리 구성이 무효화되지는 않지만 사용자 인증을 위해 검색하는 디렉토리 목록에서 사용자 디렉토리가 제거됩니다. 검색 순서에 포함되지 않은 디렉토리는 비활성화 상태로 설정됩니다. 관리자가 검색 순서에서 사용자 디렉토리를 제거하면 다른 사용자 디렉토리에 지정된 검색 순서가 자동으로 업데이트됩니다.

> **주:**
>
> Native Directory는 검색 순서에서 제거할 수 없습니다.

검색 순서에서 사용자 디렉토리를 제거하려면 다음을 수행합니다.

1. Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 검색 순서에서 제거할 디렉토리를 선택합니다.
4. **제외**를 누릅니다.
5. **확인**을 누릅니다.
6. [디렉토리 구성] 결과 화면에서 **확인**을 누릅니다.
7. Foundation Services 및 다른 EPM System 구성요소를 재시작합니다.

검색 순서 변경

각 사용자 디렉토리에 지정되는 기본 검색 순서는 디렉토리가 구성된 시퀀스를 기반으로 합니다. 기본적으로 Native Directory는 검색 순서의 마지막 디렉토리로 설정됩니다.

검색 순서를 변경하려면 다음을 수행합니다.

1. Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행를 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. 검색 순서를 변경할 디렉토리를 선택합니다.
4. **위로 이동** 또는 **아래로 이동**을 누릅니다.
5. **확인**을 누릅니다.
6. Foundation Services, 다른 EPM System 구성요소 및 Shared Services 보안 API를 사용하는 사용자정의 애플리케이션을 재시작합니다.

보안 옵션 설정

보안 옵션은 검색 순서에 포함된 모든 사용자 디렉토리에 적용되는 글로벌 매개변수로 구성됩니다.

보안 옵션을 설정하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. **보안 옵션**을 선택합니다.
4. **보안 옵션**에서 글로벌 매개변수를 설정합니다.

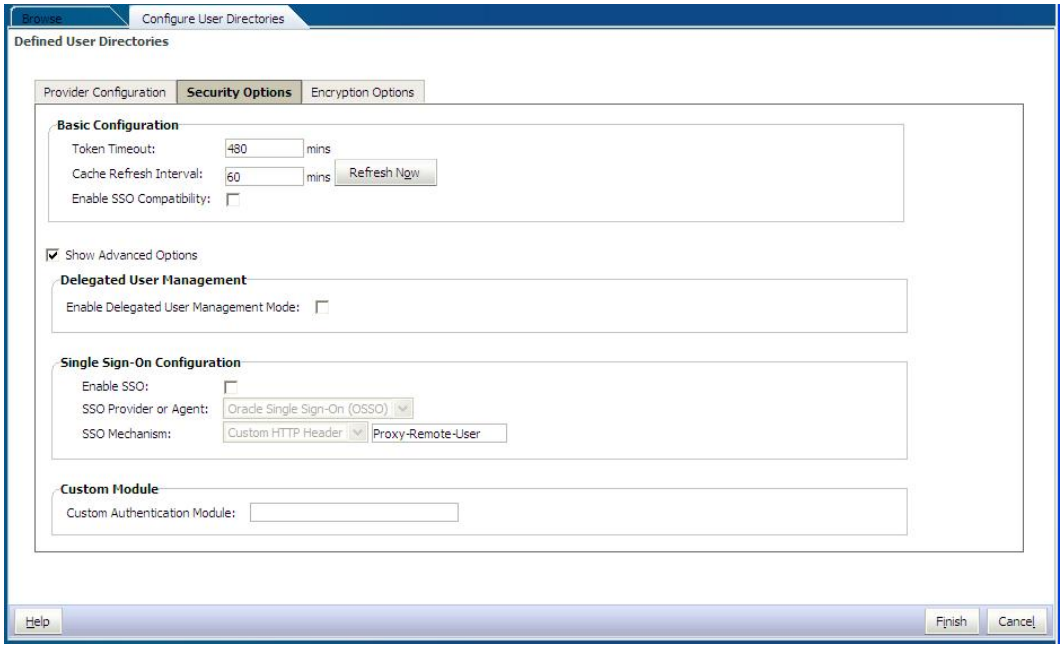


표 4-6 사용자 디렉토리의 보안 옵션

매개변수	설명
토큰 시간 초과	Oracle Enterprise Performance Management System 제품 또는 웹 ID 관리 솔루션이 실행한 SSO 토큰이 만료된 후 경과한 시간(분)입니다. 이 시간이 지나면 사용자는 다시 로그인해야 합니다. 토큰 시간 초과는 서버의 시스템 클럭을 기준으로 설정됩니다. 기본값은 480분입니다. **주:** 토큰 시간 초과는 세션 시간 초과와 동일하지 않습니다.
캐시 새로고침 간격	그룹 대 사용자 관계 데이터의 Oracle Hyperion Shared Services 캐시를 새로고치는 간격(분)입니다. 기본값은 `60`분입니다. 다음 캐시 새로고침 후에만 기존 그룹에 추가되는 새 외부 사용자 디렉토리 그룹 및 새 사용자에 대한 Shared Services 캐시 정보입니다. 새로 생성된 외부 사용자 디렉토리 그룹을 통해 프로비전닝된 사용자는 캐시를 새로 고칠 때까지 프로비전닝된 역할을 가져오지 않습니다.
지금 새로고침	그룹 대 사용자 관계 데이터가 포함된 Shared Services 캐시의 새로고침을 수동으로 시작하려면 이 버튼을 누릅니다. 외부 사용자 디렉토리에서 새 그룹을 생성하고 프로비저닝한 후 또는 새 사용자를 기존 그룹에 추가한 후 캐시 새로고침을 시작할 수 있습니다. Shared Services가 캐시에 데이터를 사용하는 호출을 요청한 후에만 캐시를 새로 고칩니다.
SSO 호환성 사용	배포가 Oracle Business Intelligence Enterprise Edition 릴리스 11.1.1.5 이전과 통합된 경우 이 옵션을 선택합니다.
위임된 사용자 관리 모드 사용	프로비저닝 작업의 분산 관리를 지원하기 위해 EPM System 제품의 위임된 사용자 관리를 활성화하는 옵션입니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "위임된 사용자 관리"를 참조하십시오.
SSO 사용	Oracle Access Manager 같은 보안 에이전트에서 SSO를 지원할 수 있도록 하는 옵션입니다.
SSO 제공자 또는 에이전트	EPM System 제품이 SSO를 받아들여야 하는 웹 ID 관리 솔루션을 선택합니다. 웹 ID 관리 솔루션(예: Kerberos)이 나열되어 있지 않은 경우에는 **기타**를 선택합니다. SSO 제공자를 선택하면 기본 SSO 메커니즘 및 이름이 자동으로 선택됩니다. 필요한 경우 SSO 메커니즘(HTTP 머리글 또는 사용자정의 로그인 클래스) 이름을 변경할 수 있습니다. SSO 제공자 또는 에이전트로 `Other`를 선택하는 경우 EPM System 지원 SSO 메커니즘을 지원하는지 확인해야 합니다. *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "지원되는 SSO 메소드"를 참조하십시오.

표 4-6 (계속) 사용자 디렉토리의 보안 옵션

매개변수	설명
SSO 메커니즘	선택한 웹 ID 관리 솔루션이 사용자의 로그인 이름을 EPM System 제품에 제공하는 데 사용하는 방법입니다. 사용할 수 있는 SSO 메소드에 대한 설명은 *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "지원되는 SSO 메소드"를 참조하십시오. • 사용자정의 `HTTP` 머리글: 보안 에이전트가 EPM System으로 전달하는 머리글 이름을 설정합니다. • 사용자정의 로그인 클래스: 인증에 대한 HTTP 요청을 처리하는 사용자정의 Java 클래스를 지정합니다. *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "사용자정의 로그인 클래스"를 참조하십시오. **주:** 사용자정의 로그인 클래스가 사용자정의 인증과 동일하지 않습니다. • `HTTP` 권한 머리글: 표준 HTTP 메커니즘입니다. • `HTTP` 요청에서 원격 사용자 가져오기: HTTP 요청에서 보안 에이전트가 원격 사용자를 채우는 경우 이 옵션을 선택합니다.
사용자정의 인증 모듈	사용자정의 인증 모듈이 선택된 모든 사용자 디렉토리의 사용자를 인증하는 데 사용되는 사용자정의 인증 모듈의 정규 Java 클래스 이름입니다(예: `com.mycompany.epm.CustomAuthenticationImpl`). 인증 모듈은 디렉토리 구성에서 해당 모듈의 사용을 활성화(기본값)한 경우에만 사용자 디렉토리에 사용됩니다. Oracle Hyperion Foundation Services에서는 사용자정의 인증 JAR 파일의 이름을 `CustomAuth.jar`로 지정해야 합니다. `CustomAuth.jar`를 *`MIDDLEWARE_HOME`*`\user_projects\domains\`*`WEBLOGIC_DOMAIN`*`\lib` 일반적으로, `C:\Oracle/Middleware/user_projects/domains/EPMSystem/lib`에서 사용할 수 있어야 합니다. 모든 클라이언트 설치에서 `CustomAuth.jar`는 *`EPM_ORACLE_HOME`*`/common/jlib/11.1.2.0` 일반적으로, `C:\Oracle\Middleware\EPMSystem11R1\common\jlib\11.1.2.0`에 있어야 합니다. JAR 파일 내의 모든 패키지 구조와 클래스 이름을 사용할 수 있습니다. 자세한 내용은 *Oracle Enterprise Performance Management System 보안 구성 가이드*의 "사용자정의 인증 모듈 사용"을 참조하십시오.

5. **확인**을 누릅니다.
6. Foundation Services 및 다른 EPM System 구성요소를 재시작합니다.

암호화 키 다시 생성

Oracle Enterprise Performance Management System은 다음 키를 사용하여 보안을 보장합니다.

- 단일 사인 온 토큰 암호화 키 - EPM System SSO 토큰을 암호화하거나 암호 해독하는 데 사용됩니다. 이 키는 Oracle Hyperion Shared Services Registry에 저장됩니다
- 인증된 서비스 키 - EPM System 구성요소에서 SSO 토큰을 요청하는 서비스의 인증을 확인하는 데 사용합니다
- 제공자 구성 암호화 키 - EPM System 보안에서 구성된 외부 사용자 디렉토리와 바인딩하기 위해 사용하는 비밀번호(LDAP 사용 가능 사용자 디렉토리에 대한 사용자 DN 비밀번호)를 암호화하는 데 사용됩니다. 이 비밀번호는 외부 사용자 디렉토리를 구성하는 동안 설정됩니다.

EPM System 보안을 강화하려면 이러한 키를 주기적으로 변경합니다. Oracle Hyperion Shared Services 및 EPM System의 보안 하위 시스템은 128비트 키 강도로 AES 암호화를 사용합니다.

> **주의:**
>
> 싱글 사인온 암호화 키를 재생성하는 경우 Oracle Hyperion Financial Management 및 Oracle Hyperion Profitability and Cost Management에서 사용하는 태스크 플로우가 무효화됩니다. 키를 다시 생성하고 난 후 태스크 플로우를 재검증하려면 해당 태스크 플로우를 열고 저장합니다.

단일 사인온 암호화 키, 제공자 구성 키 또는 인증된 서비스 키를 재생성하려면 다음을 수행합니다.

1. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다. Shared Services Console 실행을 참조하십시오.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. **암호화 옵션**을 선택합니다.
4. **암호화 옵션**에서 재생성할 키를 선택합니다.

표 4-7 EPM System 암호화 옵션

옵션	설명
단일 사인온 토큰	EPM System SSO 토큰을 암호화하거나 암호 해독하는 데 사용되는 암호화 키를 재생성하려면 선택합니다. **보안 옵션**에서 **SSO 호환성 사용**이 선택된 경우 다음 버튼 중 하나를 선택합니다. • **새 키 생성** - 새 SSO 토큰 암호화 키를 생성합니다 • **기본값으로 재설정** - 기본 SSO 토큰 암호화 키를 복원합니다 **주:** 기본 암호화 키로 되돌리는 경우 모든 EPM System 호스트 컴퓨터에서 기존 키 저장소 파일(*EPM_ORACLE_HOME*/common/CSS/ssHandlerTK)을 삭제해야 합니다.

표 4-7 (계속) EPM System 암호화 옵션

옵션	설명
인증된 서비스 키	EPM System 구성요소에서 SSO 토큰을 요청하는 서비스의 인증을 확인하는 데 사용하는 인증된 인증 키를 재생성하려면 이 옵션을 선택합니다.
제공자 구성 키	EPM System 보안에서 구성된 외부 사용자 디렉토리와 바인딩하기 위해 사용하는 비밀번호(LDAP 사용 가능 사용자 디렉토리에 대한 사용자 DN 비밀번호)를 암호화하는 데 사용되는 키를 재생성하려면 이 옵션을 선택합니다. 이 비밀번호는 외부 사용자 디렉토리를 구성하는 동안 설정됩니다.

5. **확인**을 누릅니다.
6. 새 SSO 암호화 키를 생성하도록 선택한 경우 이 단계를 완료합니다.
   a. **다운로드**를 누릅니다.
   b. **확인**을 눌러 새 SSO 암호화 키를 지원하는 키 저장소 파일인 `ssHandlerTK`를 Oracle Hyperion Foundation Services를 호스트하는 서버의 폴더에 저장합니다.
   c. 모든 EPM System 호스트 컴퓨터의 *EPM_ORACLE_HOME*`/common/CSS`에 `ssHandlerTK`를 복사합니다.
7. Foundation Services 및 다른 EPM System 구성요소를 재시작합니다.

특수 문자 사용

Active Directory 및 기타 LDAP 기반 사용자 디렉토리에서는 DN, 사용자 이름, 역할, 그룹 이름과 같은 엔티티에 특수 문자를 사용할 수 있습니다. Oracle Hyperion Shared Services에서 이러한 문자를 인식하기 위해 특수 처리가 필요할 수도 있습니다.

일반적으로 사용자 디렉토리 설정(예: 기본 DN 및 사용자와 그룹 URL)에서 특수 문자를 지정하는 경우 이스케이프 문자를 사용해야 합니다. 다음 테이블에는 사용자 이름, 그룹 이름, 사용자 URL, 그룹 URL, 사용자 DN의 OU 값에 사용할 수 있는 특수 문자가 나열되어 있습니다.

표 4-8 지원되는 특수 문자

문자	이름 또는 의미	문자	이름 또는 의미
(	여는 괄호	$	달러
)	닫는 괄호	+	더하기
"	따옴표	&	앰퍼샌드
'	작은따옴표	\	백슬래시
,	쉼표	^	캐럿
=	다음과 같음	;	세미콜론
<	다음보다 작음	#	파운드
>	다음보다 큼	@	앳

> **주:**
>
> 기본 DN 내에 있는 조직 단위 이름에는 /(슬래시)를 사용하지 마십시오.

- 로그인 사용자 속성 값에는 특수 문자를 사용할 수 없습니다.
- 별표(*)는 사용자 이름, 그룹 이름, 사용자 및 그룹 URL, 사용자 DN의 OU 이름에 사용할 수 없습니다.
- 속성 값에는 특수 문자의 조합을 포함할 수 없습니다.
- 앰퍼샌드(&)는 이스케이프 문자 없이 사용할 수 있습니다. Active Directory 설정의 경우 &는 `&`로 지정해야 합니다.
- 사용자 및 그룹 이름에 백슬래시(\)와 슬래시(/)를 함께 사용할 수 없습니다. 예를 들어, `test/\user`, `new\test/user`와 같은 이름은 지원되지 않습니다.

표 4-9 이스케이프할 필요 없는 문자

문자	이름 또는 의미	문자	이름 또는 의미
(	여는 괄호	'	작은따옴표
)	닫는 괄호	^	캐럿
$	달러	@	앳
&	앰퍼샌드		

> **주:**
>
> &는 `&`로 표시해야 합니다.

이러한 문자를 사용자 디렉토리 설정(사용자 이름, 그룹 이름, 사용자 URL, 그룹 URL, 사용자 DN)에 사용하는 경우 이스케이프해야 합니다.

표 4-10 사용자 디렉토리 구성 설정에서 특수 문자 이스케이프

특수 문자	이스케이프	예제 설정	이스케이프된 예
쉼표(,)	백슬래시(\)	`ou=test,ou`	`ou=test\,ou`
더하기 기호(+)	백슬래시(\)	`ou=test+ou`	`ou=test\+ou`
같음(=)	백슬래시(\)	`ou=test=ou`	`ou=test\=ou`
파운드(#)	백슬래시(\)	`ou=test#ou`	`ou=test\#ou`
세미콜론(;)	백슬래시(\)	`ou=test;ou`	`ou=test\;ou`
보다 작음(<)	백슬래시(\)	`ou=test<ou`	`ou=test\<ou`
보다 큼(>)	백슬래시(\)	`ou=test>ou`	`ou=test\>ou`
따옴표(")	백슬래시 2개(\\)	`ou=test"ou`	`ou=test\\"ou`
백슬래시(\)	백슬래시 3개(\\\)	`ou=test\ou`	`ou=test\\\\ou`

주:

- 사용자 DN에서 따옴표(")는 하나의 백슬래시로 이스케이프되어야 합니다. 예를 들어 `ou=test"ou`는 `ou=test\"ou`로 지정되어야 합니다.
- 사용자 DN에서 백슬래시(\)는 하나의 백슬래시로 이스케이프되어야 합니다. 예를 들어 `ou=test\ou`는 `ou=test\\ou`로 지정되어야 합니다.

주의:

사용자 URL을 지정하지 않은 경우 RDN 루트에 생성된 사용자는 /(슬래시) 또는 \(백슬래시)를 포함할 수 없습니다. 마찬가지로, 그룹 URL을 지정하지 않은 경우 RDN 루트에 생성된 그룹 이름에 이러한 문자를 사용할 수 없습니다. 예를 들어 `OU=child\ou,OU=parent/ou` 또는 `OU=child/ou,OU=parent\ou`와 같은 그룹 이름은 지원되지 않습니다. 사용자 디렉토리 구성에서 `ID` 속성으로 고유한 속성을 사용하고 있다면 이러한 이슈는 해당되지 않습니다.

Native Directory의 특수 문자

특수 문자는 Native Directory의 사용자 및 그룹 이름에서 지원됩니다.

표 4-11 지원되는 특수 문자: Native Directory

문자	이름 또는 의미	문자	이름 또는 의미
@	앳	,	쉼표
#	파운드	=	다음과 같음
$	달러	+	더하기
^	캐럿	;	세미콜론
(	여는 괄호	!	느낌표
)	닫는 괄호	%	퍼센트
'	작은따옴표		

5

사용자정의 인증 모듈 사용

참조:

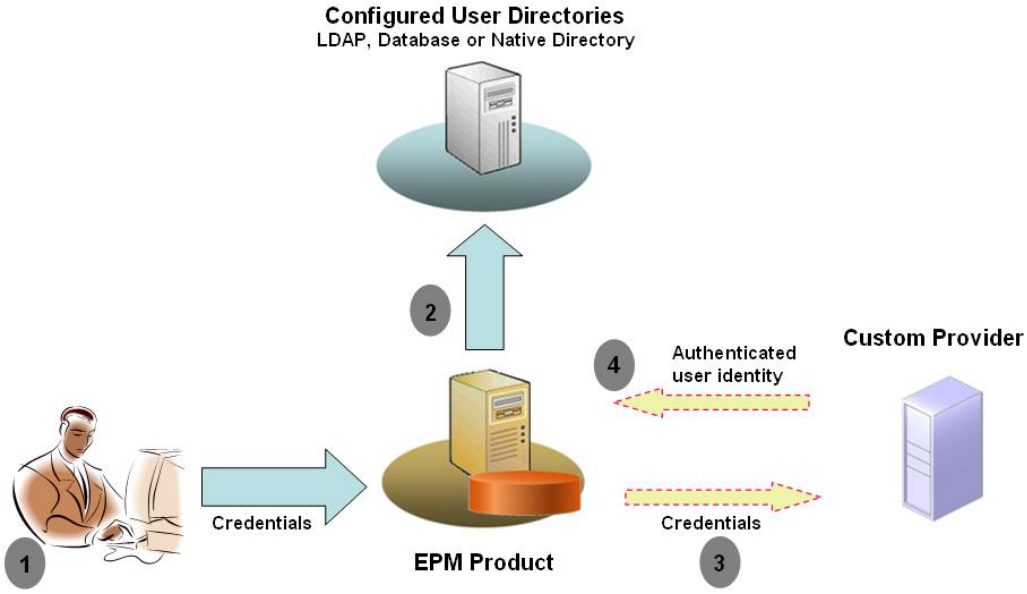
개요

사용자정의 인증 모듈은 고객이 Oracle Enterprise Performance Management System 사용자를 인증하기 위해 개발하고 구현하는 Java 모듈입니다. 일반적으로 EPM System 제품은 로그온 화면을 사용하여 사용자 인증에 사용되는 사용자 이름과 비밀번호를 캡처합니다. EPM System 인증을 사용하지 않고 사용자정의 인증 모듈을 사용하여 사용자를 인증하고 인증된 사용자 인증서를 추가 처리를 위해 EPM System에 전달할 수 있습니다. 사용자정의 인증 모듈 구현에는 EPM System 제품 수정이 포함되지 않습니다.

Thick Client(예: Oracle Smart View for Office 및 Oracle Essbase Studio)와 Thin Client(예: Oracle Hyperion Enterprise Performance Management Workspace)가 모두 포함된 사용자정의 인증 모듈을 사용할 수 있습니다.

사용자정의 인증 모듈은 EPM System 제품에 로그인할 때 사용자가 입력하는 정보를 사용합니다. 사용자 디렉토리에 대해 사용으로 설정된 경우 사용자정의 인증 모듈을 통해 사용자를 인증합니다. 사용자 인증에 성공하면 사용자정의 인증 모듈이 EPM System으로 해당 사용자 이름을 반환합니다.

다음 그림은 샘플 사용자정의 인증 시나리오를 보여 줍니다.



예를 들어 RSA SecurID 인프라를 사용자정의 제공자로 사용하여 EPM System에 대한 투명하고 강력한 인증을 보장할 수 있습니다. 개요는 다음과 같습니다.

1. 사용자가 인증서(일반적으로 사용자 이름과 비밀번호)를 입력하여 EPM System 제품에 액세스합니다. 이러한 인증서를 통해 사용자정의 인증 모듈에서 사용하는 제공자에 대해 사용자가 고유하게 확인되어야 합니다. 예를 들어 RSA SecurID 인프라를 사용하여 사용자를 인증하는 경우 사용자는 RSA 사용자 ID 및 PIN(EPM System 사용자 ID 및 비밀번호가 아님)을 입력합니다.
2. EPM System은 검색 순서(검색 순서)에 따라 구성된 사용자 디렉토리를 순환하여 해당 사용자를 찾습니다.
   - 현재 사용자 디렉토리가 사용자정의 인증에 대해 구성되지 않은 경우 EPM System은 EPM System 인증을 통해 사용자를 찾고 인증하려고 합니다.
   - 사용자 디렉토리가 사용자정의 인증에 대해 구성된 경우 EPM System은 사용자정의 모듈에 인증 프로세스를 위임합니다.
3. EPM System이 사용자정의 모듈에 인증을 위임하는 경우 사용자정의 인증 모듈은 인증서를 수락하고 자체 논리를 사용하여 사용자정의 제공자(예: RSA SecurID 인프라)에 대해 직접 사용자 인증을 수행합니다.
4. 사용자정의 인증 모듈에서 해당 제공자에 대해 사용자를 인증하는 경우 EPM System으로 사용자 이름을 반환하거나 Java 예외를 반환합니다.

   사용자정의 인증 모듈에서 반환한 사용자 이름은 사용자정의 인증에 대해 사용으로 설정된 사용자 디렉토리의 사용자 이름과 동일해야 합니다.
   - 사용자정의 인증 모듈이 사용자 이름을 반환하는 경우 EPM System은 사용자정의 인증에 대해 사용으로 설정된 사용자 디렉토리에서 사용자를 찾습니다. 이 단계에서 EPM System은 사용자정의 인증에 대해 구성되지 않은 사용자 디렉토리는 검색하지 않습니다.
   - 사용자정의 인증 모듈이 예외를 생성하거나 null 사용자를 반환하는 경우 EPM System은 사용자정의 인증에 대해 사용으로 설정되지 않은, 검색 순서의 나머지 사용자 디렉토리에서 사용자를 계속 검색합니다. 인증서와 일치하는 사용자를 찾지 못하면 EPM System에서 오류를 표시합니다.

사용 사례 예 및 제한 사항

사용자정의 인증 구현 시나리오는 다음과 같은 작업으로 구성되어 있습니다.

- 일회성 비밀번호 지원 추가
- RACF(Resource Access Control Facility)에 대한 인증 수행
- 단순 LDAP 바인드 대신 LDAP 사용 사용자 디렉토리에 SASL(Simple Authentication and Security Layer) 바인드 추가

사용자정의 인증 모듈을 구현하는 경우 시도/응답 메커니즘을 사용하는 인증이 제대로 작동하지 않을 수 있습니다. 사용자정의 인증 모듈에서 발생한 사용자정의 메시지는 클라이언트로 전달되지 않습니다. 예를 들어 Oracle Hyperion Enterprise Performance Management Workspace 같은 클라이언트는 오류 메시지를 대체하여 일반 메시지를 표시하므로 다음 시나리오는 적합하지 않습니다.

- 두 개의 연속 RSA SecurID PIN
- 비밀번호의 첫번째 문자, 마지막 문자 및 세번째 문자 입력과 같은 요청을 사용하는 비밀번호 변수

사전 필수 조건

- 사용자정의 인증 모듈 라이브러리를 포함하는, 이름이 `CustomAuth.jar`인 완전하게 테스트된 Java 아카이브. `CustomAuth.jar`는 `com.hyperion.css` 패키지에 정의된 공용 인터페이스 `CSSCustomAuthenticationIF`를 표준 Oracle Hyperion Shared Services API의 일부로 구현해야 합니다. http://download.oracle.com/docs/cd/E12825_01/epm.111/epm_security_api_11111/client/com/hyperion/css/CSSCustomAuthenticationIF.html을 참조하십시오.
- Shared Services에 Shared Services 관리자로 액세스할 수 있는 권한

디자인 및 코딩 고려 사항

검색 순서

Native Directory 외에도 여러 사용자 디렉토리를 Oracle Hyperion Shared Services에서 구성할 수 있습니다. 모든 구성된 사용자 디렉토리에 기본 검색 순서 위치가 지정됩니다. Oracle Hyperion Shared Services Console에서 검색 순서를 수정할 수 있습니다. Native Directory를 제외하고는 구성된 사용자 디렉토리를 검색 순서에서 제거할 수 있습니다. Oracle Enterprise Performance Management System에서는 검색 순서에 포함되지 않은 사용자 디렉토리를 사용하지 않습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

검색 순서에 따라 EPM System이 사용자를 인증하기 위해 사용자 디렉토리를 순환하는 순서가 결정됩니다. 사용자 디렉토리에서 사용자가 인증되면 EPM System이 검색을 중지하고 사용자를 반환합니다. 검색 순서의 사용자 디렉토리에 대해 사용자를 인증할 수 없으면 EPM System에서 인증을 거부하고 오류를 반환합니다.

사용자정의 인증이 검색 순서에 미치는 영향

사용자정의 인증은 EPM System 보안이 검색 순서를 해석하는 방법에 영향을 줍니다.

사용자정의 인증 모듈이 사용자 이름을 반환하는 경우 EPM System은 사용자정의 인증에 대해 사용으로 설정된 사용자 디렉토리에서만 사용자를 찾습니다. 이 단계에서 EPM System은 사용자정의 인증에 대해 구성되지 않은 사용자 디렉토리는 무시합니다.

사용자정의 인증 플로우 이해

사용자정의 인증 플로우를 살펴보기 위해 다음 사용 사례 시나리오를 사용합니다.

- 사용 사례 시나리오 1
- 사용 사례 시나리오 2
- 사용 사례 시나리오 3

사용 사례 시나리오 1

다음 테이블에는 EPM System 사용자 디렉토리 구성과 이 시나리오에 사용된 검색 순서가 자세히 설명되어 있습니다. 이 시나리오에서는 사용자정의 인증 모듈이 RSA 인프라를 사용하여 사용자를 인증한다고 가정합니다.

표 5-1 시나리오 1의 설정

사용자 디렉토리 유형 및 이름	검색 순서	사용자정의 인증	샘플 사용자 이름	비밀번호[1]
Native Directory	1	사용 안함	test_user_1 test_user_2 test_user_3	password
LDAP 사용 SunONE_West	2	사용 안함	test_ldap1 test_ldap_2 test_user_3 test_ldap_4	ldappassword
LDAP 사용 SunONE_East	3	사용	test_ldap1 test_ldap_2 test_user_3	SunONE에서는 ldappassword, 사용자정의 모듈에서는 RSA PIN

1 간단히 모든 사용자가 동일한 사용자 디렉토리 비밀번호를 사용한다고 가정합니다.

인증 프로세스를 시작하려면 사용자가 EPM System 제품의 로그온 화면에 사용자 이름과 비밀번호를 입력합니다. 이 시나리오에서는 사용자정의 인증 모듈이 다음 작업을 수행합니다.

- 사용자 이름과 RSA PIN을 사용자 인증서로 수락합니다.
- 사용자 이름을 *username@providername* 형식(예: test_ldap_2@SunONE_East)으로 EPM System 보안에 반환합니다.

표 5-2 사용자 상호 작용 및 결과

사용자 이름 및 비밀번호	인증 결과	로그인 사용자 디렉토리
test_user_1/password	성공	Native Directory
test_user_3/password	성공	Native Directory

표 5-2 (계속) 사용자 상호 작용 및 결과

사용자 이름 및 비밀번호	인증 결과	로그인 사용자 디렉토리
test_user_3/ ldappassword	성공	SunONE_West(검색 순서 2)[1]
test_user_3/RSA PIN	성공	SunONE_East(검색 순서 3)[2]
test_ldap_2/ ldappassword	성공	SunONE_West(검색 순서 2)
test_ldap_4/RSA PIN	실패 EPM System이 인증 오류를 표시합니다.[3]	

1 사용자가 EPM System 인증서를 입력했으므로 사용자정의 인증에서 이 사용자를 인증할 수 없습니다. EPM System은 사용자정의 인증에 대해 사용으로 설정되지 않은 사용자 디렉토리에서만 이 사용자를 확인할 수 있습니다. 이 사용자는 Native Directory(검색 순서 번호 1)에는 없고 SunONE West(검색 순서 번호 2)에서 확인됩니다.

2 EPM System은 Native Directory(검색 순서 번호 1) 또는 SunONE West(검색 순서 번호 2)에서 이 사용자를 찾지 못합니다. 사용자정의 인증 모듈에서 RSA 서버에 대해 사용자를 검증하고 test_user_3@SunONE_EAST를 EPM System에 반환합니다. EPM System은 사용자정의 인증 사용 사용자 디렉토리인 SunONE East(검색 순서 번호 3)에서 사용자를 찾습니다.

3 사용자정의 모듈에서 인증된 사용자는 모두 검색 순서에 포함된 사용자정의 인증 사용 사용자 디렉토리에 있는 것이 좋습니다. 사용자정의 인증 모듈에서 반환한 사용자 이름이 검색 순서에 포함된 사용자정의 인증 사용 사용자 디렉토리에 없으면 로그인이 실패합니다.

사용 사례 시나리오 2

다음 테이블에는 EPM System 사용자 디렉토리 구성과 이 시나리오에 사용된 검색 순서가 자세히 설명되어 있습니다. 이 시나리오에서는 사용자정의 인증 모듈이 RSA 인프라를 사용하여 사용자를 인증한다고 가정합니다.

이 시나리오에서는 사용자정의 인증 모듈이 다음 작업을 수행합니다.

- 사용자 이름과 RSA PIN을 사용자 인증서로 수락합니다.
- 사용자 이름(예: test_ldap_2)을 EPM System 보안에 반환합니다.

표 5-3 샘플 검색 순서

사용자 디렉토리	검색 순서	사용자정의 인증	샘플 사용자 이름	비밀번호[1]
Native Directory	1	사용 안함	test_user_1 test_user_2 test_user_3	password
LDAP 사용(예: SunONE)	2	사용	test_ldap1 test_ldap2 test_user_3	SunONE에서는 ldappassword, 사용자정의 모듈에서는 RSA PIN

1 간단히 모든 사용자가 동일한 사용자 디렉토리 비밀번호를 사용한다고 가정합니다.

인증 프로세스를 시작하려면 사용자가 EPM System 제품의 로그인 화면에 사용자 이름과 비밀번호를 입력합니다.

표 5-4 사용자 상호 작용 및 결과

사용자 이름 및 비밀번호	로그인 결과	로그인 사용자 디렉토리
test_user_1/password	성공	Native Directory
test_user_3/password	성공	Native Directory
test_user_3/ldappassword	실패	SunONE[1]
test_user_3/RSA_PIN	성공	SunONE[2]

1 비밀번호 불일치 때문에 Native Directory에 대한 사용자 인증이 실패합니다. 사용된 비밀번호가 적합한 RSA PIN이 아니므로 사용자정의 인증 모듈을 사용한 사용자 인증이 실패합니다. EPM System은 SunONE(검색 순서 2)에서 이 사용자를 인증하려고 시도하지 않습니다. 이 디렉토리에서는 사용자정의 인증 설정이 EPM System 인증을 대체하기 때문입니다.
2 비밀번호 불일치 때문에 Native Directory에 대한 사용자 인증이 실패합니다. 사용자정의 인증 모듈에서 사용자를 인증하고 사용자 이름 test_user_3을 EPM System으로 반환합니다.

사용 사례 시나리오 3

다음 테이블에는 EPM System 사용자 디렉토리 구성과 이 시나리오에 사용된 검색 순서가 자세히 설명되어 있습니다. 이 시나리오에서는 사용자정의 인증 모듈이 RSA 인프라를 사용하여 사용자를 인증한다고 가정합니다.

이러한 시나리오에서는 명확성을 위해 사용자정의 인증 모듈이 사용자 이름을 username@providername 형식(예: test_ldap_4@SunONE)으로 반환하는 것이 좋습니다.

표 5-5 샘플 검색 순서

사용자 디렉토리	검색 순서	사용자정의 인증	샘플 사용자 이름	비밀번호[1]
Native Directory	1	사용	test_user_1 test_user_2 test_user_3	RSA_PIN
LDAP 사용(예: MSAD)	2	사용 안함	test_ldap1 test_ldap4 test_user_3	ldappassword
LDAP 사용(예: SunONE)	3	사용	test_ldap1 test_ldap4 test_user_3	SunONE에서는 ldappassword, 사용자정의 모듈에서는 RSA_PIN

1 간단히 모든 사용자가 동일한 사용자 디렉토리 비밀번호를 사용한다고 가정합니다.

인증 프로세스를 시작하려면 사용자가 EPM System 제품의 로그온 화면에 사용자 이름과 비밀번호를 입력합니다.

표 5-6 사용자 상호 작용 및 결과

사용자 이름 및 비밀번호	인증 결과	로그인 사용자 디렉토리
test_user_1/password	성공	Native Directory
test_user_3/RSA_PIN	성공	Native Directory

표 5-6 (계속) 사용자 상호 작용 및 결과

사용자 이름 및 비밀번호	인증 결과	로그인 사용자 디렉토리
test_user_3/ldappassword	성공	MSAD(검색 순서 2)
test_ldap_4/ldappassword	성공	MSAD(검색 순서 2)
test_ldap_4/RSA PIN	성공	SunONE(검색 순서 3)

사용자 디렉토리 및 사용자정의 인증 모듈

사용자정의 인증 모듈을 사용하기 위해 EPM System 사용자 및 그룹 정보가 포함된 사용자 디렉토리에서 사용자정의 모듈에 인증을 위임하도록 개별적으로 구성할 수 있습니다.

사용자정의 모듈을 사용하여 인증된 EPM System 사용자는 검색 순서(검색 순서 참조)에 포함된 사용자 디렉토리 중 하나에 있어야 합니다. 또한, 사용자 디렉토리는 사용자정의 모듈에 인증을 위임하도록 구성되어야 합니다.

사용자정의 제공자의 사용자 ID(예: RSA SecurID 인프라의 `1357642`)가 Shared Services에 구성된 사용자 디렉토리의 사용자 이름(예: Oracle Internet Directory의 `jDoe`)과 다를 수 있습니다. 사용자를 인증한 후에는 사용자정의 인증 모듈에서 사용자 이름 `jDoe`를 EPM System으로 반환해야 합니다.

> **주:**
>
> EPM System에 구성된 사용자 디렉토리의 사용자 이름은 사용자정의 인증 모듈에서 사용하는 사용자 디렉토리에 제공된 사용자 이름과 동일한 것이 좋습니다.

`CSSCustomAuthenticationIF` Java 인터페이스

사용자정의 인증 모듈은 `CSSCustomAuthenticationIF` Java 인터페이스를 사용하여 EPM System 보안 프레임워크와 통합해야 합니다. 사용자정의 인증이 성공하면 사용자 이름 문자열을 반환하고 인증이 실패하면 오류 메시지를 반환해야 합니다. 인증 프로세스를 완료하려면 사용자정의 인증 모듈에서 반환한 사용자 이름이 Shared Services 검색 순서에 포함된 사용자 디렉토리 중 하나에 있어야 합니다. EPM System 보안 프레임워크는 *`username@providerName`* 형식을 지원합니다.

> **주:**
>
> EPM System 보안 프레임워크에서는 사용자를 검색할 때 *(별표)를 와일드카드 문자로 해석하므로 사용자정의 인증 모듈이 반환하는 사용자 이름에 별표가 포함되지 않았는지 확인하십시오.

`CSSCustomAuthenticationIF` 인터페이스 서명은 샘플 코드 1을 참조하십시오.

사용자정의 인증 모듈(클래스 파일일 수 있음)은 `CustomAuth.jar`에 포함되어 있어야 합니다. 패키지 구조는 중요하지 않습니다.

`CSSCustomAuthenticationIF` 인터페이스에 대한 자세한 내용은 보안 API 설명서를 참조하십시오.

CSSCustomAuthenticationIF의 authenticate 메소드는 사용자정의 인증을 지원합니다. authenticate 메소드는 EPM System에 입력 매개변수로 액세스하려고 시도할 때 사용자가 입력한 인증서(사용자 이름 및 비밀번호)를 수락합니다. 사용자정의 인증이 성공하면 이 메소드는 문자열(사용자 이름)을 반환합니다. 인증이 실패하면 java.lang.Exception이 발생합니다. 메소드에서 반환되는 사용자 이름으로 Shared Services 검색 순서에 포함된 사용자 디렉토리 중 하나에서 사용자가 고유하게 확인되어야 합니다. EPM System 보안 프레임워크는 *username@providerName* 형식을 지원합니다.

> **주:**
>
> JDBC 연결 풀과 같은 리소스를 초기화하려면 클래스 구성자를 사용하십시오. 이렇게 하면 일부 인증에 대해서만 리소스를 로드하므로 성능이 향상됩니다.

사용자정의 인증 모듈 배포

하나의 Oracle Enterprise Performance Management System 배포에는 하나의 사용자정의 모듈만 지원됩니다. 검색 순서에 있는 하나 이상의 사용자 디렉토리에 대해 사용자정의 인증을 사용으로 설정할 수 있습니다.

사용자정의 인증 모듈은 com.hyperion.css 패키지에 정의된 공용 인터페이스 CSSCustomAuthenticationIF를 구현해야 합니다. 이 문서에서는 사용자가 선택한 사용자 제공자에 대해 사용자를 인증하는 논리를 정의하는, 완전한 기능을 갖춘 사용자정의 모듈이 있다고 가정합니다. 사용자정의 인증 모듈을 개발하고 테스트한 후에는 EPM System 환경에서 구현해야 합니다.

단계 개요

사용자정의 인증 코드는 오류 로깅에 log4j를 사용하지 않아야 합니다. 이전 릴리스에서 사용한 코드가 log4j를 사용하는 경우 먼저 log4j를 코드에서 제거해야 이 릴리스에서 해당 코드를 사용할 수 있습니다.

사용자정의 인증 모듈을 구현하려면 다음 단계를 완료합니다.

- Oracle Hyperion Shared Services 및 Shared Services API를 사용하는 모든 시스템을 비롯한 EPM System 제품을 중지합니다.
- 사용자정의 인증 모듈 Java 아카이브 CustomAuth.jar를 다음 배포에 복사합니다.
  - **WebLogic:** CustomAuth.jar를 *MIDDLEWARE_HOME*/user_projects/domains/*WEBLOGIC_DOMAIN*/lib 일반적으로, C:/Oracle/Middleware/user_projects/domains/EPMSystem/lib에 복사합니다.

    사용자정의 인증 모듈 구현이 포함된 릴리스 11.1.2.0 또는 11.1.2.1에서 업그레이드하는 경우 CustomAuth.jar를 *EPM_ORACLE_HOME*/common/jlib/11.1.2.0에서 *MIDDLEWARE_HOME*/user_projects/domains/*WEBLOGIC_DOMAIN*/lib로 이동합니다.
  - **모든 클라이언트 배포**: CustomAuth.jar를 모든 EPM System 클라이언트 배포의 다음 위치로 복사합니다.

    EPM_ORACLE_HOME/common/jlib/11.1.2.0 일반적으로, Oracle/Middleware/common/jlib/11.1.2.0. CustomAuth.jar 파일이 항상 *EPM_ORACLE_HOME*/common/jlib/11.1.2.0 디렉토리에 배치되는지 확인합니다.

모든 서버 및 클라이언트가 사용자정의 인증을 사용하려면 `CustomAuth.jar` 파일이 다음 2개 위치에 있어야 합니다.

* *MIDDLEWARE_HOME*`/user_projects/domains/WEBLOGIC_DOMAIN/lib`
* *EPM_ORACLE_HOME*`/common/jlib/11.1.2.0`

- Shared Services에서 사용자 디렉토리 설정을 업데이트합니다. Shared Services에서 설정 업데이트을 참조하십시오.
- Shared Services,를 시작한 다음 다른 EPM System 제품을 시작합니다.
- 구현을 테스트합니다. 배포 테스트을 참조하십시오.

Shared Services에서 설정 업데이트

기본적으로 사용자정의 인증은 모든 사용자 디렉토리에 대해 사용 안함으로 설정되어 있습니다. 특정 외부 사용자 디렉토리 또는 Native Directory에 대해 사용자정의 인증을 사용으로 설정하여 기본 동작을 대체할 수 있습니다.

사용자 디렉토리 구성 업데이트

사용자정의 인증을 사용으로 설정해야 하는 사용자 디렉토리 구성을 업데이트해야 합니다.

사용자 디렉토리 구성을 업데이트하려면 다음을 수행합니다.

1. Oracle Hyperion Foundation Services를 시작합니다.
2. Oracle Hyperion Shared Services Console에 시스템 관리자로 액세스합니다.
3. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
4. 정의된 사용자 디렉토리 화면에서 사용자정의 인증 설정을 변경할 사용자 디렉토리를 선택합니다.

> **주:**
>
> EPM System은 검색 순서에 포함된 사용자 디렉토리만 사용합니다.

5. **편집**을 누릅니다.
6. **고급 옵션 표시**를 선택합니다.
7. **사용자정의 모듈**에서 **인증 모듈**을 선택하여 현재 사용자 디렉토리에 대해 사용자정의 모듈을 사용으로 설정합니다.
8. **완료**를 누릅니다.
9. 이 절차를 반복하여 검색 순서에서 다른 사용자 디렉토리의 구성을 업데이트합니다.

보안 옵션 업데이트

다음 절차를 시작하기 전에 *EPM_ORACLE_HOME*`/user_projects/domains/`*WEBLOGIC_DOMAIN*`/lib`에서 `CustomAuth.jar`를 사용할 수 있는지 확인합니다.

보안 옵션을 업데이트하려면 다음을 수행합니다.

1. Shared Services Console에 시스템 관리자로 액세스합니다.
2. **관리**, **사용자 디렉토리 구성** 순으로 선택합니다.
3. **보안 옵션**을 선택합니다.

4. **고급 옵션 표시**를 선택합니다.
5. **인증 모듈**에는 사용자정의 인증 모듈이 선택된 모든 사용자 디렉토리에서 사용자를 인증하는 데 사용해야 하는 사용자정의 인증 모듈의 전체 클래스 이름을 입력합니다. 예를 들어 `com.mycompany.epm.CustomAuthenticationImpl`을 입력합니다.
6. **확인**을 누릅니다.

배포 테스트

Native Directory가 사용자정의 인증에 대해 구성되어 있지 않은 경우 Native Directory 사용자로 사용자정의 인증을 테스트하지 마십시오.

> **주:**
>
> 사용자정의 인증 모듈의 이슈를 파악하여 정정하는 것은 사용자의 책임입니다. Oracle에서는 사용자정의 모듈이 사용하는 사용자 디렉토리의 사용자를 EPM System 검색 순서에서 사용가능한 사용자정의 인증 사용 사용자 디렉토리의 사용자에 매핑하는 작업이 사용자정의 모듈에서 원활하게 수행된다고 가정합니다.

배포를 테스트하려면 사용자정의 모듈에서 사용하는 RSA SecurID 인프라와 같은 사용자 디렉토리의 사용자 인증서를 사용하여 EPM System에 로그인하십시오. 이러한 인증서는 EPM System 인증서와 다를 수 있습니다.

EPM System 제품에서 해당 리소스에 액세스할 수 있으면 구현에 성공한 것으로 간주됩니다. 사용자를 찾지 못했다는 오류가 표시되어도 구현이 실패했음을 나타내는 것은 아닙니다. 이러한 경우 입력한 인증서가 사용자정의 사용자 저장소에 존재하며 EPM System 검색 순서의 사용자정의 인증 사용 사용자 디렉토리 중 하나에 일치하는 사용자가 존재하는지 확인하십시오.

사용자정의 인증을 테스트하려면 다음을 수행합니다.

1. EPM System 제품이 실행 중인지 확인합니다.
2. EPM System 구성요소(예: Oracle Hyperion Enterprise Performance Management Workspace)에 액세스합니다.
3. 사용자정의 인증이 사용으로 설정되어 있는 사용자 디렉토리에 정의된 사용자로 로그인합니다.
   a. **사용자 이름**에 사용자 ID(예: RSA 사용자 ID)를 입력합니다.
   b. **비밀번호**에 비밀번호(예: RSA PIN)를 입력합니다.
   c. **로그인**을 누릅니다.
4. EPM System 제품 리소스에 액세스할 수 있는지 확인합니다.

6

EPM System 보안 가이드라인

참조:

- SSL 구현
- 관리 비밀번호 변경
- 암호화 키 다시 생성
- 데이터베이스 비밀번호 변경
- 쿠키 보호
- SSO 토큰 시간 초과 감소
- 보안 보고서 검토
- 강력한 인증을 위해 인증 시스템 사용자정의
- EPM Workspace 디버깅 유틸리티 사용 안함
- 기본 웹 서버 오류 페이지 변경
- 타사 소프트웨어 지원

SSL 구현

SSL은 데이터를 암호화하는 암호화 시스템을 사용합니다. SSL은 클라이언트와 서버 간에 보안 연결을 생성하는데, 이를 통해 데이터를 안전하게 전송할 수 있습니다.

Oracle Enterprise Performance Management System 환경을 보호하려면 SSL을 사용하여 웹 애플리케이션 및 사용자 디렉토리 연결에서 사용하는 모든 통신 채널을 보호합니다. EPM System 구성요소에서 SSL을 사용으로 설정을 참조하십시오.

또한, 방화벽을 사용하여 모든 에이전트 포트(예: Oracle Hyperion Reporting and Analysis 에이전트 포트인 포트 6861)를 보호합니다. 일반 사용자는 EPM System 에이전트 포트에 액세스할 필요가 없습니다.

관리 비밀번호 변경

기본 Native Directory 관리 사용자 계정은 모든 Oracle Hyperion Shared Services 기능에 대한 액세스를 제공합니다. 이 비밀번호는 Oracle Hyperion Foundation Services를 배포할 때 설정됩니다. 이 계정의 비밀번호는 주기적으로 변경해야 합니다.

비밀번호를 변경하려면 *관리* 사용자 계정을 편집합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자 계정 수정"을 참조하십시오.

암호화 키 다시 생성

Oracle Hyperion Shared Services Console을 사용하여 정기적으로 다음 항목을 다시 생성하십시오.

- 싱글 사인온 토큰

> **주의:**
>
> 새 키 저장소를 생성하는 경우 Oracle Hyperion Financial Management 및 Oracle Hyperion Profitability and Cost Management에서 사용하는 태스크 플로우가 무효화됩니다. 키 저장소를 다시 생성하고 난 후 태스크 플로우를 다시 유효화하려면 해당 태스크 플로우를 열고 저장합니다.

- 인증된 서비스 키
- 제공자 구성 키

암호화 키 다시 생성을 참조하십시오.

> **주:**
>
> Oracle Hyperion Shared Services 및 Oracle Enterprise Performance Management System의 보안 하위 시스템은 128비트 키 강도로 AES 암호화를 사용합니다.

데이터베이스 비밀번호 변경

모든 Oracle Enterprise Performance Management System 제품 데이터베이스의 비밀번호를 주기적으로 변경하십시오. 이 절에는 Oracle Hyperion Shared Services Registry에서 데이터베이스 비밀번호를 변경하는 절차가 자세히 설명되어 있습니다.

EPM System 제품 데이터베이스 비밀번호를 변경하는 자세한 절차는 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

Shared Services 레지스트리에서 EPM System 제품 데이터베이스 비밀번호를 변경하려면:

1. 데이터베이스 관리 콘솔을 사용하여 EPM System 제품 데이터베이스를 구성하는 데 사용된 사용자 계정의 비밀번호를 변경합니다.
2. EPM System 제품(웹 애플리케이션, 서비스 및 프로세스)을 중지합니다.
3. EPM System Configurator에서 다음 절차 중 하나를 사용하여 데이터베이스를 다시 구성합니다.

   Oracle Hyperion Shared Services만 해당:

   > **주:**
   >
   > EPM System 제품이 Shared Services와 다른 머신에 있는 분산 환경에서는 모든 서버에 대해 이 절차를 수행해야 합니다.

   a. EPM System Configurator의 Foundation 태스크에서 **데이터베이스 구성**을 선택합니다.

   b. [Shared Services 및 레지스트리 데이터베이스 구성] 페이지에서 **이전에 구성한 Shared Services 데이터베이스에 연결**을 선택합니다.

c. Shared Services 데이터베이스를 구성하는 데 사용된 사용자 계정의 새 비밀번호를 지정합니다. 다른 설정은 변경하지 마십시오.

d. 구성을 계속하고 마치면 **완료**를 누릅니다.

Shared Services 이외의 EPM System 제품:

> **주:**
>
> 현재 서버에 배포된 EPM System 제품에 대해서만 다음 단계를 수행합니다.

자세한 지침은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

4. EPM System 제품 및 서비스를 시작합니다.

쿠키 보호

Oracle Enterprise Performance Management System 웹 애플리케이션은 쿠키를 설정하여 세션을 추적합니다. 쿠키, 특히 세션 쿠키를 설정한 상태에서 서버는 브라우저가 보안 채널을 통해 쿠키를 보내도록 하는 보안 플래그를 설정할 수 있습니다. 이 동작은 세션 하이재킹 위험을 줄입니다.

> **주:**
>
> EPM System 제품이 SSL 사용 환경에 배포된 경우에만 쿠키 보안을 유지합니다.

WebLogic Server 쿠키 보안을 유지하려면 Oracle WebLogic Server 세션 기술자를 수정합니다. `session-param` 요소의 `cookieSecure` 속성 값을 `true`로 설정합니다. Oracle Fusion Middleware Programming Security for Oracle WebLogic Server 11g의 웹 애플리케이션 보안을 참조하십시오.

SSO 토큰 시간 초과 감소

기본 SSO 토큰 시간 초과는 480분입니다. 토큰이 노출된 경우 토큰 재사용을 최소화하려면 SSO 토큰 시간 초과를 줄여야 합니다(예: 60분). *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "보안 옵션 설정"을 참조하십시오.

보안 보고서 검토

보안 보고서에는 감사가 구성된 보안 태스크에 관련된 감사 정보가 포함되어 있습니다. 정기적으로 Oracle Hyperion Shared Services Console에서 이 보고서를 생성하고 검토하여 특히 Oracle Enterprise Performance Management System 제품의 실패한 로그인 시도와 프로비저닝 변경사항을 확인합니다. **세부 뷰**를 보고서 생성 옵션으로 선택하여 보고서 데이터를 수정된 속성과 새 속성 값에 따라 그룹화합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "보고서 생성"을 참조하십시오.

강력한 인증을 위해 인증 시스템 사용자정의

사용자정의 인증 모듈을 사용하여 강력한 인증을 EPM System에 추가할 수 있습니다. 예를 들어 요청 없는 응답 모드로 RSA SecurID 2단계 인증을 사용할 수 있습니다. 사용자정의 인증 모듈은 씬 클라이언트와 씩 클라이언트에 투명하므로 클라이언트측 배포를 변경하지 않아도 됩니다. 사용자정의 인증 모듈 사용을 참조하십시오.

EPM Workspace 디버깅 유틸리티 사용 안함

- Oracle Hyperion Enterprise Performance Management Workspace는 문제 해결을 위해 압축되지 않은 JavaScript 파일과 함께 제공됩니다. 보안을 위해 이러한 압축되지 않은 JavaScript 파일을 프로덕션 환경에서 제거해야 합니다.
  - *EPM_ORACLE_HOME*`/common/epmstatic/wspace/js/` 디렉토리의 백업 복사본을 생성합니다.
  - *`DIRECTORY_NAME`*`.js` 파일을 제외한 `.js` 파일을 *EPM_ORACLE_HOME*`/common/epmstatic/wspace/js`의 각 하위 디렉토리에서 삭제합니다.

    각 하위 디렉토리에는 디렉토리 이름이 포함된 `.js` 파일이 있습니다. 예를 들어 *EPM_ORACLE_HOME*`/common/epmstatic/wspace/js/com/hyperion/bpm/web/common`에는 `Common.js`가 포함되어 있습니다. 디렉토리 이름이 포함된 파일을 제외하고 모든 `.js` 파일을 제거합니다(이 경우 `Common.js`).
- EPM Workspace는 일부 디버그 유틸리티 및 테스트 애플리케이션을 제공합니다. EPM Workspace가 디버그 모드로 배포되면 이러한 유틸리티 및 애플리케이션에 액세스할 수 있습니다. 보안을 위해 관리자는 EPM Workspace에서 클라이언트측 디버깅을 꺼야 합니다.

  디버깅 모드를 끄려면 다음을 수행합니다.

  1. EPM Workspace에 관리자로 로그인합니다.
  2. **탐색**, **관리**, **Workspace 서버 설정** 순으로 선택합니다.
  3. Workspace 서버 설정의 **클라이언트 디버그 사용**에서 **아니요**를 선택합니다.
  4. **확인**을 누릅니다.

기본 웹 서버 오류 페이지 변경

애플리케이션 서버가 요청을 수락할 수 없으면 백엔드 애플리케이션 서버의 웹 서버 플러그인(예: Oracle WebLogic Server의 Oracle HTTP Server 플러그인)이 플러그인 빌드 정보를 표시하는 기본 오류 페이지를 반환합니다. 웹 서버는 다른 경우에도 기본 오류 페이지를 표시합니다. 공격자가 이 정보를 사용하여 공용 웹 사이트에서 알려진 취약점을 찾을 수 있습니다.

오류 페이지에 서버 버전, 서버 유형, 플러그인 빌드 날짜, 플러그인 유형과 같은 프로덕션 시스템 구성요소에 대한 정보가 포함되지 않도록 웹 애플리케이션 서버 플러그인 및 웹 서버의 오류 페이지를 사용자정의합니다. 자세한 내용은 애플리케이션 서버 및 웹 서버 공급업체 설명서를 참조하십시오.

타사 소프트웨어 지원

Oracle은 타사 공급업체의 역호환성 어설션을 인정하고 지원합니다. 따라서 공급업체가 역호환성을 어설션하는 경우 후속 유지관리 릴리스와 서비스 팩이 사용될 수 있습니다. 비호환성 문제가 발견되면 Oracle은 배포해야 할 제품에 대한 패치 릴리스를 명시(및 지원 매트릭스에서 호환되지 않는 버전 제거)하거나 Oracle 제품에 대한 유지 관리 릴리스 또는 서비스 픽스를 제공합니다.

서버측 업데이트: 타사 서버측 구성요소에 대한 업그레이드 지원은 후속 유지 관리 릴리스 정책에 의해 관리됩니다. 일반적으로 Oracle은 현재 지원되는 릴리스 서비스 팩의 다음 유지관리 릴리스로 타사 서버측 구성요소를 업그레이드하는 것을 지원합니다. 다음 주요 릴리스에 대한 업그레이드는 지원되지 않습니다.

클라이언트측 업데이트: Oracle은 타사 클라이언트 구성요소의 다음 주요 릴리스에 대한 업데이트를 포함하여 클라이언트 구성요소에 대한 자동 업데이트를 지원합니다. 예를 들어 브라우저 JRE 버전을 현재 지원되는 JRE 버전으로 업데이트할 수 있습니다.

A

사용자정의 인증 샘플 코드

샘플 코드 1

> **주:**
>
> 사용자정의 인증 코드는 오류 로깅에 log4j를 사용하지 않아야 합니다. 이전 릴리스에서 사용한 사용자정의 인증 코드가 log4j를 사용하는 경우 먼저 log4j를 코드에서 제거해야 이 릴리스에서 해당 코드를 사용할 수 있습니다.

다음 코드 스니펫은 사용자정의 모듈의 빈 구현입니다.

```
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;
      }
}
```

입력 매개변수:

- 컨텍스트: 로케일 정보 키-값 쌍이 포함된 맵

- 사용자 이름: 사용자정의 모듈이 사용자를 인증하는 사용자 디렉토리에 대해 사용자를 고유하게 확인하는 ID. 사용자가 Oracle Enterprise Performance Management System 구성요소에 로그인할 때 이 매개변수 값을 입력합니다.
- 비밀번호: 사용자정의 모듈이 사용자를 인증하는 사용자 디렉토리의 사용자에 대해 설정된 비밀번호. 사용자가 EPM System 구성요소에 로그인할 때 이 매개변수 값을 입력합니다.

샘플 코드 2

다음 샘플 코드에서는 플랫 파일에 포함된 사용자 이름 및 비밀번호를 사용하는 사용자에 대한 사용자정의 인증을 보여 줍니다. 사용자정의 인증 작업을 수행하려면 클래스 구성자에서 사용자 및 비밀번호 목록을 초기화해야 합니다.

```
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");
          }
}
```

샘플 코드 2 데이터 파일

데이터 파일 이름이 샘플 코드에 사용된 이름인 `datafile.txt`로 지정되었고 사용자가 생성하는 Java 아카이브에 포함되어 있는지 확인하십시오.

샘플 코드 2를 통해 구현된 사용자정의 인증 모듈을 지원하기 위해 사용자정의 사용자 디렉토리로 사용되는 플랫 파일의 콘텐츠로 다음을 사용합니다(샘플 코드 2 참조).

```
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
```

사용자 이름을 *username@providername* 형식으로 반환하려면 사용자정의 사용자 디렉토리로 사용되는 플랫 파일의 콘텐츠로 다음을 사용합니다.

```
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

사용자정의 로그인 클래스 구현

Oracle Enterprise Performance Management System은 x509 인증서에서 사용자 ID(DN)를 추출하도록 com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl을 제공합니다.

DN 이외의 인증서에 있는 속성에서 사용자 ID가 파생되어야 하는 경우 이 부록에 설명된 대로 com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl과 유사한 사용자정의 로그인 클래스를 개발 및 구현해야 합니다.

사용자정의 로그인 클래스 샘플 코드

이 샘플 코드는 기본 com.hyperion.css.sso.agent.X509CertificateSecurityAgentImpl의 구현을 보여 줍니다. 일반적으로 DN이 아닌 인증서 속성에서 사용자 이름을 파생시키려면 이 구현의 parseCertificate(String sCertificate) 메소드를 사용자정의해야 합니다.

```
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 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
```

사용자정의 로그인 클래스 배포

사용자정의 로그인 클래스를 구현하려면 다음 단계를 완료합니다.

1. 사용자정의 로그인 클래스를 생성하고 테스트합니다. 코드에 `log4j`에 대한 참조가 없는지 확인합니다. 사용자정의 로그인 클래스 샘플 코드을 참조하십시오.

   사용자정의 클래스에는 어떤 이름이든 사용할 수 있습니다.

2. 사용자정의 로그인 클래스를 `CustomAuth.jar`에 패키징합니다.

3. `CustomAuth.jar`를 배포에 복사합니다.

   - **WebLogic:** `CustomAuth.jar`를 *`MIDDLEWARE_HOME`*`/user_projects/domains/`*`WEBLOGIC_DOMAIN`*`/lib` 일반적으로, `Oracle/Middleware/user_projects/domains/EPMSystem/lib`에 복사합니다.

     > **주:**
     >
     > 사용자정의 로그인 클래스 구현이 포함된 릴리스 11.1.2.0 또는 11.1.2.1에서 업그레이드하는 경우 `CustomAuth.jar`를 *`EPM_ORACLE_HOME`*`/common/jlib/11.1.2.0`에서 *`MIDDLEWARE_HOME`*`/user_projects/domains/`*`WEBLOGIC_DOMAIN`*`/lib`로 이동합니다.

   - **클라이언트 배포**: `CustomAuth.jar`를 모든 Oracle Enterprise Performance Management System 클라이언트 배포의 다음 위치로 복사합니다.

     `EPM_ORACLE_HOME``/common/jlib/11.1.2.0` 일반적으로, `Oracle/Middleware/common/jlib/11.1.2.0`

사용자정의 로그인 클래스를 사용하는 경우 클라이언트 인증서 인증을 사용으로 설정하는 것이 좋습니다.

C

사용자 디렉토리 간 사용자 및 그룹 마이그레이션

개요

프로비저닝된 Oracle Enterprise Performance Management System 사용자의 사용자 및 그룹 ID가 손상되도록 하는 시나리오가 많이 있습니다. 구성요소에 사용가능한 프로비저닝 정보가 손상된 경우 EPM System 구성요소에 액세스할 수 없습니다. 손상된 프로비저닝 데이터가 생성될 수 있는 시나리오는 다음과 같습니다.

- 사용자 디렉토리 사용 중지: 조직에서 사용자를 다른 사용자 디렉토리로 이동한 후에는 사용자 디렉토리 사용을 중지할 수 있습니다.
- 버전 업그레이드: 사용자 디렉토리 버전 업그레이드에서는 필요한 호스트 머신 이름 또는 운영 체제 환경이 변경될 수 있습니다.
- 공급업체 변경: 조직에서 다른 공급업체의 사용자 디렉토리를 위해 사용자 디렉토리 사용을 중단할 수 있습니다. 예를 들어 조직에서 해당 Oracle Internet Directory를 SunONE Directory Server로 바꿀 수 있습니다.

> **주:**
>
> - 이 부록에서는 단계적으로 중단되고 있는 사용자 디렉토리를 *소스* 사용자 디렉토리라고 하고, 사용자 계정을 이동한 대상 사용자 디렉토리를 *타겟* 사용자 디렉토리라고 합니다.
> - 이 마이그레이션 절차에서는 사용자 계정을 소스 사용자 디렉토리에서 타겟 사용자 디렉토리로 마이그레이션하는 작업을 지원하지 않으며 EPM 애플리케이션에서의 연관만 지원합니다. 타겟 사용자 디렉토리에서 사용자를 수동으로 생성해야 합니다. 이 프로세스는 Native Directory를 포함하여 소스 사용자 디렉토리의 사용자에게 적용됩니다.
>
>   Hyperion Shared Services로 구성된 소스 사용자 디렉토리에 Native Directory 그룹을 제외한 그룹이 있는 경우, 해당 그룹을 타겟 사용자 디렉토리에 생성해야 합니다.

사전 필수 조건

- 해당 프로비저닝 데이터를 사용자 디렉토리에서 마이그레이션하고 있는 Oracle Enterprise Performance Management System 사용자 및 그룹을 타겟 사용자 디렉토리에서 사용할 수 있어야 합니다.

  소스 사용자 디렉토리에 존재하는 그룹 관계는 타겟 사용자 디렉토리에서 유지관리되어야 합니다.
- EPM System 사용자의 사용자 이름은 소스 및 타겟 사용자 디렉토리에서 동일해야 합니다.

마이그레이션 절차

Native Directory 데이터 익스포트

소스 환경에서 다음 단계를 수행합니다.

Oracle Hyperion Enterprise Performance Management System Lifecycle Management를 사용하여 Native Directory에서 다음 공유 서비스 아티팩트만 익스포트합니다.

- Native Directory 그룹
- 지정된 역할
- 위임된 목록

Lifecycle Management에서는 여러 익스포트 파일을, 보통 *EPM_ORACLE_INSTANCE*/`import_export/`*USER_NAME*`/`*EXPORT_DIR*`/resource/Native Directory`에 생성합니다. 여기서, *USER_NAME*은 익스포트 작업을 수행한 사용자의 ID(예: `admin`)이며 *EXPORT_DIR*은 익스포트 디렉토리의 이름입니다. 일반적으로 생성되는 파일은 다음과 같습니다.

- `Groups.csv`
- `Assigned Roles.csv`
- `Delegated Lists.csv`
- 배포된 각 애플리케이션의 `Assigned Roles/`*PROD_NAME*`.csv`. 여기서, *PROD_NAME*은 Oracle Enterprise Performance Management System 구성요소(예: `Shared Services`)의 이름입니다.

> **주:**
>
> - Lifecycle Management를 사용하여 데이터를 익스포트하는 방법에 대한 자세한 지침은 *Oracle Enterprise Performance Management System Lifecycle Management 가이드*를 참조하십시오.
> - `Users.csv` 파일이 익스포트되지 않았는지 확인합니다.

아티팩트를 익스포트한 후 마이그레이션 상태 보고서에 마지막 익스포트 작업의 상태가 `Completed`로 표시되는지 확인하십시오.

Native Directory 데이터를 익스포트하려면:

1. Oracle Hyperion Shared Services Console의 뷰 창에 있는 **Foundation** 애플리케이션 그룹에 **Shared Services** 애플리케이션을 선택합니다.
2. 마이그레이션하려면 아래 목록에서 필수 아티팩트만 선택합니다.
   - Native Directory 그룹
   - 지정된 역할
   - 위임된 목록
3. **익스포트**를 누릅니다.

4. 익스포트 아카이브의 이름을 입력합니다. 기본값은 `admin` *DATE*(예: `admin 13-03-18`)입니다.
5. **익스포트**를 누릅니다.

Native Directory 데이터 임포트

타겟 환경에서 다음 단계를 수행합니다.

1. 수동 생성:
   a. 소스 사용자 디렉토리와 유사한 타겟 외부 사용자 디렉토리의 사용자.
   b. Native Directory 그룹을 제외하고 소스 사용자 디렉토리와 유사한 타겟 외부 사용자 디렉토리의 그룹.
2. 타겟 사용자 디렉토리 구성
   사용자 계정을 소스 사용자 디렉토리에서 다른 사용자 디렉토리로 이동한 경우 타겟 사용자 디렉토리를 EPM System의 외부 사용자 디렉토리로 추가합니다. 예를 들어 Oracle Internet Directory에서 SunONE Directory Server로 사용자 계정을 이동한 경우 SunONE Directory Server를 외부 사용자 디렉토리로 추가합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "3장, 사용자 디렉토리 구성"을 참조하십시오.

   > **주:**
   > 타겟 사용자 디렉토리에 소스 사용자 디렉토리에서 데이터가 마이그레이션되는 모든 EPM System 사용자에 대한 사용자 계정 및 그룹이 포함되어 있는지 확인합니다.

   사용자를 외부 사용자 디렉토리로 이미 정의된 사용자 디렉토리로 이동한 경우 Oracle Hyperion Shared Services에 사용자 계정이 표시되는지 확인합니다. Shared Services Console에서 사용자를 검색하여 이를 수행할 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자, 그룹, 역할, 위임된 목록 검색"을 참조하십시오.

   타겟 사용자 디렉토리를 외부 사용자 디렉토리로 구성할 때 로그인 속성 등록정보가 해당 값이 소스 사용자 디렉토리에서 사용자 이름으로 원래 사용된 속성을 가리키는지 확인하십시오. 사전 필수 조건을 참조하십시오.
3. 타겟 사용자 디렉토리를 검색 순서 맨 위로 이동

   > **주:**
   > 타겟 사용자 디렉토리 이름이 소스 디렉토리 이름과 동일한 경우 EPM System 구성에서 소스 사용자 디렉토리를 삭제해야 합니다.

   Shared Services는 기존 디렉토리에 지정된 검색 순서보다 새로 추가된 사용자 디렉토리에 더 낮은 검색 순서 우선순위를 지정합니다. 소스 사용자 디렉토리보다 타겟 사용자 디렉토리의 검색 순서 우선순위가 더 높도록 검색 순서를 변경하십시오. 이러한 순서를 통해 Shared Services는 소스를 검색하기 전에 타겟 사용자 디렉토리의 사용자를 검색할 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*의 "사용자 디렉토리 검색 순서 관리"를 참조하십시오.
4. Oracle Hyperion Foundation Services 및 다른 EPM System 구성요소를 재시작하여 변경한 내용을 적용합니다.

5. Native Directory 데이터 임포트(소스 환경에서 임포트됨)
`create/update` 옵션으로 Lifecycle Management를 실행하여 이전에 Native Directory에서 (아래 나열된 대로) 익스포트했던 데이터를 임포트합니다.
   - `Groups.csv`
   - `Assigned Roles.csv`
   - `Delegated Lists.csv`

> **주:**
>
> - Lifecycle Management를 사용하여 데이터를 임포트하는 방법에 대한 자세한 지침은 *Oracle Enterprise Performance Management System Lifecycle Management 가이드*를 참조하십시오.
> - `Users.csv` 파일이 임포트되지 않았는지 확인합니다.

데이터를 임포트한 후 마이그레이션 상태 보고서에 마지막 임포트 작업의 상태가 `Completed`로 표시되는지 확인하십시오.

Native Directory 데이터를 임포트하려면:

a. Shared Services Console의 뷰 창에서 **파일 시스템**을 확장합니다.
b. 임포트 파일의 파일 시스템 위치를 선택합니다.
c. 프로비저닝 정보를 임포트할 아티팩트 유형을 선택합니다.
d. **임포트**를 누릅니다.
e. **확인**을 누릅니다.

제품별 업데이트

> **주의:**
>
> 제품별 업데이트를 시작하기 전에 Oracle Enterprise Performance Management System 구성요소에서 사용하는 저장소의 사용자 및 그룹 데이터를 백업하는 것이 좋습니다. 로컬 제품 저장소에서 정보를 업데이트한 후에는 백업에서만 로컬 제품 저장소의 이전 사용자 및 그룹 데이터로 되돌릴 수 있습니다.

Planning

Oracle Hyperion Planning은 Planning 저장소에 프로비저닝된 사용자와 그룹에 대한 정보를 저장합니다. 사용자 디렉토리에서 사용자 및 그룹을 마이그레이션한 결과 Native Directory에서 사용자 ID가 변경된 경우 사용자/그룹 마이그레이션을 선택하여 Planning 저장소의 정보를 Native Directory의 정보와 동기화해야 합니다.이 버튼은 데이터 양식, 멤버 및 태스크 목록에 대한 액세스 권한을 지정하는 경우 Planning에서 사용할 수 있습니다.

Financial Management

Oracle Hyperion Financial Management에서는 로컬 Financial Management 저장소의 객체에 액세스하기 위해 프로비저닝된 사용자와 그룹에 대한 정보를 기록합니다. 사용자 디렉토리에서 사용자 및 그룹을 마이그레이션한 결과 Native Directory의 사용자 및 그룹 정보가 변경된 경우 Financial Management 저장소의 정보를 Native Directory의 정보와 동기화해야 합니다.