

Oracle® HTTP Server

スタンドアロン・デプロイの管理 Apache 2.0 ベース

10g リリース 2 (10.1.2)

部品番号 : B25062-01

2005 年 10 月

Oracle HTTP Server スタンドアロン・デプロイの管理 Apache 2.0 ベース , 10g リリース 2 (10.1.2)

部品番号 : B25062-01

原本名 : Oracle HTTP Server Administering a Standalone Deployment Based on Apache 2.0, 10g Release 2 (10.1.2)

原本部品番号 : B14009-02

原本著者 : Harry Schaefer

原本協力者 : Julia Pond, Sanket Atal, Warren Brieese, Olivier Caudron, Kevin Clark, Priscila Darakjian, Sander Goudswaard, Helen Grembowicz, Mathew Joy, Pushkar Kapasi, Keith Kelleman, Eric Kienle, John Lang, Bruce Lowenthal, Li Ma, Chuck Murray, Mark Nelson, Carol Orange, Bert Rich, Jon Richards, Shankar Raman, Baogang Song, Kevin Wang, Karen Wilson

Copyright © 2002, 2005, Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万一かかるとしてプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Retek は米国 Oracle Corporation 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、他社の商標の可能性があり得ます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

目次

はじめに	vii
対象読者	viii
ドキュメントのアクセシビリティについて	viii
関連ドキュメント	viii
表記規則	viii
サポートおよびサービス	ix
1 概要	
Oracle HTTP Server の機能	1-2
Oracle HTTP Server のコンポーネント	1-4
Oracle HTTP Server のモジュール	1-4
Oracle HTTP Server のサポート	1-5
Oracle HTTP Server の管理	1-5
Oracle HTTP Server の起動、停止および再起動	1-6
Oracle HTTP Server の起動	1-6
Oracle HTTP Server の停止	1-6
Oracle HTTP Server の再起動	1-6
2 Oracle Application Server に対するスタンドアロンの Oracle HTTP Server の構成	
構成チェックリスト	2-2
スタンドアロンの Oracle HTTP Server のインストール	2-2
OPMN の構成	2-3
mod_oc4j の構成	2-5
シングル・サインオンの構成	2-5
3 サーバーとファイル位置の指定	
サーバー機能と管理者機能の設定	3-2
ServerName	3-2
UseCanonicalName	3-2
ServerAdmin	3-2
ServerSignature	3-2
ServerTokens	3-3
ServerAlias	3-3
ファイル位置の指定	3-3
CoreDumpDirectory	3-3
DocumentRoot	3-4
ErrorLog	3-4

LockFile	3-4
PidFile	3-4
ScoreBoardFile	3-4
ServerRoot	3-4
4 サーバー・プロセスの管理	
Oracle HTTP Server の処理モデル	4-2
サーバー・プロセスの処理	4-2
Group	4-2
User	4-2
プロセス数と接続数の構成	4-2
StartServers	4-3
ThreadsPerChild	4-3
MaxClients	4-3
MaxRequestsPerChild	4-3
MaxSpareServers	4-3
MinSpareServers	4-4
root としての Oracle HTTP Server の実行	4-4
セキュリティに関する考慮事項	4-4
プロセス情報の取得	4-4
5 ネットワーク接続の管理	
リスナー・ポートおよびアドレスの指定	5-2
サーバーとネットワーク間の相互作用の管理	5-2
ListenBackLog	5-3
SendBufferSize	5-3
TimeOut	5-3
接続の永続性の管理	5-3
KeepAlive	5-3
KeepAliveTimeout	5-3
MaxKeepAliveRequests	5-4
クライアント IP アドレスの取得	5-4
リバース・プロキシとロード・バランサの構成	5-4
6 サーバー・ログの構成と使用	
Oracle Diagnostic Logging の使用	6-2
概要	6-2
Oracle HTTP Server の構成	6-2
ログ・レベルの指定	6-5
ログ・ファイルの指定	6-5
アクセス・ログ	6-5
カスタム・ログ	6-6
エラー・ログ	6-6
PID ファイル	6-6
パイプされたログ	6-6
リライト・ログ	6-7
スクリプト・ログ	6-7
SSL ログ	6-7

送信ログ	6-7
------------	-----

7 モジュールの理解

モジュールのリスト	7-2
mod_access	7-2
mod_actions	7-2
mod_alias	7-2
mod_asis	7-2
mod_auth	7-2
mod_auth_anon	7-3
mod_auth_dbm	7-3
mod_autoindex	7-3
mod_cern_meta	7-3
mod_certheaders	7-3
mod_cgi	7-6
mod_cgid	7-6
mod_dir	7-6
mod_env	7-6
mod_expires	7-7
mod_fastcgi	7-7
mod_file_cache	7-7
mod_headers	7-7
mod_imap	7-7
mod_include	7-7
mod_info	7-7
mod_log_config	7-8
mod_logio	7-8
mod_mime	7-8
mod_mime_magic	7-8
mod_negotiation	7-8
mod_oc4j	7-8
mod_oc4j の構成	7-9
mod_oc4j を使用したロード・バランシング	7-15
mod_oc4j と OC4J 間での SSL の有効化	7-15
mod_onsint	7-16
mod_onsint を使用するメリット	7-17
UNIX と Windows での実装上の差異	7-17
mod_oss1	7-18
mod_osso	7-18
mod_perl	7-18
データベース使用上の注意	7-18
mod_php	7-21
mod_proxy	7-21
mod_rewrite	7-21
mod_rewrite のルール処理	7-22
mod_rewrite のディレクティブ	7-22
リライト・ルールのヒント	7-24
リダイレクションの例	7-25
mod_setenvif	7-25

mod_speling	7-25
mod_status	7-25
mod_unique_id	7-25
mod_userdir	7-25
mod_usertrack	7-26
mod_vhost_alias	7-26
mod_wchandshake	7-26

8 セキュリティの管理

Oracle HTTP Server のセキュリティの概要	8-2
ユーザーのクラスとその権限	8-2
保護されるリソース	8-3
認証と認可の適用	8-3
ホストベースのアクセス制御	8-3
ユーザーの認証と認可	8-6
ポート・トンネリングの概要	8-7
ポート・トンネリングの構成	8-8
ポート・トンネリング用の SSL の構成	8-9
ポート・トンネリングの構成のリファレンス	8-10
Oracle Identity Management インフラストラクチャの利用	8-13
概要	8-13
Oracle Application Server Single Sign-On と mod_osso の使用	8-14

9 Oracle HTTP Server での SSL の有効化

概要	9-2
SSL の構成	9-2
タスク 1: 実際の Wallet の作成	9-2
タスク 2: SSL の有効化	9-2
タスク 3: (オプション) 構成のカスタマイズ	9-3
追加の SSL 機能	9-3
グローバル・サーバー ID のサポート	9-4
PKCS #11 のサポート	9-4
SSL 構成ディレクティブの使用	9-4
mod_ossl のディレクティブの使用	9-5
iasobf ユーティリティの使用	9-16
mod_proxy のディレクティブの使用	9-16

A mod_oc4j を使用したロード・バランシング

ロード・バランシング・ポリシー	A-2
ランダム	A-2
ラウンドロビン	A-2
ローカル・アフィニティを考慮したランダム	A-2
ローカル・アフィニティを考慮したラウンドロビン	A-2
ルーティングの重みを使用したランダム	A-2
ルーティングの重みを使用したラウンドロビン	A-2
メトリック・ベース	A-2
ローカル・アフィニティを考慮したメトリック・ベース	A-3

ロード・バランシング・パラメータ	A-3
Oc4jSelectMethod	A-3
Oc4jRoutingWeight	A-3
メトリック・ベースのロード・バランシング	A-4
Oracle HTTP Server の構成	A-5
OC4J の構成	A-5

B 構成ファイル

dms.conf	B-2
httpd.conf	B-2
httpd.conf のファイル構造	B-2
iaspt.conf	B-3
mime.types	B-3
mod_oc4j.conf	B-3
mod_osso.conf	B-3
opmn.xml	B-4
oracle_apache.conf	B-4
php.ini	B-4
ssl.conf	B-5

C よくある質問

アプリケーション固有のエラー・ページの作成	C-2
ISP (仮想ホスト) の顧客に対する HTTPS の提供	C-2
異なる言語およびキャラクタ・セット・バージョンのドキュメントの使用	C-2
ファイアウォールの後方にある HTTP Server へのプロキシ依存のリクエストの送信	C-2
mod_oc4j 情報	C-2
SSL を使用した mod_oc4j と OC4J との通信	C-2
Oracle HTTP Server のリリース番号	C-3
Oracle HTTP Server への Apache セキュリティ・パッチの適用	C-3
Oracle HTTP Server からの出力の圧縮	C-3
PHP のサポート	C-3
ハッカーからの Web サイトの保護	C-3

D Oracle HTTP Server のトラブルシューティング

問題および解決策	D-2
間欠的 HTTP-500 エラー	D-2
Oracle HTTP Server と OC4J ブロック間の接続におけるファイアウォール	D-2
ポートの競合により Oracle HTTP Server が起動できない	D-3
多数の HTTPD プロセスによるマシンのオーバーロード	D-3
1024 未満のポートでの Oracle HTTP Server の起動時に発生する権限拒否	D-3
PM ファイルが正しく検出されない場合に Oracle HTTP Server が起動できない	D-3
Webcache リバース・プロキシでの SSO クライアント認証の失敗	D-4
その他の解決策	D-4

E サード・パーティ・ライセンス

Apache HTTP Server	E-2
The Apache Software License	E-2

Apache SOAP	E-5
Apache SOAP License	E-5
DBI Module	E-8
Perl Artistic License	E-8
Perl	E-10
Perl Kit Readme	E-10
mod_perl License	E-10
Perl Artistic License	E-11
PHP	E-13
The PHP License	E-13
FastCGI	E-14
FastCGI Developer's Kit License	E-14
Module mod_fastcgi License	E-15

用語集

索引

はじめに

このマニュアルでは、Oracle HTTP Server を管理する方法について説明します。

対象読者

『Oracle Application Server スタンドアロン・デプロイの管理 Apache 2.0 ベース』は、アプリケーション・サーバーの管理者、セキュリティ・マネージャ、およびアプリケーション・サーバーで使用するデータベースのマネージャを対象としています。

ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

関連ドキュメント

詳細は、Oracle Application Server ドキュメント・ライブラリを参照してください。

表記規則

本文では、次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連するグラフィカル・ユーザー・インタフェース要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック	イタリックは、特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、パラグラフ内のコマンド、URL、例に記載されているコード、画面に表示されるテキスト、または入力するテキストを示します。

サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

オラクル社カスタマ・サポート・センター

オラクル製品サポートの購入方法、およびオラクル社カスタマ・サポート・センターへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.co.jp/support/>

製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://otn.oracle.co.jp/document/>

研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

<http://www.oracle.co.jp/education/>

その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.co.jp>

<http://otn.oracle.co.jp>

注意： ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

この章では、Oracle HTTP Server について、Oracle 製品とその基礎となるオープン・ソースの Apache 製品の違いを重点的に説明します。また、サーバーの起動、停止および再起動の方法についても説明します。

内容は、次のとおりです。

- Oracle HTTP Server の機能
- Oracle HTTP Server のコンポーネント
- Oracle HTTP Server のサポート
- Oracle HTTP Server の管理
- Oracle HTTP Server の起動、停止および再起動

Oracle HTTP Server の機能

Oracle HTTP Server は Oracle Application Server の Web サーバー・コンポーネントです。[Apache](#) インフラストラクチャに基づき、Oracle HTTP Server は開発者が様々な言語やテクノロジーでサイトをプログラミングすることを可能にします。Perl (mod_perl および CGI 経由)、C (CGI および FastCGI 経由)、C++ (FastCGI 経由)、PHP、Oracle の PL/SQL などが使用できます。また、Oracle HTTP Server は、フォワード・プロキシ・サーバーにも、リバース・プロキシ・サーバーにもなります。さらに、シングル・サインオン、クラスタ化された配置、高可用性などの機能により、Oracle HTTP Server の動作が拡張されます。

Apache ベース : HTTP バージョン 1.1 のサポート

Oracle HTTP Server のコードは、Apache 2.0 Web Server (<http://www.apache.org>) に基づいています。このような検証済コードベースにより、Oracle HTTP Server では Web サーバーに要求される安定性、柔軟性、およびスケーラビリティを Oracle Application Server の顧客に提供しています。

セキュリティ : SSL による暗号化

Web サイトを安全に運用するには、Secure Sockets Layer が必要です。Oracle HTTP Server では、業界標準の特許アルゴリズムに基づいた SSL 暗号化をサポートしています。SSL は、Internet Explorer および Netscape の両方のブラウザとシームレスに連動します。また、そのインフラストラクチャは、データベース・ユーザーと同じ Wallet 情報を共有するようにアップグレードされています。機能は次のとおりです。

- **SSL HW アクセラレーション・サポート** : SSL の暗号化は、ソフトウェアで実行されるときよりも低速です。このため、専用ハードウェアが (特に nCipher により) サポートされています。
- **ディレクトリ別変数セキュリティ** : この機能を使用すると、ディレクトリをそれぞれ異なる暗号化強度で保護できます。
- **OC4J に対する Oracle HTTP Server の SSL サポート** : Oracle HTTP Server および OC4J は、SSL 経由の AJP プロトコルを使用して通信できます。以前は、Oracle HTTP Server および OC4J では、AJP 1.3 非暗号化プロトコルを認証のサポートなしで使用していました。現在、Oracle HTTP Server は修正されており、SSL 経由の AJP 1.3 プロトコルに対するサポートを拡張し、暗号化および認証が行われます。

関連資料 :

- 『Oracle Application Server セキュリティ・ガイド』
- [第 8 章「セキュリティの管理」](#)
- [第 9 章「Oracle HTTP Server での SSL の有効化」](#)

セキュリティ : シングル・サインオン

Oracle HTTP Server では、Web サーバーの標準 Basic 認証機能をサポートしています。ここで使用されるユーザー名およびパスワードのソースはフラット・ファイル (暗号化パスワードを使用) です。また、モジュール mod_osso が組み込まれ、シングル・サインオンをサイト間およびアプリケーション間でサポートしています。これにより、エンド・ユーザーは 1 回しかログインする必要がないため、その使用感は大幅に向上します。また、セキュリティの大部分が宣言的であるため、開発サイクルは一層簡単になります。

関連資料 :

- 『Oracle Application Server Single Sign-On 管理者ガイド』
- [7-18 ページの「mod_osso」](#)

仮想ホスト

仮想ホスト機能により、HTTP サーバーは 1 つの IP アドレスで複数のドメイン名を処理できます。このため、仮想ホスト `www.north.com` は `www.south.com` と同じ IP アドレスを保持する場合があります。Oracle HTTP Server は、仮想ホストにコンテナ環境を提供し、ファイルの取得元となる場所に加えて、独自のセキュリティ・セットおよびその他の構成ディレクティブを仮想ホストに提供しています。これにより、何百何千ものサイトを Oracle HTTP Server の 1 つのランタイム・インスタンスから取得することが可能になり、ISP はハードウェアおよび管理のコストを削減することができます。1 つの IP アドレスでは、1 つの仮想ホストのみが SSL に対応できます。Oracle HTTP Server は複数の IP アドレスをサポートでき、各 IP アドレスは仮想ホストを 1 つのみ保持できます。

プロキシ・サーバーと URL リライティング

有効な Web サイトは、頻繁に変化します。それに伴い、ディレクトリ構造および URL も変化します。Oracle HTTP Server は、URL リライティングをサポートするエンジンを組み込むことで、このような変化に簡単に対応できます。その結果、エンド・ユーザーは自分のブックマークを変更する必要がありません。また、リバース・プロキシ機能をサポートするため、様々なサーバーから提供されるコンテンツを、1 つのサーバーから表示されるようにすることが容易になります。

サーバー・サイド・インクルード

サーバー・サイド・インクルードにより、サイトの各ページで、動的コンテンツまたは均一な静的コンテンツを簡単に追加できます。通常、ヘッダーまたはフッターの情報に使用されます。Oracle HTTP Server では、特定のファイル・タイプまたは特定の仮想ホストに対してのみサーバー側インクルードを有効化する特別なディレクティブがサポートされています。

Perl

Perl は、動的コンテンツを提供するためによく使用されるスクリプト言語です。Perl は、CGI プログラムとしてコールすることも、`mod_perl` から直接コールすることもできます。Oracle Application Server では Perl バージョン 5.8.3 を使用します。

関連項目： [7-18 ページの「`mod_perl`」](#)

PHP

PHP は、オープン・ソースで広く用いられている汎用クライアント側スクリプト言語で、標準 HTML に埋め込まれます。この言語は、動的 HTML ページの生成に使用されます。

関連項目： [7-21 ページの「`mod_php`」](#)

C/C++ (CGI および FastCGI)

CGI プログラムは、通常、Web アプリケーションのプログラミングに使用されています。Oracle HTTP Server は、リクエストのライフサイクルよりも長く有効な状態に保つメカニズムを提供することで CGI プログラムを強化し、パフォーマンスを大幅に向上させます。

Dynamic Monitoring Service

Dynamic Monitoring Service (DMS) メトリックは、Oracle HTTP Server と OC4J の両方のプロセスに関するランタイム・パフォーマンス統計を提供します。DMS は、アプリケーションの実行中に詳細なパフォーマンス統計を収集します。このデータを使用すると、重要なリクエストの処理フェーズの時間やステータス情報をモニターできます。この情報により、パフォーマンスのボトルネックを見つけ、スループットを最大化しレスポンス時間を最小化するようにアプリケーションをチューニングできます。

関連資料： 『Oracle Application Server パフォーマンス・ガイド』

Oracle Process Manager and Notification Server

Oracle Application Server は、OC4J および Oracle HTTP Server のプロセスに対するプロセス管理、障害検出およびフェイルオーバーのために、Oracle Process Manager and Notification Server (OPMN) との高可用性インフラストラクチャ統合を提供します。

関連資料：

- 『Oracle Application Server 高可用性ガイド』
- 『Oracle Process Manager and Notification Server 管理者ガイド』

ロード・バランシング

Oracle HTTP Server には、`mod_oc4j` というモジュールが組み込まれています。このモジュールは、クラスタ内の OC4J インスタンスからのリクエストをルーティングします。OPMN により、システム管理者が何も構成しなくても、`mod_oc4j` インスタンスがシステム内のすべての OC4J を確実に把握できます。

関連項目： [付録 A 「mod_oc4j を使用したロード・バランシング」](#)

Oracle HTTP Server のコンポーネント

Oracle HTTP Server は、同一プロセス内で実行される複数のコンポーネントで構成されています。これらのコンポーネントが、クライアント・リクエストの処理時に Oracle HTTP Server により提供される豊富な機能を提供しています。主なコンポーネントは、次のとおりです。

- **HTTP リスナー：**Oracle HTTP Server は、Apache HTTP リスナーに基づいてクライアント・リクエストを処理します。HTTP サーバーのリスナーは、受信リクエストを処理し、これを適切な処理ユーティリティにルーティングします。
- **モジュール (mod)：**モジュールは Oracle HTTP Server の基本機能を実装および拡張します。Oracle HTTP Server には、多くの Apache 標準モジュールが組み込まれています。また、Oracle Application Server コンポーネント固有の内部モジュールもいくつか組み込まれています。

関連項目： [1-4 ページの「Oracle HTTP Server のモジュール」](#)

- **Perl インタプリタ：**`mod_perl` を介して Oracle HTTP Server に埋め込まれている永続 Perl ランタイム環境です。

関連資料： 『Oracle Application Server 概要』

Oracle HTTP Server のモジュール

Oracle HTTP Server に同梱のモジュールを表 1-1 に示します。モジュールは Web サーバーの基本機能を拡張し、Oracle HTTP Server とその他の Oracle Application Server のコンポーネントとの統合をサポートします。このリストは、Apache オープン・ソースの配布版 (Oracle モジュールが組み込まれている場合) とは異なり、Oracle でサポートされないモジュールもあるため注意してください。

表 1-1 Oracle HTTP Server のモジュール

モジュール	注意	モジュール	注意
<code>mod_access</code>		<code>mod_log_config</code>	
<code>mod_actions</code>		<code>mod_logio</code>	
<code>mod_alias</code>		<code>mod_mime</code>	
<code>mod_asis</code>		<code>mod_mime_magic</code>	
<code>mod_auth</code>		<code>mod_negotiation</code>	
<code>mod_auth_anon</code>		<code>mod_oc4j</code>	Oracle モジュール
<code>mod_auth_dbm</code>		<code>mod_onsint</code>	Oracle モジュール
<code>mod_autoindex</code>		<code>mod_ossll</code>	Oracle モジュール
<code>mod_cern_meta</code>		<code>mod_osso</code>	Oracle モジュール

表 1-1 Oracle HTTP Server のモジュール (続き)

モジュール	注意	モジュール	注意
mod_certheaders	Oracle モジュール	mod_perl	
mod_cgi		mod_php	
mod_cgid		mod_proxy	
mod_dir		mod_rewrite	
mod_env		mod_setenvif	
mod_expires		mod_speling	
mod_fastcgi		mod_status	
mod_file_cache		mod_unique_id	UNIX システムのみ
mod_headers		mod_userdir	
mod_imap		mod_usertrack	
mod_include		mod_vhost_alias	
mod_info		mod_wchandshake	Oracle モジュール

関連項目： 第 7 章「モジュールの理解」

Oracle HTTP Server のサポート

オラクル社は、次の Oracle HTTP Server の機能および状態について、テクニカル・サポートを提供しています。

- Oracle 製品に含まれるモジュール。Apache Software Foundation などの他のソースからのモジュールは、Oracle ではサポートされません。
- サポート対象の Oracle Apache モジュールのみで構成されている Apache 構成内で再現できる問題。
- Perl インタプリタの使用 (サポート対象の Apache 構成に含まれている場合)。

Oracle HTTP Server の管理

Oracle HTTP Server は `opmnctl` を使用して管理できます。これは、プロセス管理に使用する Oracle Process Manager and Notification Server (OPMN) のコマンドライン・ユーティリティです。このファイルは次の場所にあります。

- UNIX の場合：`ORACLE_HOME/opmn/bin`
- Windows の場合：`ORACLE_HOME\opmn\bin`

関連資料：『Oracle Process Manager and Notification Server 管理者ガイド』

注意： リリース 2 (10.1.2) の Oracle HTTP Server 2.0 スタンドアロンは IPv6 をサポートしていますが、OPMN は IPv6 をサポートしていません。OPMN を使用して管理する場合、Oracle HTTP Server 2.0 の ping が正常に実行できるのであれば、Oracle HTTP Server で 1 つ以上の IPv4 アドレスをリスニングする必要があります。

Oracle HTTP Server の起動、停止および再起動

Oracle HTTP Server は、Oracle Process Manager and Notification Server (OPMN) により管理されます。サーバーの起動、停止および再起動には、必ず `opmnctl` ユーティリティを使用してください。そうしないと、構成管理インフラストラクチャで Oracle HTTP Server プロセスの検出やプロセスとの通信ができず、問題が発生する可能性があります。

注意： Oracle HTTP Server の管理に、`apachectl` ユーティリティは使用しないでください。

Oracle HTTP Server の状態を判断するには、次のコマンドを使用します。

```
opmnctl status
```

各プロセスとともに、その現行の状態 (Up、Down など) が表示されます。

Oracle HTTP Server の起動

Oracle HTTP Server を起動するには、`startproc` コマンドを使用します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] startproc ias-component=HTTP_Server`
- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] startproc ias-component=HTTP_Server`

Oracle HTTP Server の停止

Oracle HTTP Server を停止するには、`stopproc` コマンドを使用します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`
- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`

Oracle HTTP Server の再起動

Oracle HTTP Server の再起動ではグレースフル・リスタートが実行され、再起動したことがクライアントには通知されません。グレースフル・リスタートでは、UNIX 上で `USR1` シグナルが送信されます。プロセスは、このシグナルを受信すると、現行のリクエストを処理してから終了するように子プロセスに対して指示します (リクエストの処理中でない子プロセスは即時に終了します)。

親プロセスは構成ファイルを再読み取りし、ログ・ファイルを再オープンし、構成ファイルの再読み取り時に検出された設定に従って子プロセスを新規の子プロセスに置換します。この場合、常に指定のプロセス作成設定 (`MaxClients`、`MaxSpareServers`、`MinSpareServers`) が監視され、現行サーバーの負荷が考慮されます。

Oracle HTTP Server を再起動するには、`restartproc` コマンドを使用します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

関連資料： 『Oracle Process Manager and Notification Server 管理者ガイド』

Oracle Application Server に対するスタンドアロンの Oracle HTTP Server の構成

この章では、スタンドアロンの Oracle HTTP Server 2.0 を既存の Oracle Application Server 10g リリース 2 (10.1.2) 中間層と通信できるように構成する方法について説明します。Oracle Application Server のスタンドアロン・インストールによって Oracle HTTP Server 2.0 を導入した場合、Oracle Enterprise Manager 10g Grid Control コンソールおよび Distributed Configuration Management はインストールされません。これらは、Oracle Application Server でサーバー・グループ (ファーム) の構成および管理を簡単に行えるようにするためのツールです。これらの便利なツールが使用できないため、手動の構成手順を実行して、管理されている既存の Oracle Application Server 中間層と相互運用できるようにスタンドアロンの Oracle HTTP Server 2.0 を構成する必要があります。

内容は、次のとおりです。

- [構成チェックリスト](#)
- [スタンドアロンの Oracle HTTP Server のインストール](#)
- [OPMN の構成](#)
- [mod_oc4j の構成](#)
- [シングル・サインオンの構成](#)

構成チェックリスト

スタンドアロンの Oracle HTTP Server 2.0 を構成する前に、次の点を確認します。

- Oracle Application Server 10g リリース 2 (10.1.2) の標準 (管理) インスタンスが必要に応じてすべてインストールされ、構成されていることを確認します。スタンドアロンの Oracle HTTP Server 2.0 のリスナーを構成する前に、すべてのインスタンスを構成しておく必要があります。標準の Oracle Application Server OC4J 構成に変更 (クラスタのサーバーの追加または削除、新規インスタンスの追加など) が加えられている場合は、スタンドアロン (手動管理) のインストール内容の再構成が必要になります。

関連資料:

『Oracle Application Server 管理者ガイド』

『Oracle Application Server 高可用性ガイド』

- スタンドアロンの Oracle HTTP Server 2.0 のインストールおよび構成を行います。ここでの指示は、次の各項で説明する構成手順に進む前に、Oracle HTTP Server 2.0 および標準の Oracle Application Server 中間層のすべてのインスタンスがインストールされていることを前提としています。

スタンドアロンの Oracle HTTP Server のインストール

スタンドアロンの Oracle HTTP Server は、Oracle Application Server CD パック内の OracleAS Companion CD に収録されています。

スタンドアロンの Oracle HTTP Server をインストールする手順は、次のとおりです。

1. OracleAS Companion CD を挿入し、次のように Oracle Universal Installer を起動してスタンドアロンの Oracle HTTP Server をインストールします。

- UNIX の場合:

```
prompt > cd
```

```
prompt > mount_point/1012disk1/runInstaller
```

- Windows の場合:

コンピュータが自動実行機能をサポートしている場合は、インストーラが自動的に起動します。

コンピュータが自動実行をサポートしていない場合は、`setup.exe` ファイルをダブルクリックして、インストーラを起動します。

2. Oracle Universal Installer が表示されたら、「ようこそ」画面を確認して「次へ」をクリックします。
3. Oracle 製品をコンピュータにインストールするのが初めての場合は、「インベントリ・ディレクトリと資格証明の指定」画面が表示されます。

この画面で次の情報を入力します。

- インベントリ・ディレクトリのフル・パス: インストーラ・ファイルを配置するディレクトリのフルパスを入力します。製品ファイルを配置する Oracle ホーム・ディレクトリとは異なるディレクトリを入力してください。

例: `/opt/oracle/oraInventory`

- オペレーティング・システム・グループ名: インベントリ・ディレクトリへの書き込み権限を持つオペレーティング・システム・グループの名前を入力します。

例: `oinstall`

「次へ」をクリックします。`orainstRoot.sh` を実行するよう求めるメッセージがウィンドウに表示されます。`root` ユーザーとして別のシェルでスクリプトを実行します。スクリプトは、`oraInventory` ディレクトリにあります。「続行」をクリックします。

4. 「ファイルの場所の指定」画面で、次の情報を入力します。
 - **名前:** この Oracle ホームを識別する名前を入力します。名前に空白を含めることはできません。また、最大 16 文字という制限があります。
例: OH_STANDOHS
 - **インストール先パス:** インストール先ディレクトリのフルパスを入力します。これは Oracle ホームです。ディレクトリが存在しない場合は自動的に作成されます。あらかじめディレクトリを作成しておく場合は、oracle ユーザーとして作成します。root ユーザーとして作成しないでください。
例: /opt/oracle/STANDOHS
 「次へ」をクリックします。
5. 「インストールする製品の選択」画面で Web Server Services 10.1.2.0.0 を選択し、「次へ」をクリックします。
6. 「インストール・タイプの選択」画面で、スタンドアロンの Oracle HTTP Server の希望するインストール・タイプを選択し、「次へ」をクリックします。
7. 「サマリー」画面で、選択内容を確認し、「インストール」をクリックします。
8. インストール進捗状況画面にインストールの進捗状況が表示されます。
9. 「コンフィギュレーション・アシスタント」画面で、Configuration Assistant の進捗状況を監視します。Configuration Assistant により、インストールしたコンポーネントが構成されます。root.sh を実行するよう求めるメッセージが表示されます。root ユーザーとして別のシェルでスクリプトを実行します。「OK」をクリックします。
10. インストールが完了すると、インストールの終了画面が表示されます。「終了」をクリックして、インストーラを終了します。

関連資料: Oracle Universal Installer の詳細は、Oracle Application Server のインストール・ガイドを参照してください。

OPMN の構成

Oracle Process Manager and Notification Server (OPMN) は次の 2 つのコンポーネントで構成されています。これらのコンポーネントは、Oracle Application Server プロセス間で送信される通知を同じ OPMN サーバーまたは異なる OPMN サーバー内で解析および伝達します。

- **Oracle Notification Server:** Oracle Notification Server (ONS) は、障害、リカバリ、起動、その他の関連する通知を、Oracle Application Server のコンポーネント間でトランスポートするためのメカニズムです。ONS は、パブリッシュ・サブスクライブ・モデルに従って動作します。つまり、Oracle Application Server コンポーネントは、ONS のサブスクリプションを行うたびに特定のタイプの通知を受信します。通知がパブリッシュされると、ONS はそれを適切なサブスクライバに送信します。
- **Oracle Process Manager:** Oracle Process Manager (PM) は、Oracle Application Server の一元化されたプロセス管理メカニズムで、Oracle Application Server プロセスの管理に使用されます。PM は、プロセスの起動、停止、再起動および障害検出を行います。PM での管理対象として構成されている Oracle Application Server プロセスは、opmn.xml ファイルに指定されています。

関連資料: 『Oracle Process Manager and Notification Server 管理者ガイド』

Oracle Process Manager and Notification Server を構成するには、次の手順を実行します。

1. ons.conf 構成ファイルを標準の Oracle Application Server 中間層のインストール位置から Oracle HTTP Server 2.0 の対応するディレクトリにコピーします。このファイルは ORACLE_HOME/opmn/conf ディレクトリにあります。ons.conf ファイルを編集し、すべての Oracle HTTP Server 2.0 スタンドアロン・インスタンスを手動管理クラスタに追加します。新しい ons.conf ファイルには、Oracle Application Server のすべての管理インスタンスのリストだけでなく、各手動管理インスタンスも記述されている必要があります。

次に `ons.conf` ファイルの形式を示します。

```
nodes=<host_name | host_ip>[:port] [, <host_name | host_ip>[:port]] [, ...]
```

例：

```
nodes=managed1:6000,managed1:6300,unmanaged2:6400
```

- ONS の正しいリモート・リスニング・ポートを特定するには、各手動管理インスタンスの OPMN 構成ファイルを調べます (`ORACLE_HOME/opmn/conf/opmn.xml` にあります)。ONS のリモート・リスニング・ポートの値は、`opmn.xml` ファイルの `/opmn/notification-server/port` にある XML 要素の `remote` 属性で指定されています。

例：

`unmanaged2` の `opmn.xml` に次の指定があるとします。

```
<opmn>
  <notification-server>
    <port local='6100' remote='6400' request='6300' />
    ...
  </notification-server>
<...>
</opmn>
```

この場合、`ons.conf` ファイルに次のように記述する必要があります。

```
nodes=managed1:6200,unmanaged1:6300,unmanaged2:6400
```

- ホストがマルチホームの場合 (複数の IP アドレスで構成されている場合)、`opmn.xml` ファイルの `/opmn/notification-server/ipaddr` にある `remote` 属性を設定することをお勧めします。この属性は、ONS リスナーを単一の有効な IPv4 アドレスまたはホスト名にバインドします。この属性が設定されていない場合、または `ipaddr` 要素が省略されている場合には、ONS はマルチホーム・ホストのすべての IP アドレスをリスニングします。

例：

ホスト `unmanaged2` はマルチホームで、`10.1.1.1`、`10.1.2.1` という IP アドレスを持つとします。ONS のリスニング先を IP アドレス `10.1.1.1` のみに制限するには、`opmn.xml` ファイルを次のように変更します。

```
<opmn>
  <notification-server>
    <ipaddr remote='10.1.1.1' />
    <port local='6100' remote='6400' request='6300' />
    ...
  </notification-server>
<...>
</opmn>
```

`ons.conf` は次のようになります。

```
nodes-managed1.oracle.com:6200,unmanaged1.oracle.com:6300,10.1.1.1:6400
```

- `ons.conf` ファイルには、必ずクラスタ内の各 Oracle Application Server インスタンスについてエントリを作成してください。この手動で作成したファイルを、クラスタ内のその他の Oracle HTTP Server 2.0 非管理インスタンスにコピーします。各非管理ノードの `opmn.xml` 構成ファイルに `remote` および `ipaddr` が設定されている場合は、`ons.conf` 構成データをそれらの設定と一致させる必要があります。

mod_oc4j の構成

手動管理されている Oracle HTTP Server 2.0 のリスナーを次のように構成し、管理されている Oracle Application Server Containers for J2EE (OC4J) にトラフィックをルーティングする必要があります。

1. 管理クラスタで OC4J を構成します。

関連資料:

- 『Oracle HTTP Server 管理者ガイド』
- 『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』

2. スタンドアロンの Oracle HTTP Server 2.0 の各手動管理インスタンスについて、mod_oc4j.conf ファイルが管理クラスタおよびインスタンスを指すよう構成されていることを確認します。

たとえば、手動管理の Oracle HTTP Server 2.0 のリスナーがクラスタ managed1 にトラフィックをルーティングするよう構成されている場合、インスタンス名 home は mod_oc4j.conf ファイルで次のように使用されます。

```
Oc4jMount /MyApp/* cluster://managed1:home
```

3. ルーティングが必要なアプリケーションごとに、マウント・ポイントを追加する必要があります。
4. Oracle HTTP Server を再起動して、構成の変更を有効にします。
 - UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
 - Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
5. 新規アプリケーションを構成するたびに、Oracle HTTP Server 2.0 スタンドアロンの mod_oc4j.conf を変更して、これらの変更を反映させる必要があります。

シングル・サインオンの構成

スタンドアロンの Oracle HTTP Server 2.0 にシングル・サインオン機能が必要な場合、次の手順を実行して、手動管理の Oracle HTTP Server 2.0 のリスナーを Oracle Application Server Single Sign-On に登録します。

1. SSO サーバー管理ツールを使用して、パートナ・アプリケーションを構成します。

関連資料: 『Oracle Application Server Single Sign-On 管理者ガイド』

2. osso.conf ファイルを手動で作成します。そのためには、パートナ・アプリケーションの構成後に、「パートナ・アプリケーションの編集」画面から必要なデータを切り取って貼り付けます。

たとえば、osso.conf ファイルの作成に必要な構成データが「パートナ・アプリケーションの編集」画面に表示されます。次に、「パートナ・アプリケーションの編集」ページの例を示します。

```
ID: 643C32F6
Token: Q2057R2D646C20F1
Encryption Key: 3F46C27C5153B7C7
Login URL: http://foobar.us.oracle.com:7778/pls/orasso.wvssso_app_admin.ls_login
Single Sign-Off: http://foobar.us.oracle.com:7778/pls/orasso.wvssso_app_admin.ls_logout
```

この「パートナ・アプリケーションの編集」画面のデータを使用すると、次のようなクリアテキストの osso.conf 構成ファイルを手動で作成できます。

```
sso_server_version=v1.4
cipher_key=3F46C27C5153B7C7
site_id=643C32F6
site_token=Q2057R2D646C20F1
login_url=http://foobar.us.oracle.com:7778/pls/orasso.wssso_app_admin.ls_login
logout_url=http://foobar.us.oracle.com:7778/pls/orasso.wssso_app_admin.ls_logout
cancel_url=http://foobar.us.oracle.com:7778
```

3. 新たに作成したファイルを osso 構成ディレクトリにコピーします。

```
ORACLE_HOME/ohs/conf/osso.conf cleartext
```

4. 平文のファイルを不明瞭化して、暗号鍵情報を保護する必要があります。そのためには、ORACLE_HOME/Apache/Apache/bin ディレクトリにある apobfuscate ツールを次のように使用します。

```
../bin/apobfuscate osso/conf/cleartext osso.conf
```

5. スタンドアロンの Oracle HTTP Server 2.0 の mod_osso.conf を編集し、SSO による Web リソース保護を有効にします。このファイルは ORACLE_HOME/ohs/conf ディレクトリにあります。OssosConfigFile ディレクティブが、パートナー・アプリケーション登録データを含む不明瞭化された osso.conf ファイルを指していることを確認します。また、httpd.conf ファイルで #include "ORACLE_HOME/ohs/conf/mod_osso.conf" ディレクティブのコメント設定が解除されていることを確認します。

例：

```
LoadModule osso_module libexec/mod_osso.so
<IfModule mod_osso.c>
OssosConfigFile conf/osso/osso.conf
OssosIpCheck off
OssosIdleTimeout off

Alias /private/ "<ApacheServerRoot>/private/"

<Location /private>
require valid_user
AuthType Basic
</Location>

</IfModule>
```

6. Oracle HTTP Server を再起動して、構成を有効にします。

- UNIX の場合：ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server
- Windows の場合：ORACLE_HOME\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server

サーバーとファイル位置の指定

この章では、Oracle HTTP Server およびサーバー管理者オプションの設定方法と、ファイル位置の指定方法について説明します。

内容は、次のとおりです。

- [サーバー機能と管理者機能の設定](#)
- [ファイル位置の指定](#)

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

サーバー機能と管理者機能の設定

次のディレクティブを使用して、基本的な Oracle HTTP Server 機能と管理者機能を設定します。これらのディレクティブは、`httpd.conf` ファイルの Main Server Configuration 部分にあります。

関連項目： B-2 ページの「[httpd.conf のファイル構造](#)」

- [ServerName](#)
- [UseCanonicalName](#)
- [ServerAdmin](#)
- [ServerSignature](#)
- [ServerTokens](#)
- [ServerAlias](#)

ServerName

サーバーでリダイレクション URL の作成に使用するホスト名を設定できます。末尾にスラッシュ (/) がない場合も、このホスト名でディレクトリにアクセスできます。

たとえば、実際のマシンのメイン名が `main.company.com` の場合、`ServerName www.company.com` が使用されます。

関連資料： Apache Server マニュアルの「[ServerName directive](#)」

UseCanonicalName

URL を同じサーバーにリダイレクトするときに使用するホスト名とポートを指定します。

- On: サーバーでは [ServerName](#) で設定したホスト名とポートの値が使用されます。これはデフォルト設定です。
- Off: サーバーではリクエストで指定したホスト名とポートが使用されます。

例: `UseCanonicalName On`

関連資料： Apache Server マニュアルの「[UseCanonicalName directive](#)」

ServerAdmin

クライアント側で発生するすべてのデフォルト・エラー・メッセージに含まれる電子メール・アドレスを作成します。このディレクティブは、特定のサーバー用に別個の電子メール・アドレスを作成する場合に役立ちます。

例: `ServerAdmin you@your.emailaddress`

関連資料： Apache Server マニュアルの「[ServerAdmin directive](#)」

ServerSignature

エラー・メッセージなど、戻されたレスポンスを作成したサーバーを、様々なプロキシからサーバー側で認識できるようにします。

- on: 戻されたドキュメントに、[ServerName](#) やサーバーのバージョン番号などの情報を含むフッターが作成されます。これはデフォルト設定です。
- email: ドキュメントの [ServerAdmin](#) への `mailto:` 参照も作成されます。
- off: フッターおよび `mailto:` 参照は作成されません。

例: `ServerSignature On`

関連資料： Apache Server マニュアルの「[ServerSignature directive](#)」

ServerTokens

エラー・メッセージなどにおいて、クライアントに戻されるサーバー情報を制御します。この情報には、サーバーの一般的なオペレーティング・システム・タイプの説明と、コンパイルされるモジュールが含まれます。

- `min(imal)`: サーバー名およびバージョンなどの情報が提供されます。
- `OS`: サーバー名、バージョンおよびオペレーティング・システムなどの情報が提供されません。
- `full`: サーバー名、バージョン、オペレーティング・システムおよびコンパイル済モジュールなどの情報が提供されます。

例: `Server: Apache/2.0.0 (UNIX) PHP/3.0 MyMod/1.2`

関連資料: Apache Server マニュアルの「`ServerTokens directive`」

ServerAlias

現行の仮想ホストの代替名を設定します。

例:

```
<VirtualHost *>
ServerName server.domain.com
ServerAlias server server2.domain.com server2
...
</VirtualHost>
```

関連資料: Apache Server マニュアルの「`ServerAlias directive`」

ファイル位置の指定

次のディレクティブを使用して、各種サーバー・ファイルの位置を制御できます。これらのディレクティブは、`httpd.conf` ファイルの `Global Environment` セクションにあります。

関連項目: [B-2 ページの「`httpd.conf` のファイル構造](#)」

- [CoreDumpDirectory](#)
- [DocumentRoot](#)
- [ErrorLog](#)
- [LockFile](#)
- [PidFile](#)
- [ScoreBoardFile](#)
- [ServerRoot](#)

CoreDumpDirectory

サーバーによるコア・ダンプ・ディレクトリを指定します。デフォルトは [ServerRoot](#) ディレクトリです。このディレクティブは、UNIX にのみ適用されます。

例: `CoreDumpDirectory /tmp`

関連資料: Apache Server マニュアルの「`CoreDumpDirectory directive`」

DocumentRoot

httpdによってファイルが処理されるディレクトリを設定します。[Alias](#)などのディレクティブと一致しないかぎり、サーバーにより、リクエストされたURLからのパスがドキュメント・ルートに追加され、静的コンテンツ用のドキュメントへのパスが形成されます。

例: DocumentRoot "/oracle/Apache/Apache/htdocs"

関連資料: Apache Server マニュアルの「DocumentRoot directive」

ErrorLog

発生したエラーがサーバーによって記録されるファイルの名前を設定します。このファイル名の先頭にスラッシュ (/) がない場合は、[ServerRoot](#) への相対ファイル名とみなされます。このファイル名の先頭に縦線 (|) がある場合は、エラー・ログ処理用に起動されるコマンドとみなされます。

例: ErrorLog "|/private1/oracle/Apache/Apache/bin/rotatelog
/private1/oracle/Apache/Apache/logs/error_log 43200"

関連資料: Apache Server マニュアルの「ErrorLog directive」

LockFile

Oracle HTTP Server をコンパイルするときに使用するロック・ファイルへのパスを `USE_FCNTL_SERIALIZED_ACCEPT` または `USE_FLOCK_SERIALIZED_ACCEPT` で設定します。デフォルト値を使用することをお勧めします。設定を変更するのは、主に `logs` ディレクトリに NFS がマウントされている場合で、これはロック・ファイルをローカル・ディスクに格納する必要があります。

例: LockFile /oracle/Apache/Apache/logs/httpd.lock

関連資料: Apache Server マニュアルの「LockFile directive」

PidFile

サーバーによってプロセス識別番号が記録される、PID ファイルの位置を設定および変更できます。このファイル名の先頭にスラッシュ (/) がない場合は、[ServerRoot](#) への相対ファイル名とみなされます。

例: PidFile /oracle/Apache/Apache/logs/httpd.lock

関連資料: Apache Server マニュアルの「PidFile directive」

ScoreBoardFile

このディレクティブは、一部のアーキテクチャで必須です。このディレクティブを使用して、親プロセスと子プロセス間の通信用にサーバーで使用されるファイルを設定します。アーキテクチャにスコアボード・ファイルが必要かどうかを確認するには、[Oracle HTTP Server](#) を実行し、このディレクティブで指定したファイルが作成されるかどうかを調べます。アーキテクチャにスコアボード・ファイルが必要な場合は、このファイルが同時に複数のサーバー起動に使用されないことを確認する必要があります。

例: /oracle/Apache/Apache/logs/httpd.scoreboard

関連資料: Apache Server マニュアルの「ScoreBoardFile directive」

ServerRoot

`conf` および `logs` サブディレクトリを含むディレクトリを指定します。`-f` オプションを指定してサーバーを起動する場合は、[ServerRoot](#) を指定する必要があります。

例: "/oracle/Apache/Apache"

関連資料: Apache Server マニュアルの「ServerRoot directive」

サーバー・プロセスの管理

この章では、Oracle HTTP Server プロセスの概要と、これらのプロセスを制御し、モニターする方法について説明します。

内容は、次のとおりです。

- Oracle HTTP Server の処理モデル
- サーバー・プロセスの処理
- プロセス数と接続数の構成
- root としての Oracle HTTP Server の実行
- セキュリティに関する考慮事項
- プロセス情報の取得

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

Oracle HTTP Server の処理モデル

Oracle HTTP Server を起動すると、システムで `http` または `https` リクエストをリスニングしてレスポンスを返す準備ができたこととなります。リクエスト処理モデルは、UNIX と Windows で異なります。

UNIX の場合は、複数の子プロセスを管理する親プロセスが 1 つ存在します。子プロセスは、リクエストの処理を担当します。親プロセスは、構成に基づき、必要に応じて追加の子プロセスを起動します。追加の子プロセスを動的に起動することは可能ですが、最初に十分な数の子プロセスが起動されるようにサーバーを構成し、子プロセスをそれ以上作成しなくてもリクエストを処理できるようにすることをお勧めします。

Windows の場合は、親プロセスと子プロセスが 1 つずつ存在します。子プロセスは、クライアント・リクエストの処理を担当するスレッドを作成します。作成されるスレッドの数は静的であり、構成可能です。

サーバー・プロセスの処理

UNIX では、デフォルトで、メインの `httpd` 親プロセスと子プロセスが **Oracle Application Server** をインストールしたユーザーとして動作するよう構成されます。子プロセスの権限を設定するには、**User** および **Group** ディレクティブを使用します。root として実行していない場合、これらのディレクティブは無視されます。子プロセスには、処理されるすべての内容を読み取ることのできる権限が必要です。

- [Group](#)
- [User](#)

Group

サーバーがリクエストへのレスポンスに使用するグループを指定します。このディレクティブを使用するには、スタンドアロン・サーバーを root で実行する必要があります。サーバーの実行用に新規グループを作成することをお勧めします。このディレクティブは UNIX にのみ適用されます。

例: `Group myorg`

関連資料: Apache Server マニュアルの「Group directive」

User

サーバーがリクエストへのレスポンスに使用するユーザー ID を指定します。このディレクティブを使用するには、スタンドアロン・サーバーを root で実行する必要があります。任意のユーザーが使用できるファイルへのアクセス権限が必要ですが、`httpd` リクエスト用以外のコードは実行できないようにする必要があります。サーバーの実行用に新規ユーザーを設定することをお勧めします。このディレクティブは UNIX にのみ適用されます。

例: `User jdoe`

関連資料: Apache Server マニュアルの「User directive」

プロセス数と接続数の構成

次のディレクティブを使用して、クライアント・リクエストの処理方法を構成し、Oracle HTTP Server のパフォーマンスをチューニングします。これらのディレクティブは、`httpd.conf` ファイルの Global Environment セクションにあります。

関連項目: [B-2 ページの「httpd.conf のファイル構造」](#)

- [StartServers](#)
- [ThreadsPerChild](#)

- [MaxClients](#)
- [MaxRequestsPerChild](#)
- [MaxSpareServers](#)
- [MinSpareServers](#)

StartServers

Oracle HTTP Server の起動時に作成される子サーバー・プロセスの数を設定します。デフォルトは 5 です。このディレクティブは UNIX にのみ適用されます。

使用方法: `StartServers 5`

関連資料: Apache Server マニュアルの「`StartServers directive`」

ThreadsPerChild

リクエストを処理する子スレッドの最大数を制御します。デフォルトは 50 です。このディレクティブは Windows にのみ適用されます。

使用方法: `ThreadsPerChild 50`

関連資料: Apache Server マニュアルの「`ThreadsPerChild directive`」

MaxClients

一度に処理できるリクエスト数を制限します。デフォルト値と推奨値は 150 です。このディレクティブは UNIX にのみ適用されます。

使用方法: `MaxClients 150`

関連資料: Apache Server マニュアルの「`MaxClients directive`」

MaxRequestsPerChild

子プロセスで終了前に処理されるリクエスト数を制御します。値をデフォルトの 0 に設定すると、プロセスは終了しません。

Windows では、これを 0 に設定することをお勧めします。0 以外の値に設定すると、リクエスト数に達したときに子プロセスが終了し、再作成されます。その際、子プロセスは構成ファイルを再度読み取ります。そのため、構成ファイルを変更したが、変更を適用するつもりはないという場合に、予期しない処理が行われる可能性があります。

使用方法: `MaxRequestsPerChild 0`

関連資料: Apache Server マニュアルの「`MaxRequestsPerChild directive`」

MaxSpareServers

アイドル状態の子サーバー・プロセスの最大数を設定します。アイドル・プロセスは、稼働中であってもリクエストを処理していないプロセスです。親プロセスは、このディレクティブの設定値を超えるアイドル状態の子プロセスを中断します。デフォルトは 20 です。このディレクティブは UNIX にのみ適用されます。

使用方法: `MaxSpareServers 20`

関連資料: Apache Server マニュアルの「`MaxSpareServers directive`」

MinSpareServers

アイドル状態の子サーバー・プロセスの最小数を設定します。アイドル・プロセスは、稼働中であってもリクエストを処理していないプロセスです。アイドル状態のプロセス数が減少すると、親プロセスにより最大で1秒当たり1プロセスの割合で新規の子プロセスが作成されます。デフォルトは5です。このディレクティブはUNIXにのみ適用されます。

使用方法: MinSpareServers 5

関連資料: Apache Server マニュアルの「MinSpareServers directive」

root としての Oracle HTTP Server の実行

UNIX では、1024 以外のポート上で実行するには、root として実行する必要があります。

Oracle HTTP Server を root として実行する手順は、次のとおりです。

1. 次のコマンドを使用して、Oracle HTTP Server を停止します。

```
ORACLE_HOME/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server
```

2. root ユーザーに変更します。

3. ORACLE_HOME/ohs/bin にナビゲートして、次のコマンドを実行します。

```
chown root .apachectl  
chmod 6750 .apachectl
```

4. root を終了します。

5. 次のコマンドを使用して、Oracle HTTP Server を再起動します。

```
ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server
```

セキュリティに関する考慮事項

UNIX でのセキュリティ強化には、ユーザーを nobody に変更できます。子プロセスがユーザー nobody としてタスクを実行できることを確認してください。すべてのファイルがユーザー nobody により読取り可能になるように（理想的には書込み可能にならないように）静的コンテンツ（UNIX の場合は ORACLE_HOME/ohs/htdocs ディレクトリ、Windows の場合は ORACLE_HOME\ohs\htdocs ディレクトリなど）をすべて変更します。また、すべての CGI および FastCGI プログラムをユーザー nobody が実行できることも確認してください。

最後に、キャッシュ済コンテンツには機密データが含まれている可能性があるので、ファイル・システム・キャッシュの最終コンテンツはセキュリティで保護する必要があります。したがって、Oracle HTTP Server が nobody として動作する場合でも、このユーザーとしてのシステムへのアクセスは十分に保護する必要があります。

プロセス情報の取得

Oracle HTTP Server プロセスをモニターするには複数の方法があります。

1. Windows では Performance Monitor、UNIX では ps ユーティリティを使用します。

関連資料: 詳細は、『Oracle Application Server パフォーマンス・ガイド』およびオペレーティング・システムのマニュアルを参照してください。

2. `mod_status` を使用してサーバーのステータスをモニターします。デフォルトでは、ローカル・ホストからのみ使用可能です。

5

ネットワーク接続の管理

この章では、IP アドレスとポートを指定する方法、およびサーバーの相互作用とネットワーク接続の永続性を管理する方法について説明します。

内容は、次のとおりです。

- リスナー・ポートおよびアドレスの指定
- サーバーとネットワーク間の相互作用の管理
- 接続の永続性の管理
- クライアント IP アドレスの取得
- リバース・プロキシとロード・バランサの構成

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

リスナー・ポートおよびアドレスの指定

Oracle HTTP Server が起動時にリスニングするポートは、インストール・タイプによって異なります。

表 5-1 に Oracle HTTP Server のポートに関する情報を示します。

表 5-1 Oracle HTTP Server のポート

プラットフォーム	中間層インストール	Infrastructure インストール
Solaris	非 SSL: 7777 (7777 ~ 7877 の範囲)	非 SSL: 7777 (7777 ~ 7877 の範囲)
	SSL: 4443 (4443 ~ 4543 の範囲)	SSL: 4443 (4443 ~ 4543 の範囲)
Windows	非 SSL: 80 (7777 ~ 7877 の範囲)	非 SSL: 7777 (7777 ~ 7877 の範囲)
	SSL: 443 (4443 ~ 4543 の範囲)	SSL: 4443 (4443 ~ 4543 の範囲)

たとえば、ポート 7777 または 80 が占有されている場合、Oracle HTTP Server は 7777 ~ 7877 の範囲内にある、次に使用可能なポート番号でリスニングします。したがって、ポート 7778 などリスニングします。

注意： デフォルトでは、SSL は無効です。SSL を有効化する方法については、第 9 章「Oracle HTTP Server での SSL の有効化」を参照してください。

ファイル `setupinfo.txt` は、UNIX では `ORACLE_HOME/install`、Windows では `ORACLE_HOME\install` に自動的に生成されます。このファイルには、Oracle HTTP Server のポート情報が含まれます。このファイルはインストール時に生成され、その後は更新されません。Oracle HTTP Server の再起動後は、このファイル内の情報は利用できません。

Oracle HTTP Server のリスナー・ポート (SSL および非 SSL) は、インストール後に変更できます。ポートを変更した場合は、新規ポート番号を使用するように他のコンポーネントも更新する必要があります。

関連資料： 『Oracle Application Server 管理者ガイド』

サーバーを、複数のポート、選択したアドレスまたはその組合せをリスニングするように指定できます。Listen ディレクティブを使用して、リスナーのポートとアドレスを指定します。Listen ディレクティブは、`httpd.conf` ファイルの Global Environment セクションにあります。複数の Listen ディレクティブを使用して、複数のポートでリスニングできます。

例：

- Listen 7778
- Listen 12.34.56.78:80

関連資料： Apache Server マニュアルの「Listen directive」

サーバーとネットワーク間の相互作用の管理

次のディレクティブを使用して、サーバーとネットワークの相互作用を指定します。これらのディレクティブは、`httpd.conf` ファイルの Global Environment セクションにあります。

- ListenBackLog
- SendBufferSize
- TimeOut

関連項目： B-2 ページの「httpd.conf のファイル構造」

ListenBackLog

ペンディング接続のキューの最大長を指定します。サーバーで TCP SYN オーバーロードが発生し、多数の新規接続がオープンされているがタスクが完了しない場合に、このディレクティブが役立ちます。

関連資料： Apache Server マニュアルの「ListenBackLog directive」

SendBufferSize

TCP バッファのサイズを指定のバイト数まで増やしてパフォーマンスを改善します。

関連資料： Apache Server マニュアルの「SendBufferSize directive」

TimeOut

サーバーの、次の最大待機時間を秒単位で設定します。

- 1つの GET リクエストの受信にかかる合計秒数
- POST または PUT リクエストで TCP パケットを受信する間隔
- レスポンスの TCP パケットが送信されるとき ACK の間隔

デフォルトは 300 秒です。

関連資料： Apache Server マニュアルの「TimeOut directive」

接続の永続性の管理

次のディレクティブを使用して、サーバーによる永続的な接続の処理方法を決定します。これらのディレクティブは、httpd.conf ファイルの Global Environment セクションにあります。

- [KeepAlive](#)
- [KeepAliveTimeout](#)
- [MaxKeepAliveRequests](#)

関連資料：

- 『Oracle Application Server パフォーマンス・ガイド』
- [B-2 ページの「httpd.conf のファイル構造」](#)

KeepAlive

このディレクティブを「On」に設定すると、HTTP 1.1 KeepAlive のサポートが有効になり、1つのクライアントからの複数の HTTP リクエストに対して同じ TCP 接続を再利用できます。

関連資料： Apache Server マニュアルの「KeepAlive directive」

KeepAliveTimeout

サーバーが [KeepAlive](#) 接続をクローズする前に、後続のリクエストを待機する秒数を設定します。リクエストが受信されると、[TimeOut](#) ディレクティブで指定したタイムアウト値が適用されます。デフォルトは 15 秒です。

関連資料： Apache Server マニュアルの「KeepAliveTimeout directive」

MaxKeepAliveRequests

KeepAlive が「On」になっているときの、接続ごとの許容リクエスト数を制限します。「0」に設定すると、許容リクエスト数は無制限となります。デフォルトは 100 です。

関連資料： Apache Server マニュアルの「MaxKeepAliveRequests directive」

クライアント IP アドレスの取得

UseWebCacheIp は、グローバル・ディレクティブで、Oracle HTTP Server でクライアントの IP アドレスを取得できるようにします。「On」または「Off」に設定できますが、デフォルト値は「Off」です。デフォルトで「On」に設定されていないのは、状況によってセキュリティ・ホールの原因となることがあるためです。

OracleAS Web Cache が Oracle HTTP Server の前でリバース・プロキシとして機能している場合、クライアントからの TCP 接続は、OracleAS Web Cache で終了します。Oracle HTTP Server が実際に認識する TCP 接続は、OracleAS Web Cache が起点となります。Oracle HTTP Server は、クライアントの IP アドレスを取得して、次のような様々な目的に使用します。

- CGI 変数 REMOTE_ADDR への値の移入。この変数を Oracle HTTP Server の中および背後のアプリケーションで使用すると、クライアントの出所を特定できます。
- mod_access の許可 / 拒否ルールの評価。このルールにより、管理者は IP アドレスに基づいてアクセスを制限できます。

UseWebCacheIp ディレクティブを指定しないと、OracleAS Web Cache が Oracle HTTP Server の前で使用される場合、この機能は働きません。Oracle HTTP Server では、すべての接続が同じ場所（OracleAS Web Cache が稼働している IP アドレス）を起点にしていると認識するためです。

OracleAS Web Cache は、Oracle HTTP Server に転送するすべてのリクエストとともに、受信したクライアント接続の IP アドレスを含むヘッダーを送信します。UseWebCacheIp は、「On」に設定されている場合、TCP 接続の値ではなく、このヘッダーの IP 値をクライアントの IP アドレスとして使用するように Oracle HTTP Server に指示します。これにより、CGI 変数 REMOTE_ADDR は正しい値を保持し、mod_access は正しく機能することができます。

このディレクティブは、クライアントが OracleAS Web Cache 経由でしか Oracle HTTP Server に接続できないことが確実な場合にのみ設定してください。クライアントは、Oracle HTTP Server に直接接続できる場合、クライアント IP の転送に使用されるヘッダーを見つけ出し、目的の IP アドレスから取得されたかのように設定する必要があります。ファイアウォールと OracleAS Web Cache を使用する標準的な設定では、ファイアウォール経由でオープンされるポートは OracleAS Web Cache のポートのみです。したがって、クライアントから Oracle HTTP Server へのパスは必ず OracleAS Web Cache を通ります。この場合、UseWebCacheIp を「On」に設定しても問題ありません。

リバース・プロキシとロード・バランサの構成

Oracle Application Server は、デフォルトでは、Oracle HTTP Server の **ServerName** ディレクティブで設定されているローカル・ホスト名を使用してインストールします。ほとんどの Web サイトが、Web サーバーまたはアプリケーション・サーバー用に特定のホスト名またはドメイン名を使用する傾向があります。ただし、ServerName ディレクティブを使用すると、Oracle HTTP Server がローカル・ホストを使用してインスタンス化されるため、そのままでは不可能です。

例 5-1 Oracle HTTP Server とリバース・プロキシおよびロード・バランサの使用

ドメイン名 : www.oracle.com:80 123.456.7.8 (リバース・プロキシ、ロード・バランサまたはファイアウォール上に置かれる)

Oracle Application Server ホストのホスト名 : server.oracle.com 123.456.7.9

Oracle Application Server ホストのサーバー名およびポート : server.oracle.com:7777

httpd.conf ファイルに次の変更を加えます。

```
Port 80
Listen 7777
Listen 80
# Virtual Hosts
# This section is mandatory for URLs that are generated by
# the PL/SQL packages of the Oracle Portal and various other components
# These entries dictate that the server should listen on port
# 7777, but will assert that it is using port 80, so that
# self-referential URLs generated specify www.oracle.com:80
# This will create URLs that are valid for the browser since
# the browser does not directly see the host server.oracle.com.
NameVirtualHost 123.456.7.9:7777
<VirtualHost server.oracle.com:7777>
ServerName www.oracle.com
Port 80
</VirtualHost>
# Since the previous virtual host entry will cause all links
# generated by the Oracle Portal to use port 80, the server.company.com
# server needs to listen on 80 as well since the Parallel Page
# Engine will make connection requests to Port 80 to request the
# portlets.
NameVirtualHost 123.456.7.9:80
<VirtualHost server.oracle.com:80>
ServerName www.oracle.com
Port 80
</VirtualHost>
```

関連資料：『Oracle Application Server 高可用性ガイド』

サーバー・ログの構成と使用

この章では、Oracle Diagnostic Logging、ログの書式、各種ログ・ファイルおよびその位置について説明します。

内容は、次のとおりです。

- [Oracle Diagnostic Logging の使用](#)
- [ログ・レベルの指定](#)
- [ログ・ファイルの指定](#)

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

Oracle Diagnostic Logging の使用

Oracle では、診断メッセージの報告のために新しい方法を提供しています。この新しい方法は Oracle Diagnostic Logging (ODL) と呼ばれ、診断メッセージとログ・ファイルのための共通形式と、Oracle Application Server 全体の様々なコンポーネントからの全診断メッセージを相互に関係付ける仕組みを提供しています。ODL を使用して、各コンポーネントはそのコンポーネント専用のプライベート・ローカル・リポジトリにそれぞれのメッセージをログします。LogLoader というツールが、各リポジトリからメッセージを収集して共通リポジトリにロードします。メッセージは、この共通リポジトリで 1 つのログ・ストリームとして表示するか、様々な方法で分析することができます。

Oracle Application Server の診断ログ・ファイルは、テキスト・エディタを使用して表示できます。

関連資料: 『Oracle Application Server 管理者ガイド』

ODL については、次の項で詳しく説明します。

- [概要](#)
- [Oracle HTTP Server の構成](#)

概要

Oracle HTTP Server では、ログ・メッセージを生成する形式を選択できます。従来型の Apache メッセージ形式でログ・メッセージを生成するか、ODL を使用してログ・メッセージの生成を続行できます。ODL は、ログ・メッセージ生成用の新しい Oracle 標準に準拠しています。

Oracle HTTP Server の構成

Oracle HTTP Server で ODL を使用可能にするには、httpd.conf ファイルに次のディレクティブを入力します。

- `OraLogMode oracle | odl | apache`
- `OraLogSeverity module_name <msg_type>{:msg_level}`
- `OraLogDir <bus stop dir>`

モジュール固有のログ重大度が有効化してからモジュールがロギングを実行するように、これらのディレクティブは、httpd.conf ファイルでなんらかのモジュールがロードされる前に (LoadModule ディレクティブの前に) 指定することをお勧めします。

OraLogMode oracle | odl | apache

Oracle ログ書式、従来型の Apache ログ書式および ODL ログ書式を切り替えることができます。ログ書式は次のように定義されています。

- **oracle:** XML に完全に準拠した、複数行からなる XML 形式のログ・レコードです。提供される情報が最も多い書式です。
- **odl:** 標準の Apache ログ書式。リクエストに特に関連付けられたログ・レコードの ECID 情報です。これはデフォルト設定です。
- **apache:** 標準の Apache ログ書式。提供される情報が最も少ない書式です。

OraLogSeverity module_name <msg_type>{:msg_level}

メッセージ重大度を設定できます。このディレクティブで指定されるメッセージ重大度は、必要最低限のメッセージ重大度として解釈され、この重大度レベル以上のすべてのメッセージはログされます。

OraLogSeverity は、複数回指定できます。グローバル (`module_name` なし) に指定した後、モジュール固有のログ重大度が必要なモジュール 1 つにつき 1 回ずつ指定できます。

このディレクティブは、OraLogMode が `oracle` に設定されているときのみ使用されます。このディレクティブは、LogLevel ディレクティブのかわりに使用できますが、必須ではありません。OraLogSeverity が指定され、OraLogMode が `oracle` に設定されている場合、LogLevel は無視されます。

module_name この引数は、モジュール構造内に示されるモジュールの内部名です。<IfModule> ディレクティブもこの内部名を使用します。モジュール構造は、モジュール構造を定義するファイルの `_FILE_` マクロの値から（パス接頭辞を削除して）モジュール名を導出します。モジュール名を指定しない場合は、OraLogSeverity ディレクティブがグローバルに適用されます。

モジュール名を指定した場合は、指定されたモジュールで発生した、すべてのメッセージのグローバル・ディレクティブの値をこのディレクティブがオーバーライドします。ロードされないモジュールの名前を指定すると、エラーが発生します。

msg_type メッセージ・タイプは大文字でも小文字でも指定できますが、メッセージ出力は大文字で表示されます。このパラメータには、次の値のいずれかを指定する必要があります。

- INTERNAL_ERROR
- ERROR
- WARNING
- NOTIFICATION
- TRACE

msg_level このパラメータは、範囲が 1 ～ 32 の整数に指定する必要があります。1 は最高重大度、32 は最低重大度を示します。レベル 1 を使用すると、レベル 32 よりメッセージが少なくなります。

表 6-1 に、OraLogSeverity の例をいくつか示します。

表 6-1 OraLogSeverity の例

OraLogSeverity の例	処理
OraLogSeverity INTERNAL_ERROR:10	レベル 1 ～ 10 の内部エラー (INTERNAL_ERROR) タイプのメッセージをすべてログします。
OraLogSeverity WARNING:7	全レベルの内部エラー (INTERNAL_ERROR) タイプのメッセージをすべてログします。 全レベルのエラー (ERROR) タイプのメッセージをすべてログします。 レベル 1 ～ 7 の警告 (WARNING) タイプのメッセージをすべてログします。

表 6-1 OraLogSeverity の例 (続き)

OraLogSeverity の例	処理
OraLogSeverity WARNING OraLogSeverity mod_oc4j.c NOTIFICATION:4	<p>メッセージ・ソースが mod_oc4j の場合:</p> <ul style="list-style-type: none"> ■ 全レベルの内部エラー (INTERNAL_ERROR) タイプのメッセージをすべてログします。 ■ 全レベルのエラー (ERROR) タイプのメッセージをすべてログします。 ■ 全レベルの警告 (WARNING) タイプのメッセージをすべてログします。 ■ レベル 1 ~ 4 の通知 (NOTIFICATION) タイプのメッセージをすべてログします。 <p>その他のソースからのメッセージの場合:</p> <ul style="list-style-type: none"> ■ 全レベルの内部エラー (INTERNAL_ERROR) タイプのメッセージをすべてログします。 ■ 全レベルのエラー (ERROR) タイプのメッセージをすべてログします。 ■ 全レベルの警告 (WARNING) タイプのメッセージをすべてログします。

デフォルト メッセージ・レベルを指定しない場合、レベルはデフォルトの最低重大度になります。ディレクティブ全体を指定しない場合、グローバルな Apache の LogLevel ディレクティブの値が使用され、表 6-2 に示されるように、これに対応する Oracle メッセージ・タイプおよび対応する範囲内の最低レベル (最高値) に変換されます。

表 6-2 Apache ログ・レベルと Oracle メッセージ・タイプの対応

Apache ログ・レベル	Oracle メッセージ・タイプ
emerg	INTERNAL_ERROR:16
alert	INTERNAL_ERROR:32
crit	ERROR:16
error	ERROR:32
warn	WARNING:32
notice	NOTIFICATION:16
info	NOTIFICATION:32
debug	TRACE:32

関連項目: 6-5 ページの「ログ・レベルの指定」

OraLogDir <bus stop dir>

すべてのログ・ファイルを含むディレクトリへのパスを指定します。このディレクトリは存在している必要があります。

デフォルト:

- UNIX の場合: `ORACLE_HOME/Apache/Apache/logs/oracle`
- Windows の場合: `ORACLE_HOME¥Apache¥Apache¥logs¥oracle`

ログ・レベルの指定

表 6-3 に、様々なロギング・レベル、説明およびメッセージの例を示します。

表 6-3 ロギング・レベル

ロギング・レベル	説明	メッセージの例
emerg	緊急 - システムは使用不可です。	"Child cannot open lock file.Exiting."
alert	ただちに処理する必要があります。	"getpwuid: couldn't determine user name from uid"
crit	クリティカル条件。	"socket: Failed to get a socket, exiting child"
error	エラー条件。	"Premature end of script headers"
warn	警告条件。	"child process 1234 did not exit, sending another SIGHUP"
notice	正常だが重要な条件。	"httpd: caught SIGBUS, attempting to dump core in..."
info	情報メッセージ。	"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."
debug	デバッグ・レベルのメッセージ。	"Opening config file..."

注意： LogLevel ディレクティブは、OraLogMode が oracle で、OraLogSeverity が設定されている場合、省略できます。

ログ・ファイルの指定

この項では、次のログ・ファイルの機能と位置について説明します。

- [アクセス・ログ](#)
- [カスタム・ログ](#)
- [エラー・ログ](#)
- [PID ファイル](#)
- [パイプされたログ](#)
- [リライト・ログ](#)
- [スクリプト・ログ](#)
- [SSL ログ](#)
- [送信ログ](#)

中程度にビジーなサーバー上では、既存のログを移動または削除して、ログ・ファイルを定期的に切り替えることが重要です。この場合、新規ログ・ファイルがオープンされるように、ログ・ファイルを移動または削除した後にサーバーを再起動する必要があります。

関連資料： Apache Server マニュアルの「Log Rotation」

アクセス・ログ

サーバーによって処理されたすべてのリクエストが記録されます。アクセス・ログの位置と内容は、[カスタム・ログ・ディレクティブ](#)で制御します。LogFormat ディレクティブを使用すると、ログの内容を簡単に選択できます。

LogFormat の指定

LogFormat を使用して、ログ・ファイルに含める情報と書込み方法を指定します。デフォルトの書式は Common Log Format (CLF) です。CLF 書式は `host ident authuser date request status bytes` です。

- `host`: クライアントのドメイン名または IP アドレス
- `ident`: IdentityCheck が有効化されており、クライアント・マシンで `identd` が実行されている場合のクライアント識別情報
- `authuser`: 許可されたユーザーのユーザー ID
- `date`: `<day/month/year:hour:minute:second>` 書式のリクエスト日時
- `request`: 二重引用符で囲まれたクライアントからのリクエスト行
- `status`: クライアントに戻される 3 桁のステータス・コード
- `bytes`: ヘッダーを除いた、クライアントに戻されるバイト数

関連資料: Apache Server マニュアルの「Access Log」

カスタム・ログ

CustomLog ディレクティブは、サーバーへのリクエストをログするために使用されます。ログ書式が指定され、環境変数を使用して、リクエストの特性に応じてオプションでログできます。

関連資料: Apache Server マニュアルの「CustomLog directive」

エラー・ログ

診断情報を送信し、エラー・メッセージをログ・ファイルに記録します。デフォルトでは、このファイルは次の場所にあります。

- UNIX の場合: `ORACLE_HOME/ohs/logs/error_log`
- Windows の場合: `ORACLE_HOME\ohs\logs\error_log`

ファイル名は、[ErrorLog](#) ディレクティブを使用して設定できます。

関連資料: Apache Server マニュアルの「ErrorLog directive」

PID ファイル

サーバーを起動すると、親 `httpd` プロセスのプロセス ID が PID ファイルに記録されます。このファイルは、デフォルトでは次の場所にあります。

- UNIX の場合: `ORACLE_HOME/ohs/logs/httpd.pid`
- Windows の場合: `ORACLE_HOME\ohs\logs\httpd.pid`

このファイル名は、[PidFile](#) ディレクティブを使用して変更できます。管理者は、プロセス ID をデーモンの再起動と終了に使用します。プロセスが異常終了（または中断）した場合は、子 `httpd` プロセスを中断する必要があります。

関連資料: Apache Server マニュアルの「Pid File」

パイプされたログ

Oracle HTTP Server には、エラー・ログとアクセス・ログをファイルに直接書き込むのではなく、別のプロセスへのパイプを介してファイルに書き込む機能が用意されています。これによりロギングの柔軟性が高まるため、メイン・サーバーにコードを追加する必要はありません。ログをパイプに書き込むには、ファイル名を縦線「|」で置き換え、続けて標準入力でログ入力を受け入れる実行可能ファイルの名前を指定します。Oracle HTTP Server はサーバーの起動時にパイプされたログ・プロセスを開始し、サーバーの実行中にクラッシュすると再開します。

パイプされたログ・プロセスは Oracle HTTP Server の親 httpd プロセスにより作成され、そのプロセスのユーザー ID を継承します。つまり、通常、パイプされたログ・プログラムは root で実行されるため、プログラムを単純かつ安全な状態に保つ必要があります。

関連資料： Apache Server マニュアルの「Piped Log」

リライト・ログ

リライト・ログは、[mod_rewrite](#) を使用する場合のデバッグに必要です。このログ・ファイルでは、リライト・エンジンによるリクエストの変換方法の詳細分析が生成されます。詳細レベルは、RewriteLogLevel ディレクティブを使用して制御します。

関連資料： Apache Server マニュアルの「Rewrite Log」

スクリプト・ログ

CGI スクリプトからの入出力を記録できます。このファイルはテストにのみ使用し、稼働中のサーバーには使用しないでください。

関連資料： Apache Server マニュアルの「Script Log」

SSL ログ

Oracle HTTP Server を SSL モードで起動すると、ssl_engine_log および ssl_request_log が次の場所に作成されます。

- UNIX の場合 : `ORACLE_HOME/ohs/logs`
- Windows の場合 : `ORACLE_HOME\ohs\logs`

ssl_engine_log では SSL とプロトコルの問題が追跡され、ssl_request_log ではユーザー・アクティビティが記録されます。出力の制御には SSLLogFile ディレクティブを使用します。

関連項目： [第 9 章「Oracle HTTP Server での SSL の有効化」](#)

送信ログ

サイトへのアクセス・ログが格納されているファイルが指定されます。送信ログを conf ファイルに明示的に含めないと、ログは生成されません。通常、サーバーでは、各リクエストが送信ファイルに記録されます。このファイルは、デフォルトで次の場所にあります。

- UNIX の場合 : `ORACLE_HOME/ohs/logs/access_log`
- Windows の場合 : `ORACLE_HOME\ohs\logs\access_log`

ファイル名は、CustomLog ディレクティブを使用して設定できます。

モジュールの理解

この章では、Oracle HTTP Server に組み込まれているモジュール (mod) について説明します。モジュールは Web サーバーの基本機能を拡張し、Oracle HTTP Server とその他の Oracle Application Server コンポーネントとの統合をサポートします。

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

モジュールのリスト

表 7-1 に、この章で説明する Oracle HTTP Server の全モジュールを示します。

表 7-1 Oracle HTTP Server のモジュール

Oracle HTTP Server のモジュール			
mod_access	mod_actions	mod_alias	mod_asis
mod_auth	mod_auth_anon	mod_auth_dbm	mod_autoindex
mod_cern_meta	mod_certheaders	mod_cgi	mod_cgid
mod_dir	mod_env	mod_expires	mod_fastcgi
mod_file_cache	mod_headers	mod_imap	mod_include
mod_info	mod_log_config	mod_logio	mod_mime
mod_mime_magic	mod_negotiation	mod_oc4j	mod_onsint
mod_ossf	mod_osso	mod_perl	mod_php
mod_proxy	mod_rewrite	mod_setenvif	mod_speling
mod_status	mod_unique_id	mod_userdir	mod_usertrack
mod_vhost_alias	mod_wchandshake		

mod_access

ホスト名や IP アドレスなど、リクエストの特性に基づいてサーバーへのアクセスが制御されます。

関連資料： Apache Server マニュアルの「Module mod_access」

mod_actions

ファイル・タイプやリクエスト方法に基づいて CGI スクリプトを実行できます。

関連資料： Apache Server マニュアルの「Module mod_actions」

mod_alias

リクエストの処理中に URL を操作できます。このモジュールには、URL とファイル・システムのパスとのマッピングおよび URL リダイレクション機能があります。

関連資料： Apache Server マニュアルの「Module mod_alias」

mod_asis

固有の HTTP ヘッダーを含むファイルを送信できます。

関連資料： Apache Server マニュアルの「Module mod_asis」

mod_auth

ファイルベースのユーザー・リストによるユーザー認証ができます。

関連資料： Apache Server マニュアルの「Module mod_auth」

mod_auth_anon

保護付き領域への匿名ユーザー・アクセスができます（電子メール・アドレスをログインできる匿名 FTP と同様です）。

関連資料： Apache Server マニュアルの「Module mod_auth_anon」

mod_auth_dbm

DBM ファイルを使用してユーザー認証を提供します。

mod_autoindex

ディレクトリ索引が自動的に生成されます。

関連資料： Apache Server マニュアルの「Module mod_autoindex」

mod_cern_meta

CERN (Conseil European pour le Recherche Nucleaire) HTTPD メタファイルのセマンティクスがエミュレートされます。メタファイルは、サーバーがアクセスするファイルごとに通常のセットに加えて生成できる HTTP ヘッダーです。

mod_certheaders

Oracle HTTP Server の前で SSL 接続が終了するリバース・プロキシ (OracleAS Web Cache など) が、SSL クライアント証明書情報などの SSL 接続に関する情報を、Oracle HTTP Server および Oracle HTTP Server の背後で動作しているアプリケーションに送信できるようにします。この情報は、HTTP ヘッダーを使用してリバース・プロキシから Oracle HTTP Server に送信されます。情報はヘッダーから標準 CGI 環境変数に送信されます。SSL 接続が Oracle HTTP Server によって終了する場合は mod_oss1 または mod_ssl がこの環境変数を移入します。これは Oracle モジュールです。

また、特定のリクエストが HTTP 経由で受信される場合も、HTTPS リクエストとして扱うことができます。これは、SimulateHttps ディレクティブおよび AddCertHeader ディレクティブを使用して実行されます。

SimulateHttps は、それ自身が含まれるコンテナ (<VirtualHost>、<Location> など) を使用し、受信されたこのコンテナに対するすべてのリクエストを、リクエストの受信に使用された実際のプロトコルに関係なく、HTTPS 経由で受信されたものとして扱います。

AddCertHeader は、特に OracleAS Web Cache で使用するためのものです。OracleAS Web Cache 用に、OracleAS Web Cache が HTTPS 経由で受信したリクエストを Oracle HTTP Server に対して示す、特別なヘッダーを追加します。mod_certheaders は、OracleAS Web Cache が HTTPS としてリクエストを受信したケースのみを、Oracle HTTP Server が HTTPS 経由で受信したものとして扱うよう、Oracle HTTP Server に指示します。

mod_certheaders を構成するには、次の手順を実行します。

1. Oracle HTTP Server を構成して mod_certheaders をロードします。このためには、LoadModule ディレクティブを httpd.conf ファイルに追加します。
 - UNIX の場合: LoadModule certheaders_module libexec/mod_certheaders.so
 - Windows の場合: LoadModule certheaders_module modules¥ApacheModuleCertHeaders.dll

2. どのヘッダーを CGI 環境変数に変換するかを指定します。これは、AddCertHeader ディレクティブを使用して実現できます。このディレクティブは単一の引数を取ります。この引数が、受信リクエスト上の HTTP ヘッダーから移入される必要がある CGI 環境変数です。たとえば、CGI 環境変数 SSL_CLIENT_CERT を移入するには、httpd.conf に次の行を追加します。

```
AddCertHeader SSL_CLIENT_CERT
```

AddCertHeader ディレクティブは、httpd.conf のベース仮想サーバー・セクションに配置するとグローバル設定になります。このディレクティブを仮想ホスト・コンテナ内に配置すると単一仮想ホスト固有になり、httpd.conf 内の **<Directory>** または **<Location>** コンテナ・ディレクティブ内に配置すると URI セット固有になります。このディレクティブの組合せは累積的に追加されます。したがって、特定の URI に関して、URI 固有の全ディレクティブがリクエストの仮想ホスト固有の全ディレクティブに追加され、その結果が、ベース仮想ホストに対して定義されている全ディレクティブに追加されます。

表 7-2 に、サポートされているすべての CGI 環境変数と、それに対応する HTTP ヘッダー名を示します。

表 7-2 CGI 環境変数および対応するヘッダー名

CGI 変数	ヘッダー名	CGI 変数	ヘッダー名
SSL_PROTOCOL	SSL-Protocol	SSL_SESSION_ID	SSL-Session_Id
SSL_CIPHER	SSL-Cipher	SSL_CIPHER_EXPORT	SSL-Cipher-Export
SSL_CIPHER_ALGKEYSIZE	SSL-Cipher-Algkeysize	SSL_VERSION_LIBRARY	SSL-Version-Library
SSL_CLIENT_CERT	SSL-Client-Cert	SSL_VERSION_INTERFACE	SSL-Version-Interface
SSL_CLIENT_CERT_CHAIN_n	SSL-Client-Cert-Chain-n	SSL_CIPHER_USEKEYSIZE	SSL-Cipher-Usekeysize
SSL_CLIENT_VERIFY	SSL-Client-Verify	SSL_SERVER_CERT	SSL-Server-Cert
SSL_CLIENT_M_VERSION	SSL-Client-M-Version	SSL_SERVER_M_VERSION	SSL-Server-M-Version
SSL_CLIENT_M_SERIAL	SSL-Client-M-Serial	SSL_SERVER_M_SERIAL	SSL-Server-M-Serial
SSL_CLIENT_V_START	SSL-Client-V-Start	SSL_SERVER_V_START	SSL-Server-V-Start
SSL_CLIENT_V_END	SSL-Client-V-End	SSL_SERVER_V_END	SSL-Server-V-End
SSL_CLIENT_S_DN	SSL-Client-S-DN	SSL_SERVER_S_DN	SSL-Server-S-DN
SSL_CLIENT_S_DN_C	SSL-Client-S-DN-C	SSL_SERVER_S_DN_C	SSL-Server-S-DN-C
SSL_CLIENT_S_DN_ST	SSL-Client-S-DN-ST	SSL_SERVER_S_DN_ST	SSL-Server-S-DN-ST
SSL_CLIENT_S_DN_L	SSL-Client-S-DN-L	SSL_SERVER_S_DN_L	SSL-Server-S-DN-L
SSL_CLIENT_S_DN_O	SSL-Client-S-DN-O	SSL_SERVER_S_DN_O	SSL-Server-S-DN-O
SSL_CLIENT_S_DN_OU	SSL-Client-S-DN-OU	SSL_SERVER_S_DN_OU	SSL-Server-S-DN-OU
SSL_CLIENT_S_DN_CN	SSL-Client-S-DN-CN	SSL_SERVER_S_DN_CN	SSL-Server-S-DN-CN
SSL_CLIENT_S_DN_T	SSL-Client-S-DN-T	SSL_SERVER_S_DN_T	SSL-Server-S-DN-T
SSL_CLIENT_S_DN_I	SSL-Client-S-DN-I	SSL_SERVER_S_DN_I	SSL-Server-S-DN-I
SSL_CLIENT_S_DN_G	SSL-Client-S-DN-G	SSL_SERVER_S_DN_G	SSL-Server-S-DN-G
SSL_CLIENT_S_DN_S	SSL-Client-S-DN-S	SSL_SERVER_S_DN_S	SSL-Server-S-DN-S
SSL_CLIENT_S_DN_D	SSL-Client-S-DN-D	SSL_SERVER_S_DN_D	SSL-Server-S-DN-D
SSL_CLIENT_S_DN_UID	SSL-Client-S-DN-Uid	SSL_SERVER_S_DN_UID	SSL-Server-S-DN-Uid

表 7-2 CGI 環境変数および対応するヘッダー名 (続き)

CGI 変数	ヘッダー名	CGI 変数	ヘッダー名
SSL_CLIENT_S_DN_Email	SSL-Client-S-DN-Email	SSL_SERVER_S_DN_Email	SSL-Server-S-DN-Email
SSL_CLIENT_I_DN	SSL-Client-I-DN	SSL_SERVER_I_DN	SSL-Server-I-DN
SSL_CLIENT_I_DN_C	SSL-Client-I-DN-C	SSL_SERVER_I_DN_C	SSL-Server-I-DN-C
SSL_CLIENT_I_DN_ST	SSL-Client-I-DN-ST	SSL_SERVER_I_DN_ST	SSL-Server-I-DN-ST
SSL_CLIENT_I_DN_L	SSL-Client-I-DN-L	SSL_SERVER_I_DN_L	SSL-Server-I-DN-L
SSL_CLIENT_I_DN_O	SSL-Client-I-DN-O	SSL_SERVER_I_DN_O	SSL-Server-I-DN-O
SSL_CLIENT_I_DN_OU	SSL-Client-I-DN-OU	SSL_SERVER_I_DN_OU	SSL-Server-I-DN-OU
SSL_CLIENT_I_DN_CN	SSL-Client-I-DN-CN	SSL_SERVER_I_DN_CN	SSL-Server-I-DN-CN
SSL_CLIENT_I_DN_T	SSL-Client-I-DN-T	SSL_SERVER_I_DN_T	SSL-Server-I-DN-T
SSL_CLIENT_I_DN_I	SSL-Client-I-DN-I	SSL_SERVER_I_DN_I	SSL-Server-I-DN-I
SSL_CLIENT_I_DN_G	SSL-Client-I-DN-G	SSL_SERVER_I_DN_G	SSL-Server-I-DN-G
SSL_CLIENT_I_DN_S	SSL-Client-I-DN-S	SSL_SERVER_I_DN_S	SSL-Server-I-DN-S
SSL_CLIENT_I_DN_D	SSL-Client-I-DN-D	SSL_SERVER_I_DN_D	SSL-Server-I-DN-D
SSL_CLIENT_I_DN_UID	SSL-Client-I-DN-Uid	SSL_SERVER_I_DN_UID	SSL-Server-I-DN-Uid
SSL_CLIENT_I_DN_Email	SSL-Client-I-DN-Email	SSL_SERVER_I_DN_Email	SSL-Server-I-DN-Email
SSL_CLIENT_A_SIG	SSL-Client-A-Sig	SSL_SERVER_A_SIG	SSL-Server-A-Sig
SSL_CLIENT_A_KEY	SSL-Client-A-Key	SSL_SERVER_A_KEY	SSL-Server-A-Key

3. mod_certheaders を使用して、あるリクエストが HTTP 経由で受信された場合も、HTTPS 経由で受信されたものとして扱うように Oracle HTTP Server に対して指示できます。これは、Oracle HTTP Server がリバース・プロキシまたはロード・バランサのフロントエンドであるときに役立ちます。リバース・プロキシまたはロード・バランサは、SSL リクエストの終点として機能し、リクエストを HTTPS 経由で Oracle HTTP Server に転送します。

OracleAS Web Cache がロード・バランサとして使用される場合は、HTTPS 経由で受信されたすべてのリクエストを識別する HTTP ヘッダーを送信します。つまり、単にこのヘッダーを調べるのみで、HTTPS リクエストとして扱う必要があるリクエストが mod_certheaders によって自動的に検出されることとなります。このためには、httpd.conf に次のディレクティブを追加します。

```
AddCertHeader HTTPS
```

このディレクティブは、Oracle HTTP Server で処理されるすべての URL に影響します。

他のロード・バランサの場合は、HTTPS リクエストとして扱う必要があるリクエストを判別するために、mod_certheaders を明示的に構成する必要があります。これには次のディレクティブを使用します。

```
SimulateHttps on
```

SimulateHttps は、次のように仮想ホストに埋め込むことができます。

```
<VirtualHost localhost:7777>
    SimulateHttps on
    .
    .
    .
</VirtualHost>
```

ここでは、この仮想ホストで処理されるすべてのリクエストを HTTPS として扱うように `mod_certheaders` に対して指示します。または、次のように、ディレクティブを `<LocationMatch>`、`<Directory>` または `<DirectoryMatch>` ディレクティブのコンテナ内に配置できます。

```
<Location /foo/>
    SimulateHttps on
</Location>
```

これにより、対象は `/foo/` で始まる URL に限定されます。

4. `$ORACLE_HOME/sso/conf/sso_apache.conf` を編集し、次の行をコメント化します。

```
#SSLOptions +ExportCertData +StdEnvVars
```

5. 次のコマンドを実行します。

```
dcmctl updateconfig -ct ohs
```

6. 次のコマンドを実行します。

```
opmnctl restartproc type=ohs
```

7. SSO サーバーにクライアント認証でログインできることを確認します。

mod_cgi

サーバーで CGI スクリプトを実行できます。

関連資料： Apache Server マニュアルの「Module mod_cgi」

mod_cgid

最適化が施されていることと、追加の `ScriptSock` ディレクティブを除けば、このモジュールは `mod_cgi` と同様に機能します。`ScriptSock` ディレクティブは、CGI デーモンとの通信に使用されるソケットの名前を設定します。ソケットは、Apache を起動するユーザー（通常は `root`）の権限を使用してオープンされます。

関連資料： Apache Server マニュアルの「Module mod_cgid」

mod_dir

サーバーでスラッシュ (`/`) のリダイレクトを実行できます。ディレクトリ指定には後続のスラッシュを含める必要があります。後続のスラッシュがない URL リクエストを受信すると、`mod_dir` は後続のスラッシュが付いている同一の URL にリダイレクトします。次に例を示します。

```
http://myserver/documents/mydirectory
```

この URL は、次の URL にリダイレクトされます。

```
http://myserver/documents/mydirectory/
```

関連資料： Apache Server マニュアルの「Module mod_dir」

mod_env

環境変数を渡し、設定および設定解除することで、CGI スクリプトとサーバー・サイド・インクルード (SSI: Server Side Includes) ページの環境を制御できます。

`ModifyEnv` は、値を既存の `ENV` 変数の値の前または後ろに付加し、Oracle HTTP Server 環境に渡します。次に使用方法を示します。

\$FOO = "foo" の場合 :

ModifyEnv FOO "bar" modifies the value of \$FOO from "foo" to "foo:bar"

ModifyEnv FOO "+bar" modifies the value of \$FOO from "foo" to "bar:foo"

\$FOO が定義されていない場合 :

Modify Foo "bar" sets the value of \$FOO to "bar"

関連資料: Apache Server マニュアルの「Module mod_env」

mod_expires

サーバーで Expires HTTP ヘッダーを生成できます。このヘッダーは、ドキュメントの妥当性に関する情報をクライアントに提供します。期限切れ条件に基づいて、キャッシュされたコピーが期限切れになると、ドキュメントが情報源より再取得されます。

関連資料: Apache Server マニュアルの「Module mod_expires」

mod_fastcgi

FastCGI プロトコルをサポートします。このプロトコルにより、CGI アプリケーション用に実行中のサーバーのプールをメンテナンスできます。その結果、起動と初期化のオーバーヘッドがなくなります。

関連資料: Apache Server マニュアルの「Module mod_fastcgi」

mod_file_cache

頻繁にリクエストされる静的ファイルのキャッシュ方法を提供します。破損したサイトが作成されやすいため、このモジュールは慎重に使用してください。

関連資料: Apache Server マニュアルの「Module mod_file_cache」

mod_headers

HTTP レスポンス・ヘッダーをマージ、置換または削除できます。

関連資料: Apache Server マニュアルの「Module mod_headers」

mod_imap

サーバー側のイメージ・マップ処理ができます。

mod_include

SSI (サーバー・サイド・インクルード) ディレクティブ用のドキュメントを処理するフィルタを提供します。

関連資料: Apache Server マニュアルの「Module mod_include」

mod_info

すべてのインストール済モジュールとディレクティブの設定など、サーバー構成全体のサマリーが生成されます。

関連資料： Apache Server マニュアルの「Module mod_info」

mod_log_config

サーバー・アクティビティの、構成およびカスタマイズ可能なロギング機能が提供されます。ログの書式を選択し、ロギング対象となる個々のリクエストをその特性に基づいて選択または除外できます。

関連資料： Apache Server マニュアルの「Module mod_log_config」

mod_logio

各リクエストに対して送受信された入出力バイト数のロギング機能を提供します。記録される数値はネットワーク上で実際に受信されたバイト数を反映したもので、リクエストとレスポンスのヘッダーとボディから算出されます。

関連資料： Apache Server マニュアルの「Module mod_logio」

mod_mime

サーバーでファイル名からファイル・タイプを判断し、処理用のハンドラに関連付けできます。

関連資料： Apache Server マニュアルの「Module mod_mime」

mod_mime_magic

サーバーでは、ファイルの内容のうち数バイトを検査することでファイルの MIME タイプを判断できます。[mod_mime](#) でファイル・タイプを判断できない場合にこのモジュールを使用します。最初に [mod_mime](#) によってファイルが処理されるように、[mod_mime](#) が構成ファイル内で [mod_mime_magic](#) より前にあることを確認してください。

関連資料： Apache Server マニュアルの「Module mod_mime_magic」

mod_negotiation

サーバーによるコンテンツのネゴシエーション（クライアントの機能に基づくドキュメントの選択）が有効化されます。

関連資料： Apache Server マニュアルの「Module mod_negotiation」

mod_oc4j

AJP 1.3 プロトコルを介して、Oracle HTTP Server から Oracle Application Server Containers for J2EE (OC4J) にリクエストがルーティングされます。これは Oracle モジュールです。

[mod_oc4j](#) は、デフォルトで有効化されています。インストール時に、[oc4j_deploy_tool.jar](#) によって、OC4J インスタンスに配置されたアプリケーション用の [mod_oc4j.conf](#) にマウント・ポイントが追加されます。[mod_oc4j](#) の特定のマウント・ポイントに対するリクエストは、そのマウント・ポイント用の OC4J インスタンスにルーティングされます。

OC4J インスタンスは、Oracle Process Manager and Notification Server (OPMN) により起動および管理されます。

関連資料：

- 『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』
- 『Oracle Process Manager and Notification Server 管理者ガイド』

この項の内容は、次のとおりです。

- [mod_oc4j の構成](#)
- [mod_oc4j を使用したロード・バランシング](#)
- [mod_oc4j と OC4J 間での SSL の有効化](#)

mod_oc4j の構成

この項では、[httpd.conf](#) および [mod_oc4j.conf](#) 内のすべての関連ディレクティブについて説明します。また、サンプル構成も示します。

mod_oc4j の構成ファイルおよびディレクティブ

mod_oc4j のディレクティブは、mod_oc4j.conf 内に保持されます。mod_oc4j.conf ファイルは、次のディレクティブを使用して、デフォルトで httpd.conf ファイルにインクルードされます。

```
include "ORACLE_HOME/ohs/conf/mod_oc4j.conf"
```

mod_oc4j の構成には、次のディレクティブを使用します。

- [Oc4jCacheSize](#)
- [Oc4jConnTimeout](#)
- [Oc4jCookieExtension](#)
- [Oc4jExtractSSL](#)
- [Oc4jEnvVar](#)
- [Oc4jMount](#)
- [Oc4jMountCopy](#)
- [Oc4jUseOHSErrors](#)

関連項目： [9-4 ページの「SSL 構成ディレクティブの使用」](#)

LoadModule

mod_oc4j モジュールをロードします。

カテゴリ	値
構文	<code>LoadModule oc4j_module mod_oc4j shared library file</code>
必須	あり
デフォルト	<ul style="list-style-type: none"> ■ UNIX の場合：なし ■ Windows の場合：<code>LoadModule oc4j_module modules¥ApacheModuleOc4j.dll</code>
例	<ul style="list-style-type: none"> ■ UNIX の場合：<code>LoadModule oc4j_module mod_oc4j.so</code> ■ Windows の場合：<code>LoadModule oc4j_module modules¥ApacheModuleOc4j.dll</code>

Oc4jCacheSize

OC4J 接続キャッシュのサイズを指定します。

カテゴリ	値
構文	Oc4jCacheSize <size of connection cache>
必須	なし
デフォルト	<ul style="list-style-type: none"> ■ UNIX の場合 : 1 ■ Windows の場合 : 32
例	Oc4jCacheSize 64
使用方法	各 Oracle HTTP Server プロセスでキャッシュできる OC4J 同時接続の数を指定します。このディレクティブを 0 (ゼロ) に設定すると、mod_oc4j と OC4J インスタンス間の永続的な接続が無効になります。

Oc4jConnTimeout

使用されていない接続の最大アイドル時間 (秒単位) を定義します。

カテゴリ	値
構文	Oc4jConnTimeout <timeout value for AJP13 connections>
必須	なし
デフォルト	なし
例	Oc4jConnTimeout 10
使用方法	mod_oc4j と OC4J 間に接続をタイムアウトするファイアウォールがある場合に役立ちます。ファイアウォールで使用されるタイムアウト値より小さい値に設定する必要があります。

Oc4jCookieExtension

JSESSIONID_<cookie_name_extension> を Cookie 内の OC4J のセッション ID として使用するよう mod_oc4j に対して指示します。

カテゴリ	値
構文	Oc4jCookieExtension <cookie_name_extension>
必須	なし
デフォルト	なし
例	Oc4jCookieExtension MYEXT
使用方法	Cookie 内の OC4J のセッション ID として、JSESSIONID のかわりに JSESSIONID_<cookie_name_extension> を使用するよう mod_oc4j に対して指示します。前述の例では、JSESSIONID_MYEXT が OC4J のセッション ID として使用されます。

Oc4jExtractSSL

SSL 環境変数の受渡しを制御します。

カテゴリ	値
構文	Oc4jExtractSSL On/Off
必須	なし
デフォルト	Off
例	Oc4jExtractSSL On
使用方法	mod_oc4j に対して、3つの SSL 環境変数 SSL_CLIENT_CERT、SSL_CIPHER および SSL_SESSION_ID を OC4J に渡すかどうかを指示します。SSL 環境変数を OC4J にコピーする操作にはパフォーマンス・コストが関連するため、環境変数を OC4J で使用可能にする必要がある場合にのみ「On」に設定してください。

注意： mod_oc4j が構成されている場合は、mod_oss1 と mod_osso によって設定された一部のセキュリティ環境パラメータが、リクエスト時に OC4J に渡されます。

Oc4jEnvVar

mod_oc4j に対して、一部の環境変数を Oracle HTTP Server から OC4J に渡すように指示します。

カテゴリ	値
構文	Oc4jEnvVar environment variable name [environment variable default value]
必須	なし
デフォルト	なし
例	Oc4jEnvVar MY_ENV1 Oc4jEnvVar MY_ENV2 myenv_value
使用方法	OC4jEnvVar エントリごとに、Oracle HTTP Server ディレクティブ PassEnv も環境変数を使用して構成する必要があります。構成しない場合、mod_oc4j は値の取得と受渡しができません。 複数のエントリを指定できます。環境変数のデフォルト値を2番目のパラメータとして指定する方法と、何も指定しない方法があります。環境変数の値が Oracle HTTP Server 環境で見つかると、その値が OC4J に渡されます。環境変数の値が見つからない場合でも、デフォルト値が設定されていれば、その値が渡されます。 この環境変数の値が Oracle HTTP Server 環境で見つからず、デフォルト値が設定されていない場合、OC4J には何も渡されません。 mod_oc4j が一部の構成済環境変数を各リクエストで OC4J に渡すと、パフォーマンスが低下します。

注意： mod_oc4j が構成されている場合は、mod_oss1 と mod_osso によって設定された一部のセキュリティ環境パラメータが、リクエスト時に OC4J に渡されます。

Oc4jMount

mod_oc4j に対して、特定のパスを含むリクエストを宛先にルーティングするように指示します。宛先には、単一の OC4J プロセスまたは OC4J インスタンスのセットを指定できます。

カテゴリ	値
構文	<p>Oc4jMount <i>path</i> [<i>destination</i>]</p> <p><i>path</i> はコンテキストのルートです。 <i>path</i> パラメータには、OC4J 構成ファイル <code>xxx-web-site.xml</code> で指定されているものと同じアプリケーション・コンテキストのルートを指定する必要があります。 <web-site> 要素の例では、アプリケーション・コンテキストのルートは太字で示されています。</p> <pre><default-web-app application="default" name="defaultWebApp" root="/j2ee"/></pre> <p>宛先のタイプは、次のいずれかです。</p> <ul style="list-style-type: none"> ■ <code>ajp13_dest</code> ■ <code>cluster_dest</code> (デフォルトの宛先タイプ) ■ <code>instance_dest</code> <p>宛先を指定しない場合は、デフォルトの OC4J インスタンス名 <code>home</code> が使用されます。次に例を示します。</p> <pre>Oc4jMount /myApp/*</pre> <p>このディレクティブの結果は、次のディレクティブと同じになります。</p> <pre>Oc4jMount /myApp/* cluster://local_ias_cluster_name:home</pre>
必須	なし
デフォルト	なし
例	<pre>Oc4jMount /app01/* ajp13://my-sun:8888 Oc4jMount /app02/* Oc4jMount /app03/* home Oc4jMount /app04/* ias_cluster_1:home Oc4jMount /app05/* cluster://ias_cluster_1:home,ias_ cluster_2:home Oc4jMount /app06/* instance://ias_instance_1:home Oc4jMount /app07/* instance://ias_instance_1:home_1,ias_ instance_2:home_2 Oc4jMount /app08/* instance://my-sun:ias_instance_1:home</pre>

カテゴリ	値
使用方法	<p>ルーティングの宛先ごとに例を示します。</p> <p>ajp13_dest</p> <pre>Oc4jMount path ajp13://my-sun:8888</pre> <p>path で指定されたパターンを持つリクエストが、AJP 1.3 プロトコルを使用して <i>my-sun</i>、ポート 8888 でリスニング中の OC4J プロセスにルーティングされます (<i>my-sun</i> およびポート 8888 は、OC4J 構成ファイル xxx-web-site.xml で指定されている AJP 1.3 プロトコルのホストおよびポートです)。</p> <p>cluster_dest</p> <pre>Oc4jMount <path> cluster: //ias_cluster_name:OC4J_instance_name, ias_cluster_name:OC4J_instance_name...</pre> <p>path に指定したパターンを持つリクエストが、指定した 1 つ以上の OC4J インスタンスにロード・バランスされます (インスタンスはカンマで区切ってください)。</p> <p>Oracle Application Server Cluster Name はオプションです。指定する場合、宛先の OC4J インスタンスは指定したクラスタ内にある必要があります。Oracle Application Server Cluster Name を指定しない場合、宛先の OC4J インスタンスはローカルの Oracle Application Server クラスタ内にある必要があります。</p> <p>instance_dest</p> <pre>Oc4jMount <path> instance: //host:ias_instance_name:OC4J_instance_name, host:ias_instance_name:OC4J_instance_name...</pre> <p><path> に指定したパターンを持つリクエストが、指定した 1 つ以上の OC4J インスタンスにロード・バランスされます (インスタンスはカンマで区切ってください)。</p> <p>ホスト名はオプションです。ホスト名を指定する場合、宛先の OC4J インスタンスはそのホストに存在する Oracle Application Server インスタンス内にある必要があります。ホスト名を指定しない場合、宛先の OC4J インスタンスはどのホストにあってもかまいません。</p>

Oc4jMountCopy

ベース・サーバーからマウント・ポイントをコピーします。

カテゴリ	値
構文	Oc4jMountCopy On/Off
必須	なし
デフォルト	On
例	Oc4jMountCopy Off
使用方法	<p>mod_oc4j に対して、このディレクティブを指定した仮想ホストにベース・サーバーから Oc4jMount ポイントをコピーするかどうかを指示します。値が On の場合は、ベース・サーバー内に構成されたすべての Oc4jMount ポイントが仮想ホストにコピーされます。値が Off の場合は、仮想ホストの有効範囲内で構成された Oc4jMount ポイントのみが使用されます。</p>

Oc4jUseOHSErrors

OC4J から範囲内のエラーが返されたときに、ユーザーが Oracle HTTP Server のエラー・ページを使用して、エラー範囲を構成することを許可します。

カテゴリ	値
構文	Oc4jUseOHSErrors On/Off/min-max
必須	なし
デフォルト	Off
例	Oc4jUseOHSErrors 400-410
使用方法	<p>Oc4jUseOHSErrors Off: Oc4jUseOHSErrors を指定しない場合のデフォルト値です。すべてのエラー値について、OC4J のエラー・ページがクライアントに戻されます。</p> <p>Oc4jUseOHSErrors on: HTTP エラー 400 ~ 500 の Oracle HTTP Server のエラー・ページを返します。</p> <p>Oc4jUseOHSErrors min-max: HTTP エラーの最小と最大を指定します。たとえば、Oc4jUseOHSErrors 400-410 と設定すると、HTTP エラー 400 ~ 410 の Oracle HTTP Server のエラー・ページが OC4J から返されます。</p>

mod_oc4j のサンプル構成

この項では、mod_oc4j のサンプル構成について説明します。

例 7-1 mod_oc4j のサンプル構成

この構成では、URI /servlet/ で始まるすべてのリクエストが、OC4J プロセスのデフォルト・インスタンスにマウントされます。

httpd.conf ファイルに次のエントリを作成します。

```
Oc4jMount /servlet/*
```

例 7-2 mod_oc4j のサンプル構成

この構成では、Oc4jMount ディレクティブのかわりに <Location> コンテナ・ディレクティブを使用して、例 7-1 の構成と同じ動作を実行します。

httpd.conf ファイルに次のエントリを作成します。

```
<Location /servlet>
    SetHandler oc4j-handler
</Location>
```

注意： この構成では、リクエストはデフォルトの OC4J インスタンスにのみルーティングされます。

例 7-3 mod_oc4j のサンプル構成

この構成では、URI /servlet/ または /j2ee/ で始まるすべてのリクエスト、およびすべての JSP ページが、OC4J プロセスのデフォルトの OC4J インスタンスにマウントされます。

mod_oc4j.conf ファイルに次のエントリを作成します。

```
Oc4JMount /servlet/*
Oc4JMount /*.jsp
Oc4JMount /j2ee/*
```

例 7-4 mod_oc4j のサンプル構成

この構成では、次のようにマウントが行われます。

- URI /applicationA/ で始まるすべてのリクエストおよびすべての JSP ページが、oc4j_instance_A にマウントされます。このインスタンスでは、すべての OC4J プロセスが OPMN によって管理されます。
- URI /applicationB/ で始まるすべてのリクエストが、oc4j_instance_B にマウントされます。このインスタンスでは、すべての OC4J プロセスが OPMN によって管理されます。

mod_oc4j.conf ファイルに次のエントリを作成します。

```
Oc4JMount /applicationA/* oc4j_instance_A
Oc4JMount /applicationB/* oc4j_instance_B
Oc4JMount /j2ee/*
Oc4JMount /*.jsp oc4j_instance_A
```

mod_oc4j を使用したロード・バランシング

メトリック・ベースのロード・バランシングも含め、mod_oc4j によるロード・バランシングについては付録 A 「[mod_oc4j を使用したロード・バランシング](#)」で詳しく説明します。

mod_oc4J と OC4J 間での SSL の有効化

オプションで、mod_oc4j と OC4J 間の通信に直接 SSL サポートを指定できます。このためには、mod_oc4j 側と OC4J 側で SSL を有効化する必要があります。

- [mod_oc4j での SSL の有効化](#)
- [OC4J での SSL の有効化](#)

mod_oc4j での SSL の有効化

mod_oc4j に対して SSL を有効にするには、次のディレクティブを mod_oc4j.conf に追加します。

Oc4jEnableSSL

mod_oc4j が OC4J プロセスとの通信時に SSL を使用する必要があるかどうかを示します。

Oc4jASPTActive が「On」に構成されている場合、このディレクティブは「On」に構成しないでください。

カテゴリ	値
パラメータ名	Oc4jEnableSSL
パラメータ・タイプ	文字列
有効値	On または Off
デフォルト値	Off

Oc4jSSLWalletFile

Oc4jEnableSSL が「On」に設定されている場合、このディレクティブは OC4J プロセスとの SSL 通信に使用される SSL 証明書を含む、Oracle Wallet ファイルの位置を指定します。

カテゴリ	値
パラメータ名	Oc4jSSLWalletFile
パラメータ・タイプ	文字列
有効値	OC4J プロセスとの SSL 接続確立時に使用される SSL 証明書を含む Wallet ディレクトリの場所へのパス
デフォルト値	該当なし

Oc4jSSLWalletPassword

Oc4jEnableSSL が「On」に設定されている場合、この値は Wallet ファイルのオープン時に認証に使用される、不明瞭化されたパスワードです。この値は、Oracle Wallet Manager に含まれているユーティリティを使用して取得されます。

カテゴリ	値
パラメータ名	Oc4jSSLWalletPassword
パラメータ・タイプ	文字列
有効値	Oc4jSSLWalletFile により指定された Wallet ファイルのオープン時に認証に使用される不明瞭化されたパスワード
デフォルト値	該当なし

関連資料：

- Oracle Wallet Manager の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。
- 9-16 ページの「iasobf ユーティリティの使用」

注意： Wallet パスワードは使用されていません。このディレクティブを使用すると、Oracle HTTP Server ログに警告メッセージが生成されます。Wallet を保護するために、かわりに SSO Wallet を取得することをお勧めします。

OC4J での SSL の有効化

mod_oc4j と OC4J の間で SSL 通信を有効化するには、OC4J 側でも SSL を有効化する必要があります。

関連資料： OC4J 側で SSL を有効化する方法の詳細は、『Oracle Application Server Containers for J2EE セキュリティ・ガイド』を参照してください。

mod_onsint

Oracle Notification Service (ONS) および Oracle Process Manager and Notification Server (OPMN) を使用した統合サポートを提供します。これは Oracle モジュールです。

mod_onsint を使用するメリット

mod_onsint は次の機能を提供します。

- Oracle HTTP Server 内での ONS 通知にサブスクリプション・メカニズムを提供します。これは、Oracle HTTP Server がマルチプロセス・アーキテクチャを採用している UNIX で特に重要です。このようなアーキテクチャでは、1 つの Oracle HTTP Server インスタンスが最大 8192 のプロセスで構成されるため、各プロセス内に ONS サブスクリバを指定することは不可能です。mod_onsint は、Oracle HTTP Server インスタンス内のすべてのモジュールに対する通知を受信するプロセスを 1 つ提供します。
- リスナーが起動され使用可能であることが OPMN や OC4J などの他のコンポーネントに通知されるように、PROC_READY ONS 通知を発行します。また、DMS メトリックなどの情報やリスナーへの接続方法に関する情報も提供します。これらの通知は、Oracle HTTP Server インスタンスが実行されているかぎり、mod_onsint により定期的送信されます。
- 親プロセスが失敗したときに Oracle HTTP Server が単一の単位として終了できるようにする機能を提供します。親プロセスは、Oracle HTTP Server インスタンスのすべての子プロセスの起動と停止に責任があります。最初に子プロセスをシャットダウンせずに親プロセスが失敗すると、Oracle HTTP Server が一貫性のない状態になり、この状態は孤立した子プロセスをすべて手動でシャットダウンしないかぎり修正できません。これを行うまで、新しい Oracle HTTP Server インスタンスは起動できません。Oracle HTTP Server が使用するポートを、孤立した子プロセスが占有しているためです。mod_onsint は、親プロセスをモニターします。親プロセスの異常終了を検出すると、残っている子プロセスをすべて中断します。この機能と OPMN が組み合わせられると、親プロセスが失敗したときでも Oracle HTTP Server を再起動できます。mod_onsint は、Oracle HTTP Server の子プロセスがすべて中断され、新しい Oracle HTTP Server インスタンス用にポートがオープンされた状態になるようにします。OPMN は、元のインスタンスの障害が検出された後、新規インスタンスが起動されるように保証します。

UNIX と Windows での実装上の差異

UNIX と Windows では Oracle HTTP Server のアーキテクチャに違いがあるため、これらのプラットフォームでは mod_onsint の実装に多少の違いがあります。

UNIX では、mod_onsint はモジュールの初期設定時にプロセスを作成します。このプロセスでは、親プロセスの監視と ONS メッセージの送受信を行います。ONS 通知に関心がある他のモジュールからのコールバック・ファンクションは、このプロセス内に作成されます。この情報を他の Oracle HTTP Server の子プロセスと共有するには、メモリー・マップ・ファイルなどのプロセス間通信を使用する必要があります。UNIX 上で親プロセスの障害が検出されると、すべての子プロセスにシグナルが送信され、すべての子プロセスがシャットダウンします。

Windows では、Oracle HTTP Server は親プロセス、および全 HTTP リクエストを処理するマルチスレッドの子プロセスという 2 つのプロセスのみで構成されます。このモデルでは、mod_onsint は子プロセス内のスレッドとして実行されます。このスレッドが、親プロセスの監視と ONS メッセージの送受信を行います。ONS 通知に関心がある他のモジュールからのコールバック・ファンクションは、この子プロセス内に作成されます。親プロセスの障害が検出された場合、mod_onsint は子プロセスを終了し、Oracle HTTP Server を事実上シャットダウンします。

mod_onsint に対して構成できる OpmnHostPort というオプションのディレクティブがあります。このディレクティブを使用すると、mod_onsint が動作中の Oracle HTTP Server インスタンスを ping するため、OPMN が使用するホスト名とポートを指定できます。OpmnHostPort が指定されていないと、mod_onsint は HTTP ポートを自動的に選択します。状況によっては、OPMN がリスナーの ping に使用する HTTP ポートとホスト名に特定のものを選択する場合があります。

OpmnHostPort が取る引数は 1 つで、OPMN に渡す値を指定する host:port 文字列です。たとえば、次の行は、OPMN がこのリスナーを ping するために、localhost インタフェースとポート 7778 を使用する必要があることを指定します。

```
OpmnHostPort localhost: 7778
```

このディレクティブは、[httpd.conf](#) ファイルのグローバル・セクションに指定する必要があります。ロケーション・コンテナの仮想ホストに埋め込むことはできません。インストール後、OpmnHostPort ディレクティブは [dms.conf](#) に置かれます。このディレクティブは、特殊なローカルホスト専用仮想ホストである、Oracle HTTP Server の診断ポートに対する OPMN を指します。

mod_oss1

Oracle HTTP Server に対して強度の高い暗号化を有効にします。この Oracle モジュールは、サーバーが SSL を使用できるようにする Oracle HTTP Server へのプラグインです。これは、OpenSSL モジュールの `mod_ssl` と非常によく似ています。ただし、OpenSSL モジュールとは対照的に、`mod_oss1` は SSL をサポートする Oracle の SSL 実装のバージョン 3 を基盤とし、かつ Certicom および RSA セキュリティ・テクノロジーに基づいています。

関連資料:

- 『Oracle Application Server セキュリティ・ガイド』
- [8-7 ページの「ユーザー認証のための mod_oss1 の使用」](#)
- [第 9 章「Oracle HTTP Server での SSL の有効化」](#)

mod_osso

Oracle HTTP Server で[シングル・サインオン](#)が有効になります。`mod_osso` では、受信リクエストを検査して、リクエストされたリソースが保護されているかどうかを判断し、保護されている場合はユーザー用の Oracle HTTP Server Cookie を取得します。これは Oracle モジュールです。

関連資料: 『Oracle Application Server Single Sign-On 管理者ガイド』

mod_perl

Oracle HTTP Server に Perl インタプリタが埋め込まれます。これにより、起動時のオーバーヘッドが排除され、モジュールを Perl で記述できます。Oracle Application Server では、Perl バージョン 5.8.3 を使用します。

関連資料: 『Apache mod_perl Guide』

データベース使用上の注意

この項では、データベースを使用する `mod_perl` ユーザー向けに、ローカル・データベース接続をテストし、文字構成を設定する方法について説明します。

Perl を使用したデータベース・アクセス

次の項では、Perl を使用したデータベース・アクセスについて説明します。Perl スクリプトは、Oracle 用の DBI/DBD ドライバを使用してデータベースにアクセスします。DBI/DBD ドライバは Oracle Application Server に含まれています。このドライバは、Oracle Call Interface (OCI) をコールしてデータベースにアクセスします。

DBI が機能するには、[httpd.conf](#) で DBI が有効である必要があります。これには次の手順を実行します。

1. テキスト・エディタを使用して `httpd.conf` を編集します。
2. 「PerlModule Apache::DBI」を検索します。
3. 「PerlModule Apache::DBI」という行のコメントを解除します。

4. 次のコマンドを使用して、Oracle HTTP Server を再起動します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

ファイルが `ORACLE_HOME/ohs/cgi-bin` にコピーされます。

例 7-5 Perl を使用したデータベース・アクセス

```
#!<ORACLE_HOME>/perl/bin/perl -w
use DBI;
my $dataSource = "host=<hostname.domain>;sid=<orclsid>;port=1521";
my $userName = "scott";
my $password = "tiger";
my $dbhhandle = DBI->connect("dbi:Oracle:$dataSource", $userName, $password)
    or die "Can't connect to the Oracle Database: $DBI::errstr\n";
print "Content-type: text/plain\n\n";
print "Database connection successful.\n";
### Now disconnect from the database
$dbhhandle->disconnect
    or warn "Database disconnect failed; $DBI::errstr\n";
exit;
```

DBI スクリプトには次の場所からアクセスできます。

```
http://<hostname.domain>:<port>/cgi-bin/<scriptname>
http://<hostname.domain>:<port>/perl/<scriptname>
```

スクリプトに `use DBI` ではなく `use Apache::DBI` と指定されている場合、このスクリプトを実行できるのは、`http://<hostname.domain>:<port>/perl/<scriptname>` からのみです。

データベース接続のテスト

ローカル・シード・データベースのデータベース接続をテストする Perl スクリプトの例を次に示します。このスクリプトを使用して別のデータベース接続をテストするには、`scott/tiger` をターゲット・データベースのユーザー名とパスワードに置き換える必要があります。

例 7-6 ローカル・シード・データベースの接続テスト用サンプル Perl スクリプト

```
##### Perl script start #####
use DBI;
print "Content-type: text/plain\n\n";
$dbh = DBI->connect("dbi:Oracle:", "scott/tiger", "") || die $DBI::errstr;
$stmt = $dbh->prepare("select * from emp order by empno") || die $DBI::errstr;
$rc = $stmt->execute() || die $DBI::errstr;
while (($empno, $name) = $stmt->fetchrow()) { print "$empno $name\n"; }
warn $DBI::errstr if $DBI::err;
die "fetch error: " . $DBI::errstr if $DBI::err;
$stmt->finish() || die "can't close cursor";
$dbh->disconnect() || die "cant't log off Oracle";
##### Perl script End #####
```

SQL NCHAR データ型の使用

SQL NCHAR データ型は、Oracle9i 以降さらに改良され、信頼性の高い Unicode データ型と呼ばれています。NCHAR、NVARCHAR2 および NCLOB などの SQL NCHAR データ型を使用すると、あらゆる Unicode 文字をデータベースのキャラクタ・セットに関係なく格納できます。これらのデータ型のキャラクタ・セットは、各国語キャラクタ・セット、つまり AL16UTF-16 または UTF8 で指定します。

関連資料： SQL NCHAR データ型の詳細は、Oracle9i のマニュアルを参照してください。

このリリースの DBD::Oracle は SQL NCHAR データ型をサポートしており、データ・バインド用の文字構成を指定できるようにドライバ拡張機能が用意されています。次のスクリプトに、SQL NCHAR データへのアクセス例を示します。

例 7-7 SQLNCHAR データにアクセスするためのサンプル・スクリプト

```
# declare to use the constants for character forms
use DBD::Oracle qw(:ora_forms);
# connect to the database and get the database handle
$dbh = DBI->connect( ... );
# prepare the statement and get the statement handle
$stmt = $dbh->prepare( 'SELECT * FROM TABLE_N WHERE NCOL1 = :nchar1' );
# bind the parameter of a NCHAR type
$stmt->bind_param( ':nchar1', $param_1 );
# set the character form to NCHAR
$stmt->func( { ':nchar1' => ORA_NCHAR }, 'set_form' );
$stmt->execute;
```

例 7-7 に示すように、`set_form` ファンクションはプライベート・ファンクションとして提供されており、標準の DBI `func()` メソッドで起動できます。このファンクションは、どのプレースホルダをどの文字構成に関連付けるかを指定する匿名ハッシュを取ります。文字構成の有効値は、ORA_IMPLICIT または ORA_NCHAR のいずれかです。文字構成を ORA_IMPLICIT に設定すると、アプリケーションのバインド・データはデータベースのキャラクタ・セットに変換され、ORA_NCHAR に設定すると各国語キャラクタ・セットに変換されます。デフォルト構成は ORA_IMPLICIT です。

デフォルトのキャラクタ・セット構成を指定できるように、次のようにもう 1 つのファンクションも用意されています。

```
# specify the default form to be NCHAR
$dbh->func( ORA_NCHAR, 'set_default_form' );
```

このコールの後は、`set_form` のコールで特に指定しないかぎり、すべてのパラメータの構成が ORA_NCHAR になります。`set_form` ファンクションとは異なり、これはデータベース・ハンドルのファンクションであるため、指定したデフォルト構成のデータベース・ハンドルの各文は、デフォルトで選択した構成であることに注意してください。

set_form このファンクションでは、パラメータの文字構成を設定します。有効な構成は、ORA_IMPLICIT (デフォルト) または ORA_NCHAR です。この定数は、DBD::Oracle では `ora_forms` として使用できます。

例 7-8 set_form のサンプル

```
# a declaration example for the constants ORA_IMPLICIT and ORA_NCHAR
use DBD::Oracle qw(:ora_forms);
# set the character form for the placeholder :nchar1 to NCHAR
$stmt->func( { ':nchar1' => ORA_NCHAR }, 'set_form' );
# set the character form using the positional index
$stmt->func( { 2 => ORA_NCHAR }, 'set_form' );
# set the character form for multiple placeholders at once
$stmt->func( { 1 => ORA_NCHAR, 2 => ORA_NCHAR }, 'set_form' );
```

set_default_form このファンクションでは、データベース・ハンドルのデフォルトの文字構成を設定します。

例 7-9 データベース・ハンドルのデフォルトの文字構成

```
$dbh->func( ORA_NCHAR , 'set_default_form' );
```

mod_php

PHP (PHP: Hypertext Preprocessor の略) は、オープン・ソースで広く用いられている汎用クライアント側スクリプト言語で、標準 HTML に埋め込まれます。この言語は、動的 HTML ページの生成に使用されます。Oracle HTTP Server では、mod_php を介して PHP サポートが提供されます。また、Oracle Database サポートも有効になっています。使用される PHP はバージョン 4.3.9 です。

注意: `phpinfo()` は、PHP および Oracle HTTP Server の間隔の現在の状態について、機密性が非常に高い情報を出力します。PHP や `phpinfo()` に不慣れたユーザーは、`phpinfo()` という PHP スクリプトをパブリックでアクセスできる状態のままにしておかないように注意してください。

`phpinfo()` は、デバッグによく使用されます。デバッグの終了後には、このようなデバッグ・スクリプトが誤ってオープンの状態のままになっている可能性があります。

関連資料:

- <http://php.net/>
- 最初から構築したり、詳細情報が必要な場合は、<http://www.oracle.com/technology/tech/opensource/index.html> でドキュメント「Using PHP with Oracle HTTP Server (OHS)」を参照してください。

mod_proxy

FTP、CONNECT (SSL 用)、HTTP/0.9、HTTP/1.0 および HTTP/1.1 用のプロキシ機能を提供します。

関連資料:

- Apache Server マニュアルの「Module mod_proxy」
- [「mod_proxy のディレクティブの使用」 9-16 ページ](#)

mod_rewrite

Oracle HTTP Server では、URL 操作ツールとして mod_rewrite が提供されます。mod_rewrite では、リクエストされた URL をリライトするために正規表現パーサーに基づくリライト・エンジンが使用されます。URL 操作の粒度は、サーバー変数、環境変数、HTTP ヘッダーおよびタイムスタンプの書式の影響を受ける場合があります。

このモジュールは、サーバー単位のコンテキスト ([httpd.conf](#)) およびディレクトリ単位のコンテキスト ([.htaccess](#)) の両方で URL 全体 (path-info 部を含む) に対して動作し、結果の query-string 部を生成できます。

この項の内容は、次のとおりです。

- [mod_rewrite のルール処理](#)
- [mod_rewrite のディレクティブ](#)
- [リライト・ルールのヒント](#)
- [リダイレクションの例](#)

mod_rewrite のルール処理

Apache では HTTP がフェーズ単位で処理されます。これらの各フェーズ用のフックは、Apache API により提供されます。mod_rewrite では、このうちの 2 つのフックを使用します。一方は URL-to-filename 変換フックで、HTTP リクエストが読み取られてから認可が開始される間に使用されます。他方は Fixup フックで、認可フェーズの後、およびディレクトリ単位の構成ファイル (.htaccess) が読み取られてからコンテンツ・ハンドラが有効化される間にトリガーされます。

mod_rewrite は、構成構造から構成済ルールセットを読み取ります。サーバー・レベルのルールセットは起動時に最適であるように構成されますが、ディレクトリ・レベルのルールセットはカーネルによるディレクトリ・アクセス時に構成されます。

mod_rewrite はルールセット内でルールを 1 つずつループし (RewriteRule ディレクティブ)、特定のルールが一致すると、対応する条件をループします (RewriteCond ディレクティブ)。最初に、URL が各ルールの Pattern に対して照合されます。照合できなかった場合、mod_rewrite は対応しているルール条件を検索します。ルール条件が存在しない場合は、URL を文字列 Substitution からなる新規の値に単に置換して、ルールのループを継続します。ただし、条件が存在する場合は、内側のループを開始して各条件をリストされている順に処理します。

条件が存在する場合、変数を拡張して文字列 TestString を作成し、マップ参照を逆参照し、CondPattern を拡張された TestString と照合します。パターンが一致しないと、条件および対応するルールのセット全体が失敗します。パターンが一致すると、他に使用可能な条件がなくなるまで次の条件が処理されます。すべての条件が一致すると、処理が続行され、Substitution を使用して URL が置換されます。

http://yourserver//oldpath/rqstdrsrc など、複数のスラッシュ (/) を含む URL のリクエストの場合、RewriteCond および RewriteRule が正しく記述されていない場合は、//oldpath はこの 2 つのディレクティブをバイパスできます。

たとえば、次のルールがあるとします。

```
RewriteRule ^/oldpath(.*) /newpath$1 [R]
```

http://yourserver/oldpath/files のリクエストはリダイレクトされ、予想どおりのページ http://yourserver/newpath/files が戻されます。

ただし、http://yourserver//oldpath/files のリクエストはこのルールをバイパスし、予想していなかったページを提供する可能性があります。ルールで複数のスラッシュ (/) が取得されることを確認することで、この問題を回避できます。前述の例を解決するには、次のように置換を使用する必要があります。

```
RewriteRule ^/+somepath(.*) /otherpath$1 [R]
```

mod_rewrite のディレクティブ

この項では、次の mod_rewrite のディレクティブについて説明します。

- [RewriteEngine](#)
- [RewriteOptions](#)
- [RewriteLog](#)
- [RewriteLogLevel](#)
- [RewriteBase](#)

RewriteEngine

ランタイム・リライト・エンジンを有効化または無効化します。「Off」に設定すると、このモジュールではランタイム処理が実行されません。このディレクティブを使用して、すべての RewriteRule のディレクティブをコメント化するかわりにモジュールを無効にします。

リライト構成は、デフォルトで継承されません。つまり、RewriteEngine On ディレクティブを使用する各仮想ホストに対して指定する必要があります。

RewriteOptions

RewriteOptions 'inherit' を指定すると、親の構成を子に継承させることができます。仮想サーバー・コンテキストでは、これはメイン・サーバーのマップ、条件およびルールが継承されることを意味します。ディレクトリ・コンテキストでは、これは親ディレクトリの .htaccess 構成の条件とルールが継承されることを意味します。

RewriteLog

実行するリライト・アクションがサーバーによって記録されるファイルの名前を設定します。このファイル名の先頭にスラッシュ (/) がない場合は、Server Root への相対ファイル名とみなされます。ロギングを無効にするには、RewriteLog ディレクティブを削除またはコメント化するか、RewriteLogLevel 0 を使用します。ファイル名を /dev/null に設定して、ロギングを禁止しないでください。このように設定すると、サーバーが低速になり、メリットはありません。

RewriteLogLevel

リライト・ログ・ファイルの詳細レベルを設定します。デフォルト・レベルである 0 (ゼロ) はロギングなしを意味し、9 以上の値を指定すると実際には全アクションが記録されます。

RewriteBase

ディレクトリ単位のリライト用のベース URL を明示的に設定します。リライト・ルールをディレクトリ単位の構成 (.htaccess) ファイルで使用できます。新規 URL の置換が発生する場合は、サーバー処理にベース URL を追加する必要があります。これを可能にするには、対応する URL 接頭辞または URL ベースをモジュールで認識する必要があります。デフォルトでは、この接頭辞自体が対応するファイル・パスです。ただし、ほとんどの Web サイトでは、URL は物理ファイル名のパスに直接関連付けられていません。このような場合は、RewriteBase ディレクティブを使用して正しい URL 接頭辞を指定する必要があります。

Web サーバーの URL が物理ファイルのパスに直接関連付けられていない場合は、RewriteRule ディレクティブを使用する各 .htaccess ファイル内で RewriteBase を使用する必要があります。

例 7-10 RewriteBase ディレクティブ

次のディレクトリ単位の構成ファイルがあるとします。

```
## /abc/def/.htaccess - - per-dir config file for directory /abc/def
# /abc/def is the physical path of /xyz,
RewriteEngine On
RewriteBase /xyz
RewriteRule ^oldstuff%.html$ newstuff.html
```

例 7-10 では、/xyz/oldstuff.html のリクエストは物理ファイル /abc/def/newstuff.html に正確にリライトされます。

リライト・ルールのヒント

表 7-3 に、リライト・ルールを使用するためのヒントを示します。

表 7-3 リライト・ルールのヒント

値	定義
.	任意の 1 文字
[char]	大カッコで囲まれた任意の文字
b*	任意の数の文字 b からなる文字列
.*	任意の数の任意の文字からなる文字列

たとえば、/demo1、/demo2 および /demo3 からのリクエストを /alldemos にリダイレクトするには、リライト・ルールを次のどちらかとして記述します。

```
RewriteRule /demo. /alldemos [R]
```

または

```
RewriteRule /demo [123] /alldemos [R]
```

/DemoA、/DemoB および /DemoC を /alldemos にリダイレクトする場合は、次のように、前述のリライト・ルールに NC (no case) を追加します。

```
RewriteRule /demo [123] /alldemos [R, NC]
```

"." は 1 文字のみを処理するため、このリライト・ルールは /demonstration1 から /demos へのリダイレクトには機能しません。demo で始まる URL すべてを後続の文字に関係なくリダイレクト可能にするには、次のリライト・ルールを使用します。

```
RewriteRule ^/demo* /alldemos [R, NC]
```

前述の例では、^ は始まりを意味し、* は demo の後の任意の文字を意味します。

/demo1/not_just_index.html に対してリクエストがある場合、前述のすべてのリライト・ルールではリクエストは /alldemos/index.html にリダイレクトされますが、これは意図した結果でない場合があります。表 7-4 に示すように、/alldemos 内の対応するファイルにリダイレクトする必要があります。

表 7-4 リクエストのリダイレクト

リクエストの内容	リダイレクト先
/demo1/happy.html	/alldemos/happy.html
/demo1/go.jpg	/alldemos/go.jpg
/demos1/lucky.jpg	/alldemos/lucky.jpg

次のように、リライト・ルールに置換を使用する必要があります。

```
RewriteRule ^/demos1(.*)$ //alldemos/$1 [R NC]
```

このルールの内容は、次のとおりです。

happy.html、go.jpg および lucky.jpg など、demo1 の後に指定されている式の値が変数 (\$1) として使用され、/alldemos/ の後で置換されます。

リダイレクションの例

リクエストを DocumentRoot から newroot ディレクトリにリダイレクトする場合は、次の mod_rewrite のディレクティブを設定します。

```
RewriteEngine On
RewriteRule ^/(.*)$ /newroot/$1 [R]
```

あるディレクトリ (olddir) から別のディレクトリ (newdir) にファイル・リクエストを送信する場合は、次のディレクティブを設定します。

```
RewriteEngine On
RewriteRule ^/olddir(.*)$ /newdir/$1 [R]
```

どちらの場合も、リクエストされたリソースがリダイレクト先で実際に使用可能かどうかを確認する必要があります。mod_rewrite モジュールは、リクエストされたリソースが新しい場所にあるかどうかを確認しません。

HTTP TRACE メソッドを使用してリクエストをすべて無効にする場合は、次の mod_rewrite のディレクティブを設定します。

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

関連資料: Apache Server マニュアルの「Module mod_rewrite」

mod_setenvif

リクエストの特性に基づいて環境変数を設定できます。

関連資料: Apache Server マニュアルの「Module mod_setenvif」

mod_speling

スペルに誤りがある URL や、誤って大文字で記述された URL が訂正されます。

関連資料: Apache Server マニュアルの「Module mod_speling」

mod_status

サーバー・アクティビティとパフォーマンスに関する HTML ページが表示されます。

関連資料: Apache Server マニュアルの「Module mod_status」

mod_unique_id

リクエストごとに一意の ID が作成されます。このモジュールは、UNIX でのみ使用可能です。

関連資料: Apache Server マニュアルの「Module mod_unique_id」

mod_userdir

リクエストがユーザー固有のディレクトリにマップされます。

関連資料: Apache Server マニュアルの「Module mod_userdir」

mod_usertrack

ログが作成され、ユーザー・アクティビティが追跡されます。

関連資料: Apache Server マニュアルの「Module mod_usertrack」

mod_vhost_alias

動的に構成された大量の仮想ホスト設定が有効化されます。

関連資料: Apache Server マニュアルの「Module mod_vhost_alias」

mod_wchandshake

OracleAS Web Cache による Oracle HTTP Server の自動検出機能が提供されます。OracleAS Web Cache を使用しない場合は、このモジュールを無効にできます。これは Oracle モジュールです。

セキュリティの管理

この章では、Oracle HTTP Server のセキュリティ機能、およびセキュアな Web サイトを設定するための構成情報について説明します。

内容は、次のとおりです。

- [Oracle HTTP Server のセキュリティの概要](#)
- [ユーザーのクラスとその権限](#)
- [保護されるリソース](#)
- [認証と認可の適用](#)
- [ポート・トンネリングの概要](#)
- [Oracle Identity Management インフラストラクチャの利用](#)

関連資料： セキュリティの詳細は、次のマニュアルを参照してください。

- 『Oracle Application Server セキュリティ・ガイド』には、Oracle Application Server のセキュリティと基本機能の概要が記載されています。
- 『Oracle Identity Management 概要および配置プランニング・ガイド』には、Oracle セキュリティ・インフラストラクチャの管理者向けの説明が記載されています。

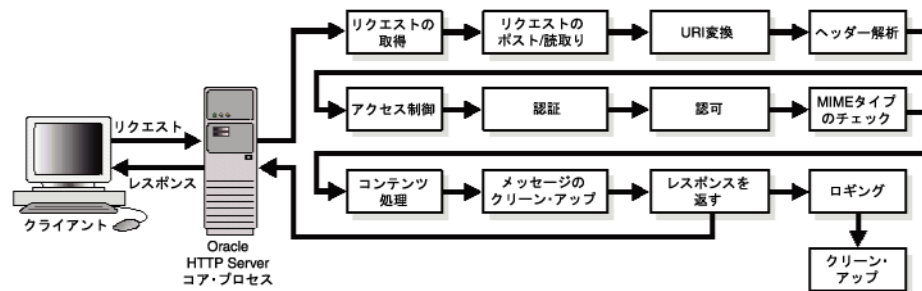
Oracle HTTP Server のセキュリティの概要

セキュリティ機能は、認証、認可および機密保護という3つのカテゴリに分類できます。Oracle HTTP Server では、この3つのカテゴリのすべてをサポートします。Oracle HTTP Server は Apache Web Server がベースで、そのセキュリティ・インフラストラクチャは、主に Apache モジュールの `mod_auth` と `mod_access`、および Oracle モジュールの `mod_oss1` と `mod_osso` により提供されています。`mod_auth` はユーザー名とパスワードのペアに基づく認証を提供し、`mod_access` はリクエストの特性（ホスト名または IP アドレスなど）に基づいてサーバーへのアクセスを制御します。`mod_oss1` は SSL を介して X.509 クライアント証明書を使用した機密保護と認証を提供し、`mod_osso` は Web アプリケーションでシングル・サインオン認証を使用可能にします。

Apache モデルに基づいて、Oracle HTTP Server ではアクセス制御、認証および認可の各メソッドを提供しています。これらのメソッドは、`httpd.conf` ファイルのアクセス制御ディレクティブを使用して構成できます。Oracle HTTP Server に URL リクエストが着信すると、サーバーのデフォルトと構成パラメータで決定される一連の手順で処理されます。URL リクエストの処理手順は、多くの Web リスナーに共通のモジュール（プラグイン）アーキテクチャを使用して実装されています。

図 8-1 は、サーバーによる URL リクエストの処理方法を示したものです。このプロセスの各手順は、サーバー・モジュールによりサーバーの構成に応じて処理されます。たとえば Basic 認証が使用される場合、図 8-1 の認証および認可という手順は、`mod_auth` モジュールの処理を示します。

図 8-1 Oracle HTTP Server での URL リクエストの処理手順



ユーザーのクラスとその権限

Oracle HTTP Server はユーザーを認可および認証します。ユーザーは、認可および認証後にサーバー上のリソースにアクセス、またはこれを変更できます。Oracle HTTP Server を使用してサーバーにアクセスするユーザーの3つのクラスとその権限は、次のとおりです。

- 認証を提供しないでサーバーにアクセスするユーザー。このユーザーがアクセスできるのは、保護されていないリソースのみです。
- Oracle HTTP Server 内のモジュールにより認証されているユーザーで、認可されている可能性もあるユーザー。これには、`mod_auth` および `mod_oss1` により認証されているユーザーが含まれます。このようなユーザーは、`httpd.conf` ファイルに定義されている URL にアクセスできます。

関連項目： 8-3 ページの「認証と認可の適用」

- `mod_osso` および Single Sign-On Server を使用して認証されているユーザー。このユーザーは、Single Sign-On により許可されているリソースにアクセスできます。

関連資料：『Oracle Application Server Single Sign-On 管理者ガイド』

保護されるリソース

Oracle HTTP Server は、次のようなリソースを保護するように構成されています。

- 静的コンテンツ。静的 HTML ページ、画像交換フォーマット、.gif ファイルおよび Oracle HTTP Server が直接提供するその他の静的ファイルなど。
- Oracle HTTP Server が直接起動する CGI/FastCGI スクリプト、単純なスクリプトまたはプログラム。
- Oracle HTTP Server 内のモジュールにより生成されるコンテンツ。mod_perl などのモジュールでは、クライアントに返されるレスポンスが生成されます。
- Oracle HTTP Server の後方に位置する Oracle Application Server コンポーネント (OC4J を使用して実行されるサーブレットと JSP で mod_oc4j を介してアクセスされるものを含む)。Oracle HTTP Server は、これらのコンポーネントの認証および認可の最前線を担当します。ただし、それ以上の認証がコンポーネント・レベルで発生することもあります。

認証と認可の適用

Oracle HTTP Server は、ユーザーの認証と認可を 2 段階で提供します。

- **ホストベースのアクセス制御** (第 1 段階) : これは、受信 HTTP リクエストおよびそのヘッダーの詳細情報 (IP アドレスやホスト名など) に基づきます。
- **ユーザーの認証と認可** (第 2 段階) : これは、HTTP サーバー構成に応じた様々な基準に基づきます。サーバーは、ユーザー名とパスワードのペアを使用してユーザーを認証するように構成できます。このペアが、既知のユーザーおよびパスワードのリストに照らしてチェックされます。また、Web アプリケーション用のシングル・サインオン認証を使用したり、SSL を介して X.509 クライアント証明書を使用するようにサーバーを構成することもできます。

ホストベースのアクセス制御

リクエスト処理サイクルの初期にアクセス制御が適用されます。これにより、ホスト名、IP アドレスまたはその他の特性 (ブラウザ・タイプなど) に基づいて、後続の処理を禁止できます。このタイプのアクセス制御を設定するには、deny、allow および order ディレクティブを使用します。この制限は Oracle HTTP Server 構成ディレクティブを使用して構成します。例 8-1 に示すように、<Files>、<Directory> および <Location> コンテナ・ディレクティブを使用し、特定のファイル、ディレクトリまたは URL 形式に基づいて制限できます。

例 8-1 ホストベースのアクセス制御

```
<Directory /internalonly/>
  order deny, allow
  deny from all
  allow from 192.168.1.* us.oracle.com
</Directory>
```

例 8-1 で、order ディレクティブは、Oracle HTTP Server が deny および allow ディレクティブの条件を読み取る順序を決定します。deny ディレクティブは、すべてのリクエストがアクセスを拒否されるようにします。次に、allow ディレクティブを使用して、192.168.1.* 範囲内の任意の IP アドレスから送信されるリクエストまたはドメイン名 us.oracle.com を持つリクエストに対して、ディレクトリ /internalonly/ 内のファイルへのアクセスが許可されます。ホスト・ベースの認証では、アクセス・ポリシーを明確にするために、allow と deny の両方を使用するのが一般的な方法です。

ファイル・システム・レベルでオブジェクトを比較する場合は、<Directory> または <Files> を使用する必要があります。URL レベルでオブジェクトを比較する場合は、<Location> を使用する必要があります。

注意： インターネット・アクセスの場合、ホスト名に基づいてアクセスを許可または制限するのは、セキュリティを提供する方法として優れているとはみなされません。ホスト名は簡単にスプーフィング（なりすまし）されるためです。IP アドレスでもこれは同じですが、妨害行為はより難しくなります。ただし、同じリスクを伴わないため、イントラネットの IP アドレス範囲を使用してアクセス制御を設定するのは合理的です。この場合は、ファイアウォールが正しく構成されているものとして扱います。

仮想ホストのアクセス制御

仮想ホストにアクセス制御を設定するには、サーバー構成ファイル `httpd.conf` 内の仮想ホスト・コンテナの中に `Include` ディレクティブを指定します。Include ディレクティブを仮想ホスト・コンテナ内で使用すると、ファイル内に含まれるアクセス制御ポリシーが指定されます。例 8-2 は、`httpd.conf` ファイルからの抜粋です。Include をこの方法で使用するための構文が示されています。

例 8-2 AccessConfig を使用したアクセス制御の設定

```
...
<VirtualHost ip_address_of_host.some_domain.com>
    ... virtual host directives ...
    Include conf/include.conf
</VirtualHost>
```

ホストベースのアクセス制御のための `mod_access` と `mod_setenvif` の使用

ホストベースのアクセス制御方法を使用すると、HTTP リクエストの発信元に基づいて制限領域へのアクセスを制御できます。Oracle HTTP Server では `mod_access` と `mod_setenvif` を使用して、ホストベースのアクセス制御を実行します。`mod_access` は、クライアントのホスト名、IP アドレスまたはクライアント・リクエストのその他の特性に基づいてアクセス制御を提供し、`mod_setenvif` は、リクエストの属性に基づいて環境変数を設定する機能を提供します。これらのモジュールを使用する構成ディレクティブを `httpd.conf` ファイルに入力すると、サーバーでは、ホストのアドレス、名前または HTTP リクエスト・ヘッダーの内容に基づいて、リクエストを実行あるいは拒否します。

ホストベース・アクセス制御を使用すると、静的な HTML ページ、アプリケーションまたはコンポーネントを保護できます。

Oracle HTTP Server は、次の 4 種類のホストベースのアクセス制御方法をサポートします。

- IP アドレスによるアクセス制御
- ドメイン名によるアクセス制御
- ネットワークまたはネットマスクによるアクセス制御
- 環境変数を使用したアクセス制御

これらの方法ではすべて、保護領域へのアクセス権が付与または拒否されるマシンを指定できます。ホストベースのアクセス制御方法のどれを選択するか（複数も可）は、制限されているコンテンツやアプリケーションをどの方法が最も効率的に保護するか、またはどの方法が最も保守しやすいかによって決まります。

IP アドレスによるアクセス制御 IP アドレスを使用したアクセス制御は、ホストベースのアクセス制御でよく使用される方法です。この方法では、DNS 参照を必要としません。DNS 参照には時間とシステム・リソースがかかり、サーバーが DNS スプーフィング攻撃を受けやすくなります。

例 8-3 IP アドレスによるアクセス制御

```
<Directory /secure_only/>
  order deny,allow
  deny from all
  allow from 207.175.42.*
</Directory>
```

例 8-3 では、207.175.42.* 範囲を除く全 IP アドレスからのリクエストは、/secure_only/ ディレクトリへのアクセスを拒否されます。

ドメイン名によるアクセス制御 ドメイン名ベースのアクセス制御を IP アドレス・ベースのアクセス制御とともに使用すると、IP アドレスが警告なしで変更される問題が解決します。この 2 つの方法を組み合わせると、IP アドレスが変更される場合でも、排除するドメイン名はアクセスを拒否されるため、サイトの制限領域が保護されます。

ドメイン名ベースと IP アドレス・ベースのアクセス制御を組み合わせるには、例 8-4 に示されている構文を使用します。

例 8-4 ドメイン名によるアクセス制御

```
<Directory /co_backgr/>
  order allow,deny
  allow from all
  # 141.217.24.* is the IP for malicious.cracker.com
  deny from malicious.cracker.com 141.217.24.*
</Directory>
```

例 8-4 では、ディレクトリ /co_backgr/ に対するリクエストは、ドメイン名 malicious.cracker.com または IP アドレス範囲 141.217.24.* からのリクエストを除いて、すべて受信されます。これは、ドメイン名または IP アドレスのスプーフィングに対する絶対的な予防措置ではありませんが、malicious.cracker.com が IP アドレスを変更している場合でも、ここからの攻撃に対してサイトを保護します。

ネットワークまたはネットマスクによるアクセス制御 ネットワークのサブセット (IP アドレスにより指定) に基づいてアクセスを制御できます。この構文を例 8-5 に示します。

例 8-5 ネットワークまたはネットマスクによるアクセス制御

```
<Directory /payroll/>
  order deny,allow
  deny from all
  allow from 10.1.0.0/255.255.0.0
</Directory>
```

例 8-5 では、ネットワークとネットマスクのペアからのアクセスが許可されます。ネットマスクは、IP アドレスをネットワーク、サブネットおよびホスト識別子に分割する方法を示したものです。ネットマスクを使用すると、IP アドレスのホスト ID 部分のみを参照できます。

例 8-5 のネットマスク 255.255.0.0 は、クラス B アドレスのデフォルト・ネットマスク設定です。バイナリの 1 (10 進の 255) がネットワーク ID をマスクし、バイナリの 0 (ゼロ) (10 進のゼロ) が、指定された IP アドレスのホスト ID を保持します。

環境変数を使用したアクセス制御 アクセス制御には、IP アドレスやドメイン名のかわりに任意の環境変数を使用できます。このタイプのアクセス制御には、BrowserMatch および SetEnvIf ディレクティブを使用します。

注意: 通常、BrowserMatch および SetEnvIf はセキュリティ・ポリシーの実装には使用されません。これらは、ブラウザのタイプとバージョンに応じて異なるリクエスト処理を提供するために使用されます。

BrowserMatch は、リクエストの送信に使用するブラウザのタイプに応じてアクセスを許可するとき 사용합니다。たとえば、Netscape ブラウザからのリクエストのみにアクセスを許可する場合は、例 8-6 に示されている構文を使用します。

例 8-6 環境変数を使用したアクセス制御

```
BrowserMatch ^Mozilla netscape_browser
<Directory /mozilla-area/>
    order deny,allow
    deny from all
    allow from env=netscape_browser
</Directory>
```

SetEnvIf は、HTTP リクエストに含まれているヘッダー情報に応じてアクセスを許可するとき 사용합니다。たとえば、HTTP バージョン 1.0 以前を使用するブラウザからのアクセスを拒否する場合は、例 8-7 に示されている構文を使用します。

例 8-7 SetEnv によるアクセス制御

```
SetEnvIf Request_Protocol ^HTTP/1.1 http_11_ok
<Directory /http1.1only/>
    order deny,allow
    deny from all
    allow from env=http_11_ok
</Directory>
```

ユーザーの認証と認可

Basic 認証では、HTTP リクエストにサービスを提供する前に、ユーザー名とパスワードを求め、プロンプトを表示します。ブラウザが保護領域のページをリクエストすると、Oracle HTTP Server は WWW-Authenticate: ヘッダーと構成ディレクティブ AuthName により構成されているレルムの名前を含む、不認可のメッセージ（ステータス・コード 401）をレスポンスとして返します。ブラウザはこのレスポンスを受信すると、ユーザー名とパスワードを求め、プロンプトを表示します。ユーザーがユーザー名とパスワードを入力した後、ブラウザはこの情報を認可ヘッダーに入れてサーバーに返します。認可ヘッダー・メッセージ内では、ユーザー名とパスワードは BASE64 エンコード文字列としてエンコードされます。

ユーザー認可では、特定のサーバー・リソース（ファイルやディレクトリなど）に関連付けられているアクセス制御リストに照らして、認証済ユーザーがチェックされます。ユーザー認可を構成するには、通常は仮想ホスト・コンテナ内にある httpd.conf ファイルに require ディレクティブを指定します。ユーザー認可は、一般に、ユーザー認証と組み合わせて使用されます。サーバーはユーザーの名前とパスワードを認証した後、リクエストされたサーバー・リソースに関連付けられているアクセス制御リストとそのユーザーを比較します。Oracle HTTP Server により、リスト上にユーザーまたはユーザー・グループが見つかったら、そのユーザーがリソースを使用できるようになります。

ユーザー認証のための mod_auth の使用

ユーザー認証はユーザー名とパスワードに基づきますが、この 2 つは既知のユーザーおよびパスワードのリストに照らしてチェックされます。このユーザー名とパスワードのペアは、テキスト・ファイル、データベースまたはディレクトリ・サービスなど様々な形で格納できます。次に、構成ディレクティブが httpd.conf ファイルで使用され、サーバー上にこのタイプのユーザー認証を構成します。mod_auth では、Basic 認証の設定に AuthUserFile ディレクティブを使用します。これは、ファイルのみをサポートします。

どの認証方法の場合にも、表 8-1 に示される構成ディレクティブを組み合わせて使用する必要があります。

表 8-1 ディレクティブの説明

ディレクティブ名	説明
AuthName	ユーザー名とパスワードが有効なレルムの名前を定義します。名前にスペースが含まれている場合は、二重引用符を使用します。
AuthType	認証タイプを指定します。ほとんどの認証モジュールでは Basic 認証を使用します。この認証タイプでは、ユーザー名とパスワードをクリアテキストで送信します。これはお薦めしません。
AuthUserFile	ユーザー名とパスワードを含むファイルのパスを指定します。
AuthGroupFile	グループの名前とメンバーを含むファイルのパスを指定します。

ユーザー認証のための mod_osso の使用

mod_osso により、Oracle HTTP Server でシングル・サインオンがオンになります。mod_osso では、受信リクエストを検査してリクエストされたリソースが保護されているかどうかを判断し、保護されている場合はユーザー用の Oracle HTTP Server Cookie を取得します。

Oracle HTTP Server は、mod_osso を使用して、シングル・サインオン (SSO) パートナ・アプリケーションになります。このアプリケーションは、ユーザーの認証とユーザー識別情報の取得に OracleAS Single Sign-On を使用し、ユーザー識別情報を Apache ヘッダー変数として Web アプリケーションに提供することができます。

Web アプリケーションは、mod_osso を使用して SSO 認証が必要な URL を登録できます。Oracle HTTP Server が URL リクエストを受信すると、mod_osso は SSO 認証に必要なリクエストを検出し、このリクエストを SSO サーバーにリダイレクトします。SSO サーバーがユーザーを認証した後、サーバーはユーザーの認証識別情報をセキュリティで保護されたトークンまたは Cookie に入れて mod_osso に返します。mod_osso は Cookie からユーザーの識別情報を取り出し、Oracle HTTP Server インスタンス内で実行中のアプリケーションにユーザーの識別情報を伝播します。mod_osso は、CGI で実行中のアプリケーションと OC4J で実行中のアプリケーションにユーザーの識別情報を伝播することができ、また、静的ファイルにアクセスするユーザーを認証することもできます。

関連資料:

- 『Oracle Application Server Single Sign-On 管理者ガイド』
- 8-13 ページの「Oracle Identity Management インフラストラクチャの利用」

ユーザー認証のための mod_oss1 の使用

mod_oss1 は Oracle HTTP Server へのプラグインで、サーバーが SSL を使用できるようにします。Oracle HTTP Server 製品では、mod_ssl が mod_oss1 に置き換わっています。mod_ssl はサポートされていません。

関連資料: mod_oss1 のディレクティブを使用した SSL の有効化および構成の詳細は、9-1 ページの「Oracle HTTP Server での SSL の有効化」を参照してください。

ポート・トンネリングの概要

ポート・トンネリングを使用すると、Oracle HTTP Server と OC4J 間のすべての通信を 1 つまたは少数のポート上で行えます。以前は、Oracle HTTP Server と複数の OC4J インスタンス間の通信を処理するには、ファイアウォール構成に多数のポートのポート情報を含める必要がありました。

関連資料: ポート・トンネリングの詳細は、『Oracle HTTP Server 管理者ガイド』を参照してください。

ポート・トンネリングの構成

ポート・トンネリングを構成するには、次の3つのタスクを実行します。

- [タスク 1: opmn.xml の構成](#)
- [タスク 2: iaspt.conf の構成](#)
- [タスク 3: mod_oc4j.conf の構成](#)

タスク 1: opmn.xml の構成

中間層（スタンドアロンの Oracle HTTP Server 2.0 ではない）に対して次の手順を実行し、1つ以上の iaspt デーモンを起動します。

1. デフォルトでは、無効化されている iaspt の opmn.xml エントリがあります。opmn.xml を編集し、status="disable" を status="enable" に変更して、iaspt を有効化します。
2. オプションとして、ポート範囲を変更して iaspt デーモンで使用される TCP/IP ポートを変え、numprocs を変更して iaspt デーモン・プロセスの数を増やすことができます。

次に、iaspt デーモンの完全な構成例を示します。このコンポーネントとともに使用できるすべての構成要素または属性が含まれています。

```
<module path="/ORACLE_HOME/opmn/lib/libopmniaspt">
  <module-id id="IASPT" />
</module>
<ias-component id="IASPT" status="enabled" id-matching="false">
  <process-type id="IASPT" module-id="IASPT">
    <port id="ajp" range="6701-6703"/>
    <process-set id="IASPT" restart-on-death="true" numprocs="3"/>
  </process-type>
</ias-component>
```

3. 次のコマンドを実行して、opmn デーモンに構成ファイルのリロードを指示します。

```
opmnctl reload
```

タスク 2: iaspt.conf の構成

中間層（スタンドアロンの Oracle HTTP Server 2.0 ではない）に対して次の手順を実行し、iaspt デーモンが使用する SSL Wallet を指定するための iaspt.conf の構成を行います。

1. mod_oc4j と iaspt 間の通信は、必ず暗号化されます。そのため、SSL Wallet ファイルは iaspt デーモン用に構成する必要があります。デフォルトでは、この Wallet は Oracle HTTP Server Wallet と同じです。このデフォルトは、iaspt.conf の次の値を編集すると変更できます。

```
wallet-file=<path to wallet file>
wallet-password=<password>
```

関連項目：

- [8-12 ページの「wallet-file」](#)
 - [8-12 ページの「wallet-password」](#)
2. 次のコマンドを使用して、iaspt デーモンを起動します。

```
opmnctl startall
```


タスク 3: mod_oc4j.conf の構成

スタンドアロンの Oracle HTTP Server 2.0 (中間層ではない) に対して次の手順を実行し、iaspt を使用してリクエストをルーティングするための mod_oc4j.conf の構成を行います。

1. 次の行を mod_oc4j.conf に追加して、ポート・トンネリングを有効にします。

```
Oc4jiASPTActive on
```

関連項目： [8-10 ページの「Oc4jiASPTActive」](#)

2. 次の 2 行を mod_oc4j.conf に追加して、SSL Wallet および Wallet パスワードを mod_oc4j.conf に指定します。

```
Oc4jiASPTWalletFile <path to wallet file>
Oc4jiASPTWalletPassword <password of wallet>
```

この Wallet は、Oracle HTTP Server または iaspt (あるいはその両方) で使用されるものと同じでもかまいません。

関連項目：

- [8-11 ページの「Oc4jiASPTWalletFile」](#)
- [8-11 ページの「Oc4jiASPTWalletPassword」](#)
- Oracle Wallet Manager の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

3. iaspt デーモンのホストおよびポート・アドレスを指定します。たとえば、次の行を mod_oc4j.conf に追加します。

```
Oc4jiASPTProcess myhost.us.oracle.com:6701
```

保持する iaspt デーモンと同じ数の Oc4jiASPTProcess の行を追加できます。ホストおよびポート・アドレスは、構成された iaspt デーモンのものと一致する必要があります。たとえば、[8-8 ページの「タスク 1: opmn.xml の構成」](#)の手順 2 の例で構成した 3 つの iaspt デーモンにリクエストをルーティングするには、次の 3 行を追加します。

```
Oc4jiASPTProcess myhost.us.oracle.com:6701
Oc4jiASPTProcess myhost.us.oracle.com:6702
Oc4jiASPTProcess myhost.us.oracle.com:6703
```

関連項目： [8-11 ページの「Oc4jiASPTProcess」](#)

4. 次のコマンドを使用して、Oracle HTTP Server を再起動し、変更を有効にします。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

ポート・トンネリング用の SSL の構成

この項では、iaspt と OC4J 間での SSL の構成について説明します。

デフォルトでは、iaspt デーモンと OC4J プロセスは、暗号化されていないデータを使用して通信を行います。これらのプロセス間の SSL 通信を構成するには、次の手順を実行します。

1. iaspt.conf で、値 destination-ssl を false から true に変更します。
2. SSL を使用するように OC4J プロセスを構成する方法については、『Oracle Application Server Containers for J2EE セキュリティ・ガイド』を参照してください。

ポート・トンネリングの構成のリファレンス

この項では、次の構成ファイルとそのパラメータについて説明します。

- [opmn.xml](#)
- [mod_oc4j.conf](#)
- [iaspt.conf](#)

opmn.xml

Oracle Application Server 内で OPMN により管理されるプロセスを記述します。

関連項目： [B-4 ページの「opmn.xml」](#)

ポート・トンネリングの一環として、起動される `iaspt` デーモン・プロセスを記述する [エン트리](#) が OPMN に存在する必要があります。このエントリーには次の記述を含めます。

- 起動する `iaspt` デーモン・プロセスの数
- これらのプロセスで使用できるポート

関連項目： [8-12 ページの「iaspt.conf」](#)

デフォルトの Oracle Application Server では、`iaspt` コンポーネントは `opmn.xml` に含まれていますが、無効化されています。

mod_oc4j.conf

Oracle HTTP Server により `mod_oc4j` を構成します。

関連項目： [B-3 ページの「mod_oc4j.conf」](#)

ポート・トンネリング用に、次を指定したディレクティブを追加する必要があります。

- `port-tunneling` を使用するかどうか
- `iaspt` デーモン・プロセスの静的位置
- `iaspt` デーモン・プロセスとの接続の確立に使用する SSL 証明書の位置

関連項目： [8-12 ページの「iaspt.conf」](#)

デフォルトでは、`mod_oc4j` は OC4J と直接通信します。ポート・トンネリング・プロセスの場合、`mod_oc4j` は `iaspt` デーモンを介して OC4J に通信する必要があります。

次のディレクティブを使用して、`mod_oc4j` を `iaspt` デーモンに接続します。

- [Oc4jiASPTActive](#)
- [Oc4jiASPTProcess](#)

Oc4jiASPTActive `mod_oc4j` がリクエストをルーティングするときにポート・トンネリングを考慮する必要があるかどうかを示します。[Oc4jEnableSSL](#) が「On」に構成されている場合、このディレクティブは「On」に構成しないでください。ポート・トンネリング・プロセスを有効化するには、このディレクティブを「On」に設定します。

カテゴリ	値
パラメータ名	Oc4jiASPTActive
パラメータ・タイプ	文字列
有効値	On または Off

カテゴリ	値
デフォルト値	Off

Oc4jiASPTProcess ポート・トンネリング・プロセスのリスニング・ホストおよびポートを記述します。mod_oc4j.conf ファイル内で、複数のポート・トンネリング・プロセス用はこのディレクティブの複数のインスタンスを指定できます。

このディレクティブの構文は、host:port です。host 値は、iaspt デーモンが実行されているマシンのホスト名と一致する必要があります。port 値は、その iaspt の opmn.xml に構成されているポートと一致する必要があります。host には、標準のホスト名と IP アドレスの両方を使用できます。

カテゴリ	値
パラメータ名	Oc4jiASPTProcess
パラメータ・タイプ	文字列
有効値	使用可能な iaspt デーモンの host:port 値
デフォルト値	該当なし
構文	host:port 例: myhost.us.oracle.com:6667

mod_oc4j は iaspt デーモンと通信するときに SSL を使用する必要があります。次に、SSL の有効化に使用されるディレクティブを示します。

- [Oc4jiASPTWalletFile](#)
- [Oc4jiASPTWalletPassword](#)

Oc4jiASPTWalletFile iaspt デーモンとの SSL 通信に使用される SSL 証明書を含む、Oracle Wallet ファイルの位置を指定します。

カテゴリ	値
パラメータ名	Oc4jiASPTWalletFile
パラメータ・タイプ	文字列
有効値	iaspt デーモンとの SSL 接続確立時に使用される SSL 証明書を含む Wallet ファイルへのパス
デフォルト値	該当なし
構文	有効なファイル名 例: /foo/bar/myfilename

Oc4jiASPTWalletPassword Wallet ファイルのオープン時に認証に使用される不明瞭化されたパスワードの値を指定します。この値は、Oracle Wallet Manager に含まれているユーティリティを使用して取得されます。

カテゴリ	値
パラメータ名	Oc4jiASPTWalletPassword
パラメータ・タイプ	文字列
有効値	Oc4jiASPTWalletFile により指定された Wallet ファイルのオープン時に認証に使用されるパスワード
デフォルト値	該当なし

関連資料： Oracle Wallet Manager の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

iaspt.conf

ポート・トンネリングを構成します。

関連項目： B-3 ページの「iaspt.conf」

このファイルは、次の情報を示します。

- Wallet ファイルおよび使用するパスワード。
- ログ・ファイルの位置とログ・レベル。
- iaspt デーモンがリスニングするポート (オプション)。このポートは、iaspt.conf 内に指定するか、ポートの範囲を指定して opmn.xml から渡すことができます。これにより、複数のポート・トンネリング・プロセスで同じ iaspt.conf ファイルを使用できます。

iaspt.conf ファイルには、一連の名前 / 値ペアが含まれます。次に、受け入れられるパラメータの名前を示します。

- [wallet-file](#)
- [wallet-password](#)
- [log-file](#)
- [log-level](#)
- [iaspt-port](#)

wallet-file ピアとの SSL 通信に使用する SSL 証明書を含む、Oracle Wallet ファイルの位置を指定します。

カテゴリ	値
パラメータ名	wallet-file
パラメータ・タイプ	文字列
有効値	他のプロセスとの SSL 接続確立時に使用される SSL 証明書を含む Wallet ファイルへのパス
デフォルト値	該当なし
構文	有効なファイル名 例: /foo/bar/myfilename

wallet-password Wallet ファイルのオープン時に認証に使用されるパスワードの値を指定します。この値は、Oracle Wallet Manager に含まれているユーティリティを使用して取得されます。

カテゴリ	値
パラメータ名	wallet-password
パラメータ・タイプ	文字列
有効値	wallet-file により指定された Wallet ファイルのオープン時に認証に使用されるパスワード
デフォルト値	該当なし

関連資料： Oracle Wallet Manager の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

log-file iaspt デーモンのログ・メッセージが書き込まれるログ・ファイルのパスを指定します。

カテゴリ	値
パラメータ名	log-file
パラメータ・タイプ	文字列
有効値	iaspt デーモンのログ・メッセージが書き込まれるログ・ファイルのパス
デフォルト値	該当なし
構文	有効なファイル名 例: /foo/bar/myfilename

log-level ログ・レベルを指定します。9 が最高で、0 はログしないことを意味します。

カテゴリ	値
パラメータ名	log-level
パラメータ・タイプ	整数
有効値	整数 (0 ~ 9)
デフォルト値	3

iaspt-port iaspt デーモンが接続を受け入れるポートの値を指定します。このパラメータはオプションです。

カテゴリ	値
パラメータ名	iaspt-port
パラメータ・タイプ	整数
有効値	有効な TCP/IP ポート値
構文	整数 例: 9898
デフォルト値	該当なし

Oracle Identity Management インフラストラクチャの利用

この項では、Oracle HTTP Server による Oracle Identity Management インフラストラクチャの使用方法を説明します。

概要

Oracle Identity Management は、分散セキュリティのために Oracle Application Server が依存する統合インフラストラクチャです。これは、Oracle Internet Directory、Oracle Directory Integration and Provisioning、Delegated Administration Services、OracleAS Single Sign-On および OracleAS Certificate Authority で構成されています。

関連資料： 『Oracle Identity Management 概要および配置プランニング・ガイド』

Oracle Application Server Single Sign-On と mod_osso の使用

Oracle Application Server は、OracleAS Single Sign-On を使用して、Web ベース・アプリケーションに対するシングル・サインオン (SSO) をサポートします。OracleAS Single Sign-On を使用して、Oracle Application Server にログインし、認可されているアプリケーションにアクセスできます。アプリケーションごとにユーザー名とパスワードを再入力する必要はありません。これは、ユーザー情報を格納する Oracle Internet Directory と完全に統合されています。Oracle Internet Directory を使用して、LDAP ベースのユーザーおよびパスワードの管理をサポートします。

Oracle HTTP Server モジュールの mod_osso を使用すると、Oracle Application Server 全体で OracleAS Single Sign-On を透過的に使用できます。Oracle HTTP Server は、mod_osso を使用して、SSO パートナ・アプリケーションになります。このアプリケーションは、ユーザーの認証とユーザー識別情報の取得に SSO を使用し、ユーザー識別情報を Apache ヘッダー変数として Web アプリケーションに提供することができます。

関連項目： [8-7 ページの「ユーザー認証のための mod_osso の使用」](#)

Oracle HTTP Server での SSL の有効化

この章では、Oracle HTTP Server での SSL の有効化および構成について説明します。内容は、次のとおりです。

- [概要](#)
- [SSL の構成](#)
- [追加の SSL 機能](#)
- [SSL 構成ディレクティブの使用](#)

概要

Secure Sockets Layer (SSL) は、インターネット上で安全にメッセージを送信するように設計されている暗号化通信プロトコルです。SSL は、アプリケーション層上の Oracle HTTP Server と TCP/IP 層との間に位置し、クライアントによりセキュアな接続が行われたときに透過的に暗号化と復号化を処理します。

SSL の一般的な用途の 1 つは、ブラウザと Web サーバー間の Web HTTP 通信を保護することです。この場合、保護されていない HTTP の使用は排除されません。保護されたバージョンは、単純に HTTP over SSL (HTTPS) と呼ばれます。違いは、HTTPS では URL スキームに `http://` ではなく `https://` を使用することと、デフォルトの通信ポートが 4443 (UNIX) または 443 (Windows) であることです。

`mod_oss1` は、サーバーが SSL を使用できるようにする Oracle HTTP Server へのプラグインです。

関連資料：『Oracle Application Server 管理者ガイド』

SSL の構成

Oracle Application Server のインストール時は、デフォルトで SSL が無効化されています。SSL を有効化および構成するには、次のタスクを実行します。

- [タスク 1: 実際の Wallet の作成](#)
- [タスク 2: SSL の有効化](#)
- [タスク 3: \(オプション\) 構成のカスタマイズ](#)

タスク 1: 実際の Wallet の作成

SSL について Oracle HTTP Server を構成するには、サーバー用の証明書が含まれる **Wallet** が必要です。Wallet には、証明書リクエスト、証明書および秘密鍵など、資格証明が格納されます。

Oracle HTTP Server によって自動的にインストールされるデフォルトの Wallet は、テスト専用のもので、本番環境のサーバー用に、実際の Wallet を作成する必要があります。デフォルトの Wallet は、`ORACLE_HOME/ohs/conf/ssl.wlt/default` にあります。新規 Wallet をその場所に置いてかまいません。また、実際の Wallet の場所を指すように `ORACLE_HOME/ohs/conf/ssl.conf` の `SSLWallet` ディレクティブを変更することもできます。

関連資料： Wallet の作成手順は、『Oracle Application Server 管理者ガイド』を参照してください。次の作業を実行することが重要です。

1. 証明書リクエストを生成します。共通名の場合は、構成中のサイトの名前または別名を指定します。
2. Wallet の自動ログイン機能を設定します。この自動ログイン機能を必ず有効にしてください。デフォルトの Wallet の場合、この機能は無効になっています。

タスク 2: SSL の有効化

SSL を手動で有効化するには、次の手順を実行します。

1. テキスト・エディタで `opmn.xml` を開きます。
2. `<ias-component id="HTTP_Server">` エントリで、起動モードを `"ssl-disabled"` から `"ssl-enabled"` に変更します。変更後のエントリは次のようになります。

```
<data id="start-mode" value="ssl-enabled"/>
```
3. `opmn.xml` を保存してクローズします。

4. 次のコマンドを使用して OPMN を再ロードします。

```
opmnctl reload
```

5. 次のコマンドを使用して、Oracle HTTP Server を停止します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`

- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`

6. 次のコマンドを使用して、Oracle HTTP Server を起動します。

- UNIX の場合: `ORACLE_HOME/opmn/bin> opmnctl [verbose] startproc ias-component=HTTP_Server`

- Windows の場合: `ORACLE_HOME\opmn\bin> opmnctl [verbose] startproc ias-component=HTTP_Server`

注意: Oracle HTTP Server の停止と起動は、必ず手順に従って行うようにしてください。Oracle HTTP Server を再起動しても、Oracle HTTP Server の停止および起動と必ずしも同じ結果にはなりません。

7. SSL ポートにナビゲートして、SSL が正常に有効化されたかどうかを確認できます。たとえば、次のように入力します。

```
HTTPS://hostname:4443
```

タスク 3: (オプション) 構成のカスタマイズ

必要に応じて、`mod_oss1` のディレクティブを使用して、構成をさらにカスタマイズできます。

関連項目: 9-5 ページの「[mod_oss1 のディレクティブの使用](#)」

注意: インストール時にインストールされるテンプレート・ファイルには、すべての必須 SSL 構成ディレクティブおよび SSL 用のデフォルト設定が含まれています。

クライアント認証を有効にするには、次のようにします。

1. サーバー側で [SSLVerifyClient](#) を指定します。
2. HTTPS 接続について、クライアント側で適切なクライアント証明書を使用します。クライアント証明書の取得および使用方法の詳細は、クライアントのマニュアルを参照してください。クライアント証明書がサーバー Wallet で信頼できるかを確認してください。

関連資料: 信頼できる証明書を Wallet にインポートする方法の手順は、『Oracle Application Server 管理者ガイド』を参照してください。

追加の SSL 機能

この項では、このリリースでサポートされる SSL 機能について説明します。

- [グローバル・サーバー ID のサポート](#)
- [PKCS #11 のサポート](#)

グローバル・サーバー ID のサポート

この機能により、「ステップアップ」、「Server Gated Cryptography (SGC)」または「グローバル・サーバー ID」など様々に称される SSL プロトコル機能のサポートが追加されます。「ステップアップ」は、古くて暗号化強度が低いブラウザをステップアップして、512 ビット超の公開鍵および 64 ビット超のバルク暗号鍵を SSL プロトコルで使用できるようにする機能です。つまり、512 ビット超の公開鍵や、ステップアップ・デジタル著作権が含まれるサーバーの X.509 証明書が Oracle Application Server で使用できます。このような証明書は、証明書そのものには通常 1024 ビットの証明書が含まれますが、しばしば「128 ビットの」証明書と呼ばれます。Verisign 社の Secure Site Pro は、Oracle Application Server で使用できるこのような証明書の一例です。

グローバル・サーバー ID 機能は、デフォルトで提供されます。構成の必要はありません。

PKCS #11 のサポート

公開鍵暗号規格 #11 (略して PKCS #11) は、ハードウェア・セキュリティ・モジュールをシステムで使用する方法的概要を定めた公開鍵暗号仕様です。ハードウェア・セキュリティ・モジュールとは、基本的に、暗号機能 (暗号化 / 復号化) が実行され、暗号鍵が格納されるボックスのことです。

Oracle HTTP Server では、nCipher の SSL 専用ハードウェアを使用できます。nCipher は、認定サード・パーティのアクセラレータで、SSL で使用される PKI 暗号化のパフォーマンスを向上させます。

関連資料:

- 『Oracle Application Server 管理者ガイド』
- <http://www.ncipher.com>

SSL 構成ディレクティブの使用

mod_oss1 は、Oracle Application Server への HTTPS プロトコル接続を標準サポートします。SSL 経由で Oracle 提供の暗号化メカニズムを使用し、Oracle HTTP Server とブラウザ・クライアント間を安全に接続できるようにします。このモジュールは、デジタル証明書技術を使用したインターネット上での認証にも使用できます。このモジュールは SSL バージョン 3.0 をサポートし、次の機能を提供します。

- **RSA** または **DES** 暗号化規格を使用した、クライアントとサーバー間の暗号化通信
- **MD5** または **SHA** チェックサム・アルゴリズムを使用した、クライアント / サーバー通信の整合性チェック
- Oracle **Wallet** を使用した証明書管理

次の mod_ssl のディレクティブは、mod_oss1 ではサポートされていません。

- SSLRandomSeed
- SSLCertificateFile
- SSLCertificateKeyFile
- SSLCertificateChainFile
- SSLCACertificateFile
- SSLCACertificatePath
- SSLVerifyDepth

注意: これらのディレクティブが使用されていると、サーバーは起動しません。

mod_oss1 のディレクティブの使用

Oracle HTTP Server に対して SSL を構成するには、使用する mod_oss1 のディレクティブを httpd.conf ファイルに入力します。

次のディレクティブについて説明します。

- SSLAccelerator
- SSLCARevocationFile
- SSLCARevocationPath
- SSLCipherSuite
- SSLEngine
- SSLLog
- SSLLogLevel
- SSLMutex
- SSLOptions
- SSLPassPhraseDialog
- SSLProtocol
- SSLRequire
- SSLRequireSSL
- SSLSessionCache
- SSLSessionCacheTimeout
- SSLVerifyClient
- SSLWallet
- SSLWalletPassword

SSLAccelerator

SSL アクセラレータが使用されるかどうかを指定します。現在サポートされているのは、nFast カードのみです。

カテゴリ	値
有効値	yes または no
構文	SSLAccelerator yes no
デフォルト	SSLAccelerator no
コンテキスト	サーバー構成

注意： SSLAccelerator ディレクティブは使用されていません。Wallet を使用した SSL アクセラレーション・サポートの有効化の詳細は、<http://www.oracle.com/technology/documentation> にある『Oracle Database Advanced Security 管理者ガイド』を参照してください。

SSLCARevocationFile

証明書を発行した **CA** (認証局) からの証明書失効リスト (CRL) をまとめるファイルを指定します。このリストは、クライアント認証に使用されます。このファイルは、**PEM** でエンコードされた様々な CRL ファイルを優先順位の順に連結したものです。このディレクティブは、**SSLCARevocationPath** のかわりに、または付加的に使用できます。

カテゴリ	値
構文	SSLCARevocationFile <i>file_name</i>
例	SSLCARevocationFile /ORACLE_HOME/ohs/conf/ssl.crl/ca_bundle.crl
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLCARevocationPath

PEM でエンコードされている証明書失効リスト (CRL) が格納されるディレクトリを指定します。CRL は、証明書の発行元の **CA** (認証局) から届きます。CRL のいずれかに記載されている証明書を使用してクライアントが自身を認証しようとする、証明書は取り消され、そのクライアントはサーバーに対して自身を認証できなくなります。

カテゴリ	値
構文	SSLCARevocationPath <i>path/to/CRL_directory/</i>
例	SSLCARevocationPath /ORACLE_HOME/ohs/conf/ssl.crl/
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLCipherSuite

クライアントが SSL ハンドシェイク時に使用できる SSL **暗号スイート** を指定します。このディレクティブでは、コロンで区切られた暗号指定文字列を使用して暗号スイートを識別します。**表 9-2** に、必要な暗号スイートを記述するためにこの文字列で使用できるタグを示します。

タグと接頭辞を組み合わせて、暗号指定文字列が作成されます。

カテゴリ	値
有効値	none: リストに暗号を追加します。 +: リストに暗号を追加し、リスト内の正しい位置に配置します。 -: リストから暗号を削除します (後で追加できます)。 !: リストから暗号を永続的に削除します。
例	SSLCipherSuite ALL:!LOW:!DH この例では、低強度暗号と Diffie-Hellman 鍵交換アルゴリズム を使用する暗号を除くすべての暗号が指定されています。
構文	SSLCipherSuite <i>cipher-spec</i>
デフォルト	ALL:!ADH:!EXPORT56:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
コンテキスト	サーバー構成、仮想ホスト、ディレクトリ

表 9-1 SSL 暗号スイートのタグ

機能	タグ	意味
鍵の交換	kRSA	RSA 鍵の交換
鍵の交換	kDhR	RSA 鍵を使用した Diffie-Hellman 鍵の交換
認証	aNULL	認証なし
認証	aRSA	RSA 認証
認証	aDH	Diffie-Hellman 認証
暗号化	eNULL	暗号化なし
暗号化	DES	DES エンコード
暗号化	3DES	Triple-DES エンコード
暗号化	RC4	RC4 エンコード
データ整合性	MD5	MD5 ハッシュ関数
データ整合性	SHA	SHA ハッシュ関数
エイリアス	SSLv3	すべての SSL V3.0 暗号
エイリアス	EXP	すべての輸出暗号
エイリアス	EXP40	すべての 40 ビット輸出暗号のみ
エイリアス	EXP56	すべての 56 ビット輸出暗号のみ
エイリアス	LOW	すべての低強度暗号（輸出暗号と Single-DES）
エイリアス	MEDIUM	128 ビット暗号化を使用したすべての暗号
エイリアス	HIGH	Triple-DES を使用したすべての暗号
エイリアス	RSA	RSA 鍵交換を使用したすべての暗号
エイリアス	DH	Diffie-Hellman 鍵交換を使用したすべての暗号

注意： 輸出版のブラウザが使用される場合は、制限があります。Oracle モジュール `mod_oss1` は、サーバーが 512 ビットの鍵サイズの Wallet を使用するときのみ、RC4-40 暗号化をサポートします。

表 9-2 Oracle Advanced Security 10g でサポートされている暗号スイート

暗号スイート	認証	暗号化	データ整合性
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES (168)	SHA
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4 (128)	SHA
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4 (128)	MD5
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES (56)	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH anon	3DES (168)	SHA
SSL_DH_anon_WITH_RC4_128_MD5	DH anon	RC4 (128)	MD5
SSL_DH_anon_WITH_DES_CBC_SHA	DH anon	DES (56)	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA	RC4 (40)	MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA	DES40 (40)	SHA

表 9-2 Oracle Advanced Security 10g でサポートされている暗号スイート (続き)

暗号スイート	認証	暗号化	データ整合性
SSL_RSA_WITH_AES_128_CBC_SHA	RSA	AES (128)	SHA
SSL_RSA_WITH_AES_256_CBC_SHA	RSA	AES (256)	SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBS_SHA	DH DSS	DES (40)	SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	DH DSS	DES (50)	SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DH DSS	3DES (168)	SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	DH RSA	DES (40)	SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	DH RSA	DES (56)	SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DH RSA	3DES (168)	SHA
SSL_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA	DH DSS	DES (40)	SHA
SSL_DHE_DSS_WITH_RC4_128_SHA	DH DSS	RC4 (128)	SHA
SSL_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA	DH DSS	RC4 (56)	SHA

SSLEngine

SSL プロトコル・エンジンの使用を切り替えます。通常は <VirtualHost> セクションの中で使用し、特定の仮想ホストに対して SSL を有効にします。デフォルトでは、SSL プロトコル・エンジンは、メイン・サーバーとすべての構成済仮想ホストの両方で無効にされています。例 9-1 に、SSLEngine ディレクティブの使用例を示します。デフォルトの SSL は、4443 (UNIX) および 443 (Windows) です。

例 9-1 SSLEngine ディレクティブの使用

```
<VirtualHost_dafault_:4443>
  SSLEngine on
  ...
</VirtualHost>
```

カテゴリ	値
構文	SSLEngine on off
デフォルト	SSLEngine off
コンテキスト	サーバー構成、仮想ホスト

SSLLog

SSL エンジンのログ・ファイルが書き込まれる場所を指定します (エラー・メッセージは、[ErrorLog](#) ディレクティブにより指定された Oracle HTTP Server 標準ログ・ファイルにも重複して書き込まれます)。

このファイルは、シンボリック・リンク攻撃に使用されないように、ルートのみが書き込める場所に配置します。このファイル名の先頭にスラッシュ (/) がない場合は、[ServerRoot](#) への相対ファイル名とみなされます。ファイル名の先頭に縦線 (|) がある場合は、縦線に続く文字列が、信頼できるパイプを確立できる実行可能プログラムへのパスと想定されます。

このディレクティブは、1 つの仮想サーバー構成につき 1 回のみ使用できます。

カテゴリ	値
構文	SSLVerifyClient <i>path/to/filename</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLLogLevel

SSL エンジン・ログ・ファイルの冗長性レベルを指定します。

カテゴリ	値
有効値	<p>レベルは次のとおりです（昇順に記述されていて、各レベルは1つ前のレベルに含まれます）。</p> <ul style="list-style-type: none"> ■ none: 専用の SSL ログは記録されません。タイプが 'error' のメッセージは、ErrorLog ディレクティブにより指定された HTTP サーバー標準ログ・ファイルに重複して書き込まれます。 ■ error: タイプが 'error'（処理を停止する状態）のメッセージのみがログに記録されます。 ■ warn: 致命的ではない問題（処理を停止しない状態）を通知するメッセージがログに記録されます。 ■ info: 主要な処理アクションを要約したメッセージがログに記録されます。 ■ trace: 重要度の低い処理アクションを要約したメッセージがログに記録されます。 ■ debug: 開発操作と低レベル I/O 操作を要約したメッセージがログに記録されます。
構文	SSLLogLevel <i>level</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLMutex

Oracle HTTP Server プロセス間で同期化する必要がある操作の、SSL エンジンによる相互排他を行うために使用するセマフォ（ロック）のタイプです。

カテゴリ	値
有効値	<ul style="list-style-type: none"> ■ none: mutex は使用されません。mutex により SSL セッション・キャッシュへの書き込みアクセスが同期化されるため、この設定はお薦めしません。mutex を構成しない場合、セッション・キャッシュが不整合になります。 ■ file: <i>path/to/mutex</i>: ロック用のファイルを使用します。ファイル名が確実に一意になるように、Oracle HTTP Server の親プロセスのプロセス ID (PID) が、ファイル名に付加されます。ファイル名の先頭にスラッシュ (/) がない場合は、ServerRoot への相対ファイル名とみなされます。この設定は、Windows では使用できません。 ■ sem: 書き込みの同期化にオペレーティング・システムのセマフォを使用します。UNIX では Sys V IPC セマフォ、Windows では Windows Mutex が使用されます。オペレーティング・システムがサポートしている場合は、これが最善の選択肢です。
例	SSLMutex file: /usr/local/apache/logs/ssl_mutex
構文	SSLMutex <i>type</i>
デフォルト	SSLMutex none

カテゴリ	値
コンテキスト	サーバー構成

SSLOptions

ディレクトリ単位で様々なランタイム・オプションを制御します。一般に、1つのディレクトリに複数のオプションが適用される場合は、より包括的なオプションが適用されます（オプションはマージされません）。ただし、SSLOptions ディレクティブのすべてのオプションの前にプラス (+) またはマイナス (-) 符号が付いている場合は、オプションがマージされます。プラスが前に付いているオプションは、現在有効なオプションに追加され、マイナスが前に付いているオプションは、現在有効なオプションから除外されます。

カテゴリ	値
有効値	<ul style="list-style-type: none"> ■ StdEnvVars: SSL に関連する CGI/SSI 環境変数の標準セットを作成します。抽出操作は CPU 時間が長くなること、また静的コンテンツを提供するときには一般に適用されないことから、これはデフォルトでは無効になっています。一般に、この値は CGI/SSI リクエストの場合にのみ有効にします。 ■ ExportCertData: 次の追加 CGI/SSI 変数を有効にします。 SSL_SERVER_CERT SSL_CLIENT_CERT SSL_CLIENT_CERT_CHAIN_n (where n= 0, 1, 2...) これらの変数には、現在の HTTPS 接続のサーバーおよびクライアント用に Privacy Enhanced Mail (PEM) でエンコードされた X.509 証明書が含まれています。CGI スクリプトではこれを使用して、より詳しい証明書チェックを行うことができます。クライアント証明書連鎖の他の証明書がすべて提供されます。このオプションを使用するとパフォーマンスに時間がかかるため、デフォルトでは「Off」になっています。 SSL_CLIENT_CERT_CHAIN_n 変数の順序は、次のようになります。SSL_CLIENT_CERT_CHAIN_0 は、SSL_CLIENT_CERT を署名する中間的な CA です。SSL_CLIENT_CERT_CHAIN_1 は、SSL_CLIENT_ROOT_CERT をルート CA として、SSL_CLIENT_CERT_CHAIN_0 などを署名する中間的な CA です。 ■ FakeBasicAuth: クライアントの X.509 証明書の対象識別名を HTTP Basic 認証のユーザー名に変換します。これは、標準の HTTP サーバー認証方式がアクセス制御に使用できることを意味します。ユーザーからはパスワードが取得されず、文字列 'password' が置き換えられることに注意してください。 ■ StrictRequire: SSLRequireSSL またはディレクティブに従ってアクセスを禁止する必要がある場合にアクセスを拒否します。StrictRequire を指定しないと、'Satisfy any' ディレクティブ設定が SSLRequire または SSLRequireSSL ディレクティブをオーバーライドして、クライアントがホスト制約を渡した場合または有効なユーザー名とパスワードを指定した場合にアクセスが許可されてしまう可能性があります。 このように、SSLRequireSSL または SSLRequire を SSLOptions +StrictRequire と組み合わせることで、mod_oss1 はあらゆる場合に 'Satisfy any' ディレクティブをオーバーライドできます。 ■ CompatEnvVars: Apache SSL 1.x、mod_ssl 2.0.x、Sioux 1.0 および Stronghold 2.x との下位互換性のために、廃止された環境変数をエクスポートします。これは、既存の CGI スクリプトに対する互換性を提供するために使用します。

カテゴリ	値
	<ul style="list-style-type: none"> OptRenegotiate: SSL のディレクティブがディレクトリ単位のコンテキストで使用されるときに、最適化された SSL 接続再ネゴシエーション処理を有効化します。
構文	SSLOptions [+] <i>option</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト、ディレクトリ

SSLPassPhraseDialog

Wallet アクセス用のパスフレーズ・ダイアログのタイプ。mod_oss1 では、Wallet にアクセスするために管理者にパスフレーズの入力が必要されます。

カテゴリ	値
有効値	<ul style="list-style-type: none"> builtin: サーバー起動時に、mod_oss1 は各 Wallet のパスワードを求めるプロンプトを表示します。 この設定は、Oracle HTTP Server が OPMN により管理されている場合は使用できません。Oracle HTTP Server が OPMN により起動される場合、ユーザー操作は許可されません。 exec:path/to/program: サーバー起動時に、mod_oss1 は各 Wallet 用に構成されている外部プログラムをコールします。このプログラムは、servername:portnumber と、RSA または DSA という 2 つの引数を使用して起動されます。
構文	SSLPassPhraseDialog <i>type</i>
例	SSLPassPhraseDialog exec:/usr/local/apache/sbin/pfilter
デフォルト	SSLPassPhraseDialog builtin
コンテキスト	サーバー構成

SSLProtocol

mod_oss1 がサーバー環境を設定するとき使用する SSL プロトコルを指定します。クライアントは、指定されたプロトコルのいずれかでのみ接続できます。

カテゴリ	値
有効値	SSLv2、SSLv3、TLSv1、ALL
例	SSL バージョン 3.0 のみを指定するには、このディレクティブを次のように設定します。 SSLProtocol +SSLv3
構文	SSLProtocol [+] <i>protocol</i>
デフォルト	SSLProtocol ALL
コンテキスト	サーバー構成、仮想ホスト

SSLRequire

任意の複合ブール式が TRUE でないかぎり、アクセスを拒否します。

カテゴリ	値
構文	SSLRequire <i>expression</i>
デフォルト	なし
コンテキスト	ディレクトリ

この *expression* は、次の構文と一致する必要があります (BNF 構文表記として示してあります)。

```

expr ::= "true" | "false"
      "!" expr
      expr "&&" expr
      expr "||" expr
      "(" expr ")"

comp ::= word "==" word | word "eq" word
      word "!=" word | word "ne" word
      word "<" word | word "lt" word
      word "<=" word | word "le" word
      word ">" word | word "gt" word
      word ">=" word | word "ge" word
      word "=~" regex
      word "!~" regex
wordlist ::= word
          wordlist "," word

word ::= digit
       cstring
       variable
       function

digit ::= [0-9]+

cstring ::= "..."

variable ::= "%{varname}"

```

表 9-3 と表 9-4 に標準変数と SSL 変数を示します。これらの値が *varname* の有効値です。

```
function ::= funcname "(" funcargs ")"
```

funcname には、次の関数を使用できます。

```
file(filename)
```

file 関数は文字列引数 (ファイル名) を 1 つ取り、そのファイルの内容に拡張します。これは、正規表現に照らしてファイルの内容を評価する場合に役立ちます。

表 9-3 には、[SSLRequire](#) の *varname* の標準変数を示します。

表 9-3 SSLRequire の varname の標準変数

標準変数	標準変数	標準変数
HTTP_USER_AGENT	PATH_INFO	AUTH_TYPE
HTTP_REFERER	QUERY_STRING	SERVER_SOFTWARE
HTTP_COOKIE	REMOTE_HOST	API_VERSION
HTTP_FORWARDED	REMOTE_IDENT	TIME_YEAR

表 9-3 SSLRequire の varname の標準変数 (続き)

標準変数	標準変数	標準変数
HTTP_HOST	IS_SUBREQ	TIME_MON
HTTP_PROXY_CONNECTION	DOCUMENT_ROOT	TIME_DAY
HTTP_ACCEPT	SERVER_ADMIN	TIME_HOUR
HTTP:headername	SERVER_NAME	TIME_MIN
THE_REQUEST	SERVER_PORT	TIME_SEC
REQUEST_METHOD	SERVER_PROTOCOL	TIME_WDAY
REQUEST_SCHEME	REMOTE_ADDR	TIME
REQUEST_URI	REMOTE_USER	ENV:variablename
REQUEST_FILENAME		

表 9-4 には、SSLRequire の varname の SSL 変数を示します。

表 9-4 SSLRequire の varname の SSL 変数

SSL 変数	SSL 変数	SSL 変数
HTTPS	SSL_PROTOCOL	SSL_CIPHER_ALGKEYSIZE
SSL_CIPHER	SSL_CIPHER_EXPORT	SSL_VERSION_INTERFACE
SSL_CIPHER_USEKEYSIZE	SSL_VERSION_LIBRARY	SSL_SESSION_ID
SSL_CLIENT_V_END	SSL_CLIENT_M_SERIAL	SSL_CLIENT_V_START
SSL_CLIENT_S_DN_ST	SSL_CLIENT_S_DN	SSL_CLIENT_S_DN_C
SSL_CLIENT_S_DN_CN	SSL_CLIENT_S_DN_O	SSL_CLIENT_S_DN_OU
SSL_CLIENT_S_DN_G	SSL_CLIENT_S_DN_T	SSL_CLIENT_S_DN_I
SSL_CLIENT_S_DN_UID	SSL_CLIENT_S_DN_S	SSL_CLIENT_S_DN_D
SSL_CLIENT_I_DN_C	SSL_CLIENT_S_DN_Email	SSL_CLIENT_I_DN
SSL_CLIENT_I_DN_O	SSL_CLIENT_I_DN_ST	SSL_CLIENT_I_DN_L
SSL_CLIENT_I_DN_T	SSL_CLIENT_I_DN_OU	SSL_CLIENT_I_DN_CN
SSL_CLIENT_I_DN_S	SSL_CLIENT_I_DN_I	SSL_CLIENT_I_DN_G
SSL_CLIENT_I_DN_Email	SSL_CLIENT_I_DN_D	SSL_CLIENT_I_DN_UID
SSL_CLIENT_CERT	SSL_CLIENT_CERT_CHAIN_n	SSL_CLIENT_ROOT_CERT
SSL_CLIENT_VERIFY	SSL_CLIENT_M_VERSION	SSL_SERVER_M_VERSION
SSL_SERVER_V_START	SSL_SERVER_V_END	SSL_SERVER_M_SERIAL
SSL_SERVER_S_DN_C	SSL_SERVER_S_DN_ST	SSL_SERVER_S_DN
SSL_SERVER_S_DN_OU	SSL_SERVER_S_DN_CN	SSL_SERVER_S_DN_O
SSL_SERVER_S_DN_I	SSL_SERVER_S_DN_G	SSL_SERVER_S_DN_T
SSL_SERVER_S_DN_D	SSL_SERVER_S_DN_UID	SSL_SERVER_S_DN_S
SSL_SERVER_I_DN	SSL_SERVER_I_DN_C	SSL_SERVER_S_DN_Email
SSL_SERVER_I_DN_L	SSL_SERVER_I_DN_O	SSL_SERVER_I_DN_ST
SSL_SERVER_I_DN_CN	SSL_SERVER_I_DN_T	SSL_SERVER_I_DN_OU

表 9-4 SSLRequire の varname の SSL 変数 (続き)

SSL 変数	SSL 変数	SSL 変数
SSL_SERVER_I_DN_G	SSL_SERVER_I_DN_I	

SSLRequireSSL

SSL を使用していないクライアントに対してアクセスを拒否します。構成エラーにより、セキュリティが弱い可能性がある SSL 対応の仮想ホストまたはディレクトリの完全保護に役立つディレクティブです。

カテゴリ	値
構文	SSLRequireSSL
デフォルト	なし
コンテキスト	ディレクトリ

SSLSessionCache

グローバル・セッションまたはプロセス間セッションのセッション・キャッシュ・ストレージ・タイプを指定します。キャッシュは、リクエストの平行処理を高速化するオプションの方法を提供します。

カテゴリ	値
有効値	<ul style="list-style-type: none"> ■ none: グローバルまたはプロセス間セッション・キャッシュを無効にします。機能に対する影響はありませんが、パフォーマンスに大きな差が出ます。 ■ shmht: /path/to/datafile[bytes]: RAM 内の共有メモリー・セグメントの中で、高パフォーマンス・ハッシュテーブルを使用します (bytes はおおよそのサイズを指定)。このテーブルは、path/to/datafile により設定されます。このハッシュテーブルは、サーバー・プロセスのローカル SSL メモリー・キャッシュと同期化します。 ■ shmcb: /path/to/datafile[bytes]: 高パフォーマンスの Shared Memory Cyclic Buffer (SHMCB) セッション・キャッシュを使用して、サーバー・プロセスのローカル SSL メモリー・キャッシュと同期化します。shmcb のパフォーマンスのほうが、shmht に比べて、すべての環境で均一です。
構文	SSLSessionCache type
例	<pre>SSLSessionCache shmht: /ORACLE_ HOME/Apache/Apache/logs/ssl_scache (512000) SSLSessionCache shmcb: /ORACLE_ HOME/Apache/Apache/logs/ssl_scache (512000)</pre>
デフォルト	SSLSessionCache none

SSLSessionCacheTimeout

セッション・キャッシュ内で SSL セッションの有効期限が満了になるまでの秒数を指定します。

カテゴリ	値
構文	SSLSessionCacheTimeout <i>seconds</i>
デフォルト	300
コンテキスト	サーバー構成

SSLVerifyClient

接続時にクライアントが証明書を提示する必要があるかどうかを指定します。

カテゴリ	値
有効値	<ul style="list-style-type: none"> ■ none: クライアント証明書は不要です。 ■ optional: クライアントは有効な証明書を提示できます。 ■ require: クライアントは有効な証明書を提示する必要があります。
構文	SSLVerifyClient <i>level</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

注意: mod_ssl に含まれているレベル optional_no_ca (クライアントは有効な証明書を提示できるが、証明書が検証可能である必要はない) は、mod_oss1 ではサポートされていません。

SSLWallet

WRL を使用して Wallet の位置を指定します。

カテゴリ	値
構文	SSLWallet <i>wrl</i> <i>wrl</i> の形式は、 <i>file:path to wallet</i> です。
例	SSLWallet <i>file:/etc/ORACLE/WALLETS/server</i> その他、Oracle SSL 製品で使用可能な <i>wrl</i> の値も使用できます。
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLWalletPassword

同一コンテキスト内に指定されている Wallet のアクセスに必要な Wallet パスワードを指定します。**クリアテキスト**の Wallet パスワードか不明瞭化されたパスワードのいずれかを選択できます。不明瞭化されたパスワードは、コマンドライン・ツール **iasobf** を使用して作成されます。通常の Wallet を使用する必要がある場合は、クリアテキスト・パスワードではなく不明瞭化されたパスワードの使用をお勧めします。

関連項目: 9-16 ページの「[iasobf ユーティリティの使用](#)」

カテゴリ	値
構文	SSLWalletPassword password パスワードが不要の場合は、このディレクティブを設定しないでください。 注意 : Oracle Wallet Manager の自動ログオン機能を使用して作成された Wallet が使用される場合、Wallet にはパスワードが不要なため、このディレクティブは設定しないでください。
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

注意： SSLWalletPassword は使用されていません。このディレクティブを使用すると、Oracle HTTP Server ログに警告メッセージが生成されます。Wallet を保護するために、自動ログインを有効にして、かわりに SSO Wallet を取得することをお勧めします。9-2 ページの「[タスク 1: 実際の Wallet の作成](#)」を参照してください。

iasobf ユーティリティの使用

iasobf ユーティリティを使用すると、不明瞭化されたパスワードを Wallet の [クリアテキスト](#)・パスワードから生成できます。

自動ログオンを使用可能にして作成された Oracle Wallet (SSO Wallet) を使用している場合は、このユーティリティを使用する必要はありません。ただし、パスワード付きの標準の Wallet を使用する必要がある場合は、ORACLE_HOME/Apache/Apache/bin にあるパスワード不明瞭化ツール iasobf を使用して、クリアテキストのパスワードから不明瞭化された Wallet パスワードを生成することをお勧めします。

不明瞭化された Wallet パスワードを生成するためのコマンド構文は、次のとおりです。

```
iasobf -p password
```

不明瞭化されたパスワードは端末に出力されます。iasobf には、httpd プロセスのオペレーティング・システム・ユーザーが必要です。したがって、UNIX の場合は root 引数、Windows の場合は system 引数を使用します。たとえば、UNIX では、コマンドは iasobf -password root となります。

注意： Windows 環境での対応ツールは opensslpassword と呼ばれ、iasobf と同じ方法で使用できます。

mod_proxy のディレクティブの使用

次のディレクティブは、[mod_proxy](#) サポート専用です。

- [SSLProxyCache](#)
- [SSLProxyCipherSuite](#)
- [SSLProxyProtocol](#)
- [SSLProxyWallet](#)
- [SSLProxyWalletPassword](#)

SSLProxyCache

プロキシ・キャッシュを使用するかどうかを指定します。プロキシは、SSL サーバーが使用するのと同じセッションを使用します。

カテゴリ	値
構文	SSLProxyCache <i>on/off</i>
デフォルト	SSLProxyCache <i>off</i>
コンテキスト	サーバー構成、仮想ホスト

SSLProxyCipherSuite

プロキシ・サーバーの暗号スイートを指定します。

カテゴリ	値
構文	SSLCipherSuite <i>cipher-spec</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLProxyProtocol

プロキシ・サーバーの SSL プロトコルを制御します。

カテゴリ	値
構文	SSLProxyProtocol [<i>+-</i>] <i>protocol</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLProxyWallet

プロキシ接続のオープン時に使用する証明書を含む、Wallet の位置を指定します。

カテゴリ	値
構文	SSLProxyWallet <i>wrl</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

SSLProxyWalletPassword

プロキシの Wallet パスワードを指定します。

カテゴリ	値
構文	SSLProxyWalletPassword <i>password</i>
デフォルト	なし
コンテキスト	サーバー構成、仮想ホスト

注意： SSLProxyWalletPassword は使用されていません。このディレクティブを使用すると、Oracle HTTP Server ログに警告メッセージが生成されます。Wallet を保護するために、かわりに SSO Wallet を取得することをお勧めします。

mod_oc4j を使用したロード・バランシング

この付録では、メトリック・ベースのロード・バランシングも含め、mod_oc4j によるロード・バランシングについて説明します。内容は、次のとおりです。

- [ロード・バランシング・ポリシー](#)
- [ロード・バランシング・パラメータ](#)
- [メトリック・ベースのロード・バランシング](#)

ロード・バランシング・ポリシー

この項では、`mod_oc4j` でサポートされているロード・バランシング・ポリシーについて説明します。

- ランダム
- ラウンドロビン
- ローカル・アフィニティを考慮したランダム
- ローカル・アフィニティを考慮したラウンドロビン
- ルーティングの重みを使用したランダム
- ルーティングの重みを使用したラウンドロビン
- メトリック・ベース
- ローカル・アフィニティを考慮したメトリック・ベース

ランダム

`mod_oc4j` は、リクエストを処理する候補である、OC4J インスタンスのリストから OC4J インスタンスをランダムに選択します。

ラウンドロビン

`mod_oc4j` は、リクエストを処理する候補である、OC4J インスタンスの順序付きリストからランダムに OC4J インスタンスを選択します。他の OC4J インスタンスは、最初に選択されたサーバーが再度選択されるまで、順序付きリストから順番に選択されます。この順序が繰り返されます。特定の OC4J インスタンスが停止されるか使用不可になると、そのインスタンスは稼働状態に戻るまでスキップされます（選択されません）。

ローカル・アフィニティを考慮したランダム

`mod_oc4j` は、リクエストを処理するローカル OC4J プロセスをランダムに選択します。使用可能なローカル OC4J プロセスがない場合、`mod_oc4j` はリモート OC4J プロセスをランダムに選択し、選択の機会を均等にします。

ローカル・アフィニティを考慮したラウンドロビン

`mod_oc4j` は、すべてのリクエストをローカル OC4J プロセスにラウンドロビン方式でルーティングします。使用可能なローカル・プロセスがない場合、`mod_oc4j` はリクエストを様々なホスト上の各 OC4J プロセスに均等にルーティングします。

ルーティングの重みを使用したランダム

`mod_oc4j` は、各ホストに対して構成されているルーティングの重みに従ってリクエストを配分します。そのホスト上の OC4J プロセスから 1 つがランダムに選択されます。

ルーティングの重みを使用したラウンドロビン

`mod_oc4j` は、各ホストに対して構成されているルーティングの重みに基づいて、リクエスト・ロードの合計を各ホスト上の OC4J プロセスに配分します。そのホスト上の OC4J プロセスの中からラウンドロビン方式で 1 つ選択されます。

メトリック・ベース

`mod_oc4j` は、OC4J プロセスに対して発生可能なロードの量を示す、OC4J プロセスからのラウンドタイム・メトリックに基づいて、リクエストをルーティングします。

ローカル・アフィニティを考慮したメトリック・ベース

mod_oc4j は、OC4J プロセスのランタイム・パフォーマンス・メトリックに基づいて、すべてのリクエストをローカル OC4J プロセスにルーティングします。使用可能なローカル OC4J プロセスがない場合、mod_oc4j はパフォーマンス・メトリックのみに従って異なるホスト上の各 OC4J プロセスにリクエストをルーティングします。

ロード・バランシング・パラメータ

この項では、次のロード・バランシング・パラメータについて説明します。

- [Oc4jSelectMethod](#)
- [Oc4jRoutingWeight](#)

Oc4jSelectMethod

ロード・バランシング用の OC4J インスタンスを選択します。

カテゴリ	値
構文	Oc4jSelectMethod roundrobin roundrobin:local roundrobin:weighted random random:local random:weighted metric metric:local
必須	なし
デフォルト	Oc4jSelectMethod を指定しないと、Oc4jSelectMethod roundrobin がデフォルトになります。
例	Oc4jSelectMethod random:local Oc4jSelectMethod metric
使用方法	<ul style="list-style-type: none"> ■ Oc4jSelectMethod random: 「ランダム」ロード・バランシング・ポリシーに従って OC4J プロセスを選択します。 ■ Oc4jSelectMethod roundrobin:weighted: 「ルーティングの重みを使用したラウンドロビン」ロード・バランシング・ポリシーに従って OC4J プロセスを選択します。 ■ Oc4jSelectMethod metric:local: 「ローカル・アフィニティを考慮したメトリック・ベース」ロード・バランシング・ポリシーに従って OC4J プロセスを選択します。

このディレクティブは、Oracle Application Server 10g リリース 2 (10.1.2) のベース・サーバーに対してのみ適用され、VirtualHost コンテナ内で指定された場合は、起動時にエラーが出力されます。

Oc4jRoutingWeight

ロード・バランシング中に、リクエストのルーティングの重みを各マシンに関連付けます。重み付けされたルーティングとは、予測されるロード処理能力に基づいて各マシンに割り当てられた、事前定義済の値に従ってリクエストを分散するロード・バランシング手段です。

カテゴリ	値
構文	Oc4jRoutingWeight <node_name> <routing_weight>
必須	なし
デフォルト	ルーティングの重みが 1 である、全ノード上の OC4J プロセスがデフォルトです。Oc4jRoutingWeight が指定され、一部のホストが指定されていない場合は、ルーティングの重みが 1 である非指定ノード上の OC4J プロセスがデフォルトになります。

カテゴリ	値
例	<ul style="list-style-type: none"> Oracle Application Server クラスタに2つのホスト Host_A および Host_B があります。各ホストでは、Oracle HTTP Server および OC4J プロセスが実行されています。 <code>Oc4jSelectMethod random:local</code> <code>Oc4jRoutingWeight Host_A 3</code> <code>Oc4jRoutingWeight Host_B 2</code> Oc4jRoutingWeight ディレクティブは無視されます。Host_A の mod_oc4j はすべてのリクエストを Host_A の OC4J プロセスにランダムにルーティングし、Host_B の mod_oc4j はすべてのリクエストを Host_B の OC4J プロセスにランダムにルーティングします。 Oracle Application Server クラスタに4つのホスト Host_A、Host_B、Host_C および Host_D があります。各ホストでは、Oracle HTTP Server および OC4J プロセスが実行されています。 <code>Oc4jSelectMethod roundrobin:weighted</code> <code>Oc4jRoutingWeight Host_A 3</code> <code>Oc4jRoutingWeight Host_B 2</code> すべてのマシン上の mod_oc4j はラウンドロビン方式に従って、Host_A で実行中の OC4J プロセスにリクエスト数の3倍をルーティングし、Host_B でリクエスト数の2倍をルーティングし、Host_C でリクエスト数をルーティングし、Host_D でリクエスト数をルーティングします。 Oracle Application Server クラスタに4つのホスト Host_A、Host_B、Host_C および Host_D があります。各ホストでは、Oracle HTTP Server および OC4J プロセスが実行されています。 <code>Oc4jSelectMethod roundrobin:weighted</code> すべてのマシン上の mod_oc4j は、Host_A、Host_B、Host_C および Host_D 上の OC4J プロセスにラウンドロビン方式でリクエストを均等にルーティングします。
使用方法	<p>Oc4jRoutingWeight は、Oc4jSelectMethod で重み付けが指定された場合にのみ考慮されます。</p> <p>Oc4jRoutingWeight <node_name> <routing_weight> は、各ノードにリクエストのルーティングの重みを関連付けます。node_name は、ホスト名形式または IP アドレス形式です。複数のインタフェースを持つホストで異なるインタフェースが指定された場合は、異なるホストとみなされます。</p>

このディレクティブは、Oracle Application Server 10g リリース 2 (10.1.2) のベース・サーバーに対してのみ適用され、VirtualHost コンテナ内で指定された場合は、起動時にエラーが出力されます。

メトリック・ベースのロード・バランシング

メトリック・ベースのロード・バランシングとは、各 OC4J から報告される負荷状態メトリックに基づいて、OC4J 間でリクエスト・ロードを分散する方法です。メトリックの範囲は 0 ~ 100 です。0 は非常にビジーな（過負荷の）状態、100 はビジーではない（負荷のかかっていない）状態を示します。メトリック・ベースのロード・バランシングが有効になっている場合、個々の OC4J から受信したメトリックをすべての OC4J から受信したメトリックの合計で割った比率に基づいて、OC4J 間でリクエストが分散されます。

たとえば、OC4J プロセス p1 がメトリック 20 を、プロセス p2 がメトリック 40 を、プロセス p3 がメトリック 90 を報告したとします。リクエストは次のように分散されます。

- p1 には、リクエスト 150 件当たり 20 件（13%）がルーティングされます。
- p2 には、リクエスト 150 件当たり 40 件（27%）がルーティングされます。
- p3 には、リクエスト 150 件当たり 90 件（60%）がルーティングされます。

メトリック・ベースのロード・バランシングを有効化するには、Oracle HTTP Server と OC4J を構成する必要があります。次の各項で、必要な構成について説明します。

- Oracle HTTP Server の構成
- OC4J の構成

Oracle HTTP Server の構成

Oracle HTTP Server 側で、`mod_oc4j.conf` に `Oc4jSelectMethod metric` または `Oc4jSelectMethod metric:local` を指定します。

関連項目：

- A-3 ページの「`Oc4jSelectMethod`」
- A-2 ページの「メトリック・ベース」
- A-3 ページの「ローカル・アフィニティを考慮したメトリック・ベース」

OC4J の構成

OC4J 側では、UNIX の場合は `ORACLE_HOME/j2ee/home/config/server.xml` に、Windows の場合は `ORACLE_HOME\j2ee\home\config\server.xml` に、メトリック・コレクタを構成する必要があります。`<metric-collector>` 要素を構成することで、`mod_oc4j` へのメトリック送信を開始するよう OC4J に指示すると、`mod_oc4j` は使用可能な OC4J インスタンスに受信リクエストを分散してロード・バランシングを行うためのルーティングの決定を下せるようになります。

OC4J から `mod_oc4j` に送信されるメトリックは、メトリック・ベースのロード・バランシングが `mod_oc4j` に対して指定されており、かつ OC4J が Oracle Application Server 環境で実行されている場合にのみ使用されます。

`mod_oc4j` にメトリック・ベースのロード・バランシングを指定し、`server.xml` に `<metric-collector>` 要素を指定しない場合、`mod_oc4j` は OC4J からメトリックが送信されることを期待しますが、OC4J はメトリックを送信しません。この場合、`mod_oc4j` は次の警告メッセージを發します。

```
No run time metrics for oc4j(opmid=%s) in notification Oc4jSelectMethod is
configured to use run time metrics, please make sure OC4J side is configured
accordingly. Default to 50.
```

この場合、`mod_oc4j` は各 OC4J プロセスに対して値 50 を使用し、続行します。

同様に、`server.xml` に `<metric-collector>` 要素を指定し、`mod_oc4j` にメトリック・ベースのロード・バランシングを指定しない場合、OC4J はメトリックを送信しますが、`mod_oc4j` はメトリックを受信するように構成されていません。この場合、`mod_oc4j` はメトリックを無視し、構成されているロード・バランシング方式が何であろうとそれを使用します。ロード・バランシング方式は `Oc4jSelectMethod` で指定します。`Oc4jSelectMethod` が指定されていない場合、`mod_oc4j` はデフォルトのラウンドロビン方式を使用します。

この Oracle HTTP Server インスタンスで使用される OC4J は、すべて同じ構成になるようにしてください。そうしないと、OC4J から返された数を比較できず、ロード・バランシングの成果がほとんど得られない可能性もあります。

`mod_oc4j` がメトリック情報を含む通知を OC4J から受信すると、リクエストのルーティング処理がただちに變更されます。OC4J からのデフォルトの通知間隔は 30 秒です。この値は、システム・プロパティ `opmnPingInterval` を使用して構成できます。このプロパティは、OC4J が OPMN によって起動されるときに、コマンドライン上で渡されます。通知間隔を變更するには、`opmn.xml` の OC4J `<process-set>` 構成要素の下に、次のように指定します。

```
<module-data>
  <category id="start-parameters">
    <data id="java-options" value="-DopmnPingInterval=<new ping interval value>"/>
  </category>
</module-data>
```

関連資料：『Oracle Process Manager and Notification Server 管理者ガイド』

OC4Jのメトリックの指定

次の2つの方法でOC4Jのメトリックを指定できます。

- [DMSMetricCollector](#) を使用するメトリック・ベースのロード・バランシングの構成
- [独自のメトリック・コレクタの作成](#)

DMSMetricCollector を使用するメトリック・ベースのロード・バランシングの構成

このデフォルトの方法では、<metric-collector> 要素は `classname` という単一の属性を取ります。この属性は、サーバー単位のメトリックを収集および計算するためのインタフェースを定義します。DMS ベースのメトリック・コレクタを使用する場合、`classname` 属性で `oracle.oc4j.server.DMSMetricCollector` を使用します。DMSMetricCollector インスタンスはいくつかのパラメータを取ります。

DMS メトリックは、次の構成例のとおり、`dms-noun` パラメータを使用して指定します。このメトリックは、DMSMetricCollector の計算のベースとなるものです。メトリック・ベースのロード・バランシングでは、DMS メトリック

`/oc4j/default/WEBS/processRequest.time` の使用をお勧めします。このメトリックは、デフォルトの Web アプリケーションにおけるサーブレットの処理時間を示します。

OC4J に送信される値は、現在の DMS メトリック値に基づいて計算された値と前回の値の送信時に計算された値の平均に重み付けしたものです。デフォルトの重みは、現在の値に対しては 0.7、前回の値に対しては 0.3 です。重みを変更するには、次の例に示すように `history-proportion` を設定します。その結果、現在の値に対する重みは 0.8 に、前回の値に対する重みは 0.2 に変更されています。

```
<metric-collector classname="oracle.oc4j.server.DMSMetricCollector">
  <init-param>
    <param-name>
      dms-noun
    </param-name>
    <param-value>
      /oc4j/default/WEBS/processRequest.time
    </param-value>
  </init-param>
  <init-param>
    <param-name>
      history-proportion
    </param-name>
    <param-value>
      0.2
    </param-value>
  </init-param>
  <init-param>
    <param-name>
      debug
    </param-name>
    <param-value>
      false
    </param-value>
  </init-param>
</metric-collector>
```

関連資料： DMS メトリックの一覧は、『Oracle Application Server パフォーマンス・ガイド』を参照してください。

DMS メトリックをメトリック・ベースのロード・バランシング用に交換する方法

`getMetrics()` がコールされると、`dms-noun` パラメータで指定した DMS メトリックの値が取得されます。前回の測定値の `delta` が計算されます。

スケールは 0 ~ 100 で、測定値は無限であるため、次の計算式が適用されます。

```
metric = 100 / (1 + (log (1 + delta)))
```

前述のとおり、このメトリックは現在の状況に一致するだけでなく、前回のメトリック履歴も反映している必要があります。新しいメトリックを作成するために、前回の履歴に 1/3 の重みを割り当て、今回収集された平均値に 2/3 の重みを割り当てることができます。

この新しいメトリックは次のメトリックにとっての履歴となり、考慮される度合いは時間がたつにつれて次第に小さくなります。構成例に示したように、`history-proportion` パラメータを 0.0 ~ 1.0 の浮動小数点値に設定することで、前回のメトリックを新しいメトリックに組み込む割合を指定できます。値が大きければ大きいほど履歴値が重要視され、メトリックの揮発性は低くなります。値が小さい場合は、最新のメトリックが反映されます。返されるメトリックは、次のとおりです。

```
smoothedmetric = ((1 - history-proportion) * metric) + (history-proportion * previousmetric)
```

`smoothedmetric` が、ロード・バランシング用に `mod_oc4j` に送信されます。

独自のメトリック・コレクタの作成

インタフェース `oracle.oc4j.api.MetricCollector` を実装すれば、`mod_oc4j` に独自のメトリックを送信することができます。メトリックは 0 ~ 100 の値であることが必要です。

メトリック・コレクタはすべて、インタフェース `oracle.oc4j.api.MetricCollector` を実装する必要があります。実際のメトリック・コレクタは、インスタンス化できるように、パラメータの指定されていないコンストラクタを必要とします。

この機能のスキーマ要素は、`server.xml` に次のように定義されています。

```
<metric-collector classname="my.package.name.MyClassName">
  <init-param>
    <param-name>
      mysetting
    </param-name>
    <param-value>
      12345
    </param-value>
  </init-param>
</metric-collector>
```

前の例のとおり、0 個以上のパラメータをメトリック・コレクタに対して設定でき、`setParameters()` メソッドがコールされると、設定したパラメータがメトリック・コレクタに渡されます。`setParameters()` の後、`setEnabled(true)` が一度コールされます。このメソッドは、必要に応じてデータの収集を開始できることをメトリック・コレクタに知らせます。

注意： カスタム・メトリック・コレクタがスレッドを開始する場合、そのスレッドはデーモン・スレッドであることが必要です。デーモン・スレッドでないと、サーバーを正常に停止できない可能性があります。

`oracle.oc4j.api.MetricCollector` の実装後、メトリックを `jar` ファイルにパッケージし、次のコマンドを使用して `server.xml` のライブラリ・パスに追加します。

```
<library path="<path to>/mymetric.jar"/>
```

oracle.oc4j.api.MetricCollector 次に、`oracle.oc4j.api.MetricCollector` インタフェースを示します。

```
package oracle.oc4j.api;

import java.util.Map;

/**
 * Defines an interface for gathering and obtaining a server-wide metric.
 * The metric is used in iAS mode, by mod_oc4j, to load balance between
 * virtual oc4j instances.
 * The metric value is relative, and should be between 0 and 100, both
 * inclusive.
 * When configured for metric load balancing,
 * Mod_oc4j will route preferably to an oc4j with the greater value.
 * <p>Concrete instances of this class must have a public empty constructor in
 * order to be loaded and instantiated.
 */
public interface MetricCollector {
    /**
     * Support for debugging: This is a property name to set to true in order
     * to display the metric that is sent from the server
     */
    String OC4J_METRIC_DEBUG_PROPERTY = "oc4j.metric.debug";

    /**
     * Debugging flag that depends on @link #OC4J_METRIC_DEBUG_PROPERTY
     */
    boolean DEBUG = Boolean.getBoolean( OC4J_METRIC_DEBUG_PROPERTY );

    /**
     * Initial metric to return when no measurement has been made (property key)
     */
    String OC4J_INITIAL_METRIC_PROPERTY = "oc4j.metric.initial";
    /**
     * Initial metric to return when no measurement has been made.
     * Default is 50
     */
    int INITIAL_METRIC = Integer.getInteger( OC4J_INITIAL_METRIC_PROPERTY, 50
    ).intValue();

    /**
     * Enabled flag for the collector.
     * @return true if the collector is collecting data
     */
    boolean isEnabled();

    /**
     */
    void setEnabled( boolean enabled );

    /**
     * The parameters the metric collector is configured with.
     * This method will be called even when the set of parameters is null.
     * @param params the key/value pairs the metric collector is configured with,
     * or <code>null</code> if none
     */
    void setParameters( Map params );

    /**
     * @return a metric between 0 and 100, inclusive. 100 is better, 0 is worse
     */
    int getMetric();
}
```

構成ファイル

この付録では、一般に使用される Oracle HTTP Server 構成ファイルについて説明します。
内容は、次のとおりです。

- [dms.conf](#)
- [httpd.conf](#)
- [iaspt.conf](#)
- [mime.types](#)
- [mod_oc4j.conf](#)
- [mod_osso.conf](#)
- [opmn.xml](#)
- [oracle_apache.conf](#)
- [php.ini](#)
- [ssl.conf](#)

dms.conf

Oracle の Dynamic Monitoring Service (DMS) を使用してサイト・コンポーネントのパフォーマンスをモニターできます。

このファイルは次の場所にあります。

- UNIX の場合 : `ORACLE_HOME/ohs/conf`
- Windows の場合 : `ORACLE_HOME\ohs\conf`

関連資料: 『Oracle Application Server パフォーマンス・ガイド』

httpd.conf

これはサーバー構成ファイルであり、通常は、使用するユーザー ID とグループ ID、および他のファイルの位置など、サーバーの実行方法に影響するディレクティブが含まれています。サーバー構成ファイルはサーバーの起動に使用されるメイン・ファイルであるため、Oracle HTTP Server にはその位置を指定するディレクティブは含まれていません。位置は、サーバーの起動時にコマンドラインで渡されます。

このファイルは次の場所にあります。

- UNIX の場合 : `ORACLE_HOME/ohs/conf/httpd.conf`
- Windows の場合 : `ORACLE_HOME\ohs\conf\httpd.conf`

単一の構成ファイルを管理する方が容易であるため、`srm.conf` または `access.conf` ではなく、このファイルのみを使用する必要があります。

注意: `/home/your_directory/orahome` にインストールした Oracle Application Server が `/private/your_directory/orahome` にリンクされている場合、インストール済のファイルには `/home/your_directory/orahome` と `/private/your_directory/orahome` のどちらからもアクセスできます。

インストール後、元の Oracle ホーム・パスを使用した `dms.conf` ファイルのエントリが `httpd.conf` ファイルに作成されます。次に例を示します。

```
include /home/your_directory/orahome/Apache/Apache/conf/dms.conf
```

元の Oracle ホーム・パスを、リンク先の Oracle ホーム・パスに置き換えな
いでください。

httpd.conf のファイル構造

`httpd.conf` に次のセクションがあります。

- [Global Environment](#)
- [Main Server Configuration](#)
- [Virtual Hosts Parameters](#)

Global Environment

これは、`httpd.conf` ファイルのセクション 1 です。このセクションには、Oracle HTTP Server を扱う構成ディレクティブが含まれています。

関連項目:

- [3-3 ページの「ファイル位置の指定」](#)
- [4-2 ページの「プロセス数と接続数の構成」](#)
- [5-2 ページの「リスナー・ポートおよびアドレスの指定」](#)

Main Server Configuration

これは、httpd.conf ファイルのセクション2です。このセクションには、デフォルト・サーバーのディレクティブが含まれています。

関連項目： [3-2 ページの「サーバー機能と管理者機能の設定」](#)

Virtual Hosts Parameters

これは、httpd.conf ファイルのセクション3です。このセクションには仮想ホスト固有のパラメータが含まれており、メイン・サーバー構成のデフォルトの一部がオーバーライドされます。

iaspt.conf

ポート・トンネリング・プロセスを構成します。ポート・トンネリングを使用すると、Oracle HTTP Server と OC4J 間のすべての通信を1つまたは少数のポート上で行えます。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/iaspt/conf
- Windows の場合：ORACLE_HOME\iaspt\conf

関連項目： [8-7 ページの「ポート・トンネリングの概要」](#)

mime.types

特定のファイル拡張子についてクライアントに送信される MIME (Multi Internet media) タイプを制御します。クライアントがファイルのコンテンツの処理方法を認識できるように、クライアントに正しいメディア・タイプを送信することが重要です。mime.type ファイルにタイプを追加するか、構成ファイルに AddType ディレクティブを追加できます。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/ohs/conf
- Windows の場合：ORACLE_HOME\ohs\conf

関連項目： [7-8 ページの「mod_mime」](#)

mod_oc4j.conf

mod_oc4j モジュールを構成およびロードします。これは、デフォルトで有効です。リクエストを Oracle HTTP Server から OC4J にルーティングするため、このファイルにはルーティング情報が含まれています。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/ohs/conf
- Windows の場合：ORACLE_HOME\ohs\conf

関連項目： [7-8 ページの「mod_oc4j」](#)

mod_osso.conf

Oracle HTTP Server のシングル・サインオンを有効化する mod_osso を構成します。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/ohs/conf
- Windows の場合：ORACLE_HOME\ohs\conf

関連項目： [7-18 ページの「mod_osso」](#)

opmn.xml

Oracle Application Server 内で Oracle Process Manager and Notification Server (OPMN) により管理されるプロセスを記述します。

opmn.xml ファイルは、OPMN のメイン構成ファイルです。ONS、PM および Oracle Application Server コンポーネント固有の構成情報が含まれます。opmn.xml ファイルには、システム上で OPMN により管理されている Oracle Application Server コンポーネントが表示されます。このファイルには、Oracle Application Server コンポーネントのエントリが次の階層構造で配置されています。

```
<ias-component>
  <process-type>
    <process-set>
```

- **<ias-component>**: このエントリは Oracle Application Server コンポーネントを表します。起動や停止などのプロセスに対するコンポーネントの管理を有効化します。
- **<process-type>**: これは <ias-component> エントリのサブコンポーネントであり、特定の PM モジュールとの関連付けにより実行されるプロセスのタイプを宣言します。
- **<process-set>**: これは <ias-component> エントリのサブコンポーネントであり、Oracle Application Server コンポーネント用にオプションのランタイム引数と環境の様々なセットを宣言します。

opmn.xml は次の場所にあります。

- UNIX の場合：ORACLE_HOME/opmn/conf
- Windows の場合：ORACLE_HOME\opmn\conf

関連資料：『Oracle Process Manager and Notification Server 管理者ガイド』

oracle_apache.conf

サポート対象モジュールの構成ファイルを格納します。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/ohs/conf
- Windows の場合：ORACLE_HOME\ohs\conf

注意： Oracle Application Server の「インフラストラクチャ」インストール・タイプの場合は、oracle_apache.conf によりもう 1 つの構成ファイル oracle_ocm.conf がインクルードされます。この構成ファイルには、Oracle Application Server Certificate Authority の構成が含まれています。

php.ini

mod_php を構成します。このファイル名は PHP で検索されるため、変更しないでください。

このファイルは次の場所にあります。

- UNIX の場合：ORACLE_HOME/ohs/conf
- Windows の場合：ORACLE_HOME\ohs\conf

関連項目： [7-21 ページの「mod_php」](#)

ssl.conf

SSL 定義と仮想ホスト・コンテナが含まれています。これは、デフォルトでは無効です。

このファイルは次の場所にあります。

- UNIX の場合 : `ORACLE_HOME/ohs/conf`
- Windows の場合 : `ORACLE_HOME\ohs\conf`

よくある質問

この章では、Oracle HTTP Server に関してよくある質問とそれに対する回答について説明します。

関連資料： Apache Server マニュアルの「Frequently Asked Questions」

該当する場合は、Apache Software Foundation のマニュアルを参照しています。

アプリケーション固有のエラー・ページの作成

Oracle HTTP Server には、エラー処理用のデフォルトのコンテンツ・ハンドラが用意されています。ErrorDocument ディレクティブを使用すると、デフォルトを上書きできます。

関連資料： Apache Server マニュアルの「ErrorDocument directive」

ISP（仮想ホスト）の顧客に対する HTTPS の提供

HTTP の場合、Oracle HTTP Server では名前ベースおよび IP ベースという 2 種類の仮想ホストがサポートされます。HTTPS でサポートされるのは、IP ベースの仮想ホストのみです。

HTTP に IP ベースの仮想ホストを使用している場合、顧客の仮想サーバーは顧客別 IP アドレスのポート 80 でリスニングしています。このような顧客に HTTPS を提供するには、同じ顧客別 IP アドレスのポート 4443 でリスニングするユーザーごとに仮想ホストを追加し、[mod_ossll のディレクティブの使用](#)などの SSL ディレクティブを使用して顧客ごとの SSL 特性を指定します。各顧客が専用 Wallet とサーバー証明書を使用できることに注意してください。

HTTP に名前ベースの仮想ホストを使用している場合、顧客の仮想サーバーは共有 IP アドレスのポート 80 でリスニングしています。このような顧客に HTTPS を提供するには、共有 IP アドレスのポート 4443 でリスニングする共有 IP の仮想ホストを 1 つ追加します。すべての顧客は、Wallet や ISP のサーバー証明書などの SSL 構成を共有します。

関連項目： 4-4 ページの「root としての Oracle HTTP Server の実行」

異なる言語およびキャラクタ・セット・バージョンのドキュメントの使用

Apache サーバーの機能に与えられた汎用名である Multiviews を使用すると、リクエストに対するレスポンスで様々なバージョンの言語と文字固有のドキュメントを提供できます。

ファイアウォールの後方にある HTTP Server へのプロキシ依存のリクエストの送信

Cache ディレクティブではなく Proxy ディレクティブを使用して、ファイアウォール間でプロキシ依存のリクエストを送信する必要があります。

mod_oc4j 情報

mod_oc4j は、Web サーバー（通常は Oracle HTTP Server）と統合するモジュールであり、リクエストをバックエンドの OC4J プロセスにルーティングします。OPMN モジュールにより、mod_oc4j は様々な OC4J プロセスのステータスを認識できるため、稼働中のプロセスに対してのみルーティングを行います。また、mod_oc4j は Oracle Application Server のクラスタおよび OC4J アイランドの概念も認識し、それに応じてルーティングすることで最大限の透過的フェイルオーバーを提供します。

関連項目： 7-8 ページの「mod_oc4j」

SSL を使用した mod_oc4j と OC4J との通信

mod_oc4j と OC4J プロセス間の AJP 通信は、AJP/SSL を介して実行できます。これに対する妨害は以前からありませんでした。また、mod_oc4j と OC4J が通信するたびに SSL ネゴシエーションが発生することではなく、パフォーマンスへの影響はより少なくなっています。

関連項目： 7-15 ページの「mod_oc4j と OC4J 間での SSL の有効化」

Oracle HTTP Server のリリース番号

Oracle HTTP Server は、Apache バージョン 2.0.52 をベースとしています。

Oracle HTTP Server への Apache セキュリティ・パッチの適用

次の理由で、Oracle HTTP Server には Apache セキュリティ・パッチを適用できません。

- オラクル社は Oracle HTTP Server ユーザーにリリースする前に、セキュリティ・パッチをテストして適切に変更しています。
- オラクル社はセキュリティ・パッチのコンポーネントを使用中のスタックから削除しているため、多くの場合、openssl アラートなどのアラートは該当しません。
- オラクル社では、同社からパッチを入手する場合の遅延の影響がオープン・ソース組織から入手する場合に比べて最小限ですみ、サポート面で大きなメリットが得られるようにこれらのパッチをできるだけ早期にリリースしています。

Oracle HTTP Server からの出力の圧縮

通常、この目的には OracleAS Web Cache の使用をお勧めします。mod_gzip など、この目的のためにプラグインできる他のフリーウェア・モジュールもありますが、その使用はサポートされていません。これらのモジュールを使用すると、EAPI に関してエラー・メッセージが表示されることがありますが、通常は無視できます。

PHP のサポート

リリース 2 (10.1.2) では mod_php が完全にサポートされています。

関連項目： [7-21 ページの「mod_php」](#)

ハッカーからの Web サイトの保護

多数の攻撃があり、日々新たな攻撃が発生しています。次にサイトの保護に関する一般的なガイドラインを示します。サイトを完全に保護することはできませんが、安易なターゲットとなることは回避できます。

- ISP と Web サーバーの間に、Checkpoint FW-1 や Cisco PIX のような市販のファイアウォールを使用します。ただし、すべてのハッカーが組織の外にいるとは限らないことを認識してください。
- 切替式のイーサネットを使用して、信頼できないサーバーから傍受される可能性のある通信量を制限します。Web サーバー・マシンと、データベースやエンタープライズ・アプリケーションを実行中の機密性の高い内部サーバーの間に、追加のファイアウォールを使用します。
- RPC、Finger、telnet など、不要なネットワーク・サービスをサーバー・マシンから削除します。
- Web フォームからのすべての入力を慎重に検証します。特に、長い入力文字列や、印刷不能文字、HTML タグまたは JavaScript タグを含む入力には注意してください。
- 機密情報を含む Cookie の内容を暗号化つまりランダム化します。たとえば、ハッカーによる有効なセッションのハイジャックを防止するために、有効なセッション ID を推測しにくくする必要があります。
- すべてのシステムとアプリケーション・ソフトウェアのセキュリティ・パッチを頻繁にチェックし、入手後すぐにインストールします。このようなパッチが善意のソースからのものかどうかを確認し、信頼できるサイトからダウンロードして、暗号チェックサムを検証します。

- 侵入検出パッケージを使用して、改変された Web ページ、ウイルスおよびハッカーに侵入されたことを示す rootkit の有無をモニターします。可能な場合は、システムの実行可能ファイルと Web のコンテンツを読み取り専用ファイル・システムにマウントしてください。
- 法的に認知された解析パッケージ (forensic analysis) を入手して、侵入が検出されたらすぐにその証拠を取得します。これは、ハッカーを告発する際に役立ちます。

Oracle HTTP Server のトラブルシューティング

この付録では、Oracle HTTP Server の使用時に発生する可能性がある一般的な問題およびその解決方法について説明します。

内容は、次のとおりです。

- [問題および解決策](#)
- [その他の解決策](#)

問題および解決策

この項では、一般的な問題および解決策について説明します。内容は、次のとおりです。

- 間欠的 HTTP-500 エラー
- Oracle HTTP Server と OC4J ブロック間の接続におけるファイアウォール
- ポートの競合により Oracle HTTP Server が起動できない
- 多数の HTTPD プロセスによるマシンのオーバーロード
- 1024 未満のポートでの Oracle HTTP Server の起動時に発生する権限拒否
- PM ファイルが正しく検出されない場合に Oracle HTTP Server が起動できない
- Webcache リバース・プロキシでの SSO クライアント認証の失敗

間欠的 HTTP-500 エラー

Oracle HTTP Server で KeepAlive ディレクティブが「On」に設定されていると、Microsoft Internet Explorer の特定のセキュリティ・パッチが、MOD_OC4J_0145 エラー、MOD_OC4J_0119 エラー、MOD_OC4J_0013 エラーなどの間欠的 HTTP-500 エラーの原因となります。

問題

間欠的 HTTP-500 エラーの原因は、Microsoft Internet Explorer の不具合にあります。

解決策

この問題の考えられる解決策として、次の 2 つがあげられます。

- すべてのクライアントの Internet Explorer ブラウザにパッチを適用します。
- 前述の解決策が実際的ではない場合、Oracle HTTP Server で KeepAlive を「Off」に設定します。

この問題の詳細は、<http://metalink.oracle.com> で Metalink Note 269980.1 を参照してください。この Metalink Note にアクセスするには、OracleMetalink サイトの上部にある「Advanced Search」ボタンをクリックして、Doc ID 「269980.1」を検索するのが最も簡単な方法です。

関連項目： [5-3 ページの「KeepAlive」](#)

Oracle HTTP Server と OC4J ブロック間の接続におけるファイアウォール

特定のファイアウォールが Oracle HTTP Server と OC4J の間で使用されている場合、Oracle HTTP Server はリクエストを OC4J に転送できません。

問題

Oracle HTTP Server プロセスでは、OC4J プロセスとの永続的な接続を維持します。Oracle HTTP Server より前にファイアウォールによって接続がタイムアウトされると、ファイアウォールおよびオペレーティング・システムの構成方法に応じて、OC4J プロセスに対するリクエストはエラーとなるか、処理の時間が非常に長くなります。

解決策

Oracle HTTP Server のディレクティブ OC4JConnTimeout をファイアウォールのタイムアウト値（この値はファイアウォール固有）より小さい値に設定します。

関連項目： [7-10 ページの「Oc4jConnTimeout」](#)

ポートの競合により Oracle HTTP Server が起動できない

ポートの競合が原因で Oracle HTTP Server が起動できない場合、次のエラーが発生する可能性があります。

```
[crit] (98) Address already in use: make_sock: could not bind to port 7778
```

問題

ポート番号が別のプロセスで使用されているため、Oracle HTTP Server が起動できません。

解決策

Oracle HTTP Server に割り当てられているアドレスをブラウザに指定して結果を確認し、どのプロセスでポートがすでに使用されているかを特定します。結果に応じて、Oracle HTTP Server または競合するプロセスの IP: ポート・アドレスを変更します。

多数の HTTPD プロセスによるマシンのオーバーロード

1 台のマシン上で実行中の httpd プロセスが多すぎると、レスポンス時間が急激に低下します。

問題

httpd プロセスを起動しすぎると、通常の処理を行うためのリソースが不足します。

解決策

ハードウェア・ボックスが対処できる値まで MaxClients の値を下げます。

関連項目: [4-3 ページの「MaxClients」](#)

1024 未満のポートでの Oracle HTTP Server の起動時に発生する権限拒否

1024 未満のポートで Oracle HTTP Server を起動しようとする、次のエラーが発生します。

```
Bind errors on ports below 1024: PERMISSION DENIED: MAKE_SOCKET: COULD NOT BIND TO PORT 443.
```

問題

1024 未満のポートでは、Oracle HTTP Server は起動しません。これは、これらのポートをバインドするには、root 権限が必要になるためです。また、.apachectl を構成する手順も実行されていません。

解決策

次の手順を実行すると、1024 未満のポートで root として Oracle HTTP Server を実行できます。

1. root としてログインします。
2. 中間層の Oracle ホームで次のコマンドを実行します。

```
cd $ORACLE_HOME/Apache/Apache/bin
chown root .apachectl
chmod 6750 .apachectl
```

PM ファイルが正しく検出されない場合に Oracle HTTP Server が起動できない

Oracle HTTP Server では次のエラーが発生し、起動できないことがあります。

```
"[error] Can't locate mod_perl.pm in @INC (@INC contains:$ORACLE_HOME/perl/...)
```

または

```
[error] Can't locate Apache::Registry.pm in @INC (@INC contains: $ORACLE_HOME/per/...)
```

問題

mod_perl は、ORACLE_HOME/ohs/mod_perl ディレクトリの下に保存される PM ファイルを検出する必要があります。これらの PM ファイルがないと、mod_perl は起動しません。

解決策

UNIX の場合、apachectl によって変数 PERL5LIB が正しく定義されているかどうかをチェックします。この変数は次の場所を指している必要があります。

```
ORACLE_HOME/perl/lib/5.8.3:ORACLE_HOME/perl/lib/site_perl/5.8.3:ORACLE_HOME/ohs/mod_perl/lib/site_perl/5.8.3/sun4-solaris-thread-multi
```

Windows の場合、opmn.xml の HTTP Server セクションの環境サブセクションに PERL5LIB の正しいエントリがあるかどうかをチェックします。この変数は次の場所を指している必要があります。

```
$(ORACLE_HOME)\ohs\mod_perl\bin\MSWin32-x86-multi-thread;$(ORACLE_HOME)\perl\5.8.3\bin;$(PERL5LIB)
```

Webcache リバース・プロキシでの SSO クライアント認証の失敗

Webcache リバース・プロキシで SSO クライアント認証が失敗します。

問題

SSO クライアントのログイン時には、ブラウザからのクライアント証明書が SSO サーバーで認証された後、そのサーバーへの接続に成功します。ただし、ssoServer.log に示されているように、ブラウザではなく Webcache の Wallet に格納されている証明書を認証しようとしているため、接続に失敗します。

解決策

次の手順を実行します。

1. \$ORACLE_HOME/Apache/Apache/conf/httpd.conf を編集し、次の行を含めるようにします。

```
LoadModule certheaders_module libexec/mod_certheaders.so
AddCertHeader HTTPS
AddCertHeader SSL_CLIENT_CERT
```

2. \$ORACLE_HOME/sso/conf/sso_apache.conf を編集し、次の行をコメント化します。

```
#SSLOptions +ExportCertData +StdEnvVars
```

3. dcmctl updateconfig -ct ohs を実行します。
4. opmnctl restartproc type=ohs を実行します。
5. クライアント認証で SSO サーバーにログインできることを確認します。

その他の解決策

Oracle MetaLink (<http://metalink.oracle.com>) には、さらに多くの解決策が掲載されています。問題の解決策が見つからない場合は、サービス・リクエストを作成してください。

関連資料： Oracle Application Server のリリース・ノート (Oracle Technology Network (<http://www.oracle.com/technology/documentation/index.html>) で入手可能)

サード・パーティ・ライセンス

この付録には、Oracle Application Server に付属するすべてのサード・パーティ製品のサード・パーティ・ライセンスが記載されています。

内容は、次のとおりです。

- [Apache HTTP Server](#)
- [Apache SOAP](#)
- [DBI Module](#)
- [Perl](#)
- [PHP](#)
- [FastCGI](#)

Apache HTTP Server

Apache のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (Apache ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、Apache ソフトウェアは現状のままで Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または Apache から提供されません。

The Apache Software License

```
/* =====  
                                Apache License  
                                Version 2.0, January 2004  
                                http://www.apache.org/licenses/  
  
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION  
  
1. Definitions.  
  
"License" shall mean the terms and conditions for use, reproduction,  
and distribution as defined by Sections 1 through 9 of this document.  
  
"Licensor" shall mean the copyright owner or entity authorized by  
the copyright owner that is granting the License.  
  
"Legal Entity" shall mean the union of the acting entity and all  
other entities that control, are controlled by, or are under common  
control with that entity. For the purposes of this definition,  
"control" means (i) the power, direct or indirect, to cause the  
direction or management of such entity, whether by contract or  
otherwise, or (ii) ownership of fifty percent (50%) or more of the  
outstanding shares, or (iii) beneficial ownership of such entity.  
  
"You" (or "Your") shall mean an individual or Legal Entity  
exercising permissions granted by this License.  
  
"Source" form shall mean the preferred form for making modifications,  
including but not limited to software source code, documentation  
source, and configuration files.  
  
"Object" form shall mean any form resulting from mechanical  
transformation or translation of a Source form, including but  
not limited to compiled object code, generated documentation,  
and conversions to other media types.  
  
"Work" shall mean the work of authorship, whether in Source or  
Object form, made available under the License, as indicated by a  
copyright notice that is included in or attached to the work  
(an example is provided in the Appendix below).  
  
"Derivative Works" shall mean any work, whether in Source or Object  
form, that is based on (or derived from) the Work and for which the  
editorial revisions, annotations, elaborations, or other modifications  
represent, as a whole, an original work of authorship. For the purposes  
of this License, Derivative Works shall not include works that remain  
separable from, or merely link (or bind by name) to the interfaces of,  
the Work and Derivative Works thereof.  
  
"Contribution" shall mean any work of authorship, including  
the original version of the Work and any modifications or additions  
to that Work or Derivative Works thereof, that is intentionally
```


submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache SOAP

Apache のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (Apache ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、Apache ソフトウェアは現状のままで Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または Apache から提供されません。

Apache SOAP License

Apache SOAP license 2.3.1

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

DBI Module

Oracle はサード・パーティ・ライセンスの本文を表示することが求められていますが、サード・パーティ・プログラムは Oracle ライセンスの対象となり、Oracle はサード・パーティのテクノロジーに対する保証および技術サポートを提供しないものとします。

このプログラムには、DBI のサード・パーティ・コードが組み込まれています。DBI のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (DBI ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、DBI ソフトウェアは現状のまま Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または DBI から提供されません。

DBI モジュールの著作権 (1994 ~ 2002) は、アイルランドの Tim Bunce 氏に属しており、無断複写および転載を禁じます。

Perl README ファイルに指定されているように、GNU General Public License または Artistic License の条件に従って配布できます。

Perl Artistic License

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b. use the modified Package only within your corporation or organization.
 - c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b. accompany the distribution with the machine-readable source of the Package with your modifications.
 - c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.
7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.
8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Perl

Oracle はサード・パーティ・ライセンスの本文を表示することが求められていますが、サード・パーティ・プログラムは Oracle ライセンスの対象となり、Oracle はサード・パーティのテクノロジーに対する保証および技術サポートを提供しないものとします。

このプログラムには、Perl のサード・パーティ・コードが組み込まれています。Perl のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (Perl ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、Perl ソフトウェアは現状のまま Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または Perl から提供されません。

Perl Kit Readme

Copyright 1989-2001, Larry Wall

All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of either:

1. the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
2. the "Artistic License" which comes with this Kit.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See either the GNU General Public License or the Artistic License for more details.

You should have received a copy of the Artistic License with this Kit, in the file named "Artistic". If not, I'll be glad to provide one.

You should also have received a copy of the GNU General Public License along with this program in the file named "Copying". If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA or visit their Web page on the internet at <http://www.gnu.org/copyleft/gpl.html>.

For those of you that choose to use the GNU General Public License, my interpretation of the GNU General Public License is that no Perl script falls under the terms of the GPL unless you explicitly put said script under the terms of the GPL yourself. Furthermore, any object code linked with perl does not automatically fall under the terms of the GPL, provided such object code only adds definitions of subroutines and variables, and does not otherwise impair the resulting interpreter from executing any standard Perl script. I consider linking in C subroutines in this manner to be the moral equivalent of defining subroutines in the Perl language itself. You may sell such an object file as proprietary provided that you provide or offer to provide the Perl source, as specified by the GNU General Public License. (This is merely an alternate way of specifying input to the program.) You may also sell a binary produced by the dumping of a running Perl script that belongs to you, provided that you provide or offer to provide the Perl source as specified by the GPL. (The fact that a Perl interpreter and your code are in the same binary file is, in this case, a form of mere aggregation.) This is my interpretation of the GPL. If you still have concerns or difficulties understanding my intent, feel free to contact me. Of course, the Artistic License spells all this out for your protection, so you may prefer to use that.

mod_perl License

```
/* =====
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1996-2000 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
```



```

* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in
*   the documentation and/or other materials provided with the
*   distribution.
*
* 3. The end-user documentation included with the redistribution,
*   if any, must include the following acknowledgment:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*   Alternately, this acknowledgment may appear in the software itself,
*   if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Apache" and "Apache Software Foundation" must
*   not be used to endorse or promote products derived from this
*   software without prior written permission. For written
*   permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*   nor may "Apache" appear in their name, without prior written
*   permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* =====
*/

```

Perl Artistic License

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b. use the modified Package only within your corporation or organization.
 - c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b. accompany the distribution with the machine-readable source of the Package with your modifications.
 - c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 - d. make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the

so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.
8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

PHP

Oracle はサード・パーティ・ライセンスの本文を表示することが求められていますが、サード・パーティ・プログラムは Oracle ライセンスの対象となり、Oracle はサード・パーティのテクノロジーに対する保証および技術サポートを提供しないものとします。

このプログラムには、PHP のサード・パーティ・コードが組み込まれています。PHP のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (PHP ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、PHP ソフトウェアは現状のまま Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または PHP から提供されません。

The PHP License

The PHP License, version 3.0

Copyright (c) 1999-2004 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.
4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in

conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"

5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number.
Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes PHP, freely available from
<<http://www.php.net/>>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

FastCGI

Oracle はサード・パーティ・ライセンスの本文を表示することが求められていますが、サード・パーティ・プログラムは Oracle ライセンスの対象となり、Oracle はサード・パーティのテクノロジーに対する保証および技術サポートを提供しないものとします。

このプログラムには、FastCGI のサード・パーティ・コードが組み込まれています。FastCGI のライセンス条件に基づき、Oracle は次のライセンス文書を表示することが求められています。ただし、Oracle プログラム (FastCGI ソフトウェアを含む) を使用する権利は、この製品に付随する Oracle プログラム・ライセンスによって決定され、次のライセンス文書に含まれる条件でこの権利が変更されることはありません。反対の内容が Oracle プログラム・ライセンス内にあった場合でも、FastCGI ソフトウェアは現状のままで Oracle から提供されるものであり、いかなる種類の保証またはサポートも Oracle または FastCGI から提供されません。

FastCGI Developer's Kit License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and

Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS". OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

Module mod_fastcgi License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses.

Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS". OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

用語集

Apache

米国立スーパー・コンピュータ応用研究所 (National Center for Supercomputing Applications: NCSA) から導入された、公開の HTTP サーバー。

CA

「[認証局](#)」を参照。

CGI

Common Gateway Interface (CGI)。Web サーバー、および CGI 仕様に準拠したデータを受け入れて戻すように設計されたプログラム間で、情報を送信する業界標準の技術。

DAD

「[データベース・アクセス記述子](#)」を参照。

DES

データ暗号化規格。一般的に使用される、56 ビット・キーを使用した対称[鍵暗号化](#)方法。

Diffie-Hellman 鍵交換アルゴリズム (Diffie-Hellman key negotiation algorithm)

非保護チャンネルで通信を行う二者間で、当事者のみにわかる乱数を取り決める方法。Diffie-Hellman 鍵交換アルゴリズムの実行中は、当事者は非保護チャンネルで情報を交換するが、攻撃者がネットワーク通信を分析し、当事者の間で取り決めた乱数を計算によって推定するのはほぼ不可能である。Oracle Advanced Security では、セッション鍵の生成に Diffie-Hellman 鍵交換アルゴリズムが使用されている。

<Directory>

指定したディレクトリとそのサブディレクトリにのみ適用するディレクティブのグループを囲むために使用。ディレクトリのコンテキストで許可される任意のディレクトリを使用できる。ディレクトリ指定には、ディレクトリへのフル・パスまたはワイルドカード文字列を使用できる。ワイルドカード文字列の場合、? は任意の 1 文字と一致し、* は任意の文字列と一致する。<Directory /> はファイル・システム全体を指し、<Directory dir> は絶対ディレクトリを指すことに注意。<Directory> コンテナは相互にネストできないが、ドキュメント・ルートでネストしているディレクトリを指すことはできる。

<DirectoryMatch>

正規表現を指定するときは、ディレクトリ指定にワイルドカードとともに <Directory> のテイルダ形式を使用するかわりに、これを使用する必要がある。次の 2 つの例は同じ結果になり、web で始まって 1～9 の数字で終わるディレクトリが一致となる。

```
<Directory ~/web[1-9]/>
<DirectoryMatch "/web[1-9]/">
```

DIT

「[ディレクトリ情報ツリー](#)」を参照。

DMZ

「[非武装地帯](#)」を参照。

DN

「[識別名](#)」を参照。

<Files>

<Files *file*> および </Files> ディレクティブでは、ファイル名によるアクセス制御がサポートされる。この2つは、<Directory> および <Location> ディレクティブに相当する。このセクションで指定したディレクティブは、指定したファイル名と一致するベース名（ファイル名の最後の構成要素）に含まれる、任意のオブジェクトに適用できる。<Files> セクションは、構成ファイルに表示されている順序で、すなわち、<Directory> セクションの後に .htaccess ファイルが読み取られてから（ただし <Location> セクションの前に）処理される。<Files> ディレクティブを <Directory> セクション内でネストして、ファイル・システムの適用部分を限定できることに注意。

<FilesMatch>

<Files> ディレクティブと同様に、ファイル名によるアクセス制御を提供する。ただし、正規表現を使用できる。

HTTP

「[Hypertext Transfer Protocol \(HTTP\)](#)」を参照。

Hypertext Transfer Protocol (HTTP)

メッセージを書式化して送信し、各種コマンドへのレスポンスのために Web サーバーとブラウザで実行する必要がある処理を決定するために Web で使用される、基礎となる形式。HTTP は Oracle Application Server とクライアントの間で使用されるプロトコルである。

Keytool

[鍵](#)と[証明書](#)の管理ユーティリティ。

LDAP

「[Lightweight Directory Access Protocol](#)」を参照。

Lightweight Directory Access Protocol

拡張可能な標準ディレクトリ・アクセス・プロトコル。LDAP クライアントとサーバーが通信で使用される共通言語。業界標準のディレクトリ製品（Oracle Internet Directory など）をサポートする設計規則のフレームワーク。

<Limit>

<Limit *method*> では、受信リクエストの HTTP メソッドに従ってブロックを定義する。

<LimitExcept>

アクセス制御の対象を、指定した HTTP メソッドを除くすべての HTTP メソッドに限定する。

<Location>

<Directory> ディレクティブとは異なり、ブロック内のディレクティブの適用対象を物理ファイルの位置ではなく指定の URL に限定する。<Location> セクションは、構成ファイルに表示される順序、すなわち、<Directory> セクションと .htaccess ファイルが読み取られた後、および <Files> セクションの後に処理される。<Location> には、ワイルドカード・ディレクトリおよびティルダ文字による正規表現を使用できる。

<LocationMatch>

機能は <Location> と同じ。位置指定にワイルドカードを使用するティルダ形式の <Location> のかわりに、このディレクティブを使用して正規表現を指定する必要がある。

MD5

デジタル署名を作成するために、32 ビット・マシンで使用されるハッシュ・アルゴリズム。MD5 は **一方向ハッシュ関数** であり、これによりメッセージは **メッセージ・ダイジェスト** という固定長の数値に変換される。

OPMN

「[Oracle Process Manager and Notification Server](#)」を参照。

Oracle Process Manager and Notification Server (OPMN)

アプリケーション・サーバー・インスタンス内で Oracle HTTP Server および OC4J のプロセスを管理するコンポーネント。様々なコンポーネントからの全イベントを、その受信に関連する全コンポーネントにチャネルする。

PEM

プライバシー保護が強化された電子メールの形式。暗号化、認証、メッセージ整合性および鍵管理を行う **暗号化** 技術。

RSA

RSA Data Security 社によって開発された **公開鍵暗号** 技術。RSA アルゴリズムは、非常に大きな数の因数分解は困難であることに基づいている。RSA **鍵** のデコードに必要な処理能力および時間を考えると、数学的には実現不可能である。

Secure Hash Algorithm

与えられたデータから 160 ビットの暗号メッセージ・ダイジェストを生成することでデータの整合性を確保するアルゴリズム。1 ビットでもデータが変更されると、そのデータの Secure Hash Algorithm チェックサムが変化する。与えられたデータ・セットを偽造して、元のデータと同じ結果を Secure Hash Algorithm で生成することはコンピュータではほぼ不可能と考えられる。

長さが 264 ビット未満のメッセージを取得して、160 ビットのメッセージ・ダイジェストを生成するアルゴリズム。このアルゴリズムは MD5 よりも若干遅いが、メッセージ・ダイジェストが大きくなることで、総当たり攻撃や反転攻撃に対してより強力に保護できる。

Secure Shell

Secure Shell (SSH) は広く知られたプロトコルで、その実装は幅広く使用され、ポート・トンネリングに非常に似た安全な接続トンネリング・ソリューションを提供する。SSH には、接続のクライアント側およびサーバー側の両方にデーモンが提供されている。クライアントは、サーバーに直接接続せず、ローカル・デーモンに接続する。次に、ローカルの SSH デーモンはサーバー側のデーモンに対して安全な接続を確立する。通信は、クライアントからクライアント側のデーモンを経由してサーバー側のデーモン、さらに実際のサーバーにルーティングされる。これにより、安全性の低いプロトコルを使用するクライアント / サーバー・プログラムでも、安全なチャネルを経由してトンネルできる。SSH の短所は、ホップが 2 つ必要であること、および使用可能な実装のパフォーマンスと拡張性が十分でないことである。SSH の詳細は、次のサイトから入手できる。

<http://www.ssh.org>

Secure Sockets Layer (SSL)

HTTPS (セキュアな HTTP) を使用した、インターネット経由でドキュメントを安全に送信するための標準。SSL は、デジタル署名を使用して送信データが改ざんされていないことを保証する。

SHA

「[Secure Hash Algorithm](#)」を参照。

SSH

「[Secure Shell](#)」を参照。

SSL

「[Secure Sockets Layer](#)」を参照。

<VirtualHost>

Oracle HTTP Server には、多数の異なる Web サイトを同時に処理する機能がある。ディレクトティブを <VirtualHost> セクションに置くことで有効範囲を指定し、特定の Web サイトに関するリクエストにのみ適用することもできる。

仮想ホストは、その明示的なホスト名で区別されるように、あるマシン上の複数のサーバーをメンテナンスする手段である。たとえば、通常、複数の企業が 1 台の Web サーバーを共有している場合は独自のドメインを持つ必要があり、余分なパス情報を知る必要がない場合も、Web サーバーには [www.oracle1.com](#) や [www.oracle2.com](#) などとしてアクセスできる。

Wallet

[デジタル Wallet](#) ともいう。Wallet とは、個々のエンティティに対するセキュリティ資格証明の格納と管理に使用されるデータ構造である。様々な暗号化サービスで使用できるように、資格証明の格納と取出しを実現する。[Wallet Resource Locator](#) (WRL) は、Wallet の位置を特定するために必要な情報をすべて提供する。

Wallet Resource Locator

Wallet Resource Locator (WRL) は、Wallet の位置を特定するために必要な情報をすべて提供する。Wallet の保存場所であるオペレーティング・システムのディレクトリへのパスである。

WRL

「[Wallet Resource Locator](#)」を参照。

X.509

公開鍵は様々なデータ形式で作成できる。X.509 v3 形式は、そのような一般的な形式の 1 つである。

暗号化 (cryptography)

読むことのできない形式に変換 (暗号化) することで情報を保護する技法。「[暗号化 \(encryption\)](#)」を参照。

暗号化 (encryption)

意図した受信者以外はだれも判読できないようにメッセージを変換する処理。暗号化はデータを秘密のコードに変換することによって行われる。暗号化には、主に 2 つの種類がある。[公開鍵暗号](#) (非対称鍵暗号化) および対称鍵暗号化である。

暗号スイート (cipher suite)

ネットワークのノード間でメッセージ交換に使用される認証、暗号化およびデータ整合性アルゴリズムのセット。たとえば、SSL ハンドシェイク時には、メッセージを送受信するときに使用する暗号スイートを確認するために 2 つのノード間で折衝が行われる。

暗号文 (ciphertext)

暗号化されたデータ。暗号文は、鍵を使用して平文に変換 (復号化) しないかぎり読むことができない。「[復号化](#)」を参照。

一方向ハッシュ関数 (one-way hash function)

メッセージを 1 つの数値に変換するアルゴリズム。一方向とは、数値から元のメッセージを生成するのがほぼ不可能であることを意味する。計算済の[メッセージ・ダイジェスト](#)は、メッセージが改ざんされていないことを確認するため、[公開鍵](#)を使用して復号化されたメッセージ・ダイジェストと比較される。

エン트리 (entry)

ディレクトリ・サービスのコンテキストでは、エント리는ディレクトリの基本構成要素である。エント리는、ディレクトリ内のオブジェクトに関する情報の集まりである。各エント리는、オブジェクトのある1つの特性を表す属性のセットで構成される。たとえば、ディレクトリ・エント리가人物を示す場合、エントリには姓、名、電話番号、電子メール・アドレスなどの属性が含まれる。

鍵 (key)

エンコードされたデータの復号化に必要なパスワードまたは表。

可用性 (availability)

スケジュールされた時間のうち、コンピューティング・システムがアプリケーション・サービスを提供する割合または時間。

キーストア (Keystore)

企業での**鍵**および**証明書**を保持する保護データベース。キーストアへのアクセスは、パスワード（キーストアの作成時に作成者により定義され、現在のパスワードを指定した場合にのみ変更可能）により保護される。さらに、キーストア内の**秘密鍵**は、それぞれ独自のパスワードにより保護できる。

クリアテキスト (cleartext)

「[平文](#)」を参照。

公開鍵 (public key)

公開鍵暗号において一般に公開される鍵。主に暗号化に使用されるが、署名の確認にも使用される。「[公開鍵と秘密鍵のペア](#)」を参照。

公開鍵暗号 (public-key cryptography)

2つの異なる乱数（**鍵**）を使用する暗号化手法。「[公開鍵](#)」および「[公開鍵暗号 \(public-key encryption\)](#)」を参照。

公開鍵暗号 (public-key encryption)

メッセージの送信側が受信側の**公開鍵**でメッセージを暗号化する処理。配信されたメッセージは、受信側の**秘密鍵**で復号化される。

公開鍵と秘密鍵のペア (public/private key pair)

暗号化および**復号化**に使用される2つの数字のセットで、1つは**秘密鍵**、もう1つは**公開鍵**と呼ばれる。公開鍵は通常広く使用可能であるのに対して、秘密鍵はその所有者のみにより保持される。数学的に関連付けられてはいるが、計算によって公開鍵から秘密鍵を求めるのはほぼ不可能と考えられている。公開鍵と秘密鍵は、非対称型暗号化アルゴリズム（**公開鍵暗号**アルゴリズムともいう）または公開鍵暗号システムでのみ使用される。鍵のペアの公開鍵または秘密鍵を使用して暗号化されたデータは、鍵のペアによってそれに関連付けられている鍵で復号化できる。ただし、公開鍵で暗号化されたデータを同じ公開鍵で復号化したり、秘密鍵で暗号化されたデータを同じ秘密鍵で復号化することはできない。

識別名 (distinguished name)

ディレクトリ・エントリの一意の名前。**ディレクトリ情報ツリー**内のルートまでのすべての親エントリの名前から構成される。

証明書 (certificate)

デジタル証明ともいう。公開鍵に対して識別情報をセキュアにバインドする ITU x.509 v3 の標準データ構造。

証明書は、あるエンティティの**公開鍵**が信頼できる機関（**認証局**）によって署名されたときに作成され、そのエンティティの情報が正しいこと、および公開鍵が実際にそのエンティティに属していることを保証する。

証明書には、エンティティの名前、識別情報および公開鍵が含まれる。シリアル番号、有効期限、ならびにその証明書に関連する権利、使用および権限についての情報が含まれていることもある。最後に、発行元の認証局に関する情報が含まれる。

シングル・サインオン (single sign-on)

ユーザーが1度認証を受けると、その後の他のデータベースやアプリケーションへの接続時には厳密認証が透過的に発生する機能のこと。シングル・サインオンにより、ユーザーは1回の接続時に入力した1つのパスワードで、複数のアカウントおよびアプリケーションにアクセスできるようになる。

スケーラビリティ (scalability)

ビジネス・ニーズの変化に、ソフトウェアまたはハードウェア製品を対応させる手段。

データベース・アクセス記述子 (database access descriptor: DAD)

HTTP リクエストを実行するためにアプリケーションから Oracle Database に接続する方法を指定する一連の値。DAD には、ユーザー名 (スキーマと権限も指定)、パスワード、接続文字列、エラー・ログ・ファイル、標準エラー・メッセージおよび National Language Support (NLS) パラメータ (NLS 言語、NLS 日付書式、NLS 日付言語、NLS 通貨など) の情報が含まれる。

ディレクトリ情報ツリー (directory information tree)

ディレクトリ・エントリの DN で構成されるツリー形式の階層構造。「[識別名](#)」を参照。

デジタル Wallet (digital wallet)

「[Wallet](#)」を参照。

デジタル証明 (digital certificate)

「[証明書](#)」を参照。

認証 (authentication)

ユーザー、デバイス、またはホスト・システムにおけるその他のエンティティの識別情報を検証する処理で、しばしば、システム内のリソースへのアクセスを付与するための前提条件とされる。認証されたメッセージの受信者は、メッセージの起点 (送信者) を信頼できる。認証は、別人が送信者になりすましているという可能性を排除すると考えられる。

認証局 (certificate authority)

他のエンティティ (ユーザー、データベース、管理者、クライアント、サーバー) が本物であることを証明する、信頼できる第三者。ユーザーを証明するとき、認証局は最初にそのユーザーが証明書失効リスト (CRL) に掲載されていないことを確認してからそのユーザーの識別情報を検証し、証明書を付与し、認証局の秘密鍵を使用してその証明書に署名する。認証局には自身の証明書と公開鍵があり、公開されている。サーバーおよびクライアントは、これらを使用して認証局の署名を検証する。認証局は、証明書サービスを提供する外部の会社であったり、企業の MIS 部門のような内部の組織の場合もある。

非武装地帯 (de-militarized zone: DMZ)

ファイアウォールにより一方をインターネットから、もう一方を会社のイントラネットから隔離されたマシン群。このマシン群は、中程度のセキュリティを持つとみなされる。オープンなインターネットからは保護されているが、2つ目のファイアウォールの内側にあって会社のイントラネットの一部を構成するマシンほど完全には信頼できない。DMZ を使用した一般的なアプリケーション・サーバーの設定では、Web リスナーと Web サイトの静的コンテンツのみが DMZ に置かれる。イントラネット内のビジネス・ロジック、データベースおよびその他の重要なデータとシステムは、すべて保護される。

秘密鍵 (private key)

[公開鍵暗号](#)における秘密鍵。主に復号化に使用されるが、デジタル署名の場合、暗号化にも使用される。「[公開鍵と秘密鍵のペア](#)」を参照。

平文 (plaintext)

クリアテキストともいう。暗号化されていない ASCII 形式のデータ。

フェイルオーバー (failover)

コンポーネントに障害が発生したときに、類似するアクティブな代替コンポーネントを利用して、コンピューティング・システムを再構成する機能。

復号化 (decryption)

暗号化されたメッセージ (**暗号文**) の内容を、元の判読可能な書式 (**平文**) に変換する処理。

ポート (port)

特定のプログラムとの間で送信データをルーティングするため、TCP で使用される番号。

メッセージ・ダイジェスト (message digest)

単一行の連続した文字列として表されるテキスト表現。 **一方向ハッシュ関数** という式を使用して作成される。

モジュール (module)

モジュールは、Web サーバーの基本機能を拡張し、Oracle HTTP Server とその他の Oracle Application Server コンポーネントとの統合をサポートする。

索引

A

access.conf, B-2
ACKS, 5-3
AddCertHeader, 7-4
AddType, B-3
All16UTF-16, 7-19
alert, 6-4
Apache, 用語集 -1
 セキュリティ・パッチ, C-3
Apache HTTP Server, 1-2
 ライセンス, E-2
Apache SOAP
 ライセンス, E-5
apachectl, 1-6
Apache ソフトウェア
 ライセンス, E-2
AuthGroupFile, 8-7
AuthName, 8-7
AuthType, 8-7
AuthUserFile, 8-6

B

BrowserMatch, 8-5

C

CA, 用語集 -1
CERN, 7-3
CGI, 用語集 -1
 環境変数, 7-4
CompatEnvVars, 9-10
CondPattern, 7-22
conf, 3-4
CoreDumpDirectory, 3-3
crit, 6-4

D

DAD, 用語集 -1
DBI Module
 ライセンス, E-8
debug, 6-4
DES, 9-4, 用語集 -1
Diffie-Hellman 鍵交換アルゴリズム, 9-6, 用語集 -1
DIT, 用語集 -2
dms.conf, B-2

DMSMetricCollector を使用する MBLB の構成, A-6
DMZ, 用語集 -2
DN, 用語集 -2
DocumentRoot, 3-4, 7-25
Dynamic Monitoring Service, B-2

E

emerg, 6-4
error, 6-4
ErrorLog, 3-4
ExportCertData, 9-10

F

FakeBasicAuth, 9-10
FAQ, C-1
 Apache セキュリティ・パッチ, C-3
 ISP の顧客に対する HTTPS の提供, C-2
 mod_oc4j, C-2
 SSL の使用, C-2
Oracle HTTP Server
 リリース番号, C-3
Web サイトの保護
 ハッカー, C-3
 圧縮
 出力, C-3
 サポート
 PHP, C-3
 プロキシ依存のリクエスト, C-2
FastCGI
 ライセンス, E-14
-f オプション, 3-4

G

GET, 5-3
Global Environment, B-2
Group, 4-2

H

HTTP, 用語集 -2
httpd.conf, B-2
 Global Environment, B-2
 Main Server Configuration, B-3
 Virtual Hosts Parameters, B-3
HTTP リスナー, 1-4

I

iasobf, 9-16
iaspt.conf, 8-12, B-3
iaspt-port, 8-13
identd, 6-6
IdentityCheck, 6-6
IfModule ディレクティブ, 6-3
info, 6-4
IP アドレス
 アクセス制御, 8-4

K

KeepAlive, 5-3
KeepAliveTimeout, 5-3
Keytool, 用語集 -2

L

LDAP, 用語集 -2
Lightweight Directory Access Protocol, 用語集 -2
ListenBackLog, 5-3
LoadModule, 7-9
LoadModule ディレクティブ, 7-3
LockFile, 3-4
log-file, 8-13
LogFormat, 6-6
LogLevel, 6-4
log-level, 8-13
LogLoader, 6-2

M

Main Server Configuration, B-3
MaxClients, 1-6, 4-3
MaxKeepAliveRequests, 5-4
MaxRequestsPerChild, 4-3
MaxSpareServers, 1-6, 4-3
MD5, 9-4, 用語集 -3
mime.types, B-3
MinSpareServers, 1-6, 4-4
mod_access, 7-2, 8-2, 8-4
 ホストベースのアクセス制御, 8-4
mod_actions, 7-2
mod_alias, 7-2
mod_asis, 7-2
mod_auth, 7-2, 8-2, 8-6
 ユーザー認証, 8-6
mod_auth_anon, 7-3
mod_auth_dbm, 7-3
mod_autoindex, 7-3
mod_cern_meta, 7-3
mod_certheaders, 7-3
 CGI
 環境変数, 7-4
mod_cgi, 7-6
mod_cgid, 7-6
mod_dir, 7-6
mod_env, 7-6
mod_expires, 7-7

mod_fastcgi, 7-7
mod_file_cache, 7-7
mod_headers, 7-7
mod_imap, 7-7
mod_include, 7-7
mod_info, 7-7
mod_log_config, 7-8
mod_logio, 7-8
mod_mime, 7-8
mod_mime_magic, 7-8
mod_negotiation, 7-8
mod_oc4j, 7-8, 8-3, C-2
 SSL, 7-15
 構成ファイル, 7-9
 サンプル構成, 7-14
 ディレクティブ, 7-9
 ロード・バランシング, 7-15
mod_oc4j.conf, 8-10, B-3
mod_onsint
 実装上の差異, 7-17
 メリット, 7-17
 モジュール
 mod_onsint, 7-16
mod_ossl, 7-11, 7-18, 8-2, 8-7, 9-2
 使用方法, 9-4
 ディレクティブ, 9-5
 SSLCARevocationFile, 9-5, 9-6
 SSLCARevocationPath, 9-6
 SSLCipherSuite, 9-6
 SSLEngine, 9-8
 SSLLog, 9-8
 SSLLogLevel, 9-9
 SSLMutex, 9-9
 SSLOptions, 9-10
 SSLPassPhraseDialog, 9-11
 SSLProtocol, 9-11
 SSLRequire, 9-12
 SSLRequireSSL, 9-14
 SSLSessionCache, 9-14
 SSLSessionCacheTimeout, 9-15
 SSLVerifyClient, 9-15
 SSLWallet, 9-15
 SSLWalletPassword, 9-15
 ユーザー認証, 8-7
mod_osso, 7-11, 7-18, 8-2, 8-7, 8-14, B-3
 Oracle Identity Management, 8-14
 ユーザー認証, 8-7
mod_osso.conf, B-3
mod_perl, 1-4, 7-18, 8-3
 データベース使用上の注意, 7-18
 データベース接続のテスト, 7-19
mod_php, 7-21
mod_proxy, 7-21, 9-16
 ディレクティブ, 9-16
 SSLProxyCache, 9-16
 SSLProxyCipherSuite, 9-17
 SSLProxyProtocol, 9-17
 SSLProxyWallet, 9-17
 SSLProxyWalletPassword, 9-17
mod_rewrite, 7-21
 CondPattern, 7-22
 TestString, 7-22
 ディレクティブ, 7-22

- RewriteBase, 7-23
- RewriteEngine, 7-22
- RewriteLog, 7-23
- RewriteLogLevel, 7-23
- RewriteOptions, 7-23
- リダイレクションの例, 7-25
- ルール処理, 7-22
- ルールのヒント, 7-24
- mod_setenvif, 7-25, 8-4
 - ホストベースのアクセス制御, 8-4
- mod_speling, 7-25
- mod_ssl, 7-18, 8-7
- mod_status, 4-4, 7-25
- mod_unique_id, 7-25
- mod_userdir, 7-25
- mod_usertrack, 7-26
- mod_vhost_alias, 7-26
- mod_wchandshake, 7-26
- Multiviews, C-2

N

- nCipher, 9-4
- nFast, 9-5
- notice, 6-4

O

- OC4J
 - SSL, 7-15, 7-16
 - oc4j_deploy_tool.jar, 7-8
 - Oc4jCacheSize, 7-10
 - Oc4jConnTimeout, 7-10
 - Oc4jCookieExtension, 7-10
 - Oc4jEnableSSL, 7-15
 - Oc4jEnvVar, 7-11
 - Oc4jExtractSSL, 7-11
 - Oc4jiASPTActive, 8-10
 - Oc4jiASPTProcess, 8-11
 - Oc4jiASPTWalletFile, 8-11
 - Oc4jiASPTWalletPassword, 8-11
 - Oc4jMount, 7-12
 - Oc4jMountCopy, 7-13
 - Oc4jRoutingWeight, A-3
 - Oc4jSelectMethod, A-3
 - Oc4jSSLWallet, 7-16
 - Oc4jSSLWalletPassword, 7-16
 - Oc4jUseOHSErrors, 7-14
- OPMN, 用語集 -3
- opmn.xml, 8-10, 9-2, B-4
 - ias-component, B-4
 - process-set, B-4
 - process-type, B-4
- OptRenegotiate, 9-11
- ORA_IMPLICIT, 7-20
- ORA_NCHAR, 7-20
- Oracle Application Server
 - Certificate Authority
 - oracle_ocm.conf, B-4
- Oracle Diagnostic Logging, 6-2
 - LogLoader, 6-2
 - 概要, 6-2
 - 構成

- Oracle HTTP Server, 6-2
- 従来型の Apache メッセージ形式, 6-2
- ディレクティブ
 - OraLogMode, 6-2
 - OraLogSeverity, 6-2
- Oracle HTTP Server
 - C/C++, 1-3
 - Dynamic Monitoring Service, 1-3
 - FAQ, C-1
 - OPMN, 1-3
 - Perl, 1-3
 - PHP, 1-3
 - 圧縮
 - 出力, C-3
 - 仮想ホスト, 1-3
 - 管理, 1-5
 - 概要, 1-1
 - 起動, 1-6
 - 機能, 1-2
 - 構成ファイル, B-1
 - コンポーネント, 1-4
 - HTTP リスナー, 1-4
 - Perl インタプリタ, 1-4
 - モジュール, 1-4
 - サード・パーティ・ライセンス, E-1
 - Apache HTTP Server, E-2
 - Apache SOAP, E-5
 - DBI Module, E-8
 - FastCGI, E-14
 - Perl, E-10
 - PHP, E-13
 - サーバー・サイド・インクルード, 1-3
 - サーバー・プロセスの処理, 4-2
 - 再起動, 1-6
 - サポート, 1-5
 - シングル・サインオン, 1-2
 - スタンドアロン, 2-1
 - セキュリティ, 1-2
 - 仮想ホストのアクセス制御, 8-4
 - 概要, 8-2
 - 認可, 8-3
 - 認証, 8-3
 - 保護されるリソース, 8-3
 - ホストベースのアクセス制御, 8-3
 - ユーザー認可, 8-6
 - ユーザー認証, 8-6
 - ユーザーのクラス, 8-2
 - ユーザーの権限, 8-2
 - 停止, 1-6
 - プロキシ・サーバーと URL リライティング, 1-3
 - プロセス・モデル, 4-2
 - セキュリティに関する考慮事項, 4-4
 - モジュール, 1-4, 7-1
 - ユーティリティ
 - iasobf, 9-16
 - リリース番号, C-3
 - ロード・バランシング, 1-4
- Oracle Identity Management
 - セキュリティ, 8-13
- Oracle Process Manager and Notification Server, B-4,
用語集 -3
- Oracle Wallet, 8-12
- Oracle Wallet Manager, 8-11

oracle_apache.conf, B-4
oracle_ocm.conf, B-4
oracle_oc4j.api.MetricCollector, A-7
OraLogMode, 6-2
OraLogSeverity, 6-2

P

PEM, 9-6, 用語集 -3
Performance Monitor, 4-4
Perl
 データベース・アクセス, 7-18
 ライセンス, E-10
Perl インタプリタ, 1-4
PHP, C-3
 ライセンス, E-13
php.ini, B-4
PidFile, 3-4
PID ファイル, 6-6
PKCS #11 のサポート, 9-4
POST, 5-3
PROC_READY, 7-17
ps ユーティリティ, 4-4
PUT, 5-3

R

restartproc, 1-6
RewriteBase, 7-23
RewriteEngine, 7-22
RewriteLog, 7-23
RewriteLogLevel, 6-7, 7-23
RewriteOptions, 7-23
root, 4-4
RSA, 9-4, 用語集 -3

S

ScoreBoardFile, 3-4
ScriptSock, 7-6
Secure Hash Algorithm, 用語集 -3
Secure Shell, 用語集 -3
Secure Sockets Layer, 9-2, 用語集 -3
SendBufferSize, 5-3
ServerAdmin, 3-2
ServerAlias, 3-3
ServerName, 3-2, 5-4
ServerRoot, 3-4
ServerSignature, 3-2
ServerTokens, 3-3
set_default_form, 7-20
set_form, 7-20
SetEnvIf, 8-6
setupinfo.txt, 5-2
SHA, 9-4, 用語集 -3
SimulateHttps, 7-5
SQL NCHAR データ型, 7-19
SQLNCHAR, 7-20
srm.conf, B-2
SSH, 用語集 -4
SSL, 9-2, 用語集 -4
 mod_oc4j, 7-15
 OC4J, 7-15, 7-16

 バージョン 3.0, 9-4
 ログ, 6-7
ssl_engine_log, 6-7
ssl_request_log, 6-7
SSLAccelerator
 nFast, 9-5
SSLCACertificateFile, 9-4
SSLCACertificatePath, 9-4
SSLCARevocationFile, 9-5, 9-6
SSLCARevocationPath, 9-6
SSLCertificateChainFile, 9-4
SSLCertificateFile, 9-4
SSLCertificateKeyFile, 9-4
SSLCipherSuite, 9-6
 タグ, 9-7
ssl.conf, B-5
SSLEngine, 9-8
SSLLog, 9-8
SSLLogFile, 6-7
SSLLogLevel, 9-9
SSLMutex, 9-9
SSLOptions, 9-10
 CompatEnvVars, 9-10
 ExportCertData, 9-10
 FakeBasicAuth, 9-10
 OptRenegotiate, 9-11
 StdEnvVars, 9-10
 StrictRequire, 9-10
SSLPassPhraseDialog, 9-11
SSLProtocol, 9-11
SSLProxyCache, 9-16
SSLProxyCipherSuite, 9-17
SSLProxyProtocol, 9-17
SSLProxyWallet, 9-17
SSLProxyWalletPassword, 9-17
SSLRandomSeed, 9-4
SSLRequire, 9-12
 変数
 SSL, 9-13
 標準, 9-12
SSLRequireSSL, 9-14
SSLSessionCache, 9-14
SSLSessionCacheTimeout, 9-15
SSLVerifyClient, 9-15
SSLVerifyDepth, 9-4
SSLWallet, 9-15
SSLWalletPassword, 9-15
SSL の構成
 SSL の有効化, 9-2
 構成のカスタマイズ, 9-3
 実際の Wallet の作成, 9-2
SSL の有効化, 9-1
 概要, 9-2
startproc, 1-6
StartServers, 4-3
StdEnvVars, 9-10
stopproc, 1-6
StrictRequire, 9-10

T

TCP, 5-3
TCP SYN, 5-3

TCP バッファ, 5-3
TestString, 7-22
ThreadsPerChild, 4-3
TimeOut, 5-3

U

UseCanonicalName, 3-2
User, 4-2
UseWebCacheIp, 5-4
USR1, 1-6
UTF8, 7-19

V

Virtual Hosts Parameters, B-3

W

Wallet, 9-4, 用語集 -4
デジタル, 用語集 -6
Wallet Resource Locator, 用語集 -4
wallet-file, 8-12
wallet-password, 8-12
warn, 6-4
WRL, 用語集 -4

X

X.509, 用語集 -4

あ

アクセス制御
IP アドレス, 8-4
環境変数, 8-5
ドメイン名, 8-5
ネットマスク, 8-5
ネットワーク, 8-5
アクセス・ログ, 6-5
アフィニティを考慮したランダム, A-2
アプリケーション固有のエラー・ページ, C-2
暗号化, 用語集 -4
暗号スイート, 用語集 -4
暗号文, 用語集 -4
一方向ハッシュ関数, 用語集 -4
エラー・ログ, 6-6
エントリ, 用語集 -5

か

鍵, 用語集 -5
カスタム・ログ, 6-6
仮想ホスト
アクセス制御, 8-4
可用性, 用語集 -5
環境変数
アクセス制御, 8-5
間欠的 HTTP-500 エラー, D-2
管理, 1-5
サーバーとネットワークの相互作用, 5-2
サーバー・プロセス, 4-1
接続の永続性, 5-3

ネットワーク接続, 5-1
概要, 1-1
キーストア, 用語集 -5
起動, 1-6
機能, 1-2
機密保護, 8-2
クライアント IP アドレス
取得, 5-4
クリアテキスト, 用語集 -5
グレースフル・リスタート, 1-6
グローバル・サーバー ID のサポート, 9-4
公開鍵, 用語集 -5
公開鍵暗号, 用語集 -5
公開鍵と秘密鍵のペア, 用語集 -5
構成

iaspt.conf
iaspt-port, 8-13
log-file, 8-13
log-level, 8-13
wallet-file, 8-12
wallet-password, 8-12
mod_oc4j, 7-9
Oc4jASPTActive, 8-10
Oc4jASPTProcess, 8-11
Oc4jASPTWalletFile, 8-11
Oc4jASPTWalletPassword, 8-11
SSL の有効化, 7-15
OC4J
OC4J での SSL, 7-16
SSL, 9-2
サーバー・ログ, 6-1
リバース・プロキシ, 5-4
ロード・バランサ, 5-4
構成ファイル, B-1
access.conf, B-2
dms.conf, B-2
httpd.conf, B-2
ファイル構造, B-2
iaspt.conf, B-3
mime.types, B-3
mod_oc4j.conf, B-3
mod_osso.conf, B-3
opmn.xml, B-4
oracle_apache.conf, B-4
oracle_ocm.conf, B-4
php.ini, B-4
srm.conf, B-2
ssl.conf, B-5

コマンド
-f, 3-4
restartproc, 1-6
startproc, 1-6
stopproc, 1-6
コンポーネント, 1-4

さ

サード・パーティ・ライセンス, E-1
サーバー・プロセス, 4-1
サーバー・ログ, 6-1
再起動, 1-6
サポート, 1-5
PHP, C-3

識別名, 9-10, 用語集 -5
指定, 3-3
 サーバー位置, 3-1
 ファイル位置, 3-1
 リスナー・アドレス, 5-2
 リスナー・ポート, 5-2
 ログ・ファイル, 6-5
 PID ファイル, 6-6
 SSL ログ, 6-7
 アクセス・ログ, 6-5
 カスタム・ログ, 6-6
 スクリプト・ログ, 6-7
 送信ログ, 6-7
 パイプされたログ, 6-6
 リライト・ログ, 6-7
 ロットの切替え, 6-5
 ログ・ファイルの位置, 6-5
 ログ・レベル, 6-5
証明書, 用語集 -5
 X.509, 9-10
 デジタル, 用語集 -6
証明書失効リスト, 9-6
シングル・サインオン, 8-2, 8-14, 用語集 -6
 Oracle Identity Management, 8-14
 パートナ・アプリケーション, 8-7
実行
 root, 4-4
順序, 8-3
スクリプト・ログ, 6-7
スケーラビリティ, 用語集 -6
スタンドアロンの Oracle HTTP Server
 mod_oc4j の構成, 2-5
 インストール, 2-2
 構成
 OPMN, 2-3
 Oracle Application Server, 2-1
 シングル・サインオン, 2-5
 チェックリスト, 2-2
セキュリティ
 Oracle Identity Management, 8-13
 mod_osso, 8-14
 概要, 8-13
 シングル・サインオン, 8-14
 機密保護, 8-2
 認可, 8-2
 認証, 8-2
 保護されるリソース, 8-3
 ユーザーのクラス, 8-2
 ユーザーの権限, 8-2
接続の永続性, 5-3
送信ログ, 6-7

た

追加の SSL 機能, 9-3
 PKCS #11 のサポート, 9-4
 グローバル・サーバー ID のサポート, 9-4
停止, 1-6
ディレクティブ
 AddCertHeader, 7-4
 AddType, B-3
 AuthGroupFile, 8-7
 AuthName, 8-7

AuthType, 8-7
AuthUserFile, 8-6
CoreDumpDirectory, 3-3
DocumentRoot, 3-4
ErrorLog, 3-4
Group, 4-2
Include, 8-4
KeepAlive, 5-3
KeepAliveTimeOut, 5-3
ListenBackLog, 5-3
LockFile, 3-4
LogFormat, 6-6
MaxClients, 4-3
MaxKeepAliveRequests, 5-4
MaxRequestsPerChild, 4-3
MaxSpareServers, 4-3
MinSpareServers, 4-4
mod_ossll, 8-7, 9-2
mod_ssl, 8-7
Oc4jMount
 ajp13_dest, 7-13
 cluster_dest, 7-13
 instance_dest, 7-13
OraLogMode, 6-2
OraLogSeverity, 6-2
 module_name, 6-2, 6-3
 msg_level, 6-3
 msg_type, 6-3
PidFile, 3-4
RewriteBase, 7-23
RewriteEngine, 7-22
RewriteLog, 7-23
RewriteLogLevel, 6-7, 7-23
RewriteOptions, 7-23
ScoreBoardFile, 3-4
SendBufferSize, 5-3
ServerAdmin, 3-2
ServerAlias, 3-3
ServerName, 3-2
ServerRoot, 3-4
ServerSignature, 3-2
ServerTokens, 3-3
SimulateHttps, 7-5
SSLCACertificateFile, 9-4
SSLCACertificatePath, 9-4
SSLCertificateChainFile, 9-4
SSLCertificateFile, 9-4
SSLCertificateKeyFile, 9-4
SSLLogFile, 6-7
SSLRandomSeed, 9-4
SSLVerifyDepth, 9-4
StartServers, 4-3
ThreadsPerChild, 4-3
TimeOut, 5-3
UseCanonicalName, 3-2
User, 4-2
UseWebCacheIp, 5-4
ディレクトリ情報ツリー, 用語集 -6
データベース・アクセス記述子, 用語集 -6
データベース使用上の注意, 7-18
デジタル Wallet, 用語集 -6
デジタル証明, 用語集 -6
トラブルシューティング, D-1

- 1024 未満のポートでの Oracle HTTP Server の起動時に発生する権限拒否, D-3
- Oracle HTTP Server と OC4J ブロック間の接続におけるファイアウォール, D-2
- PM ファイルが正しく検出されない場合に Oracle HTTP Server が起動できない, D-3
- Webcache リバース・プロキシでの SSO クライアント認証の失敗, D-4
- 間欠的 HTTP-500 エラー, D-2
- 多数の HTTPD プロセスによるマシンのオーバーロード, D-3
- ポートの競合により Oracle HTTP Server が起動できない, D-3
- 独自のメトリック・コレクタの作成, A-7
 - ドメイン名
 - アクセス制御, 8-5

な

- 認可, 8-2
- 認証, 8-2, 用語集 -6
- 認証局, 用語集 -6
- ネットマスク
 - アクセス制御, 8-5
- ネットワーク
 - アクセス制御, 8-5

は

- ハッカー, C-3
- パイプされたログ, 6-6
- 非武装地帯, 用語集 -6
- 秘密鍵, 用語集 -6
- 平文, 用語集 -7
- ファイル位置, 3-3
- フェイルオーバー, 用語集 -7
- 復号化, 用語集 -7
- プロセス情報, 4-4
 - mod_status, 4-4
 - Performance Monitor, 4-4
 - ps ユーティリティ, 4-4
- 変更
 - ポート, 5-2
- 保護
 - Web サイト, C-3
 - 保護されるリソース, 8-3
 - ホストベースのアクセス制御, 8-3
 - IP アドレス, 8-4
 - mod_access, 8-4
 - mod_setenvif, 8-4
 - 環境変数, 8-5
 - ドメイン名, 8-5
 - ネットマスク, 8-5
 - ネットワーク, 8-5
- ポート, 用語集 -7
 - 変更, 5-2
- ポート・トンネリング, 8-7, B-3
 - 構成
 - iaspt.conf, 8-8
 - mod_oc4j.conf, 8-9
 - opmn.xml, 8-8
 - SSL, 8-9
 - 構成のリファレンス, 8-10

ま

- メッセージ・ダイジェスト, 用語集 -7
- メトリック・コレクタ, A-5
- メトリック・ベースのロード・バランシング, A-4
 - DMSMetricCollector を使用する MBLB の構成, A-6
 - DMS メトリックを MBLB に変換する方法, A-7
- OC4J の構成, A-5
 - OC4J のメトリックの指定, A-6
- Oracle HTTP Server の構成, A-5
 - oracle.oc4j.api.MetricCollector, A-7
 - 独自のメトリック・コレクタの作成, A-7
- モジュール, 1-4, 7-1, 用語集 -7
 - mod_access, 7-2
 - mod_actions, 7-2
 - mod_alias, 7-2
 - mod_asis, 7-2
 - mod_auth, 7-2
 - mod_auth_anon, 7-3
 - mod_auth_dbm, 7-3
 - mod_autoindex, 7-3
 - mod_cern_meta, 7-3
 - mod_certheaders, 7-3
 - mod_cgi, 7-6
 - mod_cgid, 7-6
 - mod_dir, 7-6
 - mod_env, 7-6
 - mod_expires, 7-7
 - mod_fastcgi, 7-7
 - mod_file_cache, 7-7
 - mod_headers, 7-7
 - mod_imap, 7-7
 - mod_include, 7-7
 - mod_info, 7-7
 - mod_log_config, 7-8
 - mod_logio, 7-8
 - mod_mime, 7-8
 - mod_mime_magic, 7-8
 - mod_negotiation, 7-8
 - mod_oc4j, 7-8
 - mod_oss, 7-18
 - mod_osso, 7-18
 - mod_perl, 7-18
 - mod_php, 7-21
 - mod_proxy, 7-21
 - mod_rewrite, 7-21
 - mod_setenvif, 7-25
 - mod_speling, 7-25
 - mod_ssl, 7-18
 - mod_status, 7-25
 - mod_unique_id, 7-25
 - mod_userdir, 7-25
 - mod_usertrack, 7-26
 - mod_vhost_alias, 7-26
 - mod_wchhandshake, 7-26

や

- 有効化
 - SSL
 - mod_oc4j, 7-15
 - mod_oc4j と OC4J, 7-15
 - OC4J, 7-16

- Oc4jEnableSSL, 7-15
- Oc4jSSLWalletFile, 7-16
- Oc4jSSLWalletPassword, 7-16
- ユーザー認可, 8-6
- ユーザー認証, 8-6
 - mod_auth, 8-6
 - mod_ossl, 8-7
 - mod_osso, 8-7
- ユーティリティ
 - iasobf, 9-16
- よくある質問, C-1

ら

- ラウンドロビン, A-2
- ランダム, A-2
- リスナー・アドレス, 5-2
- リスナー・ポート, 5-2
- リバース・プロキシ, 5-4
- リライト・ログ, 6-7
- ルーティングの重みを使用したラウンドロビン, A-2
- ルーティングの重みを使用したランダム, A-2
- ローカル・アフィニティを考慮したメトリック・ベース, A-3
- ローカル・アフィニティを考慮したラウンドロビン, A-2
- ロード・バランサ, 5-4
- ロード・バランシング, 7-15
 - mod_oc4j の使用, A-1
 - パラメータ, A-3
 - Oc4jRoutingWeight, A-3
 - Oc4jSelectMethod, A-3
- ポリシー, A-2
 - アフィニティを考慮したランダム, A-2
 - メトリック・ベース, A-2
 - ラウンドロビン, A-2
 - ランダム, A-2
 - ルーティングの重みを使用したラウンドロビン, A-2
 - ルーティングの重みを使用したランダム, A-2
 - ローカル・アフィニティを考慮したメトリック・ベース, A-3
 - ローカル・アフィニティを考慮したラウンドロビン, A-2
- メトリック・ベース, A-4
- ロギング
 - エラー, 6-6
- ログ, 3-4
- ログの切替え, 6-5
- ログの書式
 - authuser, 6-6
 - bytes, 6-6
 - Common Log Format, 6-6
 - data, 6-6
 - host, 6-6
 - ident, 6-6
 - request, 6-6
 - status, 6-6
- ログ・ファイル, 6-5, 6-6
 - 位置, 6-5
- ログ・レベル, 6-5