

Oracleホワイト・ペーパー
2014年10月

Oracle HTTP Server 12c Release 1 – 技術概要

目次

はじめに.....	3
Oracle HTTP Server機能概要.....	3
Oracle HTTP Serverアーキテクチャ概要.....	5
Oracle HTTP Serverのモジュール.....	6
Oracle HTTP Server 12c管理概要.....	8
WebLogic Management Framework.....	8
WebLogic ServerドメインによるOracle HTTP Server 12cの管理.....	9
スタンドアロン・ドメインによるOracle HTTP Server 12cの管理.....	10
WebLogic ServerドメインによるOracle HTTP Server 12cの構成.....	11
開発/テスト・シナリオ.....	12
本番シナリオ.....	15
スタンドアロン・ドメインによるOracle HTTP Server 12cの構成.....	15
WebLogic ServerドメインによるOracle HTTP Server 12cの管理.....	16
WebLogic管理サーバーの起動.....	16
WebLogicノード・マネージャの起動.....	17
Oracle HTTP Server 12cインスタンスの起動.....	18
WebLogic Scripting ToolによるOracle HTTP Server 12cの管理.....	18
Oracle HTTP Server 12c (スタンドアロン・ドメイン).....	18
Oracle HTTP Server 12c (WebLogic Serverドメイン).....	19
Oracle HTTP Server 12cインスタンスの作成.....	19
Oracle HTTP Server 12cインスタンスの削除.....	20
Oracle HTTP Server 12cインスタンスの起動.....	21
Oracle HTTP Server 12cインスタンスの停止.....	22
Oracle HTTP Server 12cインスタンスのソフト・リスタート.....	23
Oracle HTTP Server 12cインスタンスのステータスの取得.....	23
Oracle HTTP Server 12c監視統計の取得.....	23
Oracle HTTP Server構成のプロパティへのアクセス.....	24
付録.....	25
WebLogic JRF 12c Release 1のインストール.....	25
Oracle HTTP Server 12c Release 1 (12.1.x) のインストール (連結インストール).....	25
Oracle HTTP Server 12c Release 1 (12.1.x) のインストール (スタンドアロン).....	26
結論.....	28

はじめに

Oracle HTTP Server 12cは、インターネットのHTTP/HTTPSプロトコルを扱い、堅牢なコンテンツ提供機能とリバース・プロキシ機能によってWeb層の責務を果たすための主要インフラストラクチャを確立します。

Oracle HTTP Serverのおもな特徴は次のとおりです。

- **テクノロジー** : Oracle HTTP Serverは、実績あるオープンソース・テクノロジーのApache HTTP Serverをベースとしています。Oracle HTTP Server 11g (11.1.1.x) および12c Release 1 (12.1.2/12.1.3) リリースは、Apache 2.2リリースの最新バージョンをベースとしています。Oracle HTTP Serverには、オープンソースのApache HTTP Serverでは通常利用できないような、Fusion Middlewareデプロイメントと関連の深い追加機能も搭載されています。
- **コンテンツ提供** : Oracle HTTP Serverには、静的Webコンテンツ (HTML5、画像など) をエンドユーザーに提供する機能や、スループット向上のために静的Webコンテンツをメモリ内にキャッシュする機能があります。さらに、PHP、Python、RubyなどのWebスクリプト技術に基づいて、組込みのCGI/FastCGIモジュールによって動的Webコンテンツを提供する機能もあります。
- **リバース・プロキシ** : ‘リバース・プロキシ’ として動作し、受信したHTTP/HTTPSリクエストをバックエンド・コンテンツ・サーバー (アプリケーション・サーバーまたは単に ‘オリジン’ サーバー) に透過的に受け渡します。管理者はこの機能により、受信したHTTP/HTTPSリクエストの認証と認可を行ってから、1台以上のコンテンツ・サーバー (アプリケーション・サーバーまたは単に ‘オリジン’ サーバー) に効率的にルーティングするように、Oracle HTTP Server内のルールを構成できます。
- **統合** : Oracle HTTP Server 12cは、Fusion Middlewareアプリケーションとエンド・ツー・エンドで統合されます。管理者はFusion Middlewareアプリケーション・デプロイメント全体のプロビジョニング、構成、管理、**監視**を行うことができます。また、Oracle HTTP Serverは、スタンドアロン・デプロイメントのシナリオにも対応し、組込みのリバース・プロキシ用の ‘mod_proxy’ モジュールによって、Oracle HTTP Server以外のサーバーもフロントエンドとして使用できます。

Oracle HTTP Server 12c以降では**WebLogic Management Framework**を利用することで、Oracle HTTP Server、Oracle WebLogic Server、他のFusion Middlewareアプリケーション内で管理の一貫性が維持されます。

このドキュメントでは、Oracle HTTP Serverの技術概要と、Oracle HTTP Server 12c Release 1 (12.1.x) リリースで利用できる管理関連の新機能について説明します。

Oracle HTTP Server 機能概要

Oracle HTTP Server 12.1.3は、Oracle HTTP Server 12c Release 1の最新リリースです。

ここでは、最新リリースであるOracle HTTP Server 12.1.3に搭載されるすべての機能について、概要を簡単に説明します。

- **プロトコル** : Oracle HTTP Serverは、複数のユーザーから同時に送信されるHTTP 1.0/1.1およびHTTPS (SSL 3.0、TLS 1.2) のトラフィックを処理し、応答できます。

- **大規模仮想ホスティング**：顧客（大企業またはISP）が、単一のOracle HTTP Serverインスタンス内で名前の異なる複数のWebサイトをホストできるようにします。管理者は、複数のWebサイトや関連するビジネス・アプリケーションを単一のOracle HTTP Serverインスタンス内に統合できます。通常はこの機能を実現するために、Apache HTTP ServerベースのOracle HTTP Server内に複数の**仮想ホスト**（バーチャルホスト）を構成します。詳しくは、Apache HTTP Serverドキュメントの[バーチャルホスト](#)に関する説明を参照してください。
- **URLリライト**：管理者はこの機能により、外部に示されるURLに影響を及ぼさずに、Webサイトをすばやく再構成できます。詳しくは、Apache HTTP Serverドキュメントの[URLリライト](#)に関する説明を参照してください。
- **コンテンツ提供**：Oracle HTTP Serverは通常、クライアントからリクエストされたコンテンツを次のいずれかの方法で提供します。ディスクのコンテンツを送信するか、CGI/FastCGIスクリプトの実行により生成された出力を送信するか、バックエンド・アプリケーション・サーバーによって生成された出力を送信します（リバース・プロキシ）。
- **コンテンツ・キャッシュ**：管理者はApache HTTP Serverが提供する組込みのキャッシュ機能を利用し、ディスクまたはメモリにコンテンツをキャッシュして、Webサイトのスループットや応答時間を改善できます。詳しくは、Apache HTTP Serverドキュメントの[コンテンツ・キャッシュ](#)に関する説明を参照してください。
- **リバース・プロキシ**：Oracle HTTP Serverには、Apache HTTP Serverが提供する組込みのリバース・プロキシ・モジュール（mod_proxyなど）が搭載されています。これを利用すれば、フロントエンドとして構成して、受信リクエストをバックエンド・アプリケーション・サーバーに受け渡して、レスポンスをクライアントに返すことができます。このセットアップでは、Oracle HTTP Serverで企業のポリシー（認証、認可など）を適用してから、バックエンドにあるHTTP準拠のアプリケーション・サーバーにリクエストを送信できます。
- **WebLogicプロキシ・プラグイン**：Oracle HTTP Serverには、Apache用のWebLogic Webサーバー・プロキシ・プラグインがバンドルされています。この機能によって、Oracle HTTP ServerはバックエンドのWebLogic管理対象サーバー/クラスタに対して、受信トラフィックのロードバランシングを効率的に実行できます。さらに、Oracle HTTP Server 12cでアクセスできる管理コンソールには、このモジュールの管理機能も備わっています。
- **認証と認可**：Oracle HTTP Serverには、Apache HTTP Serverが提供する、認証および認可のポリシー適用に関連したモジュールがバンドルされています。各種認証/認可ポリシーの適用方法について詳しくは、Apache HTTP Serverドキュメントの[認証と認可](#)に関する説明を参照してください。
- **シングル・サインオン**：従業員や企業のITチームは、企業のイントラネット内にデプロイされた複数のWebアプリケーションで、従業員の認証情報（ユーザー名、パスワード）を共有し、アクセスしたいと考える傾向にあります。管理者は、Oracle HTTP Server内にバンドルされたWeb Gateモジュール（mod_webgate）を利用して、Oracle Access Managementによるシングル・サインオンを有効にすることができます。この機能によって、Oracle Single Sign-Onソリューションのプロビジョニング、構成、管理を行うことができます。

- **SSL** : Oracle HTTP Serverは、HTTPSトラフィックの処理の際にSSL接続に対応し、SSL接続を終了できる設計になっています。Apache HTTP Serverのオープンソース・バージョンとは異なり、Oracle HTTP ServerはそのようなSSLトラフィックを独自のSSLライブラリ (mod_oss1モジュール) によって処理して、企業の厳格なセキュリティ要件に対応します。これらのSSLライブラリでは、TLS 1.0/1.1プロトコルを含む最新バージョンのSSLプロトコルをサポートしています。そのため、管理者はOracle HTTP Server 12cを利用して、SSL v3プロトコルまたはTLS 1.0/1.1プロトコルによってWebサイトのHTTPSトラフィックを処理できます。詳しくは、[SSL暗号化の製品ドキュメント](#)を参照してください。Oracle HTTP Server 12cには以下の追加機能が搭載されています。
 - **ハードウェア・オフロードのサポート** - AES暗号化トランザクションを、Intel x86プロセッサやSPARC T4/T5プロセッサ上のハードウェア暗号化機構にオフロードして、HTTPトラフィックの全体的なスループットを大幅に改善できます。
 - **FIPS 140-2 Level 1** - FIPS 140-1認定のランタイム・ライブラリが提供されます。Oracle HTTP ServerベースのWeb Tierソリューションを利用することで、FIPS 140-1 (Level 1) 認定に準拠できます。このFIPS 140-1認定ランタイム・ライブラリをOracle HTTP Serverで有効にする方法について詳しくは、Oracle HTTP Serverドキュメントの[SSL構成](#)に関する説明を参照してください。
 - **リソースごとの可変セキュリティ** - 異なるWebリソース (Webコンテンツをホストするディレクトリなど) を、異なる強度の暗号化によって保護できます。
 - **リバース・プロキシ・シナリオでのSSL** - 組込みのmod_proxyを使用するかWebLogic Webサーバー・プラグインのmod_wl_ohsを使用してOracle HTTP Serverをリバース・プロキシ・モードでデプロイした場合に、受信したSSL接続を終了できるだけだけでなく、バックエンドのオリジン・サーバーとのSSL接続を開始することもできます。
- **管理** : Oracle HTTP Server 12cの管理ではWebLogic Management Frameworkを利用することで、他のFusion Middlewareアプリケーションとの管理の一貫性が維持されます。Oracle HTTP Server 12cの管理では、1台以上のホスト (マシン) で稼働する1つ以上のOracle HTTP Serverインスタンスを管理できます。ユーザーは管理コンソールを使用してOracle HTTP Serverを管理できます。このコンソールは一般的に**Fusion Middleware Control**と呼ばれています。このコンソールを利用するには、WebLogic Serverドメイン内にOracle HTTP Serverをインストールするシナリオ (略して '連結インストール' シナリオ) を選択する必要があります。詳しくは、次のセクションを参照してください。

WebLogic ServerドメインによるOracle HTTP Server 12cの管理

Oracle HTTP Serverアーキテクチャ概要

Oracle HTTP Server 12c Release 1 (12.1.2/12.1.3) は、Apache HTTP Server 2.2アーキテクチャをベースとしており、Apache HTTP Server 2.2の高度にモジュール化されたアーキテクチャを利用しています。

Apache HTTP Serverアーキテクチャは、端的に言えばコアにある[マルチプロセッシング・モジュール \(MPM\)](#)のいずれかを利用するアーキテクチャであり、HTTPトラフィックの処理機能を含む他の機能はすべてモジュールとして設計されます。Oracle HTTP ServerはApache HTTP Serverとは異なり、コア・アーキテクチャとして、HTTPトラフィックを処理するための[Worker MPM](#)のみを利用します。このようなアーキテクチャによって、Oracle HTTP Serverはコンテンツ提供に加えて、リバース・プロキシのユースケースにもシームレスに対処できます。

Apache HTTP Serverが受信HTTPリクエストを処理する際に、適切なリクエスト/レスポンス・ライフサイクルで、さまざまなモジュール（およびそのAPI）が適切に呼び出されます。

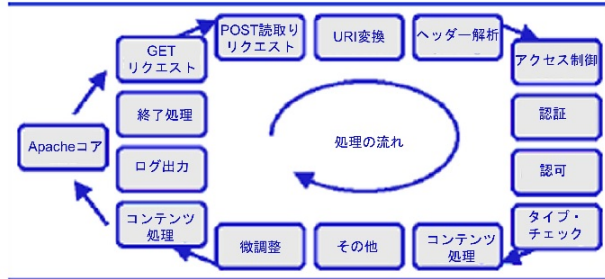


図 - HTTPリクエスト/レスポンス・サイクル

顧客はこのモジュール式アーキテクチャを利用し、Apache HTTP Server 2.xとの互換性がある独自のカスタム・モジュールを開発して、Oracle HTTP Serverが提供する機能を補完できます。ただし、Oracle Supportがお客様からOracle HTTP Server関連の問題について報告を受けた際に、トラブルシューティングの実行時にそのようなカスタム・モジュールがその問題と干渉すると考えられる十分な根拠がある場合は、カスタム・モジュールをOracle HTTP Server構成から削除するように依頼することがあります。

Oracle HTTP Serverのモジュール

Apache HTTP Serverのモジュールは、コアへのプラグイン・モジュールであり、Apache HTTP Serverの基本機能を、コンテンツ・サーバーまたは（リバース）プロキシ・サーバーとして拡張するためのものです。

Oracle HTTP Serverには、標準のApache 2.2リリースに搭載される標準モジュールのほとんどが搭載されています。それに加えて、Oracle HTTP Serverは、Fusion Middlewareアーキテクチャとシームレスに統合するための以下の追加モジュールを提供しています。

詳しくは、[Oracle HTTP Server 12cのモジュールに関するドキュメント](#)を参照してください。

表 - Oracle HTTP Server 12c Release 1で利用できる追加モジュール

モジュール名	モジュールのバージョン	モジュールの説明
Mod_wl_ohs	12.1.x.0.0	Oracle HTTP Server用WebLogic Webサーバー・プロキシ・プラグイン - Oracle HTTP Serverをフロントエンドとして利用し、HTTPトラフィックをバックエンドのWebLogic管理対象サーバーまたはクラスタに効率的に受け渡すことができます。 このモジュールは、従来の 'mod_proxy' モジュールの機能を受け継ぎ、さらにWebLogic Serverアプリケーションのフロントエンドとして機能します。
Mod_ossll	12.1.x.0.0	Oracle SSLモジュール。Oracle HTTP Serverに送信されたSSLトラフィックをインターセプトして、SSLハンドシェイクを処理し、その後SSLトラフィックを適切に終了させます。 このモジュールは従来の 'mod_ssl' モジュールを受け継ぎ、さらにSSLト

		ラフィックの処理も行います。このモジュールは、RSAが提供する暗号化ライブラリを利用するため、FIPS 140-2 (Level 1) 認定となります。詳しくは、製品ドキュメントを参照してください。
Mod_Perl		Oracle HTTP Server内でPerlスクリプトを実行するためのモジュールです。 このモジュールは現在「非推奨」であり、今後のリリースで削除される予定です。
Mod_plsql	2.1.22	Oracle HTTP ServerからOracle Databaseに接続するためのモジュールです。管理者は、Oracle Databaseで提供されるストアド・プロシージャ機能を利用してWebアプリケーションを作成できます。 このモジュールは現在「非推奨」であり、今後のリリースで削除される予定です。
Mod_fastcgi	2.4.6	FastCGIプロトコルをサポートし、ユーザーが従来のCGIアプリケーションを効率的に実行できるようにするモジュールです。起動時および初期化時のオーバーヘッドが除去されます。 このモジュールは現在「非推奨」であり、今後のリリースで削除されるか、別のモジュールに置き換わる予定です。
Mod_security	2.6	オープンソースのModSecurityモジュールをベースとしたWebアプリケーション・ファイアウォール・モジュール。 管理者はModSecurityのルール機能を利用して、SQLインジェクション、Cross Site Request Forgery (CSRF)、コマンド・インジェクションといった既知のセキュリティの脆弱性からWebアプリケーションを保護するための適切なルールを開発し、デプロイできます。 また、オープンソースの OWASP Core Rule Set (事前作成済みルール・セット) を基盤として利用し、既知のセキュリティの脆弱性からアプリケーションを保護することもできます。
Mod_dms	12.1.x.0.0	すべてのOracle HTTP Serverインスタンスの監視情報をOracle Fusion Middleware ControlコンソールまたはOracle Enterprise Manager Cloud Controlに送信し、ファーム全体にわたって統一的な監視情報を提供します。

Oracle HTTP Server 12c管理概要

ここでは、Oracle HTTP Server 12cの管理に関する概要を説明し、*Oracle HTTP Server 12c*を*WebLogic Management Framework*によって管理する際のさまざまなシナリオや手順について紹介します。

Oracle HTTP Server 12cでは、Oracle HTTP Serverや他のCloud Application Foundation 12cインフラストラクチャを、WebLogic Management Frameworkを使用して管理できます。このような統合管理手法によって、管理者がインフラストラクチャ全体を一貫性のあるインターフェースで管理できるようにしています。

WebLogic Management Framework

WebLogic Management Framework 12cは、Oracle HTTP Serverや他のFusion Middlewareアプリケーション (WebLogic Serverなど) のようなシステム・コンポーネントを管理するためのフレームワークであり、Oracle HTTP Serverについては2種類の管理方式に対応しています。

- **WebLogic Server ドメイン** - 1台のWebLogic管理サーバー、0台以上のWebLogic管理対象サーバー、0台以上のシステム・コンポーネント・インスタンス (例: Oracle HTTP Serverインスタンス) により構成されます。複数の物理マシンにわたって1つのWebLogic Serverドメインを構成でき、管理サーバーがそれらのインスタンスやマシンを一元管理します。これらの特徴から、WebLogic Serverドメインはご使用のシステム・コンポーネント (Oracle HTTP Serverなど) とJava EEコンポーネント (Fusion Middlewareアプリケーションなど) の最適な統合を実現できます。スタンドアロン・ドメインを使用したOracle HTTP Server管理を選択するに至るおもなユースケースは以下のとおりです。
 - Oracle HTTP ServerをFusion Middlewareドメインのフロントエンドとして使用したい (SOAアプリケーションなど)。
 - ブラウザベースのFusion Middleware Controlコンソールが提供する管理/監視機能が必要である。
 - WebLogic/JRFライブラリとOracle HTTP Serverを同じ場所にインストールできる (連結インストール)。
- **スタンドアロン・ドメイン** - 1つ以上のシステム・コンポーネント (Oracle HTTP Serverなど) のコンテナとして機能します。このドメインは、1つのOracle WebLogic Serverドメインと同様のディレクトリ構造になりますが、管理サーバーや管理対象サーバーは含まれません。スタンドアロン・ドメインの場合、WebLogic Management Frameworkが、ノード・マネージャ、WebLogic Scripting Tool (WLST)、構成ウィザード、パック/アンパックなどのWebLogic管理ツールをサポートします。スタンドアロン・ドメインを使用したOracle HTTP Server管理を選択するに至るおもなユースケースは以下のとおりです。
 - Oracle HTTP ServerをFusion Middlewareドメインのフロントエンドとして使用したくない。ただしこの場合でも、WebLogicプラグイン構成ファイルを手動で構成することで、WebLogic管理対象サーバー/クラスタのフロントエンドとして使用できる。
 - Fusion Middleware Controlコンソールが提供する管理/監視機能は不要である。ただしこの場合でも、管理者はWebLogic Scripting Tool (WLST) を利用してOracle HTTP Serverを管理でき、Apache HTTP Serverが提供するツール ('mod_status' モジュールなど) を使用してOracle HTTP Serverのパフォーマンスを監視できる。

- 。 Oracle HTTP Serverを“非武装地帯”（DMZ：内部ファイアウォールと外部ファイアウォールの間にあるゾーン）にプロビジョニングしたい。そのため、インストールのフットプリントと依存性を最小限に抑えたい。また、余分な管理ポート（一般的にはWebLogic Management Framework内のWebLogicノード・マネージャが使用するポート）を開きたくない。

WebLogic Server ドメインによるOracle HTTP Server 12cの管理

このシナリオでは、Fusion Middleware ControlコンソールまたはWLSTを使用してOracle HTTP Server構成とサーバー・ライフ・サイクルを管理できます。

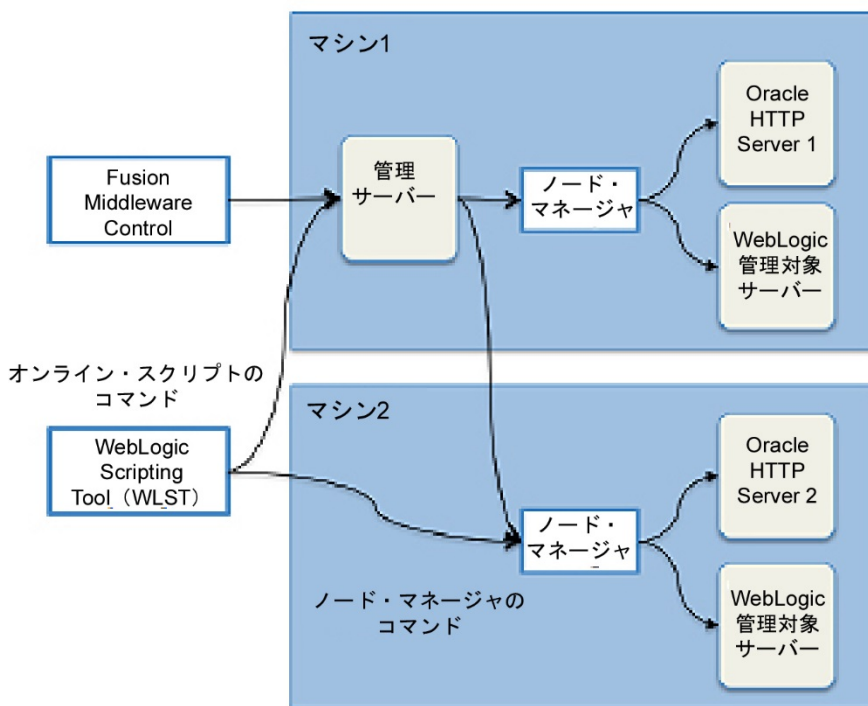


図 - WebLogic Server ドメインによるOracle HTTP Server の管理

このシナリオは、ユーザーがOracle HTTP Server、WebLogic、JRFを同じ場所にインストールする場合に適用され、ブラウザベースの管理コンソール（Fusion Middleware Controlコンソール）またはコマンドラインベースのWebLogic Scripting Tool (WLST) によってOracle HTTP Serverを管理できます。

このシナリオを実装するには、以下の流れでOracle HTTP Serverを管理する必要があります。

- システムにJDK7がインストールされていない場合、Java SE 7 Development Kitをダウンロードして、ご使用のマシンにインストールする必要があります。Java SE 7は[こちら](#)からダウンロードできます。
- インストール先（例：Oracle Middlewareホーム）にJRFを有効にしたWebLogic 12.1.3がない場合は、JRFを有効にしたWebLogic (Fusion Middleware Infrastructure 12.1.3) を[こちら](#)からダウンロードできます。詳しくは、[Fusion Middleware Infrastructureドメインの構成](#)に関する説明を参照してください。
- 上記のJRFを有効にしたWebLogic Serverと同じインストール先（同じOracle Middlewareホーム）に、Oracle HTTP Server 12c (12.1.3) をインストールします。Oracle HTTP Serverのインストールについて詳しくは、「[Oracle HTTP Server 12c Release 1のインストール](#)」を参照してください。
- [Fusion Middleware Infrastructureドメインの構成](#)に関する説明に従って、Oracle HTTP Server 12cドメインを構成します。詳しくは、「[Oracle HTTP Server 12c Release 1 \(12.1.x\) のインストール \(連結インストール\)](#)」を参照してください。
- WebLogic管理サーバーを起動し、<http://localhost:<wls-admin-server-port>/em>にアクセスして、Fusion Middlewareコンソールを起動します。これで、Fusion Middleware Controlコンソールを使用してOracle HTTP Serverを管理できるようになります。詳しくは、次のセクションを参照してください。

スタンドアロン・ドメインによるOracle HTTP Server 12cの管理

このシナリオでは、依存性を最小限に抑えてOracle HTTP Serverをプロビジョニングし、WebLogicノード・マネージャの統合インターフェースを使用してサーバー・ライフ・サイクルを管理できます。

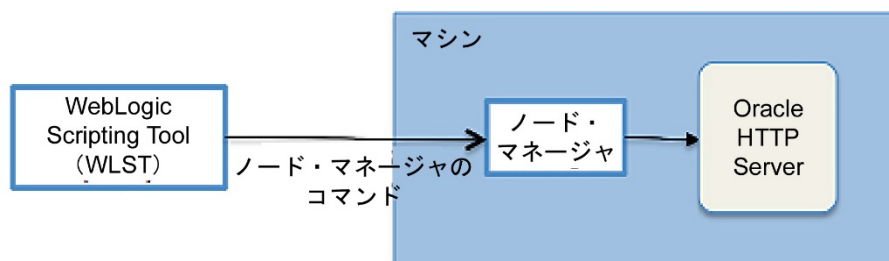


図 - スタンドアロン・ドメインによるOracle HTTP Server の管理

このシナリオは、ユーザー（開発者およびシステム管理者）が最小限のフットプリントと依存性でOracle HTTP Serverをインストールしたいと考えており、ブラウザベースの管理コンソールによってOracle HTTP Serverを（ローカルまたはリモートで）管理する必要はない場合に適用されます。また、システム管理者は、Oracle HTTP Serverを非武装地帯（DMZ）にプロキシとしてインストールしてプロビジョニングする場合に、このシナリオを選択することになるでしょう。

このシナリオを実装するには、以下の流れでOracle HTTP Serverを管理する必要があります。

- システムにJDK7がインストールされていない場合、Java SE 7 Development Kitをダウンロードして、ご使用のマシンにインストールする必要があります。Java SE 7はこちらからダウンロードできます。
- Oracle HTTP Server 12c Release 1バイナリはこちらからダウンロードできます。Oracle HTTP Server 12.1.3を（オプションとしてスタンドアロンを選択して）、特定の場所（例：Oracle Middlewareホーム）にインストールします。Oracle HTTP Serverのインストールについて詳しくは、「Oracle HTTP Server 12c Release 1 (12.1.x) のインストール（スタンドアロン）」を参照してください。
- Fusion Middleware 12.1.3構成ウィザードを使用して、Oracle HTTP Serverを構成します。詳しくは、「スタンドアロン・ドメインによるOracle HTTP Server 12cの構成」を参照してください。
- WebLogicノード・マネージャを起動し、WebLogic Management Frameworkのコマンドライン・インタフェース（WLST）を使用して1つ以上のOracle HTTP Serverインスタンスをプロビジョニングし、これらのOracle HTTP Serverインスタンスの起動や停止を行います。詳しくは、次のセクションを参照してください。

WebLogic Server ドメインによるOracle HTTP Server 12cの構成

ここでは、開発と本番の両方のシナリオについて、WebLogic Serverドメイン内にOracle HTTP Server 12cを構成する手順を段階的に説明します。管理者は、*WebLogic Management Framework*を利用して、Oracle HTTP Serverや他の*Fusion Middleware*アプリケーションを管理できるようになります。製品ドキュメントの[WebLogic Server ドメインでのOracle HTTP Server 12c Release 1の構成](#)に関する説明も参照してください。

ここでは、ユーザーがすでに以下の手順を完了しているものとします。

- JRFを有効にしたWebLogic 12.1.3 (Fusion Middleware Infrastructure 12.1.3) のインストール
- [Oracle HTTP Server 12.1.3のインストール](#)（JRFを有効にしたWebLogic Server12.1.3との連結インストールとして）

WebLogic Server 12.1.3とOracle HTTP Server12.1.3をインストールしたら、WebLogic構成ウィザードを使用してドメインをセットアップし、WebLogicノード・マネージャを使用して1つ以上のOracle HTTP Serverインスタンスを管理できるようになります。

これで、以下のいずれかの方法を使用して*WebLogic Server*ドメイン内でOracle HTTP Serverを構成できるようになりました。

- Fusion Middleware 12cの構成ウィザードを起動し、「Expanded Domain」モードでWebLogicをプロビジョニングして、Oracle HTTP Serverテンプレートを選択します。詳しくは、[WebLogic ServerドメインでのOracle HTTP Serverの構成](#)に関する説明を参照してください。このオプションは、本番デプロイメント用にサポートされており、データベースが必要になります。
- WebLogic Scripting Tool (WLST) を使用して、「OHS Test Domain」を作成します。このオプションでは、データベースへの依存性は発生しません。このオプションは開発環境とステージング環境のみを対象として設計されています。そのため、このオプションは、本番使用が認定されたデプロイメント構成ではありません。

これ以降、両方のシナリオでOracle HTTP Serverを構成する手順を段階的に説明します。

開発/テスト・シナリオ

JRFを有効にしたWebLogic ServerおよびOracle HTTP Server 12.1.3のインストールの終了時に、Oracle HTTP Serverインスタンスを作成し管理するためのWebLogicドメインをプロビジョニングする必要があります。

開発およびテストのシナリオの場合、Oracle HTTP Server付属のコマンドライン・インタフェースをWebLogic Scripting Tool (WLST) に対して使用して、Oracle HTTP Serverドメインや対応するサーバー・インスタンスをWebLogic管理サーバー内にプロビジョニングできます。この操作は、2つの手順で実行します。

Oracle HTTP Serverドメインのプロビジョニング

まず、Oracle HTTP ServerドメインをWebLogic内にプロビジョニングします。

ユーザーはコマンドライン・ターミナル・ウィンドウで以下のコマンドを実行して、Oracle HTTP Serverドメインを（WebLogic管理サーバー内の‘ベース・ドメイン’として）プロビジョニングできます。

(Linuxの場合)

```
<ORACLE_MIDDLEWARE_HOME>/ohs/common/bin/wlst.sh
```

(Windowsの場合)

```
<ORACLE_MIDDLEWARE_HOME>%ohs%common%bin%wlst.cmd
```

以下のコマンドを実行します。

```
wls:/offline> createOHSTestDomain(adminAccountPass=' welcome1' , , nmAccountName=' nm' ,  
nmAccountPass=' welcome1' )  
wls:/offline> exit()
```

コマンド説明：

adminAccountPass -> WebLogic管理サーバーへのログインに使用するパスワード

nmAccountName -> WebLogicノード・マネージャで使用するユーザー名

nmAccountPass -> WebLogicノード・マネージャで使用するパスワード

このコマンドによって、WebLogic管理サーバーがベース・ドメインとともに構成されます。

次に、このWebLogic管理サーバーを使用して管理可能なOracle HTTP Serverインスタンスを作成する必要があります。この手順では、この‘ベース・ドメイン’内で関連付けられたWebLogicノード・マネージャ・サービスと管理サーバー・サービスを有効にして稼働状態にする必要があります。

そのためには、まず以下のコマンドを実行して、WebLogicノード・マネージャとWebLogic管理サーバーを起動する必要があります。

WebLogicノード・マネージャの起動

(Linuxの場合)

```
<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/base_domain/bin/startNodeManager.sh &
```

(Windowsの場合)

```
<ORACLE_MIDDLEWARE_HOME>%user_projects%domains%base_domain%bin%startNodeManager.cmd &
```

WebLogic管理サーバーの起動

(Linuxの場合)

```
<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/base_domain/bin/startWebLogic.sh &
```

(Windowsの場合)

```
<ORACLE_MIDDLEWARE_HOME>%user_projects%domains%base_domain%bin%startWebLogic.cmd &
```

Oracle HTTP Serverインスタンスのプロビジョニング

次に、Oracle HTTP Serverインスタンスをプロビジョニングします。この手順では、WebLogic管理サーバーとWebLogicノード・マネージャが稼働している必要があります。WebLogic管理サーバーとWebLogicノード・マネージャの起動方法について詳しくは、前のセクションを参照してください。

WebLogic管理サーバーとWebLogicノード・マネージャが稼働状態になれば、Oracle HTTP Server付属のコマンドライン・インタフェースをWebLogic Scripting Tool (WLST) に対して利用して、以下のコマンドを実行することで、Oracle HTTP Serverインスタンスを作成できます。

(Linuxの場合)

```
<ORACLE_MIDDLEWARE_HOME>/ohs/common/bin/wlst.sh
```

(Windowsの場合)

```
<ORACLE_MIDDLEWARE_HOME>%ohs%common%bin%wlst.cmd
```

以下のコマンドを実行します。

```
wls:/offline> connect()
```

```
wls:/base_domain/serverConfig> createOHSInstance(instanceName=' ohs1' , machine=' localmachine' ,  
listenPort=' 7777' , sslPort=' 4443' , adminPort=' 9999' )
```

コマンド説明 :

instanceName -> ' <ohs-instance-name>' (Oracle HTTP Serverインスタンスがこの名前で作成されます)

machine -> ' <configured-machine-name>' (前のコマンドでOracle HTTP Serverテスト・ドメインの作成に使用したマシン名です)

このコマンドでは、インスタンス名、マシン名、ポート情報について基本的なエラー・チェックを実行してから、Oracle HTTP Serverインスタンスを構成します。

これ以降、Oracle HTTP Serverインスタンスのホームの場所については、**OHS_INSTANCE_HOME**と表します。この場所は、<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/base_domainに変換できます。

上記のコマンドでは、`'instanceName'` と `'machine'` のみが必須のオプションであり、その他は必須ではありません。`'listenPort'`、`'sslPort'`、`'adminPort'` などの必須ではないオプションを省略した場合、`'createOHSInstance'` では、それぞれのデフォルト・ポートを使用して Oracle HTTP Server インスタンスを作成する最善の努力をします。

ユーザーがこの‘ベース・ドメイン’と関連付けられた‘マシン’の情報を忘れた場合は、このドメイン構成で使用されている‘マシン名’の情報を、以下のコマンドを実行して把握できます。

- `<OHS_INSTANCE_HOME>/config/config.xml` 内で `<machine>` 要素を探します。
- または WLST で、以下のコマンドを実行します。
 - `connect()`
 - `ServerConfig()`
 - `cd('Machines')`
 - `ls()`
- このコマンドによって、以下の場所に Oracle HTTP Server インスタンスが作成されます。
- `<ORACLE_INSTANCE_HOME>/config/fmwconfig/components/OHS/ohs1` → このドメイン上で稼働している WebLogic 管理サーバー用の“構成レポジトリ”として機能します。この“構成レポジトリ”は、Oracle HTTP Server 構成を 2 台以上のマシンにレプリケートする場合に使用されます（クラスタ・デプロイメント）。
- `<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/instances/ohs1` → WebLogic ノード・マネージャが現在のマシン上の構成を管理する場所であり、通常はクラスタ・デプロイメントのすべてのマシン上に作成されます。Oracle HTTP Server の `httpd` サーバー・プロセスにも、この構成ディレクトリが使用されます。

注：WebLogic ノード・マネージャが利用できない状態の場合、この作成操作は一部のみ実行されます。この場合、`'<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/ohs1'` 以下のファイルのみが作成されます。

WebLogic ノード・マネージャが再度オンラインになったときに、残りのファイルが `<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/instances/ohs1` 以下に適切に作成されます。

また、ユーザーはこの時点より、Fusion Middleware Control コンソールというブラウザベースの管理コンソールにアクセスできます。このコンソールを使用して、Oracle HTTP Server のサーバー・ライフ・サイクルや構成を管理できます。コンソールを使用するには、<http://localhost:7777/em/console> にアクセスします。この URI は、<http://<hostname>:<OHS-HTTP-LISTENER-PORT>/em/console> と表すことができます。

`<hostname>` → Oracle HTTP Server がインストールされているマシンのホスト名
`<OHS-HTTP-LISTENER-PORT>` → HTTP リスナー・ポート（デフォルトは 7777）

この後の手順については、Oracle HTTP Server 製品ドキュメントの [WebLogic ドメインでの Oracle HTTP Server 構成後の次のステップ](#) に関する説明を参照してください。特に、[WebLogic Server ドメインでの Oracle HTTP Server 向けの次のステップ](#) に関するセクションを参照してください。

本番シナリオ

JRFを有効にしたWebLogic ServerおよびOracle HTTP Server 12.1.3のインストールの終了時に、Oracle HTTP Serverインスタンスを作成し管理するためのWebLogicドメインをプロビジョニングする必要があります。

本番シナリオの場合、Fusion Middleware 12.1.3の構成ウィザードを利用して、WebLogic内にOracle HTTP Serverドメインを構成し、このドメイン内に対応するOracle HTTP Serverインスタンスを構成する必要があります。この手順では、データベースへの依存性が発生します。

段階的なインストール手順については、製品ドキュメントの [WebLogic ServerドメインでのOracle HTTP Serverの構成](#)に関する説明を参照してください。

Oracle HTTP ServerドメインとOracle HTTP Serverインスタンスをプロビジョニングしたら、WebLogic Management Frameworkを使用してサーバー・インスタンスを管理できるようになります。

また、ユーザーはこの時点より、Fusion Middleware Controlコンソールというブラウザベースの管理コンソールにアクセスできます。このコンソールを使用して、Oracle HTTP Serverのサーバー・ライフ・サイクルや構成を管理できます。コンソールを使用するには、<http://localhost:7777/em/console>にアクセスします。このURIは、<http://<hostname>:<OHS-HTTP-LISTENER-PORT>/em/console>と表すことができます。
<hostname> -> Oracle HTTP Serverがインストールされているマシンのホスト名
<OHS-HTTP-LISTENER-PORT> -> HTTPリスナー・ポート（デフォルトは7777）

この後の手順については、Oracle HTTP Server製品ドキュメントの[WebLogicドメインでのOracle HTTP Server構成後の次のステップ](#)に関する説明を参照してください。特に、[WebLogic ServerドメインでのOracle HTTP Server向けの次のステップ](#)に関するセクションを参照してください。

スタンドアロン・ドメインによるOracle HTTP Server 12cの構成

ここでは、WebLogic Management Frameworkおよび関連するデータベースへの依存性を最小限に抑えてOracle HTTP Serverを管理する手順を段階的に説明します。

このユースケースは、DMZ環境内にデプロイされたOracle HTTP Serverを管理するのに適しています。

この環境でOracle HTTP Serverを管理するには、Oracle HTTP Server 12.1.3を（*Oracle Middleware*ホームの下などに）正常にインストールする必要があります。Oracle HTTP Serverのインストールについて詳しくは、「Oracle HTTP Server 12c Release 1 (12.1.x) のインストール」を参照してください。

Oracle HTTP Server 12.1.3のインストール後、Oracle Fusion Middleware構成ウィザードを使用してOracle HTTP Serverを構成できます。その際には、「Standalone」オプションを選択します。

Fusion Middleware 12.1.3の構成ウィザードは以下のコマンドで起動できます。

(Linuxの場合)

```
<ORACLE_MIDDLEWARE_HOME>/oracle_common/common/bin/config.sh
```

(Windowsの場合)

```
<ORACLE_MIDDLEWARE_HOME>%oracle_common%common%bin%config.cmd
```

注：Linuxマシンでは、構成ウィザードのGUIを表示するために、有効なX11ディスプレイが必要になります。

このウィザードによって、WebLogicノード・マネージャとOracle HTTP Serverインスタンスをプロビジョニングします。WebLogicノード・マネージャは、Oracle HTTP Serverのサーバー・ライフ・サイクルを管理するために使用します。

このFusion Middleware構成ウィザードを使用したOracle HTTP Serverの構成方法については、[スタンドアロン・ドメインでのOracle HTTP Serverの構成](#)に関する説明を参照してください。

後の手順については、Oracle HTTP Server製品ドキュメントの[スタンドアロン・ドメインでのOracle HTTP Server 12c向けの次のステップ](#)に関する説明を参照してください。

WebLogic Server ドメインによるOracle HTTP Server 12cの管理

WebLogic Management Frameworkを使用してOracle HTTP Serverインスタンスを管理するには、WebLogic管理サーバーとWebLogicノード・マネージャを起動し、Fusion Middleware ControlコンソールまたはWebLogic Scripting Tool (WLST) を使用してOracle HTTP Serverインスタンスのライフ・サイクルを管理する必要があります。

最初の手順として、WebLogicノード・マネージャとWebLogic管理サーバーを起動する必要があります。

WebLogic管理サーバーの起動

通常は以下の方法で起動できます。

- JAVA_HOME環境変数に、ご使用のマシンにあるJDK/JRE6またはJDK/JRE7のインストール先を設定します。
 - JAVA_HOME=<jdk-install>; export JAVA_HOME
- ORACLE_MIDDLEWARE_HOME環境変数に、Oracle HTTP ServerとWebLogic Serverが連結してインストールされている場所を指定します。
 - ORACLE_MIDDLEWARE_HOME=<ohs12.1.3-installation>; export ORACLE_HOME
 - JAVA_OPTIONS環境変数を、SSLホスト名検証を無視できるように設定します。
 - JAVA_OPTIONS=" -Dweblogic.security.SSL.ignoreHostnameVerification=true" ; export JAVA_OPTIONS

注：Windowsの場合、ユーザーはコマンドライン・ウィンドウから‘setコマンドを使用して’この環境変数を適切に設定し、同じコマンドライン・ターミナル・ウィンドウからWebLogic Serverを起動する必要があります。

これで、以下のコマンドによりWebLogic管理サーバーを起動できるようになります。

<OHS_INSTANCE_HOME>/bin/startWebLogic.sh (Linuxの場合)

<OHS_INSTANCE_HOME>%bin%startWebLogic.cmd (Windowsの場合)

コマンド説明：

<OHS_INSTANCE_HOME> -> <ORACLE_MIDDLEWARE_HOME>/user_projects/domains/<ohs-domain>

WebLogic管理サーバーのログは通常、<OHS_INSTANCE_HOME>/servers/AdminServer/logsディレクトリにあります。

注：WebLogic構成ウィザードでは、WebLogic管理サーバーを実行するためのデフォルトのポートとして‘7001’が設定されます。ただし、このポートが別のアプリケーションで使用されていると検出された場合は、ランダムなポート番号が割り当てられます。この情報は、WebLogic構成ウィザードの最後に表示される構成のサマリーで確認できます。

WebLogicノード・マネージャの起動

通常は以下の方法で起動できます。

- JAVA_HOME環境変数に、ご使用のシステムにあるJDK/JRE6またはJDK/JRE7のインストール先を設定します。
 - JAVA_HOME=<jdk-install>; export JAVA_HOME
- ORACLE_MIDDLEWARE_HOME環境変数に、Oracle HTTP ServerとWebLogic Serverが連結してインストールされている場所を指定します。
 - ORACLE_MIDDLEWARE_HOME=<ohs12.1.3-installation>; export ORACLE_HOME
JAVA_OPTIONS環境変数を、SSLホスト名検証を無視できるように設定します。
 - JAVA_OPTIONS=" -Dweblogic.security.SSL.ignoreHostnameVerification=true" ;
export
JAVA_OPTIONS

注：Windowsの場合、ユーザーはコマンドライン・ウィンドウの‘set JAVA_HOMEなど’でこの環境変数を適切に設定し、同じコマンドライン・ターミナル・ウィンドウからWebLogicノード・マネージャを起動する必要があります。

これで、以下のコマンドによりWebLogicノード・マネージャを起動できるようになります。

<OHS_INSTANCE_HOME>/bin/startNodeManager.sh (Linuxの場合)

<OHS_INSTANCE_HOME>%bin%startNodeManager.cmd (Windowsの場合)

コマンド説明：

<OHS_INSTANCE_HOME>は<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/<ohs- domain>を表します

WebLogicノード・マネージャは通常、<OHS_INSTANCE_HOME>/nodemanager/nodemanager.logにあります。

これらのログ・ファイルを参照して、WebLogic管理サーバーとWebLogicノード・マネージャが正常に稼働していることを確認してください。

注：WebLogic構成ウィザードでは、WebLogicノード・マネージャを実行するためのデフォルトのポートとして5556が設定されます。ユーザーはWebLogic構成ウィザード内で適切なポートを選択できます。

Oracle HTTP Server 12cインスタンスの起動

Oracle HTTP Serverインスタンスの管理における最後の手順は、Oracle HTTP Serverインスタンスを起動することです。そのためには通常、コマンドラインから直接（つまりWLSTは起動せずに）Oracle HTTP Serverを起動します。以下のコマンドを入力します。

Linuxの場合：<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/base_domain/bin/startComponent.sh
<componentName>

Windowsの場合：<ORACLE_MIDDLEWARE_HOME>%user_projects%domains%base_domain%bin%startComponent.cmd
<componentName>

コマンド説明：

*componentName*はOracle HTTP Serverインスタンスの名前です（'ohs1' など）。

詳しくは、Oracle HTTP Serverドキュメントの[基本的なOracle HTTP Serverタスクの実行](#)に関する説明を参照してください。

WebLogic Scripting ToolによるOracle HTTP Server 12cの管理

Oracle HTTP Server 12cでは、管理タスクにWebLogic Scripting Tool (WLST) を利用します。WLSTは、スクリプト可能なコマンドライン管理インタフェースです。ユーザーはWLSTによって、Oracle HTTP ServerやWebLogic ServerのようなFusion Middleware Infrastructureコンポーネントを、一貫性のある方法でシームレスに管理できます。詳しくは、[WLST構文リファレンス](#)を参照してください。また、特にWLSTでの[Oracle HTTP Server 12c用のカスタム管理インタフェース](#)に関する説明を参照してください。

WLSTでどこまで管理できるかは、Oracle HTTP Serverのプロビジョニング方法によって異なります。ここでは、WLSTコマンド・インタフェースを使用してOracle HTTP Server 12.1.3内の一般的な管理タスクを実行する際の制限事項とその手順について説明します。

Oracle HTTP Server 12c（スタンドアロン・ドメイン）

Oracle HTTP Serverをスタンドアロン・ドメインにプロビジョニングしている構成では、WebLogic管理サーバーは存在しません。そのため、WebLogic Scripting Tool (WLST) が使用できるのは、Oracle HTTP Serverのサーバー・ライフ・サイクル操作（起動、停止、ソフト・リスタートなど）の監視と管理のみです。

Oracle HTTP Server 12c (WebLogic Server ドメイン)

Oracle HTTP ServerをWebLogic Serverドメインにプロビジョニングしている構成では、WebLogic管理サーバーが存在します。そのため、WebLogic Scripting Tool (WLST) を使用して、サーバー・ライフ・サイクル操作 (起動、停止、ソフト・リスタートなど)、構成管理、サーバー・インスタンスの監視を含むOracle HTTP Serverのすべての管理タスクを実行できます。

Oracle HTTP Serverのサーバー・ライフ・サイクルを管理するためのWLSTコマンドを起動するには、以下のコマンドを実行する必要があります。

```
$ORACLE_HOME/oracle_common/common/bin/wlst.sh (Linuxの場合)
```

```
$ORACLE_HOME\oracle_common\common\bin\wlst.cmd (Windowsの場合)
```

続けて、'Oracle HTTP Server ドメイン' に接続します。

```
connect( '<weblogic-admin-username>' , '<weblogic-admin-password>' , '<host:port>' )
```

```
例 : connect( 'weblogic' , 'welcome1' , 'adc6260179:7001' )
```

Oracle HTTP Serverインスタンスを管理するには、WebLogicノード・マネージャ (Oracle HTTP Serverスタンドアロン・ドメインの場合) またはWebLogic管理サーバー (WebLogic Serverドメインの場合) に接続する必要があります。この接続が、以降のWLSTコマンドを実行する際の最初の手順になります。

Oracle HTTP Server 12cインスタンスの作成

WLSTによってこのタスクを実行できるのは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合に限られます。

WLSTインタフェースで、以下のコマンドを実行して新しいOracle HTTP Serverインスタンスを作成する必要があります。

```
createOHSInstance(instanceName=' ohs1' , machine=' new_Machine_1' , listenPort=' 7777' , sslPort=' 4443' , adminPort=' 9999' )
```

または

```
createOHSInstance(instanceName=' <ohs-instance-name>' , machine=' <configured-machine-name>' ) ->
```

このコマンドでは、デフォルトのHTTPポート、HTTPSポート、管理ポートを使用してOracle HTTP Serverインスタンスを構成する最善の努力をします (保証はされません)。

例 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

```
wls:/offline> connect( ' weblogic ' , ' welcome1 ' , ' <host>:7001 ' ) -> コマンド説明 :  
connect( ' <user-name>' ,
```

```
' <password>' , ' <host>:<port>' )
```

```
wls:/ohs_domain> createOHSInstance(instanceName=' ohs1' , machine=' new_Machine_1' ) -> コマンド  
説明 :
```

‘ohs1’は適切な‘<ohs-instance-name>’であり、new_Machine_1は、WebLogic構成ウィザードの‘Machines’で使用した適切な名前です。

ここで、上記のコマンドでは、instanceNameとmachineのみが必須のオプションであり、その他は必須ではありません。

‘listenPort’、‘sslPort’、‘adminPort’などの必須ではないオプションを省略した場合、‘createOHSInstance’では、それぞれのデフォルト・ポートを使用してOracle HTTP Serverインスタンスを作成する最善の努力をします。

ユーザーがWebLogic構成ウィザードで使用した‘マシン’の情報を忘れた場合は、Oracle HTTP Serverドメイン構成で使用されている‘マシン名’の情報を、以下のコマンドを実行して把握できます。

- <OHS_INSTANCE_HOME>/config/config.xml内で<machine>要素を探します。または
- WLSTで、以下のコマンドを実行します。
 - connect()
 - ServerConfig()
 - cd(‘Machines’)
 - ls()

注：このコマンドでは、インスタンス名、マシン名、ポート情報について基本的なエラー・チェックを実行します。

また、このコマンドによって、以下の場所にOracle HTTP Server構成が作成されます。

<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/ohs1 → このドメイン上で稼働しているWebLogic管理サーバー用の“構成レポジトリ”として機能します。この“構成レポジトリ”は、Oracle HTTP Server構成を2台以上のマシンにレプリケートする場合に使用されます（クラスタ・デプロイメント）。

<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/instances/ohs1 → WebLogicノード・マネージャが現在のマシン上の構成を管理するところであり、通常はクラスタ・デプロイメントのすべてのマシン上に作成されます。Oracle HTTP Serverのhttpdサーバー・プロセスにも、この構成ディレクトリが使用されます。

注：WebLogicノード・マネージャが利用できない状態の場合、この作成操作は一部のみ実行されます。

したがって、‘<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/ohs1’以下のファイルのみが作成されます。

WebLogicノード・マネージャが再度オンラインになったときに、残りのファイルが<OHS_INSTANCE_HOME>/config/fmwconfig/components/OHS/instances/ohs1以下に適切に作成されます。

Oracle HTTP Server 12cインスタンスの削除

WLSTによってこのタスクを実行できるのは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合に限られます。

Oracle HTTP Serverインスタンスを削除するには、WLSTで以下のコマンドを実行する必要があります。

```
deleteOHSInstance(instanceName=' <ohs-instance-name>' )
```

例 : Linuxの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic ' , ' welcome1 ' , ' <host>:7001 ' ) -> コマンド説明 :  
connect(' <user-name>' , ' <password>' , ' <host>:<port>' )
```

```
wls:/ohs_domain/ServerConfig> deleteOHSInstance(instanceName=' ohs1' ) -> コマンド説明 : ' ohs1' は  
適切な '<ohs-instance-name>' です。
```

注 : このコマンドを実行するには、WebLogicノード・マネージャが稼働中であり、Oracle HTTP Serverアプリケーションがすでに停止している必要があります。

Oracle HTTP Server 12cインスタンスの起動

このタスクは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合も、スタンドアロン・ドメインにプロビジョニングされている場合も、WLSTによって実行できます。

また、単に 'startComponent.sh' コマンド (Linuxの場合) または 'startComponent.cmd' コマンド (Windowsの場合) を使用して、コマンドラインからOracle HTTP Serverインスタンスを起動することもできます。詳しくは、[Oracle HTTP Server 12cのWLST管理インタフェース](#)に関する説明を参照してください。

Oracle HTTP Serverインスタンスを起動するには、WLSTで以下のコマンドを実行する必要があります。

Linuxの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic ' , ' welcome1 ' , ' <host>:7001 ' ) -> コマンド説明 :  
connect(' <user-name>' ,
```

```
' <password>' , ' <host>:<port>' )
```

```
wls:/ohs_domain/ServerConfig> start( ' ohs1' ) -> コマンド説明 : start( '<ohs-instance-name>' )
```

また、WLSTで 'state(ohs1)' コマンドを実行して、Oracle HTTP Serverアプリケーションの現在のステータスを取得できます。

Oracle HTTP Serverインスタンスが正常に起動した場合：

- そのインスタンスのOracle HTTP Serverログ・ファイルは<OHS_INSTANCE_HOME>/servers/ohs1/logsにあります。
- Oracle HTTP Server監査ログは<OHS_INSTANCE_HOME>/auditlogs/OHS/ohs1にあります。
- Oracle HTTP Serverの状態に関する情報は、ノード・マネージャによって以下の場所にログ出力されます。
<OHS_INSTANCE_HOME>/system_components/OHS/ohs1/

ここで、'ohs1' は、このOracle HTTP Serverインスタンスの作成時に使用したOracle HTTP Serverインスタンス名です。

これで、ブラウザからhttp://<host>:<http-port>/またはhttps://<host>:<ssl-port>にアクセスして、このOracle HTTP Serverインスタンスにアクセスできるようになります。

注：Oracle HTTP ServerドメインのWebLogicノード・マネージャが利用できない状態の場合、このOracle HTTP Serverインスタンスは、ノード・マネージャが再度オンラインになるまで起動できません。

Oracle HTTP Server 12cインスタンスの停止

このタスクは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合も、スタンドアロン・ドメインにプロビジョニングされている場合も、WLSTによって実行できます。

単に 'startComponent.sh' コマンド (Linuxの場合) または 'startComponent.cmd' コマンド (Windowsの場合) を使用して、コマンドラインからOracle HTTP Serverインスタンスを起動することもできます。詳しくは、製品ドキュメントの[Oracle HTTP Serverインスタンスの停止](#)に関する説明を参照してください。

Oracle HTTP Serverインスタンスを停止するには、WLSTで以下のコマンドを実行する必要があります。

Linuxの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( 'weblogic' , 'welcome1' , '<host>:7001' ) -> コマンド説明 :  
connect('<user-name>',
```

```
'<password>', '<host>:<port>')
```

```
wls:/ohs_domain/ServerConfig> shutdown( 'ohs1' ) -> コマンド説明 : shutdown( '<ohs-instance-  
name>' )
```

警告：'shutdown()' をパラメータなしで呼び出すと、Oracle HTTP Serverドメイン内のWebLogic管理サーバーがシャットダウンされ、さらにWLSTシェルも終了しますが、Oracle HTTP Serverの 'httpd' プロセスはシャットダウンされません。

注：Oracle HTTP ServerドメインのWebLogicノード・マネージャが利用できない状態の場合、このOracle HTTP Serverインスタンスは、ノード・マネージャが再度オンラインになるまで起動できません。

Oracle HTTP Server 12cインスタンスのソフト・リスタート

このタスクは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合も、スタンドアロン・ドメインにプロビジョニングされている場合も、WLSTによって実行できます。

Oracle HTTP Serverインスタンスの再起動を実行するには、WLSTで以下のコマンドを実行する必要があります。

Linuxの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic ' , ' welcome1 ' , ' <host>:7001 ' ) -> コマンド説明 :  
connect( ' <user-name> ' ,
```

```
' <password> ' , ' <host>:<port>' )
```

```
wls:/ohs_domain/ServerConfig> softRestart( ' ohs1 ' ) -> コマンド説明 : softRestart( ' <ohs-instance-  
name>' )
```

注：Oracle HTTP ServerドメインのWebLogicノード・マネージャが利用できない状態の場合、このOracle HTTP Serverインスタンスは、ノード・マネージャが再度オンラインになるまで起動できません。

Oracle HTTP Server 12cインスタンスのステータスの取得

このタスクは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合も、スタンドアロン・ドメインにプロビジョニングされている場合も、WLSTによって実行できます。

Oracle HTTP Serverインスタンスのステータスを取得するには、WLSTで以下のコマンドを実行する必要があります。

Linuxの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合：

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic ' , ' welcome1 ' , ' <host>:7001 ' ) -> コマンド説明 :  
connect( ' <user-name> ' , ' <password> ' , ' <host>:<port>' )
```

```
wls:/ohs_domain/ServerConfig> state( ' ohs1 ' ) -> コマンド説明 : state( ' <ohs-instance-name>' )
```

httpdプロセスが実行中の場合、このコマンドの実行により 'RUNNING' と表示されます。

Oracle HTTP Server 12c監視統計の取得

このタスクは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合も、スタンドアロン・ドメインにプロビジョニングされている場合も、WLSTによって実行できます。

パフォーマンス監視統計を取得するには、WLSTを以下のように利用します。

Linuxの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic' , ' welcome1' , ' <host>:7001' ) -> コマンド説明 :  
connect('<user-name>', ' <password>', ' <host>:<port>')
```

```
wls:/base_domain/ServerConfig> displayMetricTables(' ohs_*')
```

Oracle HTTP Server構成のプロパティへのアクセス

WLSTによってこのタスクを実行できるのは、Oracle HTTP ServerがWebLogic Serverドメインにプロビジョニングされている場合に限られます。Oracle HTTP Server構成内の編集可能プロパティのリストを取得するには、WLSTコマンドを利用し、以下のコマンドを実行します。

Linuxの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.sh
```

Windowsの場合 :

```
<ORACLE_HOME>/oracle_common/common/bin/wlst.cmd
```

```
wls:/offline> connect( ' weblogic' , ' welcome1' , ' <host>:7001' ) -> コマンド説明 :  
connect('<user-name>', ' <password>', ' <host>:<port>')
```

```
wls:/ohs_domain/ServerConfig> editCustom()
```

```
wls:/ohs_domain/ServerConfig> cd( ' oracle. ohs' )
```

```
wls:/ohs_domain/ServerConfig> cd( ' oracle. ohs:type=OHSInstance, name=ohs1' )
```

```
wls:/ohs_domain/ServerConfig> ls()
```


付録

WebLogic JRF 12c Release 1のインストール

WebLogic Management FrameworkによってOracle HTTP Server 12cを管理するには、JRFを有効にしたWebLogicとOracle HTTP Serverを同じインストール先（例：Oracle Middlewareホーム）にインストールする必要があります。ここでは、JRFを有効にしたWebLogic 12.1.xのインストール手順を段階的に説明します。

JRFを有効にしたWebLogic 12.1.3は[こちら](#)からダウンロードできます。

- JAVA_HOME環境変数にJDK/JRE7を設定し、以下のコマンドを実行して、WebLogic Server 12.1.3のインストールを開始してください。
 - `$JAVA_HOME/bin/java -jar wls_jrf_generic.jar`
- 適切なオプションを選択して、インベントリ・ディレクトリを指定します。デフォルトのオプションでも十分に機能します。
- 次の画面で、「Skip Software Updates」を選択してそのまま進み、前提条件のチェックを完了します。
- “Oracleホーム”のインストール先を指定する画面で、**WebLogic 12.1.3サーバーとOracle HTTP Server 12.1.3**
- サーバーの両方をホストする新しいインストール先を選択し、「Next」をクリックします。
- セキュリティ・アップデートを取得せず、インストーラでバイナリの配置を開始するように選択してかまいません。

詳しくは、製品ドキュメントの[WebLogic 12.1.3のインストール](#)に関する説明を参照してください。

Oracle HTTP Server 12c Release 1 (12.1.x) のインストール (連結インストール)

ここでは、Oracle HTTP Server 12c Release 1を、JRFを有効にしたWebLogicと（同じ場所に）連結してインストールする手順を段階的に説明します。

- [WebLogic JRF 12c Release 1のインストール](#)で説明したとおりに、JRFを有効にしたWebLogic（Fusion Middleware Infrastructure）を特定のOracleホームにインストールする必要があります。
- Oracle HTTP Server 12c Release 1は、E-Delivery（本番目的の場合に推奨）または[Oracle Technology Network](#)（評価および開発目的の場合に推奨）からダウンロードできます。ダウンロード後、WebLogic JRF（Fusion Middleware Infrastructure）がインストールされているOracleホームにインストールします。
- 以下のコマンドを実行して、Oracle HTTP Server 12.1.3をインストールできます。
 - `ohs_linux64.bin` (Linux 64ビットのターミナルから実行)

- setup_ohs_win64.exe (Windows 64ビットのコマンドライン・ウィンドウから実行)
- インベントリ・ディレクトリに関するデフォルトのオプションを選択します。
- 次の画面で、「Skip Software Updates」を選択してそのまま進み、前提条件のチェックを完了します。
- 次の画面で、「OHS managed through a pre-installed WebLogic server」オプションを選択します。

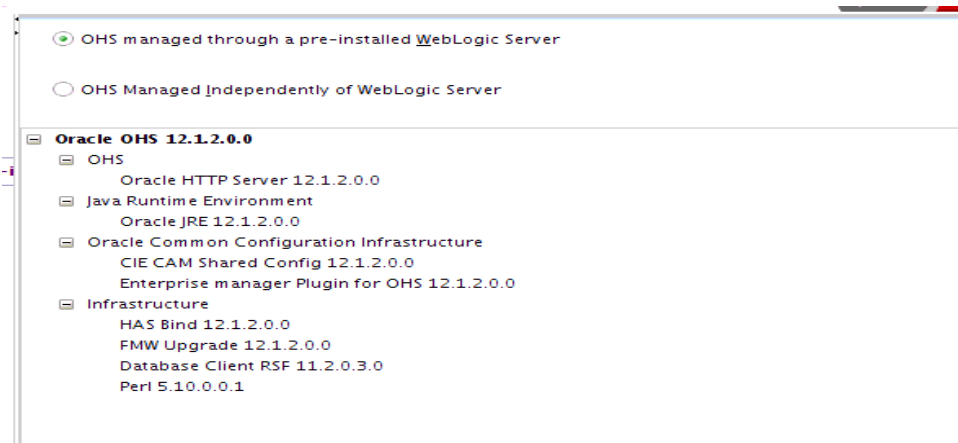


図 - Oracle HTTP Serverの連結インストール

- “Oracleホーム” のインストール先として、WebLogic 12.1.3をインストールした際に使用したインストール先を選択し、「Next」をクリックします。
- ここでは、セキュリティ・アップデートを取得せず、インストーラでバイナリの配置を開始するように選択してかまいません。

このインストール先について、以降は単に ‘ORACLE_MIDDLEWARE_HOME’ と表します。

詳しくは、製品ドキュメントの[Oracle HTTP Server 12c Release 1のインストール](#)に関する説明を参照してください。

Oracle HTTP Server 12c Release 1 (12.1.x) のインストール (スタンドアロン)

ここでは、Oracle HTTP Server 12.1.3をスタンドアロン・シナリオでインストールする手順を段階的に説明します。

- Oracle HTTP Server 12c Release 1は、E-Delivery (本番目的の場合に推奨) または[Oracle Technology Network](#) (評価および開発目的の場合に推奨) からダウンロードできます。以下のコマンドを実行して、Oracle HTTP Server 12.1.3をインストールできます。
 - ohs_linux64.bin (Linux 64ビットのターミナルから実行)
 - setup_ohs_win64.exe (Windows 64ビットのコマンドライン・ウィンドウから実行)
インベントリ・ディレクトリに関するデフォルトのオプションを選択します。
- 次の画面で、「Skip Software Updates」を選択してそのまま進み、前提条件のチェックを完了します。

- 次の画面で、「OHS standalone」オプションを選択します。

詳しくは、製品ドキュメントのOracle HTTP Server 12c Release 1のインストールに関する説明を参照してください。

結論

Oracle HTTP Server 12c Release 1 (12.1.x) は、広く普及しているApache HTTP Serverインフラストラクチャをベースとしています。Fusion Middleware内にホストされたアプリケーションへのHTTP/HTTPSトラフィックを、フロントエンドとして安全かつ確実に処理することで、高い信頼性で‘Web層’の責務を果たすことができます。以下に、Oracle Web Tier製品以外には見当たらない、Oracle HTTP Server 12.1.3の特徴的な機能を示します。

- 他のFusion Middlewareアプリケーションとの一貫性を維持した統合管理/監視機能
- 複数のプロビジョニング・モデル - *WebLogic Server*ドメイン（統合管理向け）またはスタンドアロン・ドメイン（リモート管理機能を持つ簡潔かつ単純なプロビジョニング向け）
- WebLogicドメインによる、複数のマシンで稼働するOracle HTTP Serverファームの管理

これらの機能により、Oracle HTTP Serverは静的コンテンツを提供でき、WebLogic管理対象サーバー/クラスタへのHTTP/HTTPSリクエストをリバース・プロキシとして処理できます。そして、‘Web層’と‘アプリケーション層’を同じWebLogicドメイン内で管理できるため、プロビジョニングやサーバー・ライフ・サイクル管理などの共通の管理タスクを容易に実行できます。



Oracle HTTP Server 12c Release 1 - 技術概要
2013年7月

著者 : Sriram Natarajan

共著者 : Frances

Zhao

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U. S. A.

海外からのお問い合わせ窓口 : 電話 : +1. 650. 506. 7000

ファクシミリ : +1. 650. 506. 7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

本文書は情報提供のみを目的として提供されており、記載内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含め、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。

0113

Hardware and Software, Engineered to Work Together