

Oracle® Containers for J2EE

構成および管理ガイド

10g (10.1.3.4.0)

部品番号 : B50866-01

2008 年 9 月

Oracle Containers for J2EE 構成および管理ガイド, 10g (10.1.3.4.0)

部品番号: B50866-01

Oracle Containers for J2EE Configuration and Administration Guide, 10g (10.1.3.4.0)

原本部品番号: E12288-01

原著者: Bonnie Vaughan

原本協力者: Dan Hynes, Sheryl Maring, Brian Wright, Bryan Atsatt, Ellen Barnes, Angela Barone, Julie Basu, Steve Button, Olivier Caudron, Lars Ewe, Jose Alberto Fernandez, Marcelo Goncalves, Sumathi Gopalakrishnan, Ping Guo, Hal Hildebrand, James Kirsch, Alex Kosowski, Sunil Kunisetty, Clement Lai, Philippe Le Mouel, Adam Leftik, Mike Lehmann, Sharon Malek, Sheryl Maring, Kuassi Mensah, Jasen Minton, Rama Notowidigdo, John O'Duinn, Debu Panda, Shiva Prasad, Chaya Ramanujam, Vinay Shukla, Sanjay Singh, Gael Stevens, Kenneth Tang, Frances Zhao, Helen Zhao, Serge Zlot

Copyright © 2006, 2008, 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、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 USA, Inc., 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, Siebel 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ります。

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	xi
対象読者	xii
ドキュメントのアクセシビリティについて	xii
関連ドキュメント	xiii
表記規則	xiv
サポートおよびサービス	xiv
1 Oracle Containers for J2EE (OC4J) の概要	
OC4J の概要	1-2
OC4J における J2EE サポート	1-3
OC4J の新機能および変更された機能	1-3
OC4J の新機能	1-3
Web サービスのサポート	1-4
アプリケーションの管理およびデプロイに関する新しい J2EE 1.4 仕様のサポート	1-4
Enterprise JavaBeans 3.0 のサポート	1-4
Oracle Application Server TopLink のサポート	1-4
Oracle Job Scheduler	1-5
新しい 2 フェーズ・コミット・トランザクション・コーディネータ機能	1-5
汎用 JMS リソース・アダプタの拡張機能	1-5
admin_client.jar の新しいコマンドおよびリモート・クライアント	1-5
旧リリースからの構成ファイルの変更点	1-5
スタンドアロン構成での OC4J	1-7
Oracle Application Server 構成での OC4J	1-8
OC4J のアプリケーション階層の概要	1-10
system アプリケーション	1-10
default アプリケーション	1-11
グローバル Web アプリケーション	1-11
J2EE アプリケーション	1-11
2 スタンドアロン OC4J のインストール	
スタンドアロン OC4J サーバーのインストール前提条件への準拠	2-2
スタンドアロン OC4J ディストリビューションのインストール	2-3
3 OC4J の管理用ツール	
Oracle Enterprise Manager 10g Application Server Control	3-2
スタンドアロン OC4J での Application Server Control へのアクセス	3-2

Oracle Application Server での Application Server Control へのアクセス	3-2
Application Server Control インタフェースの機能の概要	3-3
admin_client.jar コマンドライン・ユーティリティ	3-4
admin.jar コマンドライン・ユーティリティ	3-4
oc4j 実行可能スクリプト	3-5
Oracle Process Manager and Notification Server (OPMN)	3-6
oc4jadmin アカウントのパスワードの変更	3-7

4 OC4J ランタイムの構成

JDK のバージョンの指定	4-2
スタンドアロン構成での JDK の指定	4-2
管理構成での JDK の指定	4-2
起動時の OC4J ランタイム・オプションの設定	4-3
スタンドアロン OC4J 構成でのランタイム・オプションの設定	4-3
管理 OC4J 構成でのランタイム・オプションの設定	4-3
OC4J ランタイム・オプションの概要	4-4
起動時のシステム・プロパティの設定	4-5
スタンドアロン OC4J 構成でのシステム・プロパティの設定	4-5
OPMN 管理 OC4J 構成でのシステム・プロパティの設定	4-6
manifest.dependencies.warn.only を使用した共有ライブラリでの異なる依存性の許容	4-6
シンボリック・リンク使用の防止	4-7
stdout および stderr ログ・ファイルの管理	4-8
一般的なシステム・プロパティの概要	4-8
デバッグ・プロパティの概要	4-13
統合開発環境からのリモート・デバッグの有効化	4-13
Application Server Control を使用した OC4J インスタンスのリモート・デバッグの有効化	4-14
opmn.xml ファイルへのデバッグ開始パラメータの指定	4-14
起動コマンドラインへのデバッグ開始パラメータの指定	4-14
リモートからの Web アプリケーションのデバッグ	4-15
サーブレットのリモート・デバッグ用の設定	4-15
JSP のリモート・デバッグ用の設定	4-15

5 OC4J の起動および停止

スタンドアロン環境での OC4J の起動	5-2
oc4j スクリプトによる OC4J の起動	5-2
oc4j.jar による OC4J の起動	5-2
Oracle Application Server 環境での OC4J の起動	5-2
スタンドアロン環境での OC4J の停止	5-3
admin_client.jar によるスタンドアロン OC4J の停止	5-3
admin.jar による OC4J の停止	5-4
oc4j スクリプトによる OC4J の停止	5-4
Oracle Application Server 環境での OC4J の停止	5-5
スタンドアロン環境での OC4J インスタンスの再起動	5-5
Oracle Application Server 環境での OC4J インスタンスの再起動	5-6

6 admin_client.jar ユーティリティの使用法

admin_client.jar の使用の準備	6-2
admin_client.jar の構文および URI の指定の概要	6-2
クラスタ内の OC4J インスタンスのグループでの操作の実行	6-3
特定の OC4J インスタンスでの操作の実行	6-4
スタンドアロン OC4J サーバーでの操作の実行	6-5
URI の検証	6-5
リモート管理クライアントのダウンロードおよび抽出	6-5
コンソールへの使用方法の出力	6-7
ロギングの有効化	6-7
アーカイブのデプロイ	6-8
J2EE アプリケーション (EAR) のデプロイ	6-9
リモート・クライアントからの J2EE アプリケーションのデプロイ	6-11
スタンドアロン Web モジュール (WAR) のデプロイ	6-12
スタンドアロン・リソース・アダプタ (RAR) のデプロイ	6-13
バッチ・デプロイでのスクリプト・ファイルの使用	6-14
デプロイ後の Web サイトへの Web モジュールのバインド	6-14
単一の Web サイトへの全 Web モジュールのバインド	6-14
特定 Web モジュールの Web サイトへのバインドおよびコンテキスト・ルートの設定	6-15
アーカイブの再デプロイ	6-15
クラスタ内の順次再デプロイ間の遅延の指定	6-16
スケジュール済ジョブを含むアプリケーションの再デプロイ	6-17
アーカイブのアンデプロイ	6-17
EAR またはスタンドアロン WAR のアンデプロイ	6-17
スタンドアロン RAR のアンデプロイ	6-17
デプロイ済 EJB モジュールの変更されたクラスの更新	6-18
共有ライブラリの作成および管理	6-18
共有ライブラリのインストール	6-19
既存の共有ライブラリの変更	6-20
共有ライブラリの内容の表示	6-21
すべての共有ライブラリのリスト出力	6-21
共有ライブラリの削除	6-21
アプリケーションの起動、再起動および停止	6-22
OC4J インスタンスの再起動および停止	6-22
OC4J インスタンスまたはインスタンスのグループの再起動	6-22
1 つ以上の OC4J インスタンスの停止	6-22
データソースの管理	6-23
データソース接続プールの追加、テストおよび削除	6-23
データソース接続プールの追加	6-23
データソース接続プールのテスト	6-24
データソース接続プールの削除	6-24
データソースの追加、テストおよび削除	6-25
マネージド・データソースの追加	6-25
マネージド・データソースの削除	6-26
ネイティブ・データソースの追加	6-26
ネイティブ・データソースの削除	6-27
データベース接続のテスト	6-27
データソースのテスト	6-28

アプリケーションのデータソース・ディスクリプタの取得	6-28
JMS リソースの管理	6-28
JMS コネクション・ファクトリの管理	6-28
JMS コネクション・ファクトリの追加	6-29
JMS コネクション・ファクトリの削除	6-29
JMS コネクション・ファクトリの情報の取得	6-30
JMS 宛先の管理	6-30
JMS 宛先の追加	6-30
JMS 宛先の削除	6-31
JMS 宛先の情報の取得	6-31
リモート・クライアントを使用した OC4J の管理	6-31
リモートからの admin_client.jar コマンドの使用法	6-31
JConsole を使用したリモートの Oracle Application Server インスタンスへの接続	6-32
JMX プログラム・クライアントを使用したリモートからの OC4J の管理	6-33

7 admin.jar ユーティリティの使用法

admin.jar の使用法の概要	7-2
admin.jar の構文の概要	7-2
コンソールへのヘルプ・テキストの出力	7-3
スタンドアロン OC4J インスタンスの管理	7-3
スタンドアロン環境での OC4J の停止および再起動	7-3
OC4J による変更済ファイルのチェック	7-4
アプリケーションのデプロイまたはアンデプロイ	7-4
アプリケーションの管理	7-7
アプリケーションの起動、停止または再起動	7-7
アプリケーション内の EJB モジュールの更新	7-7
データソースの管理	7-8
アプリケーション固有のデータソースの作成	7-8
既存のデータソースのリスト出力、テストおよび削除	7-10
新しい構成への既存のデータソースの変換	7-10
スタンドアロン OC4J が稼働中または停止中の data-sources.xml ファイルの変換	7-10
アプリケーションおよび新しい data-sources.xml ファイル間での一貫性の確認	7-11
コネクタのデプロイまたはアンデプロイ	7-12

8 クラスタおよび OC4J グループの構成と管理

クラスタリングの概要	8-2
クラスタリングの動作	8-2
サポートされるクラスタリング・モデル	8-3
クラスタリングにおける変更点	8-3
Oracle Application Server クラスタ内の OC4J グループの作成および管理	8-4
OC4J インスタンスのグループの作成	8-6
グループ内の OC4J インスタンスの管理	8-7
OC4J インスタンスの追加作成	8-8
Application Server Control を使用した OC4J インスタンスの作成	8-8
createinstance ユーティリティを使用した OC4J インスタンスの作成	8-8
新規インスタンスへのアクセスおよび管理	8-10
グループからの OC4J インスタンスの削除	8-10
Application Server Control を使用した OC4J インスタンスの削除	8-10

removeinstance ユーティリティを使用した OC4J インスタンスの削除	8-10
クラスタでの変更のレプリケート	8-11
クラスタの構成	8-13
マルチキャストを使用した動的ノード検出の構成	8-13
opmnctl によるマルチキャスト検出の構成	8-15
opmnassociate によるマルチキャスト検出の構成	8-16
静的検出サーバーの構成	8-17
opmnctl による静的検出サーバー接続の構成	8-18
トポロジ間ゲートウェイの構成	8-19
ネットワーク接続の有無にかかわらず動作するマシンの構成	8-20
静的ノードツーノード通信の構成	8-21
クラスタのステータスの表示	8-22
opmnctl によるクラスタのステータスの表示	8-22
Application Server Control でのクラスタのステータスの表示	8-22
Oracle HTTP Server を使用したルーティングおよびロード・バランシングの構成	8-23
Web サーバーのルーティング ID を使用した OC4J リクエスト・ルーティングの制御	8-23
Application Server Control を使用したルーティング ID の変更	8-24
opmn.xml ファイルのルーティング ID の変更	8-25
mod_oc4j ロード・バランシング・オプションの設定	8-26
アプリケーション・マウント・ポイントの構成	8-29
アプリケーション・マウント・ポイントの動的構成の有効化	8-29
マウント・ポイント構成アルゴリズムの変更点	8-30
マウント・ポイントの構成データの表示	8-31
複数の JVM での OC4J インスタンスの稼働	8-32
OC4J インスタンス用の追加の JVM の作成	8-33
Application Server Control を使用して OC4J インスタンスに追加の JVM を作成する方法	8-33
opmn.xml ファイルの OC4J インスタンスに追加の JVM を作成する方法	8-33
複数の JVM の監視	8-34
ダイナミック・モニタリング・サービス JVM メトリックの監視	8-35
J2SE JVM 5.0 メトリックを監視するための jmxremote システム・プロパティの設定	8-35
Oracle Application Server 環境の J2SE 5.0 JVM メトリックの監視	8-36
スタンドアロン OC4J 環境の J2SE 5.0 JVM メトリックの監視	8-36

9 OC4J でのアプリケーションのクラスタリング

OC4J におけるアプリケーションのクラスタリングの概要	9-2
アプリケーションのクラスタリングにおける旧リリースの OC4J との相違点	9-2
サポートされなくなったアイランド	9-2
使用されなくなった loadbalancer.jar	9-2
アプリケーションのクラスタリング固有の非推奨の XML 要素	9-3
アプリケーションのクラスタリングの構成	9-3
アプリケーションのクラスタリングの有効化	9-3
レプリケーション・ポリシーの設定	9-4
アプリケーションの状態データをレプリケートする JVM の数の管理	9-6
同期レプリケーションまたは非同期レプリケーションの使用	9-6
マルチキャスト・レプリケーションの構成	9-7
マルチキャスト・レプリケーションに対する既存の JavaGroups 構成の使用	9-7
peer-to-peer レプリケーションの構成	9-8
動的 OPMN 管理の peer-to-peer レプリケーションの構成	9-8

静的 peer-to-peer レプリケーションの構成	9-9
OPMN のステート・レプリケーション用ポートの指定	9-10
データベース・レプリケーションの構成	9-10
アプリケーションの JVM、OC4J インスタンスおよび Application Server インスタンスの確認 ...	9-11
クラスタリングの無効化	9-12
<cluster> 要素の指定	9-12
10 タスク・マネージャとスレッド・プールの構成	
OC4J タスク・マネージャの構成	10-2
OC4J スレッド・プールの構成	10-2
スレッド・プール構成の変更	10-5
Application Server Control を使用したスレッド・プール構成の変更	10-6
MBeans を使用したスレッド・プール構成の変更	10-6
server.xml への <thread-pool> 要素の追加	10-6
アプリケーションのカスタム・スレッド・プールの構成	10-8
古いスレッド・プール形式からの変換	10-8
11 OC4J でのロギング	
OC4J により生成されるログ・ファイルの概要	11-2
プレーン・テキスト・ファイル・ロギングの使用法	11-3
テキスト・ファイル・ロギングの有効化または無効化	11-4
テキスト・ログ・ファイルの管理	11-4
テキスト・ログ・ファイルの表示	11-4
Oracle Diagnostic Logging (ODL) の使用法	11-5
ODL の有効化または無効化	11-5
ODL ログ・ファイルの管理	11-6
サイズベースのログ循環	11-6
時間ベースのログ循環	11-7
ODL ログ・ファイルの表示	11-8
OC4J ロギングの構成	11-8
OC4J コンポーネント・ログ出力の使用法と構成	11-8
OC4J ログ・ファイルの表示	11-9
oracle ログ出力の構成	11-10
LogViewer を使用した OC4J ログのアプリケーション・メッセージの表示	11-11
Application Server Control を使用したアプリケーション・ログ出力の構成	11-16
アプリケーションの log4j メッセージの OC4J ログへのリダイレクト	11-16
アプリケーションへの Oracle log4j アペンドの構成	11-17
アプリケーションによる log4j クラス・ライブラリの使用	11-20
12 OC4J での MBean の使用法	
OC4J での MBean および Java Management Extensions (JMX) のサポート	12-2
MBeans の概要	12-2
トップレベルの OC4J システム MBean の概要	12-3
MBean を介して加えた変更を有効にする時期	12-4
MBean データの永続性	12-5
システム MBean ブラウザの使用法	12-5
JMX 通知のサブスクリプト	12-6

アプリケーション固有の MBean の使用方法	12-6
-------------------------------	------

13 OC4J での Web サイトの管理

OC4J の Web サイトの概要	13-2
Web サイト接続データの構成	13-3
スタンドアロン OC4J インストールでの Web サイト・データの構成	13-3
OPMN 管理 OC4J インスタンスでの Web サイト・データの構成	13-3
Application Server Control でのポート範囲の変更	13-3
opmn.xml におけるプロトコルおよびポート範囲の変更	13-4
opmnctl による Web サイトの構成	13-5
OC4J での Web サイトの新規作成	13-8
Web サイト構成ファイルの作成	13-8
server.xml での Web サイト構成ファイルの参照	13-9
opmn.xml での Web サイトの接続データの定義	13-10
Web サイト間での Web アプリケーションの共有	13-10
Cookie ドメインの指定	13-11
OC4J でのセキュアな Web サイトの構成	13-11
セキュアな Web サイトの構成ファイルの作成	13-12
クライアント認証の要求	13-13
OC4J によるクライアント認証のリクエスト	13-13
Web サイトの起動と停止	13-14
Web サイトのアクセス・ロギングの構成	13-14
テキストベースのアクセス・ロギングの構成	13-15
テキスト・アクセス・ログ・ファイルの表示	13-16
ODL アクセス・ロギングの構成	13-16
ODL アクセス・ログ・ファイルの表示	13-17
Web モジュールまたはアプリケーションに対するアクセス・ロギングの有効化または無効化 ...	13-17

14 OC4J への DTD および XSD の登録

登録する XSD の検証	14-2
DTD または XSD の登録	14-2

A OC4J のトラブルシューティング

問題と解決策	A-2
最大同時タイマーに関する警告	A-2
java.lang.OutOfMemory エラー	A-3
アプリケーションのパフォーマンスがガベージ・コレクションの一時停止の影響を受ける	A-3
無効または不要なライブラリ要素によるパフォーマンスの低下	A-4
ClassCastException および ClassNotFoundException エラー	A-4
OC4J が起動できない: Java コンパイラを検出できない	A-4
アプリケーションのクラスタリング時にエラーが発生する	A-5
JDK 5.0 から JDK 1.4.2 へのダウングレード時にエラーが発生する	A-5
JMX MBeanServer および MBeanServerConnection インタフェースでサポートされていない メソッド	A-6
Oracle Application Server でアプリケーションを起動する際に OC4J がハングする	A-7
その他のヘルプ	A-7

B OC4J で使用される構成ファイル

OC4J で使用される XML 構成ファイルの概要	B-2
OC4J サーバー構成ファイル (server.xml) の要素	B-6
server.xml ファイルの例	B-6
<application-server>	B-7
<application>	B-8
<code-source>	B-9
<custom-thread-pool>	B-9
<execution-order>	B-10
<global-application>	B-10
<global-thread-pool>	B-11
<global-web-app-config>	B-13
<import-shared-library>	B-13
<init-param>	B-14
<j2ee-logging-config>	B-14
<java-compiler>	B-15
<javacache-config>	B-16
<jms-config>	B-16
<log>	B-16
<rmi-config>	B-17
<shared-library>	B-17
<shutdown-class>	B-18
<startup-class>	B-18
<thread-pool>	B-19
<transaction-manager-config>	B-20
<web-site>	B-21
<work-manager-thread-pool>	B-21
Web サイト構成ファイル (*-web-site.xml) の概要	B-22
<web-site>	B-22
<description>	B-24
<frontend>	B-25
<web-app>	B-25
<default-web-app>	B-28
<user-web-apps>	B-28
<access-log>	B-29
<odl-access-log>	B-29
<ssl-config>	B-29

C セッション状態表の概要

D サード・パーティ・ライセンス

ANTLR	D-2
The ANTLR License	D-2
Apache	D-2
The Apache Software License	D-3
Apache SOAP	D-7
Apache SOAP License	D-7
DBI モジュール	D-10
Perl Artistic License	D-10

Preamble	D-10
Definitions	D-11
FastCGI	D-12
FastCGI Developer's Kit License	D-12
Module mod_fastcgi License	D-13
Info-ZIP Unzip パッケージ	D-14
The Info-ZIP Unzip Package License	D-14
JSR 110	D-14
Jaxen	D-15
The Jaxen License	D-15
JGroups	D-16
The GNU License	D-16
mod_mm および mod_ssl	D-23
OpenSSL	D-23
OpenSSL License	D-23
Perl	D-25
Perl Kit Readme	D-25
mod_perl 1.29 License	D-26
mod_perl 1.99_16 License	D-27
Perl Artistic License	D-30
Preamble	D-30
Definitions	D-30
SAXPath	D-32
The SAXPath License	D-32
W3C DOM	D-33
The W3C License	D-33

索引

はじめに

このマニュアルは、スタンドアロン環境と OPMN 管理 (Oracle Application Server) 環境の両方における Oracle Containers for J2EE (OC4J) の構成および管理に関する主要なリファレンスです。このマニュアルは実質的に、旧バージョンの OC4J とともにリリースされた『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』および『Oracle Application Server Containers for J2EE スタンドアロン・ユーザーズ・ガイド』のかわりとなります。

この章の内容は次のとおりです。

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)
- [サポートおよびサービス](#)

対象読者

このマニュアルは、次の読者を対象としています。

- OC4J インストールの構成および管理を担当するシステム管理者
- スタンドアロン環境で OC4J を使用する Java アプリケーション開発者

このマニュアルでは、次の内容を理解していることを前提にしています。

- Java 2 Platform Enterprise Edition (J2EE) 環境
- 一般的なサーバーおよびシステム管理の概念
- 一般的な Web テクノロジ
- Java プログラミング言語

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。

Oracle サポート・サービスへの TTY アクセス

アメリカ国内では、Oracle サポート・サービスへ 24 時間年中無休でテキスト電話 (TTY) アクセスが提供されています。TTY サポートについては、(800)446-2398 にお電話ください。アメリカ国外からの場合は、+1-407-458-2479 にお電話ください。

関連ドキュメント

詳細は、次の Oracle ドキュメントを参照してください。

次の OC4J ドキュメントを参照できます。

- 『Oracle Containers for J2EE デプロイメント・ガイド』
このマニュアルでは、OC4J 環境にアプリケーションをデプロイするための情報および手順を示します。Oracle Enterprise Manager 10g Application Server Control に付属のデプロイ・プラン・エディタについても説明します。
- 『Oracle Containers for J2EE 開発者ガイド』
このマニュアルでは、サーブレット、EJB、JSP コンテナなど特定のコンテナに固有の問題（クラスのロードなど）以外の、OC4J で稼働するアプリケーションを作成する開発者が一般に関心を持つ項目について説明します。
- 『Oracle Containers for J2EE サーブレット開発者ガイド』
このマニュアルでは、OC4J におけるサーブレットおよびサーブレット・コンテナの使用（基本的なサーブレット開発、JDBC および EJB の使用など）に関する情報をサーブレット開発者に示します。
- 『Oracle Containers for J2EE JavaServer Pages 開発者ガイド』
このマニュアルでは、OC4J における JavaServer Pages の開発と JSP の実装およびコンテナに関する情報を示します。コマンドライン・トランスレータや OC4J 固有の構成パラメータなど、Oracle の機能についても説明します。
- 『Oracle Containers for J2EE JSP タグ・ライブラリおよびユーティリティ・リファレンス』
このマニュアルでは、OC4J に付属するタグ・ライブラリ、Enterprise JavaBeans (EJB) モジュール、その他の Java ユーティリティの概念と、詳しい構文情報および使用方法を示します。
- 『Oracle Containers for J2EE サービス・ガイド』
このマニュアルでは、OC4J に付属する標準ベースの Java サービス（JTA、JNDI、JMS、JAAS、Oracle Application Server Java Object Cache など）に関する情報を示します。
- 『Oracle Containers for J2EE セキュリティ・ガイド』
OC4J 固有のセキュリティ機能および実装について説明します。Java Authentication and Authorization Service (JAAS) とその他の Java セキュリティ・テクノロジーの使用方法についても説明します。
- 『Oracle Containers for J2EE Enterprise JavaBeans 開発者ガイド』
このマニュアルでは、OC4J における Enterprise JavaBeans (EJB) モジュールの開発と EJB の実装およびコンテナに関する情報を示します。
- 『Oracle Containers for J2EE リソース・アダプタ管理者ガイド』
J2EE Connector Architecture 機能の概要と、OC4J におけるリソース・アダプタの構成および監視方法を説明します。

Oracle Application Server のマニュアル：

- 『Oracle Application Server Web Services 開発者ガイド』
このマニュアルでは、OC4J および Oracle Application Server における Web サービスの開発および構成について説明します。

- 『Oracle Application Server Web Services アドバンスド開発者ガイド』
このマニュアルでは、高度な Web サービス・アセンブリについて説明します。たとえば、一般的な相互運用性の問題を診断する方法、Web サービス管理機能（信頼性、監査、ロギングなど）を有効にする方法、Java の値タイプのカスタム・シリアライズを使用する方法について説明します。
また、Web Service Invocation Framework (WSIF)、Web サービス・プロバイダ API、メッセージ添付ファイル、管理機能（信頼性、ロギング、監査）の使用方法についても説明します。さらに、転送手段として JMS を使用する方法など、Web サービスの代替方法についても説明します。
- 『Oracle Application Server Web Services セキュリティ・ガイド』
このマニュアルでは、OC4J および Oracle Application Server における Web サービスのセキュリティおよび構成について説明します。

表記規則

このマニュアル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 Graphical User Interface 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。
イタリック体	イタリックは、ユーザーが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、段落内のコマンド、URL、サンプル内のコード、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。

Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービス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.com/lang/jp/support/index.html>

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://www.oracle.com/technology/global/jp/documentation/index.html>

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://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3

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.com/lang/jp/index.html>

<http://www.oracle.com/technology/global/jp/index.html>

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。

Oracle Containers for J2EE (OC4J) の概要

この章では、Oracle Containers for J2EE (OC4J) の概要を説明します。この章の内容は次のとおりです。

- OC4J の概要
- OC4J における J2EE サポート
- OC4J の新機能および変更された機能
- スタンドアロン構成での OC4J
- Oracle Application Server 構成での OC4J
- OC4J のアプリケーション階層の概要

OC4J の概要

Oracle Containers for J2EE 10g (10.1.3.4.0)、すなわち OC4J は、Java 2 Enterprise Edition (J2EE) 1.4 に完全準拠した環境を提供します。OC4J には、J2EE 仕様で規定されたコンテナ、API およびサービスがすべて装備されています。

OC4J は、次の 2 つの構成で配布されます。

- **スタンドアロン構成** : OC4J は単一のスタンドアロン・インスタンスとしてインストールされ、自己完結型コンポーネントとして直接起動、管理および停止されます。

この構成の詳細は、1-7 ページの「[スタンドアロン構成での OC4J](#)」を参照してください。

- **管理構成** : OC4J は OC4J インスタンスのグループの一部としてインストールされ、Oracle Application Server のコンポーネントとして管理されます。

Oracle Application Server 10g リリース 3 (10.1.3.4.0) では、**グループ**は同じ**クラスタ・トポロジ**に属する OC4J インスタンスの同期セットです (クラスタ・トポロジは、2 つ以上の粗結合の Oracle Application Server ノードで構成されています)。グループのすべての OC4J インスタンスで、構成、管理およびデプロイ操作を同時に実行できます。

管理 OC4J インストールには、少なくとも Oracle Process Manager and Notification Server (OPMN) が含まれます。OPMN は、OC4J などの Oracle Application Server の各種コンポーネントを管理します。

通常、このインストールには、1 つ以上の Oracle HTTP Server (OHS) インスタンスも含まれます。このインスタンスは、Web 通信およびロード・バランシング機能を提供します。

詳細は、1-8 ページの「[Oracle Application Server 構成での OC4J](#)」を参照してください。

OC4J は、完全に Java で記述されており、Java Platform Standard Edition (Java SE) Development Kit (JDK) 6、Java Platform 2 Standard Edition (J2SE) Development Kit (JDK) 5.0 (JDK 1.5 と呼ばれる)、または JDK 1.4.2 の Java 仮想マシン (JVM) で稼働します。OPMN 管理の OC4J では、JDK 5.0 はサーバー・バイナリとともにインストールされ、デフォルトで OC4J インスタンスを起動するために使用されます。スタンドアロン OC4J の場合は、JDK を指定する必要があります。複数の JVM で実行する OC4J インスタンスを構成できます。

OC4J ドキュメントでは、Java プログラミング、J2EE テクノロジ、Web アプリケーションおよび EJB アプリケーションのテクノロジについて基礎知識があることを前提にしています。これには、/WEB-INF および /META-INF ディレクトリなどのデプロイ規則が含まれます。

OC4J における J2EE サポート

OC4J 10g (10.1.3.4.0) では、表 1-1 に示す標準 J2EE 仕様がサポートされています。

表 1-1 サポートされる J2EE 仕様

J2EE 仕様	OC4J でサポートされるバージョン
JavaServer Pages (JSP)	2.0
Servlet	2.4
Enterprise JavaBeans (EJB)	2.1、3.0 (完全な EJB 3.0 および JPA 実装)
Java Management Extensions (JMX)	1.2
J2EE Management	1.0
J2EE Application Deployment	1.1
Java Transaction API (JTA)	1.0
Java Message Service (JMS)	1.1
Java Naming and Directory Interface (JNDI)	1.2
Java Mail	1.2
Java Database Connectivity (JDBC)	3.0
Java Authentication and Authorization Service (JAAS) Provider	1.0
J2EE Connector Architecture	1.5
Enterprise Web Services	1.1
Java API for XML-Based RPC (JAX-RPC)	1.1
SOAP with Attachments API for Java (SAAJ)	1.2
Java API for XML Processing (JAXP)	1.2
Java API for XML Registries (JAXR)	1.0.5

OC4J の新機能および変更された機能

次の各項では、Oracle Containers for J2EE 10g (10.1.3.x) の新機能の他に、旧リリースからの機能の変更点について説明します。

OC4J の新機能

Oracle Containers for J2EE 10g (10.1.3.x) には、次の各項で説明する多くの新機能および追加機能があります。

- [Web サービスのサポート](#)
- [アプリケーションの管理およびデプロイに関する新しい J2EE 1.4 仕様のサポート](#)
- [Enterprise JavaBeans 3.0 のサポート](#)
- [Oracle Application Server TopLink のサポート](#)
- [Oracle Job Scheduler](#)
- [新しい 2 フェーズ・コミット・トランザクション・コーディネータ機能](#)
- [汎用 JMS リソース・アダプタの拡張機能](#)

Web サービスのサポート

OC4J では、JAX-RPC 1.1 などの J2EE 1.4 標準に従って Web サービスを完全サポートします。Web サービスの相互運用性も同様にサポートします。

- Enterprise Web Services 1.1 仕様のサポート
- EJB 2.1 Web サービス・エンド・ポイント・モデル
- JSR 109 クライアントおよびサーバーのデプロイメント・モデル
- CORBA Web サービス : 既存の基本 CORBA サーバントの Web サービスとしてのラッピング、および IDL からの WSDL の自動生成のサポート
- 呼出しおよび終了の形式 (RPC/ リテラル、RPC/ エンコード、ドキュメント/ リテラル) などの Web サービスの動作をカスタマイズするためのソース・コードのアノテーション、Java と XML のマッピングのカスタマイズ、およびセキュリティ強化のサポート
- データベースおよび JMS Web サービス

アプリケーションの管理およびデプロイに関する新しい J2EE 1.4 仕様のサポート

OC4J では、J2EE 環境におけるアプリケーションのデプロイおよび管理について新しい標準を定義する次の仕様をサポートします。

- J2EE Application Deployment API (JSR-88) : J2EE アプリケーションおよびモジュールを構成し、J2EE と互換性のある環境にデプロイするための標準 API を定義します。OC4J 実装には、コンポーネントを OC4J にデプロイするために必要な OC4J 固有の構成データを含むデプロイ・プランを作成または編集する機能が含まれます。
- Java Management Extensions (JMX) 1.2 仕様 : J2EE 環境でリソース (サービス、アプリケーションなど) を管理するための標準インタフェースを作成できるようにします。JMX の OC4J 実装には、OC4J サーバーとその内部で稼働するアプリケーションを完全に管理するために使用できる JMX クライアントが装備されます。
- J2EE Management 仕様 (JSR-77) : J2EE 環境でアプリケーションを管理するための標準コンポーネントを作成できるようにします。

Enterprise JavaBeans 3.0 のサポート

OC4J 10g (10.1.3.4.0) では、EJB の注釈および依存性インジェクションのサポートを含め、Enterprise JavaBeans 3.0 の最終仕様を完全にサポートしています。この最終仕様は、次の Web サイトで入手できます。

<http://java.sun.com/products/ejb/>

注意 : EJB 3.0 のサポートを有効にするには、OC4J で JDK 6 または 5.0 を使用する必要があります。JDK 5.0 は、OPMN 管理の OC4J インスタンスにおいてデフォルトで JDK5.0 が使用される 10g (10.1.3.4.0) リリースに同梱されています。

Oracle Application Server TopLink のサポート

Oracle Application Server TopLink は、様々な Java 2 Enterprise Edition (J2EE) および Java アプリケーションのアーキテクチャで使用するための高度なオブジェクト永続性フレームワークです。Oracle TopLink には、Bean 管理の永続性 (BMP) の開発を容易にする OC4J のコンテナ管理の永続性 (CMP) コンテナおよびベース・クラスのサポートが含まれます。

Oracle Job Scheduler

Oracle Job Scheduler は、J2EE アプリケーションに非同期スケジュール・サービスを提供します。主な特長として、ジョブの送信、制御および監視を行う機能があり、各ジョブは作業実行時に動作する作業単位として定義されます。

新しい 2 フェーズ・コミット・トランザクション・コーディネータ機能

OC4J の新しい分散トランザクション・マネージャは、Oracle だけでなく他のベンダーや JMS プロバイダ (IBM WebSphere MQ など) のデータベースを含むあらゆるタイプの XA リソース間で 2 フェーズ・コミット・トランザクションを調整できます。障害時の自動トランザクション・リカバリもサポートされます。

汎用 JMS リソース・アダプタの拡張機能

汎用 JMS リソース・アダプタを、IBM WebSphere MQ JMS バージョン 5.3 のみではなく、OC4J 10g (10.1.3.4.0) に付属している Oracle Enterprise Messaging Service (OEMS) の OC4J プラグインとして使用できるようになりました。

遅延トランザクション登録のサポートが追加されているため、JMS 接続をキャッシュすることができます。また、JMS 接続は、これまでどおりグローバル・トランザクションに正しく参加できます。

汎用 JMS リソース・アダプタのエラー処理が改善されています。エンドポイントは、プロバイダまたはシステムの障害後に自動的に再試行され、onMessage () エラーは正しく処理されるようになりました。

admin_client.jar の新しいコマンドおよびリモート・クライアント

admin_client.jar ユーティリティには、データソース、および OC4J JMS コネクション・ファクトリや接続先を管理するための新しいコマンドがあります。admin_client.jar コマンドは、リモートの管理クライアント・ユーティリティでも使用できます。これらのコマンドをコマンドライン・ツールまたは関連する JMX MBeans を介して使用し、データソース、および JMS コネクション・ファクトリや接続先に関する情報の追加、削除および取得を実行できます。詳細は、第 6 章「admin_client.jar ユーティリティの使用方法」を参照してください。

旧リリースからの構成ファイルの変更点

Oracle Application Server のコンポーネントとしてインストールされた OC4J インスタンス、およびスタンドアロン OC4J で使用される構成ファイルに、次の変更が加えられています。ここに示すファイルはすべて ORACLE_HOME/j2ee/instance/config にデフォルトでインストールされます。ここで、instance は OC4J インスタンス名を表します。

application.xml

- <persistence> 要素は、新しい system-application.xml ファイルに移動されました。
- <jazn> 要素で指定する OC4J インスタンスのセキュリティ構成ファイルは、新しい system-jazn-data.xml ファイルになりました。<jazn> の詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。
- <orion-application> ルート要素の default-data-source 属性で指定するデフォルトのデータソースは、スタンドアロン OC4J と Oracle Application Server の両方で jdbc/OracleDS になりました。
- PortComponentLinkResolver の <ejb-module> 要素は、削除されました。
- default アプリケーションの ODL ロギングを有効にするための <odl> 要素が <log> のサブ要素として追加されましたが、コメント・アウトされています。

ascontrol-web-site.xml

- このファイルは、スタンドアロン OC4J と Oracle Application Server の両方から削除されました。OC4J にデプロイされる Application Server Control インスタンスは、デフォルトで default-web-site.xml にバインドされ、/em コンテキスト・ルートからアクセスできます。

default-web-site.xml

- このファイルは、スタンドアロン OC4J と Oracle Application Server の両方で使用されるデフォルト Web サイトを構成します。OC4J インスタンスにデプロイされる Application Server Control インスタンスを含むすべてのアプリケーションは、デフォルトで、このファイルに指定されたコンテキスト・ルートを使用してデフォルト Web サイトを介してアクセスされます。

global-web-application.xml

- <dtd> 要素は、Oracle Application Server 用のこのファイルから削除されました。
- rmi-tunnel サブレット定義の <url-pattern> 要素は、スタンドアロン OC4J と Oracle Application Server の両方で rmiTunnel/* を指定しています。

http-web-site.xml

- このファイルは、スタンドアロン OC4J と Oracle Application Server の両方から削除されました。OC4J インスタンスにデプロイされるアプリケーションはすべて、デフォルトで default-web-site.xml にバインドされます。

j2ee-logging.xml

- この新しいファイルは、Java ログ出力 (oracle ログ出力など) を構成するために使用されます。

jazn-data.xml

- このファイルには、OC4J インスタンスのセキュリティ構成が含まれなくなりました。この構成は、新しい system-jazn-data.xml ファイルで定義されるようになりました。ただし、アプリケーション・レベルでユーザーおよびロールを定義するよう、jazn-data.xml ファイルを指定できます。jazn-data.xml および system-jazn-data.xml ファイルの詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。

oc4j-connectors.xml

- <connector> 要素の location 属性は、datasources および Oracle Enterprise Messaging Service (OEMS) コネクタに対して指定されなくなりました。

server.xml

- http-web-site.xml および ascontrol-web-site.xml を示す <web-site> 要素は削除されました。1つの要素が、デフォルト Web サイトの構成ファイルである default-web-site.xml を示すようになりました。
- 複数の <shared-library> 要素が追加され、それぞれが OC4J とともにインストールされる共有ライブラリを参照しています。
- OC4J プロセスおよび OC4J インスタンスにデプロイされるアプリケーションで使用できるようにスレッド・プールを定義するための <thread-pool> 要素が、server.xml に追加されました。この要素は、OC4J 10g (10.1.3.4.0) では非推奨の <global-thread-pool> および <work-manager-thread-pool> 要素を置き換えます。
- アプリケーションに個別のカスタム・スレッド・プールを定義するための <custom-thread-pool> 要素が、server.xml ファイルに追加されました。

system-application.xml

- このファイルは、system アプリケーションの構成を指定するために新しく追加されたファイルです。この新しい内部コンポーネントの詳細は、1-10 ページの「[system アプリケーション](#)」を参照してください。

system-jazn-data.xml

- この新しいファイルに、OC4J インスタンスのセキュリティ構成が含まれています。このファイルは実質的に、jazn-data.xml のかわりとなります。jazn-data.xml および system-jazn-data.xml ファイルの詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。

スタンダロン構成での OC4J

スタンダロン（非管理）の OC4J 構成は、管理しやすく堅牢な J2EE 準拠のコンテナを提供します。この構成では、単一の OC4J インスタンスが 1 つの `ORACLE_HOME` ディレクトリ（Oracle ソフトウェアがインストールされるルート・ディレクトリ）にインストールされます。スタンダロン OC4J 構成には次のコンポーネントが含まれます。

- Oracle Containers for J2EE 10g (10.1.3.4.0)。
- Oracle Enterprise Manager 10g Application Server Control。デフォルトで OC4J とともにインストールされる Web ベースの管理アプリケーションです。

Application Server Control は、インストールと同時に使用できます。この管理インタフェースの使用方法は、3-2 ページの「[Oracle Enterprise Manager 10g Application Server Control](#)」を参照してください。

インストール

スタンダロン OC4J のディストリビューション（Application Server Control を含む）は、ZIP アーカイブとして提供されます。手順は、[第 2 章「スタンダロン OC4J のインストール」](#)を参照してください。

管理

スタンダロン OC4J は、インスタンスとともにインストールされる Application Server Control アプリケーション、組込みコマンドライン・ユーティリティの 1 つである `admin_client.jar`、または OC4J Ant タスクを使用して、スタンダロン OC4J インスタンスとして管理されます。これらのツールの詳細は、[第 3 章「OC4J の管理用ツール」](#)を参照してください。

スタンダロン OC4J は、OC4J Ant タスクおよび `admin_client.jar` が含まれる管理クライアント・ユーティリティを使用して、リモートで管理することもできます。リモート管理の詳細は、6-5 ページの「[リモート管理クライアントのダウンロードおよび抽出](#)」を参照してください。

OC4J で提供される `admin.jar` ツールは、スタンダロン OC4J サーバーでのみ管理タスクを実行できます。このツールの使用方法の詳細は、[第 7 章「admin.jar ユーティリティの使用法」](#)を参照してください。

起動、停止および再起動

スタンダロン構成では、OC4J インスタンスは、`oc4j` コマンド・スクリプト、または実行可能な `oc4j.jar` アーカイブを使用して起動します。起動オプションおよびシステム・プロパティは、コマンド・スクリプトの場合は起動前に、`oc4j.jar` 直接実行モデルを使用する場合は起動時に設定します。

詳細は、5-2 ページの「[スタンダロン環境での OC4J の起動](#)」を参照してください。

スタンダロン OC4J サーバーは、`admin_client.jar` または `admin.jar` コマンドライン・ユーティリティか、`oc4j` コマンド・スクリプトを使用して停止および再起動できます。詳細は、5-3 ページの「[スタンダロン環境での OC4J の停止](#)」、5-5 ページの「[スタンダロン環境での OC4J インスタンスの再起動](#)」または 7-3 ページの「[スタンダロン環境での OC4J の停止および再起動](#)」を参照してください。

バックアップ、リストアおよび障害リカバリ機能

スタンドアロン OC4J 構成には、バックアップ、リストアおよび障害リカバリ機能はありません。

Web 通信

スタンドアロン環境での Web 通信は、組込み OC4J Web サーバーを介して提供されます。このサーバーは、Oracle HTTP Server を使用せずに HTTP および HTTPS 通信をネイティブにサポートします。

デフォルト Web サイトは、default-web-site.xml ファイルに定義されています。このファイルは、ポート 8888 にデフォルトの HTTP リスナーを指定します。このファイルを変更して使用することで、追加の Web サイトを定義できます。OC4J で追加の Web サイトを作成する手順は、[第 13 章「OC4J での Web サイトの管理」](#)を参照してください。

Oracle Application Server 構成での OC4J

この構成では、OC4J は、Oracle Application Server クラスタ内の 1 つ以上の OC4J インスタンスのグループに、Oracle Application Server のコンポーネントとしてインストールされます。通常の構成には、次のコンポーネントが含まれます。

- Oracle Containers for J2EE 10g (10.1.3.4.0)。1 つ以上のグループの 1 つ以上のインスタンスです。
- Oracle Enterprise Manager 10g Application Server Control。デフォルトで OC4J とともにインストールされる Web ベースの管理アプリケーションです。
- Oracle HTTP Server 1.3。フロントエンドの Web 通信およびロード・バランシング機能を提供します。
- Oracle Process Manager and Notification Server (OPMN)。インストールされた他のコンポーネント (OC4J や Oracle HTTP Server など) の起動や停止、および監視に使用されません。OPMN には、コンポーネント間の通信を管理する Oracle Notification Server (ONS) が含まれます。

Oracle Application Server では、クラスタ・トポロジ内の OC4J インスタンスのグループでの、HTTP セッションおよびステートフル・セッション Enterprise JavaBean (EJB) のレプリケーションとロード・バランシングをサポートします。詳細は、[第 9 章「OC4J でのアプリケーションのクラスタリング」](#)を参照してください。

Oracle Application Server クラスタ内での接続は、Oracle Application Server コンポーネント (OC4J や Oracle HTTP Server など) 間の通信を管理する Oracle Notification Server (ONS) の機能により提供されます。ONS サーバーは、Oracle Process Manager and Notification Server (OPMN) のコンポーネントで、すべての Oracle Application Server ホストにデフォルトでインストールされます。

Oracle Universal Installer には、次のような複数のインストール・オプションがあります。

■ 統合された Web サーバー、J2EE Server およびプロセス管理

この構成では、すべてのコンポーネント (OC4J、Oracle HTTP Server および OPMN) が単一の `ORACLE_HOME` ディレクトリにインストールされます。

この `ORACLE_HOME` ディレクトリ内には、複数の OC4J インスタンスを作成できます。それぞれ 1 つ以上の OC4J インスタンスのホストとなる複数のホスト・マシンを、1 つの Oracle Application Server クラスタに含めることができます。

■ J2EE Server およびプロセス管理

このインストールには、OC4J および OPMN が含まれます。開発またはテスト用に、スタンドアロン OPMN 管理 OC4J インスタンスとして使用できます。また、Oracle Application Server クラスタに含めることができます。

■ Web サーバーおよびプロセス管理

このインストールには、Oracle HTTP Server および OPMN が含まれます。通常は Oracle Application Server クラスタのフロントエンド Web リスナーとして機能する、スタンドアロン Oracle HTTP Server インスタンスとして使用できます。

インストール

Oracle Universal Installer を使用して、様々なコンポーネントがインストールされます。インストールされた各コンポーネントを監視できるようにするには、OPMN をすべての `ORACLE_HOME` ディレクトリにインストールする必要があります。

管理

管理タスクは、次のツールを使用して実行できます。

- Web ベースの Application Server Control のユーザー・インタフェース
- `admin_client.jar` コマンドライン・ツール
- OC4J Ant タスク
- `admin.jar` コマンドライン・ツール (スタンドアロン OC4J サーバーに対してのみ)

これらのツールの詳細は、第 3 章「OC4J の管理用ツール」を参照してください。

Oracle Application Server のクラスタ化された環境では、単一の Application Server Control インスタンスを使用してクラスタ内のすべての OC4J インスタンスを管理できます。このアプリケーションの詳細は、3-2 ページの「Oracle Enterprise Manager 10g Application Server Control」を参照してください。

OC4J には、Oracle Application Server クラスタ内の OC4J インスタンスのグループ、OPMN 管理の OC4J インスタンス、またはスタンドアロン OC4J サーバーで管理タスクを実行するための一連の Ant タスクが含まれています。Ant タスクの詳細、およびアプリケーション作成プロセスにそれらのタスクを統合するためのガイドラインは、『Oracle Containers for J2EE デプロイメント・ガイド』の「OC4J Ant タスクによるデプロイ」を参照してください。

OC4J で提供されている `admin_client.jar` ツールは、Oracle Application Server クラスタ内の OC4J インスタンスのグループ、または OC4J インスタンスで管理タスクを実行できます。また、管理クライアント・ユーティリティのディストリビューションである `oc4j_admin_client_101340.zip` には、次に示す 3 つの方法でリモート・クライアントから管理操作を実行する際に必要なクライアント・サイドの JAR が含まれています。

- OPMN 管理またはスタンドアロン OC4J インスタンスに対するリモートからの `admin_client.jar` コマンドの使用
- OPMN 管理またはスタンドアロン OC4J インスタンスに対するリモートからの OC4J Ant タスクの使用
- JMX プログラム・クライアントを使用したリモートからの OC4J の管理

リモート管理の詳細は、6-5 ページの「リモート管理クライアントのダウンロードおよび抽出」を参照してください。

起動および停止

管理環境では、すべてのコンポーネント (OC4J など) の起動および停止に OPMN を使用する必要があります。詳細は、5-2 ページの「Oracle Application Server 環境での OC4J の起動」を参照してください。

OC4J ランタイム・オプションおよびシステム・プロパティは、OPMN 構成ファイル `opmn.xml` に手動で設定できます。詳細は、第 4 章「OC4J ランタイムの構成」を参照してください。

バックアップ、リストアおよび障害リカバリ機能

これらの機能は、管理 Oracle Application Server 構成で使用できます。

Web 通信

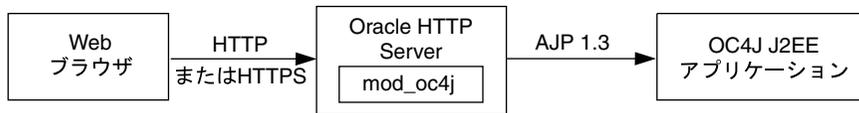
スタンドアロン OPMN 管理の OC4J インスタンス（「J2EE Server およびプロセス管理」インストール・タイプ）は、組込み OC4J Web サーバーを使用して、HTTP (S) リクエストを直接受信し応答します。

また、OC4J との Web 通信は Oracle HTTP Server および `mod_oc4j` モジュールを使用して管理できます。OHS はフロント・エンド・リスナーとして機能し、`mod_oc4j` モジュールは Apache JServ Protocol (AJP) 1.3 を使用して HTTP リクエストを OC4J インスタンスに転送します。

Oracle HTTP Server と OC4J 間のリクエスト / レスponse・フローは次のとおりです。

1. 受信 HTTP リクエストが Oracle HTTP Server リスナーによって受信されます。
2. Oracle HTTP Server は、`mod_oc4j` モジュールを介してリクエストを OC4J インスタンスに渡します。Oracle HTTP Server と OC4J 間の接続では、OC4J の起動時に決定されたポート番号で Apache JServ Protocol (AJP) が使用されます。

図 1-1 OHS を使用した OC4J の Web 通信



リクエスト元アプリケーションとして機能する OC4J インスタンスにリクエスト URL をマップするマウント・ポイントは、アプリケーションのデプロイ時に `mod_oc4j` に動的に作成されます。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・ポイントに対応する OC4J インスタンスにルーティングされます。

OHS および `mod_oc4j` モジュールの構成および管理の詳細は、『Oracle HTTP Server 管理者ガイド』を参照してください。

OC4J のアプリケーション階層の概要

この項では、OC4J インスタンス内のアプリケーション階層の概要を説明します。

system アプリケーション

`system` アプリケーションは、Oracle Containers for J2EE 10g (10.1.3.4.0) の内部コンポーネントです。このアプリケーションは、OC4J の初回起動時に、OC4J インスタンスまたはスタンドアロン OC4J に自動デプロイされます。

このアプリケーションは、主に OC4J へのアプリケーションのデプロイまたは再デプロイに関連する問題に対処するために追加されました。アプリケーション階層のルートに存在し、OC4J の起動時に必要なクラスおよび構成を提供します。たとえば、その他のすべてのデプロイ済アプリケーション（Oracle JDBC ドライバ実装や XML パーサー実装など）によってデフォルトでインポートされる共有ライブラリを提供します。

`system` アプリケーションは、OC4J の内部コンポーネントにすぎません。アプリケーションをデプロイすることも、別のアプリケーションの親として宣言することもできません。`default` アプリケーションは、すべてのデプロイ済アプリケーションのデフォルトの親として引き続き機能します。

`system` アプリケーションの構成は、`system-application.xml` に定義されています。このファイルは、`ORACLE_HOME/j2ee/instance/config` にデフォルトでインストールされません。

重要: system は、OC4J の起動に不可欠な主要な内部コンポーネントであるため、<jazn> および <log> タグを除き、system-application.xml ファイルは変更しないでください。

<jazn> タグは、セキュリティ・プロバイダ、OC4J セキュリティ構成ファイル (system-jazn-data.xml) の場所、あるいはその両方への変更を指定するために、必要に応じて変更できます。<jazn> および system-jazn-data.xml ファイルの詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。

<log> タグを変更して、システム・ログ・ファイルを循環できます。

default アプリケーション

default アプリケーションは、アプリケーション階層の system の直下に存在します。OC4J インスタンスにデプロイされたその他すべての J2EE アプリケーションのデフォルトの親として引き続き機能します。したがって、default アプリケーションに対して定義された構成パラメータはすべて、アプリケーション・レベルで明示的に上書きされないかぎり、その他すべてのアプリケーションによって継承されます。

スタンドアロン Web モジュール (WAR ファイル) を default アプリケーションにデプロイすることもできます。

default アプリケーションの構成は、application.xml に定義されています。このファイルは、ORACLE_HOME/j2ee/instance/config にデフォルトでインストールされます。

グローバル Web アプリケーション

グローバル Web アプリケーションは、default アプリケーションの Web モジュール・コンポーネントです。OC4J インスタンスにデプロイされたすべての Web モジュールにデフォルトで適用される構成データを提供します。また、すべてのサーブレットにデフォルトで適用される初期化パラメータも含まれます。

デフォルトの Web アプリケーションの構成ファイルは、global-web-application.xml です。このファイルは、ORACLE_HOME/j2ee/instance/config にデフォルトでインストールされます。このファイルには、OC4J インスタンスにデプロイされたすべての Web モジュールにデフォルトで適用されるパラメータの他に、すべてのサーブレットに適用されるサーブレット初期化パラメータも含まれます。これらのパラメータ値はいずれも、Web モジュールの orion-web.xml ファイルの対応する値によって上書きできます。

スタンドアロン OC4J インストールでは、デフォルトの Web アプリケーションのルート・ディレクトリは、j2ee/home/default-web-app です。デフォルトの Web アプリケーションにデプロイするには、Web アプリケーションの標準のディレクトリ構造で、JSP ページおよびクラス・ファイルをこのディレクトリ内に配置します。

J2EE アプリケーション

デフォルトでは、OC4J インスタンスにデプロイされたアプリケーションは、指定された親アプリケーションから、あるいは他の親が指定されていない場合は default アプリケーションから、構成パラメータを継承します。ただし、アプリケーションの orion-application.xml ディスクリプタに設定されたパラメータ値は、親から継承された該当するパラメータに優先します。

Web モジュールは親 J2EE アプリケーションに含まれている必要があります。WAR ファイルは通常、親 J2EE アプリケーションを定義する EAR ファイルとともにパッケージおよびデプロイされます。しかし、WAR ファイルは、スタンドアロン Web モジュールとして default アプリケーションにデプロイできます。

スタンドアロン OC4J のインストール

この章では、スタンドアロン OC4J ディストリビューション (oc4j_extended.zip アーカイブとして配布) のインストールに関する前提条件およびプロセスについて説明します。

OC4J を Oracle Application Server のコンポーネントとしてインストールする手順は、環境固有の Oracle Application Server のインストール・ガイドを参照してください。

この章の内容は次のとおりです。

- [スタンドアロン OC4J サーバーのインストール前提条件への準拠](#)
- [スタンドアロン OC4J ディストリビューションのインストール](#)

スタンドアロン OC4J サーバーのインストール前提条件への準拠

スタンドアロン OC4J サーバーをインストールする前に、次の前提条件が満たされていることを確認してください。

JDK 6、5.0 または 1.4.2 のインストール

スタンドアロン OC4J をインストールする前に、OC4J ホスト・マシンに、次のいずれかの JDK リリースをインストールする必要があります。

- Java Platform Standard Edition (Java SE) Development Kit (JDK) 6
- Java Platform 2 Standard Edition (J2SE) Development Kit (JDK) 5.0 (JDK1.5 と呼ばれる)
- Java Platform 2 Standard Edition (J2SE) Development Kit (JDK) 1.4

JDK のリリースは、<http://java.sun.com/j2se/> からダウンロードできます。

注意：スタンドアロン OC4J の場合は、JDK を指定する必要があります。
OPMN 管理 OC4J の場合は、JDK 5.0 がサーバー・バイナリに付属しています。

環境変数の設定

J2SE のインストール後に、`JAVA_HOME` および `ORACLE_HOME` 環境変数が設定されていることを確認してください。`J2EE_HOME` 環境変数も設定できます。

表 2-1 環境変数の設定

環境変数	値
<code>JAVA_HOME</code>	JDK の場所に設定します。この変数は、OC4J サーバーの起動に必要です。次に例を示します。 <code>JAVA_HOME=/java/j2se15</code> 使用される JDK は、ホスト・マシンの <code>PATH</code> 環境変数に追加する必要があります。
<code>ORACLE_HOME</code>	OC4J ディストリビューションをインストールするルート・ディレクトリに設定します。この変数は、 <code>oc4j</code> または <code>oc4j.cmd</code> 実行可能スクリプトを実行しようとする場合に定義する必要があります。 たとえば、OC4J を <code>C:\oracle</code> にインストールする場合、 <code>ORACLE_HOME</code> 変数の値をこのディレクトリに設定します。
<code>J2EE_HOME</code>	必要に応じてこの環境変数を作成し、 <code>ORACLE_HOME/j2ee/home</code> (<code>admin_client.jar</code> 、 <code>oc4j.jar</code> および <code>admin.jar</code> のインストール場所) に設定します。 <code>ORACLE_HOME</code> の値は、OC4J ディストリビューションをインストールするルート・ディレクトリです。 <code>J2EE_HOME</code> 環境変数または <code>oracle.j2ee.home</code> システム・プロパティを J2EE ホーム・ディレクトリに設定すると、 <code>admin_client.jar</code> 、 <code>oc4j.jar</code> または <code>admin.jar</code> を任意のディレクトリから起動できるようになります。

環境変数 `ORACLE_HOME` および `J2EE_HOME` のかわりに、システム・プロパティ `oracle.home` および `oracle.j2ee.home` を使用して、Oracle および J2EE ホーム・ディレクトリに設定できます。

オペレーティング・システムのデフォルト・ロケール以外のロケールを使用する場合も、`LC_ALL` および `LANG` 環境変数を同じ値に設定します。

スタンドアロン OC4J ディストリビューションのインストール

スタンドアロン OC4J ディストリビューションは、ユーザー自身が選択したアーカイブ・ユーティリティを使用して、OC4J のインストール・ディレクトリとなるディレクトリ（このマニュアルでは `ORACLE_HOME` と呼ぶ）に `oc4j_extended.zip` ファイルを解凍してインストールします。インストーラにより、次に示す必須ディレクトリ構造が自動的に作成されます。

```
ORACLE_HOME
  /ant
  /bin
  /diagnostics
  /j2ee
  /javacache
  /javavm
  /jdbc
  /jlib
  /lib
  /opmn
  /rdbms
  /sqlj
  /toplink
  /webservices
  /xqs
```

注意： jar ユーティリティをスタンドアロン OC4J の解凍に使用すると、ファイル権限は適切に設定されません。ファイル権限を修正するには、unzip ユーティリティを再度使用してディストリビューションを抽出するか、実行可能ファイルを実行できるよう各ファイルに権限を設定します。

OC4J の初回起動時に、OC4J 管理者アカウントのパスワードを設定するように要求されます。このアカウントのユーザー名は、デフォルトで `oc4jadmin` に設定されます。

`jazn.jar` ツールを使用して、OC4J を起動する前に `oc4jadmin` アカウントをアクティブにすることもできます。このツールは `ORACLE_HOME /j2ee/home` ディレクトリにあります。構文は次のとおりです。

```
jazn.jar -activateadmin password
```

注意： 以前は `oc4j.jar -install` コマンドが `oc4jadmin` アカウントをアクティブにする他、このアカウントにパスワードを設定するために使用されましたが、OC4J 10g (10.1.3.4.0) では推奨されていません。

スタンドアロン OC4J ディストリビューションは、デフォルト構成でインストールされます。この構成には、アプリケーションにアクセスできるデフォルトの Web サイトと、Application Server Control を使用できる Web サイトが含まれます。これらは、OC4J をすぐに使用し始めることができるように用意されています。詳細は、[第 13 章「OC4J での Web サイトの管理」](#)を参照してください。

OC4J の管理用ツール

この章では、OC4J に装備された管理機能の概要を説明します。この章の内容は次のとおりです。

- Oracle Enterprise Manager 10g Application Server Control
- `admin_client.jar` コマンドライン・ユーティリティ
- `admin.jar` コマンドライン・ユーティリティ
- `oc4j` 実行可能スクリプト
- Oracle Process Manager and Notification Server (OPMN)
- `oc4jadmin` アカウントのパスワードの変更

Oracle Enterprise Manager 10g Application Server Control

Oracle Enterprise Manager 10g Application Server Control は、JMX に準拠する Web ベースのユーザー・インタフェースで、OC4J 内のアプリケーションのデプロイ、構成および監視の他、スタンドアロン OC4J サーバー、Oracle Application Server クラスタ内の OC4J インスタンスのグループ、およびアプリケーションで使用される Web サービスの管理を目的とします。この項の内容は次のとおりです。

- [スタンドアロン OC4J での Application Server Control へのアクセス](#)
- [Oracle Application Server での Application Server Control へのアクセス](#)
- [Application Server Control インタフェースの機能の概要](#)

このインタフェースの使用方法は、Application Server Control に付属するオンライン・ヘルプを参照してください。

注意： Application Server Control の現行リリースでは、OPMN の構成や Oracle HTTP Server の起動と停止はサポートされていますが、Oracle HTTP Server の構成はサポートされていません。OPMN および Oracle HTTP Server の構成の詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

スタンドアロン OC4J での Application Server Control へのアクセス

Application Server Control は、OC4J ソフトウェアのインストール時に自動的にインストールおよび構成されます。OC4J を起動するとデフォルトで起動されます。

コンソールには、HTTP リクエストをポート 8888 でリスニングするように構成されたデフォルト Web サイトを介してアクセスします。コンソールにアクセスするには、次の URL を Web ブラウザに入力するだけです。

```
http://hostname:8888/em
```

Oracle Application Server での Application Server Control へのアクセス

Application Server Control は、OC4J の埋込みコンポーネントとしてインストールおよび構成されます。コンソールは、OPMN コマンドライン・ツール `opmnctl` を使用して、その他のすべてのインストール済 Oracle Application Server コンポーネントとともに起動されます。このツールは、各サーバー・ノードの `ORACLE_HOME/opmn/bin` ディレクトリにインストールされています。

次のコマンドを発行して、すべてのインストール済コンポーネントを起動します。

```
cd ORACLE_HOME/opmn/bin
opmnctl startall
```

複数の OC4J インスタンスを含むクラスタ・トポロジでは、クラスタの OPMN 構成ファイル `opmn.xml` に `sequential` オプションが含まれない場合には、コマンドで `-sequential` フラグを使用する必要があります。

```
cd ORACLE_HOME/opmn/bin
opmnctl startall -sequential
```

`sequential` オプションを使用すると、OC4J インスタンスが順番に起動されます。コンポーネントを同時に起動すると、リソース競合の問題が発生する可能性があります。`opmn.xml` ファイルに `sequential` オプションを指定する方法の詳細は、4-3 ページの「[管理 OC4J 構成でのランタイム・オプションの設定](#)」を参照してください。

通常の Oracle Application Server インストールでは、Application Server Control を含むすべての Web アプリケーションには、Oracle HTTP Server を介してアクセスします。次の URL を使用してコンソールにアクセスします。

```
http://ohs_host_address:port/em
```

- `ohs_host_address`: Oracle HTTP Server ホスト・マシンのアドレス。たとえば `server07.company.com` です。
- `port`: OPMN によって Oracle HTTP Server に割り当てられた HTTP リスナー・ポート。割り当てられたリスナー・ポートのリストを OPMN から取得するには、次の `opmnctl` コマンドを Oracle HTTP Server ホスト・マシンで実行します。

```
opmnctl status -l
```

OPMN ステータス出力で `http1` に指定されたポートを、`port` の値として指定します。

```
HTTP_Server | HTTP_Server | 6412 | Alive | 1970872013 | 1
6396 | 0:48:01 | https1:4443,http2:722,http1:7779
```

Application Server Control インタフェースの機能の概要

Application Server Control は、次に説明するいくつかの機能領域で構成されています。

アプリケーション

- OC4J インスタンスまたは Oracle Application Server クラスタ内のインスタンスのグループにデプロイされたアプリケーション、モジュールまたはスタンドアロン・リソース・アダプタの起動と停止
- アプリケーションまたはモジュールのデプロイ、アンデプロイまたは再デプロイ
- アプリケーションのデプロイの一環としてのデプロイ・プランの作成または編集
- HTTP リクエストおよびアクティブな EJB メソッド・コールに関する統計の表示

管理

- J2EE サービス (JMS や JTA など) の管理
- JNDI 名の表示および検索
- JDBC データソースおよびデータベース・アクセスを提供する接続プールの作成
- JSP コンテナのプロパティの設定
- セキュリティ・プロバイダの構成およびユーザーとロールの管理
- JMX MBean ブラウザを介した MBean へのアクセス
- イベント・ドリブンの JMX 通知のサブスクライブ

パフォーマンス

- OC4J と他のアクティブなアプリケーションによる CPU およびメモリー・リソースの使用状況と、OC4J ヒープ使用状況を示すグラフの表示
- データベース接続とトランザクション・アクティビティ、JVM 使用状況、JSP とサーブレットのリクエストおよび EJB メソッドに関する統計の表示
- 最もリクエストが多かった JSP、サーブレットおよび EJB モジュールに関するシステムへの問合せ

Web サービス

- Web サービスの有効化または無効化
- インスタンス内で稼働している Web サービスのメトリックおよび統計の表示
- Web サービスの WSDL の表示
- Web サービスのテスト
- Web サービスの監査、ロギング、信頼性およびセキュリティの構成

ログ

- OC4J インスタンスにデプロイされた特定のアプリケーションのログ・ファイルの表示
- デフォルト・アプリケーション（グローバル Web アプリケーションを含む）および Application Server Control のログの表示
- 特定のメッセージ・タイプおよび文字列についてのログの検索
- Oracle Diagnostic Logging (ODL) フレームワークを使用した、コンポーネントの XML 形式のログ・ファイルの表示
- Web サービスのログの取得

OC4J によって提供されるロギング機能の詳細は、[第 11 章「OC4J でのロギング」](#)を参照してください。

admin_client.jar コマンドライン・ユーティリティ

OC4J には `admin_client.jar` というコマンドライン・ユーティリティが用意されています。このユーティリティを使用すると、Oracle Application Server のクラスタ化された環境およびスタンドアロン OC4J サーバーのアクティブな OC4J インスタンスで操作を実行できます。

このユーティリティを使用して実行できるタスクは次のとおりです。

- 特定の OC4J インスタンスまたは Oracle Application Server クラスタ内のインスタンス・グループに対するアプリケーション (EAR)、スタンドアロン Web モジュール (WAR)、スタンドアロン EJB モジュール (EJB JAR)、スタンドアロン・リソース・アダプタ (RAR) のデプロイ
- アプリケーション、Web モジュール、EJB モジュールまたはリソース・アダプタのアンデプロイ
- 変更されたクラスを保持するデプロイ済 EJB モジュールの増分的な更新
- 共有ライブラリの新規作成
- JDBC および JMS リソースの作成
- OC4J インスタンスの停止、起動または再起動
- 特定の OC4J インスタンスまたはクラスタ全体のインスタンス・グループでの特定のアプリケーションの停止、起動または再起動

このツールの使用方法は、[第 6 章「admin_client.jar ユーティリティの使用法」](#)を参照してください。

admin.jar コマンドライン・ユーティリティ

OC4J には `admin.jar` というコマンドライン・ユーティリティが用意されています。このユーティリティを使用すると、アクティブなスタンドアロン OC4J インスタンスで操作を実行できます。

注意： `admin.jar` ユーティリティは、スタンドアロン OC4J インストールの単一の OC4J インスタンスを管理するためだけに使用できます。

より高度な機能を装備しているため、`admin.jar` のかわりに `admin_client.jar` ユーティリティを使用することをお勧めします。このユーティリティの使用法は、[第 6 章「admin_client.jar ユーティリティの使用法」](#)を参照してください。

特に、このユーティリティを使用してできるタスクは次のとおりです。

- スタンドアロン OC4J インスタンスの停止および再起動
- 特定のアプリケーションの再起動
- スタンドアロン OC4J インスタンスに対するアプリケーションのデプロイまたはアンデプロイ
- グローバル・データソースまたはアプリケーション固有のデータソースの追加、削除またはテスト

このユーティリティは、`ORACLE_HOME/j2ee/instance/` にデフォルトでインストールされます。データソースをアップグレードする場合を除き、このユーティリティを使用するには、OC4J を起動する必要があります。また、このユーティリティを使用して OC4J を起動することはできません。このツールの使用方法は、第 7 章「[admin.jar ユーティリティの使用方法](#)」を参照してください。

oc4j 実行可能スクリプト

OC4J ディストリビューションには、実行可能スクリプトが含まれています。これらのスクリプトをスタンドアロン OC4J 構成で使用すると、ローカルの OC4J インスタンスの起動および停止、OC4J バージョンの取得、OC4J インストール・プロセスの完了が可能です。これらのスクリプトには、Linux と UNIX 環境用のシェル・スクリプト、および Windows 環境用のバッチ・ファイルが含まれています。

oc4j 実行可能スクリプトは、`ORACLE_HOME/bin` ディレクトリにあります。スクリプトは、次のように環境固有です。

- oc4j シェル・スクリプトは、Linux または UNIX 環境で使用します。
- oc4j.cmd バッチ・ファイルは、Windows 環境で使用します。

これらのスクリプトを使用するには、2-2 ページの「[環境変数の設定](#)」で説明されているように、`ORACLE_HOME` および `JAVA_HOME` 環境変数が設定されている必要があります。

どちらの実行可能ファイルも、次のような同じ構文を使用します。

```
oc4j [options]
```

実行可能ファイルに渡すことができる一連のオプションは、[表 3-1](#) にまとめられているようにどちらについても同じです。

表 3-1 oc4j 実行可能ファイルのオプション

オプション	説明
-start	OC4J インスタンスを起動します。
-shutdown	OC4J インスタンスを停止します。
-port <i>ormiport</i>	-port <i>ormiport</i> : OC4J がデフォルトの ORMI ポート (23791) で稼働している場合は、ポートを指定する必要はありません。
-password <i>password</i>	-password <i>password</i> : oc4jadmin アカウントのパスワードを指定します。
-version	OC4J のバージョン番号を返します。
-help	構文および一連のオプションを表示します。

Oracle Process Manager and Notification Server (OPMN)

管理 OC4J 環境では、すべてのインストール済 Oracle Application Server コンポーネント (すべての OC4J インスタンスを含む) の管理と起動および停止に、OPMN が使用されます。また、OPMN は OC4J および関連するコンポーネント (Oracle HTTP Server など) も監視します。そのため、インストール済 Oracle Application Server コンポーネントを監視するには、OPMN を各 `ORACLE_HOME` ディレクトリにインストールする必要があります。

OPMN を構成および使用する方法は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

OPMN デーモンの制御には、コマンドライン・ユーティリティ `opmnctl` が使用されます。このユーティリティは、Oracle Application Server のホスト・コンポーネントのホストとなるマシンの `ORACLE_HOME/opmn/bin` ディレクトリにデフォルトでインストールされます。

注意： Application Server Control の現行リリースでは、OPMN の構成や Oracle HTTP Server の起動と停止はサポートされていますが、Oracle HTTP Server の構成はサポートされていません。OPMN および Oracle HTTP Server の構成の詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

OPMN は、`opmn.xml` 構成ファイルを使用して構成されます。このファイルは、`ORACLE_HOME/opmn/conf` ディレクトリにあります。現行リリースの Application Server Control にはファイル編集機能が装備されていないため、ほとんどの場合、このファイルの編集は手動で行う必要があります。

次の例に、構成ファイル `opmn.xml` での OC4J 構成データの構造を示します。

- 各コンポーネントの構成データは、`<ias-component>` 要素内に設定されます。この要素の `id` 属性はコンポーネント名 (この場合は `default_group`) です。
- ホスト・マシン上に作成された各 OC4J インスタンスは、`<process-type>` 要素内で個別に構成されます。`id` 属性は、インスタンスを一意に識別します。
- `<process-set>` 要素は、起動時に作成される OC4J プロセスのグループを定義します。
`id` 属性の値はグループを識別し、管理しやすくするためにグループ内のプロセスに対して生成されたログ・ファイルの最後に追加されます。

次の要素は、`opmn.xml` における OC4J 構成データの構造の簡略化された例です。

```
<opmn>
. . .
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-Djava.awt.headless=true"/>
        <data id="java-bin" value="/jdk/bin"/>
        <data id="oc4j-options" value="-validateXML -verbosity 10"/>
      </category>
      <category id="stop-parameters">
        <data id="java-options" value="-Djava.awt.headless=true"/>
      </category>
    </module-data>
    <start timeout="600" retry="2"/>
    <stop timeout="120"/>
    <restart timeout="720" retry="2"/>
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="rmi" range="12401-12500"/>
    <port id="jms" range="12601-12700"/>
    <port id="rmis" range="12701-12800"/>
    <process-set id="default_group" numprocs="1"/>
  </process-type>
</ias-component>
</opmn>
```

oc4jadmin アカウントのパスワードの変更

OC4J 管理者アカウントは、デフォルトでユーザー名 `oc4jadmin` を使用して作成されます。このアカウントは、OC4J に付属する各種ツール (`admin_client.jar` コマンドライン・ユーティリティなど) を使用してコマンドを起動する際に必要です。また、Application Server Control にログインする際にも使用できます。

oc4jadmin アカウントには `oc4j-administrators` ロールが割り当てられます。このロールは、アカウントでユーザーおよびロールを管理するために必要です。また、MBeanServer サーバーに接続するためにもこのロールが必要です。

このアカウントの初期パスワードは、OC4J のインストール時に設定できます。設定しない場合は、OC4J の初回起動時に設定するように要求されます。Oracle Application Server クラスタ内のグループの全 OC4J インスタンスでは、Application Server Control を使用してすべてのインスタンスにアクセスしグループ操作を実行できるよう、oc4jadmin アカウントのパスワードが同一である必要があります。また、Oracle Application Server クラスタのすべての OC4J インスタンスでも、OPMN 関連の問題を防ぐために、oc4jadmin アカウントのパスワードが同一である必要があります。

『Oracle Application Server 管理者ガイド』の付録 A で説明されているように、パスワードは後から変更できます。oc4jadmin パスワードの変更には次のガイドラインが適用されます。

- ベスト・プラクティスとして、oc4jadmin アカウントは、Application Server Control への最初のログインにのみ使用することをお勧めします。その後、日常業務用で使用するための新規アカウント（およびその他の管理者のアカウント）を作成する必要があります。oc4jadmin アカウントおよびそのパスワードは、ascontrol アプリケーションでの内部的な使用に制限してください。このアプリケーションでは、そのアカウントを使用して、Oracle Application Server クラスタ内のその他の OC4J インスタンスを管理します。
- oc4jadmin パスワードを変更する場合には、クラスタ内のすべての OC4J インスタンスでパスワードを変更する必要があります。このパスワードの変更では、『Oracle Application Server 管理者ガイド』の付録 A で説明されている様々な手順を実行します。特に、アクティブな ascontrol アプリケーションを実行する管理 OC4J インスタンスの oc4jadmin パスワードを変更する手順は、リモートで管理されている OC4J インスタンスの oc4jadmin パスワードの変更手順とは異なります。

OC4J ランタイムの構成

この章では、OC4J の起動時に設定できるランタイム・オプションおよびシステム・プロパティについて詳しく説明します。次の項目が含まれます。

- [JDK のバージョンの指定](#)
- [起動時の OC4J ランタイム・オプションの設定](#)
- [起動時のシステム・プロパティの設定](#)
- [統合開発環境からのリモート・デバッグの有効化](#)

JDK のバージョンの指定

OC4J には、次のいずれかの JDK リリースが必要です。

- Java Platform Standard Edition (Java SE) Development Kit (JDK) 6
- Java Platform 2 Standard Edition (J2SE) Development Kit (JDK) 5.0 (JDK1.5 と呼ばれる)
- Java Platform 2 Standard Edition (J2SE) Development Kit (JDK) 1.4

スタンドアロン OC4J 構成に使用する JDK バージョンを指定する必要があります。各 OC4J インスタンスに JDK バージョンを指定できます。Oracle Application Server インストールでは、OC4J インスタンスにデフォルトで JDK 5.0 が使用されます。

スタンドアロン構成での JDK の指定

スタンドアロン OC4J 構成では、`JAVA_HOME` 環境変数に OC4J で使用される JDK の場所を設定します。使用される JDK は、ホスト・マシンの `PATH` 環境変数に追加する必要があります。

あるいは、OC4J の起動時に使用する JDK を指定できます。次に例を示します。

```
C:¥ORACLE_HOME¥j2ee¥home¥C:¥jdk¥bin¥java -jar oc4j.jar
```

管理構成での JDK の指定

Oracle Application Server のコンポーネントとしてインストールされた OPMN 管理の OC4J インスタンスは、デフォルトでは JDK リリース 5.0 を使用します。EJB 3.0 を使用するには、JDK 5.0 または JDK 6 が必要です。OC4J にデプロイされるアプリケーションに JDK 6 または JDK 1.4.2 のリリースが必要な場合は、別のバージョンに切り替える必要があります。

JDK 5.0 から JDK 6 または JDK 1.4.2 に切り替えるには、すべてのコンパイル済アプリケーション・ファイルを OC4J インスタンスから削除する必要があります。

1. OC4J インスタンスを停止します。
2. `ORACLE_HOME/j2ee/instance/application-deployments` ディレクトリを削除します。

このディレクトリを削除すると、OC4J が JDK 6 または JDK 1.4.2 で再起動される際に、アプリケーション・ファイルは再コンパイルされます。

`opmn.xml` 構成ファイルを手動で編集して、各 OC4J インスタンスに使用する JDK を指定できます。JDK とともにインストールされた、`JAVA_HOME` 環境変数に定義されている `javac` コンパイラを使用する場合は、`server.xml` ファイルから `<java-compiler>` 要素も削除し、OC4J がデフォルト設定を再検出できるようにします。

`id` 属性が `java-bin` である `<data>` 要素に Java システム・プロパティを設定します。この `<data>` 要素は、XML 構造の `<ias-component id="default_group">` 要素の `<category id="start-parameters">` サブ要素に囲まれています。次に例を示します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-bin" value="c:/myhost/jdk/bin/java"/>
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

起動時の OC4J ランタイム・オプションの設定

OC4J の起動時に、多くの OC4J ランタイム・オプションを OC4J インスタンスに設定できますが、ほとんどは標準出力メッセージを管理するためのオプションです。これらのオプションの設定方法は、スタンドアロン OC4J 構成と管理 Oracle Application Server 構成では異なります。

- [スタンドアロン OC4J 構成でのランタイム・オプションの設定](#)
- [管理 OC4J 構成でのランタイム・オプションの設定](#)
- [OC4J ランタイム・オプションの概要](#)

スタンドアロン OC4J 構成でのランタイム・オプションの設定

OC4J ランタイム・オプションは、OC4J の起動時に `oc4j.jar` コマンドラインで引数を渡して設定できます。`oc4j.jar` の構文は次のとおりです。

```
java [props] -jar oc4j.jar [args]
```

構文の `oc4j.jar` の後に、ランタイム・オプション（`[args]`）を指定します。次に例を示します。

```
java -jar oc4j.jar -userthreads
```

管理 OC4J 構成でのランタイム・オプションの設定

OC4J が Oracle Application Server のコンポーネントとしてインストールされる場合、OC4J ランタイム・オプションは `opmn.xml` 構成ファイルに手動で追加する必要があります。オプションは、起動時に管理 OC4J インスタンスに渡されます。

`id` 属性が `oc4j-options` である `<data>` 要素に OC4J ランタイム・オプションを設定します。この `<data>` 要素は、XML 構造の `<ias-component id="default_group">` 要素の `<category id="start-parameters">` サブ要素に囲まれています。次に例を示します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="oc4j-options" value="-userthreads"/>
        ...
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

OC4J ランタイム・オプションの概要

表 4-1 に、OC4J ランタイム・オプションを示します。

表 4-1 OC4J 起動オプション

コマンドライン引数	説明
-quiet	コンソールへの標準出力を抑止します。
-config path	server.xml ディスクリプタ・ファイルへのパスを指定します。デフォルトの場所は、ORACLE_HOME/j2ee/instance/config ディレクトリです。
-validateXML	読取り時に XSD によって定義された XML 構成ファイルを検証します。 orion-ejb-jar.xml ファイルの <entity-deployment> または <session-deployment> 要素に tx-retry-wait 属性を追加した場合は、OC4J 起動コマンドラインで -validateXML オプションを使用しないでください。
-out [file]	標準出力のルーティング先となるファイルを指定します。このファイルには、System.out に出力されるメッセージの他、サーブレット・ロギング・インタフェースを介して出力に送信されるメッセージが格納されます。指定しない場合は、すべての出力が標準出力に書き込まれます。 stdout ファイルを管理するために設定できるその他のシステム・プロパティは、4-8 ページの「 stdout および stderr ログ・ファイルの管理 」を参照してください。 OPMN 管理の構成では、このファイルは、指定されたパスの最後に追加される instance_default_group_1 ディレクトリ内に生成されます。 たとえば、opmn.xml に次の要素を指定したとします。 <data id="oc4j-options" value="...-out /mypath/mylog.log"/> mylog.log ファイルは、実際には次のファイルに生成されます。 /mypath/instance_default_group_1/mylog.log
-err [file]	標準エラー出力のルーティング先となるファイルを指定します。このファイルには、System.err に出力されるメッセージが格納されます。指定しない場合は、すべてのエラーが標準エラーに書き込まれます。 stderr ファイルを管理するために設定できるその他のシステム・プロパティは、4-8 ページの「 stdout および stderr ログ・ファイルの管理 」を参照してください。 OPMN 管理の構成では、このファイルは、指定されたパスの最後に追加される instance_default_group_1 ディレクトリ内に生成されます。詳細は、-out の説明を参照してください。
-verbosity int	メッセージ出力の詳細レベルを設定する 1 ~ 10 の整数を定義します。値 10 は、最も詳細な出力を生成します。次に例を示します。 java -jar oc4j.jar -verbosity 10
-monitorResourceThreads	スレッド・リソースのバックアップ・デバッグを有効にします。コードの重要な部分でスレッドがスタックする問題が発生した場合にのみ、有効にします。
-userThreads	ユーザー作成のスレッドからのコンテキスト参照のサポートを有効にします。
-http.sessionInvalidatingThreads	HTTP セッションを無効化する最大スレッド数を指定します。デフォルト値は、3 です。
-listProperties	OC4J の起動時に JVM に設定できるすべての OC4J 固有のシステム・プロパティのリストを出力した後、終了します。次の例では、作業ディレクトリ内のテキスト・ファイルに出力をリダイレクトしています。 java -jar oc4j.jar -listProperties > props.txt

表 4-1 OC4J 起動オプション (続き)

コマンドライン引数	説明
-sequential	Oracle Application Server クラスタ内の各 OC4J インスタンスを順番に起動します。
-version	インストールされた OC4J のバージョンを返した後、終了します。
-? -help	これらのオプションのヘルプ・テキストをコンソールに出力します。

起動時のシステム・プロパティの設定

次の各項で説明するように、OC4J の起動時に、多くの OC4J 固有のシステム・プロパティを JVM に設定できます。

- スタンドアロン OC4J 構成でのシステム・プロパティの設定
- OPMN 管理 OC4J 構成でのシステム・プロパティの設定
- manifest.dependencies.warn.only を使用した共有ライブラリでの異なる依存性の許容
- シンボリック・リンク使用の防止
- stdout および stderr ログ・ファイルの管理
- 一般的なシステム・プロパティの概要
- デバッグ・プロパティの概要

注意: `oc4j.jar -listProperties` オプションを使用すると、OC4J の起動時に JVM に設定できるすべての OC4J 固有のシステム・プロパティのリストを出力できます。次の例では、作業ディレクトリ内のテキスト・ファイルに出力をリダイレクトしています。

```
java -jar oc4j.jar -listProperties > props.txt
```

スタンドアロン OC4J 構成でのシステム・プロパティの設定

システム・プロパティは、起動時に OC4J コマンドラインを使用して JVM に設定できます。OC4J が稼働中の場合は、新しいプロパティ設定を有効にするために、インスタンスを再起動する必要があります。

構文は次のとおりです。

```
java [props] -jar oc4j.jar [args]
```

構文の `oc4j.jar` の前に、すべてのシステム・プロパティ (`props`) を指定します。コマンドラインでは、各システム・プロパティを `-D` で始める必要があります。次に例を示します。

```
java -Doc4j.formauth.redirect=true -jar oc4j.jar
```

OPMN 管理 OC4J 構成でのシステム・プロパティの設定

OC4J が Oracle Application Server のコンポーネントとしてインストールされる場合、`opmn.xml` 構成ファイルに OC4J システム・プロパティを手動で追加できます。オプションは、起動時に管理 OC4J インスタンスに渡されます。

`id` 属性が `java-options` である `<data>` 要素に Java システム・プロパティを設定します。この `<data>` 要素は、XML 構造の `<ias-component id="default_group">` 要素の `<category id="start-parameters">` サブ要素に囲まれています。すべてのシステム・プロパティは、`-D` で始めます。次に例を示します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-Doc4j.formauth.redirect=true
          -Dhttp.session.debug=true"/>
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

manifest.dependencies.warn.only を使用した共有ライブラリでの異なる依存性の許容

OC4J が初期化される際には、MANIFEST.MF ファイルに指定されているバージョンを検証することで、コンテナ・レベル構成に定義されている共有ライブラリの依存性が検証されます。依存性チェック失敗のデフォルトの動作では、エラーが発行されて初期化プロセスが停止されます。同じ共有ライブラリの異なるバージョンを使用する場合は、起動時に `manifest.dependencies.warn.only` プロパティを `true` に設定し、依存性チェックが失敗した際に、OC4J が警告を発行し、初期化が続行されるようにできます。

このプロパティは、同じアプリケーション・ライブラリに対する依存性が異なるプロジェクト・リリースおよびビルドを同時に実行する場合に便利です。

`manifest.dependencies.warn.only` が `true` に設定されていると、OC4J により、デフォルトで警告メッセージが発行されます。記録されたメッセージは、`oracle` ログ出力の `ORACLE_HOME/j2ee/instance/config/j2ee-logging.xml` 構成ファイルに指定された場所に保存されます。このログ出力には、コンソールと、`j2ee/instance/log/oc4j/log.xml` の OLD 形式の XML ファイルという 2 つの出力場所があります。スタンドアロン OC4J では、`instance` 名は `home` です。OPMN 管理の Oracle Application Server 環境では、すべてのコンソール出力が `ORACLE_HOME/opmn/logs/instance_default_group_1` ファイル（またはこの名前の変換形）にリダイレクトされます。

シンボリック・リンク使用の防止

OC4J 10g (10.1.3.4.0) は、デフォルトで、`http.file.allowAlias` プロパティが `false` に設定されて出荷されます。この設定により、シンボリック・リンクの使用が防止されています。場合によっては、JSP のソース・コードがエンドユーザーに表示される可能性があるため、この設定を `true` に変更しないことをお勧めします。

プロパティ設定を変更するかわりに、次のいずれかの回避策を使用できます。

- OC4J 軽量 HTTP リスナーの使用を、Oracle HTTP Server を介して OC4J アプリケーションがフロントエンドとなる構成に一時的に切り替え、ブラウザが HTTP を介して直接ページにアクセスするのではなく、`mod_oc4j` および Apache JServ Protocol (AJP) を介して間接的にアクセスできるようにします。
- アプリケーション内のすべてのシンボリック・リンクを、それらが提供する実際のファイル名で置き換えます。

シェル・スクリプトを使用すると、シンボリック・リンクの置換えを自動化できます。次に例を示します。

```
#!/bin/ksh

PROGNAME="${0##*/}"
LN_EXTN="ln"

function displaySyntax
{
echo "${PROGNAME}! SYNTAX: ${PROGNAME} some_dir_path"
exit 1
}

if [[ $# < 0 ]]
then
displaySyntax
fi

DIR="$1"

if [[ ! -d ${DIR}
then
displaySyntax
fi

find ${DIR} -type l|while read filepath
do
echo "FIXING: ${filepath} (=> ${filepath}.${LN_EXTN})"
mv ${filepath} ${filepath}.${LN_EXTN}
cp -L ${filepath}.${LN_EXTN} ${filepath}
done
```

この例の KSH スクリプトは、UNIX 環境では次のようにして起動します。

```
$ fixLinks web_module_root
```

このスクリプトにより、すべてのディレクトリが再帰的に調査され、追加の `.ln` 拡張子が付くシンボリック・リンクである、検出されたすべてのファイル名が変更されます。その後、スクリプトにより、リンクが検出された元の場所にリンク・ターゲットのコピーが配置されます。

stdout および stderr ログ・ファイルの管理

次のプロパティは、stderr および stdout 標準ログ・ファイルの管理に使用されます。oc4j.jar コマンドの -out および -err オプションを使用して、プロパティが属するログ・ファイルのタイプと場所を指定できます。

たとえば、次のコマンドは、ファイル・サイズが 2.5MB に達したときに stdout および stderr ファイルを循環させることを指定しています。ログ・ファイルは、d:¥logs ディレクトリに出力されます。

```
java -Dstdout.filesize=2.5 -jar oc4j.jar -out d:¥logs¥oc4j.out -err d:¥logs¥oc4j.err
```

次の例では、stdout ログ・ファイルを毎日午後 2:30 に循環させ、ログ・アーカイブを最大 10 ファイルに制限することを指定しています。

```
java -Dstdout.rotatetime=14:30 -Dstdout.filenumber=10 -jar oc4j.jar -out d:¥logs¥oc4j.out
```

表 4-2 stdout および stderr ファイルのアーカイブ管理プロパティ

プロパティ	説明
stdout.filesize=max_file_size	アーカイブ内のファイルの最大サイズ (MB 単位)。この最大サイズに達するとファイルが循環します。
stdout.filenumber=max_files	アーカイブとして保持される最大ファイル数。この制限を超えると、最も古いファイルが自動的に削除されます。
stdout.rotatetime=HH:mm	ログ・ファイルが毎日循環する時刻。

一般的なシステム・プロパティの概要

表 4-3 に、OC4J に対して設定できる一般的なシステム・プロパティを示します。

表 4-3 OC4J の一般的な -D システム・プロパティ

プロパティ	説明
java.ext.dirs	コンパイル時にクラスが検索される外部ディレクトリを設定します。
java.io.tmpdir=new_tmpdir	デプロイ・ウィザード用の一時ディレクトリを設定します。デフォルトは /tmp/var です。 デプロイ・ウィザードは、デプロイ・プロセス時に情報を格納するために、一時ディレクトリのスワップ領域を 20MB 使用します。完了すると、デプロイ・ウィザードは一時ディレクトリをクリーンアップします。 ただし、ウィザードが中断されると、一時ディレクトリをクリーンアップする時間またはタイミングが失われることがあります。その場合、ユーザー自身がこのディレクトリから追加デプロイメント・ファイルをクリーンアップする必要があります。クリーンアップしないと、このディレクトリが満杯になり、それ以降のデプロイが実行できなくなります。 Out of Memory エラーが出力された場合は、一時ディレクトリの使用可能領域を確認してください。
java.awt.headless=true false	true の場合は、環境でディスプレイ、キーボードおよびマウスがサポートされているかどうかを確認されます。false の場合、確認は行われません。デフォルトは true です。
oracle.home	OC4J ディストリビューションをインストールするルート・ディレクトリを設定します。 oracle.home システム・プロパティを使用するかわりに、ORACLE_HOME 環境変数の値をルート・ディレクトリに設定できます。

表 4-3 OC4J の一般的な -D システム・プロパティ (続き)

プロパティ	説明
oracle.j2ee.home	J2EE ホーム・ディレクトリを、oc4j.jar および admin.jar ファイルのインストール・ディレクトリである ORACLE_HOME/j2ee/instance に設定します。ORACLE_HOME の値は、OC4J ディストリビューションをインストールするルート・ディレクトリです。 このシステム・プロパティまたは J2EE_HOME 環境変数を J2EE ホーム・ディレクトリに設定すると、oc4j.jar および admin.jar を任意のディレクトリから起動できるようになります。
GenerateIIOP=true false	IIOP スタブの生成を有効にします。デフォルトは false です。
KeepIIOPCode=true false	生成された IIOP スタブ / タイ・コードを保持するかどうかを設定します。デフォルトは false です。
oracle.arraylist.deepCopy=true false	true の場合は、配列リストのクローニング中に、ディープ・コピーが実行されます。false の場合は、配列リストに対してシャロー・コピーが実行されます。デフォルトは true です。
dedicated.rmicontext=true false	非推奨の dedicated.connection プロパティはこのプロパティに置き換えられました。デフォルトは false です。 同じプロセス内の 2 つ以上のクライアントが InitialContext を取得すると、OC4J はキャッシュされているコンテキストを返します。したがって、各クライアントは、そのプロセスに割り当てられている同じ InitialContext を受け取ります。サーバー参照は、サーバーのロード・バランシングを招きますが、これはクライアントが独自の InitialContext を取得する場合にのみ発生します。 dedicated.rmicontext=true と設定すると、各クライアントは共有コンテキストではなく独自の InitialContext を受け取ります。各クライアントが独自の InitialContext を保持する場合、クライアントの負荷分散が可能で このプロパティは、JNDI プロパティに設定することもできます。 OC4J 10g (10.1.3.4.0) で非推奨になった dedicated.rmicontext、dedicated.connection および LoadBalanceOnLookup プロパティは、oracle.j2ee.rmi.loadBalance プロパティに置き換えられました。
manifest.dependencies.warn.only	このプロパティを使用すると、システム・ライブラリの JAR 依存性が一致していない場合に、OC4J の初期化が停止されるデフォルトの動作を上書きできます。デフォルトのプロパティ値は false です。 OC4J が初期化される際には、MANIFEST.MF ファイルに指定されているバージョンを検証することで、コンテナ・レベル構成に定義されているライブラリの依存性が検証されます。 このプロパティを true に設定すると、問題を説明した警告メッセージがシステムにより発行され、初期化は続行されます。このプロパティの使用方法の詳細は、4-6 ページの「manifest.dependencies.warn.only を使用した共有ライブラリでの異なる依存性の許容」を参照してください。

表 4-3 OC4J の一般的な -D システム・プロパティ (続き)

プロパティ	説明
<code>oracle.j2ee.rmi.loadBalance</code>	<p>このプロパティは、次のいずれかの設定で、レプリケーション・ベースのロード・バランシングを構成します。</p> <ul style="list-style-type: none"> ■ <code>client</code>: クライアントが通信全体の初期参照で最初に選択された OC4J プロセスと通信します。 ■ <code>context</code>: 別のコンテキストが使用されると (非推奨の <code>dedicated.rmicontext</code> プロパティと同じように)、クライアントは新規サーバーに切り替えます。 ■ <code>lookup</code>: クライアントは参照ごとにサーバーを切り替えます。 <p>デフォルト設定は <code>client</code> です。</p> <p>非推奨の <code>dedicated.rmicontext</code>、<code>dedicated.connection</code> および <code>LoadBalanceOnLookup</code> プロパティは、<code>oracle.j2ee.rmi.loadBalance</code> プロパティに置き換えられました。</p>
<code>oracle.mdb.fastUndeploy=int</code>	<p>OC4J が基礎となるデータベースをポーリングして、MDB セッションが停止しているかどうかをチェックする間隔 (秒単位) を設定します。このプロパティを使用すると、Windows 環境で MDB を実行している場合または Windows 環境でバックエンド・データベースが稼働している場合に、OC4J をクリーンに停止できます。</p> <p>通常、MDB を使用する場合、MDB は受信メッセージを待機する受信状態でブロックされます。しかし、Windows 環境で MDB が待機状態のときに OC4J を停止すると、MDB がブロックされているため、OC4J インスタンスは停止されず、アプリケーションはアンデプロイされません。</p> <p>このプロパティを設定すると、MDB が受信メッセージを処理しておらず待機状態であるときに、OC4J はデータベースをポーリングして、セッションが停止しているかどうかを確認できるようになります。このプロパティを設定せずに、<code>[Ctrl]+[C]</code> を使用して OC4J を停止しようとする、OC4J プロセスは 2.5 時間以上ハングします。</p> <p>このポーリング・プロセスは、パフォーマンスに対して高価となる可能性があるため、頻繁に起動しすぎないように設定してください。</p>
<code>oracle.dms.sensors=none normal heavy all</code>	<p>Oracle 組込みパフォーマンス・メトリックの値は、次のように設定できます。</p> <ul style="list-style-type: none"> ■ <code>none</code>: メトリックを無効化 ■ <code>normal</code>: 中程度のメトリック (デフォルト) ■ <code>heavy</code>: 高いメトリック ■ <code>all</code>: 可能なすべてのメトリック <p>このパラメータは、OC4J サーバーで設定する必要があります。これらのパフォーマンス・メトリックを有効にするための以前の方法である <code>oracle.dms.gate=<true false></code> は、<code>oracle.dms.sensors</code> 変数で置き換えられています。ただし、<code>oracle.dms.gate</code> を引き続き使用する場合、この変数を <code>false</code> に設定すると、<code>oracle.dms.sensors=none</code> の設定と同じ意味になります。</p>
<code>associateUsingThirdTable=true false</code>	<p>エンティティ Bean におけるコンテナ管理の関係について、第 3 のデータベース表を使用して関係を管理するかどうかを指定できます。第 3 の関連表を必要としない場合は、<code>false</code> に設定します。デフォルトは <code>false</code> です。</p> <p>このプロパティは非推奨で、OrionCMP でのみ機能します。</p>

表 4-3 OC4J の一般的な -D システム・プロパティ (続き)

プロパティ	説明
DefineColumnType=true false	<p>リリース 9.2.0 より前の Oracle JDBC ドライバを使用している場合は、true に設定します。これらのドライバの場合、この変数を true に設定すると、Oracle JDBC ドライバに対して Select を実行する場合のラウンドトリップを回避できます。このパラメータは、OC4J サーバーで設定する必要があります。デフォルトは false です。</p> <p>このオプションの値を変更して OC4J を再起動すると、この変更の後にデプロイされるアプリケーションに対してのみ有効になります。変更前にデプロイされたアプリケーションには影響はありません。</p> <p>true の場合、DefineColumnType 拡張機能により、通常は表の記述に必要なデータベースへのラウンドトリップが節約されます。Oracle JDBC ドライバが問合せを実行する場合、最初にデータベースへのラウンドトリップを使用して、結果セットの列に使用する必要がある型を判別します。次に、JDBC は問合せからデータを受け取ると、結果セットに移入する際に、データを必要に応じて変換します。</p> <p>DefineColumnType 拡張機能を true に設定して問合せの列の型を指定すると、Oracle データベースへの最初のラウンドトリップが回避されます。そのように最適化されているサーバーは、必要な型変換を実行します。</p>
oc4j.formauth.redirect=true false	<p>このプロパティは、Web アプリケーションでフォームベース認証が使用される場合に適用できます。</p> <p>true に設定した場合、リソースへのアクセス時にユーザーが有効な資格証明を入力すると、OC4J はリクエスト URL へのクライアント・サイド・リダイレクトを実行します。ユーザーに有効な資格証明がない場合、Web ブラウザは、Web アプリケーションに定義されたフォーム認証エラー・ページにリダイレクトされます。</p> <p>false に設定した場合、ユーザーが有効な資格証明を入力すると、ブラウザには /j-security-check の URL が表示されます。デフォルトは false です。</p>
file.encoding=value	<p>このプロパティは、ファイル・エンコーディングのデフォルトのキャラクタ・セットを指定します。エンコーディングが指定されていない場合、OC4J では、デフォルトのキャラクタ・セットとして ISO-8859-1 が使用されます。</p> <p>Windows 環境では、ファイル・エンコーディングのデフォルトのキャラクタ・セットは、ISO 8859-1 キャラクタ・セットのスーパーセットである Windows-1252 であることが必要です。デフォルトとして Windows-1252 キャラクタ・セットを使用するには、-Dfile.encoding=Cp1252 を OC4J の起動プロパティとして指定します。これにより、PrintWriter 出力で Windows-1252 の文字が疑問符 (?) に置き換えられなくなります。</p>
http.file.allowAlias	<p>このプロパティにより、シンボリック・リンクの使用が制御されます。デフォルト値は false で、シンボリック・リンクの使用を防止します。</p> <p>場合によっては、JSP のソース・コードがエンドユーザーに表示される可能性があるため、この設定を true に変更しないことをお勧めします。</p> <p>詳細は、4-7 ページの「シンボリック・リンク使用の防止」を参照してください。</p>

表 4-3 OC4J の一般的な -D システム・プロパティ (続き)

プロパティ	説明
http.maxFileInfoCacheEntries	<p>このプロパティにより、クライアントがパス情報を使用してアプリケーションにアクセスする際に、OC4J が収集するファイル情報が格納されるキャッシュが制御されます。次のように設定すると、キャッシュを制御できます。</p> <ul style="list-style-type: none"> ■ http.maxFileInfoCacheEntries < 0 と設定すると、ファイル情報はキャッシュされません。 ■ http.maxFileInfoCacheEntries == 0 と設定すると、すべてのファイル情報が格納され、オブジェクトはキャッシュから解放されません。 ■ http.maxFileInfoCacheEntries > 0 と設定すると、キャッシュされるエントリの最大数を指定できます。 <p>デフォルト値は 2000 です。</p>
http.proxyHost=proxy_host http.proxyPort=proxy_port	<p>HTTP トラフィックがプロキシ Web サーバーを通過する場合、プロキシ・ホストと必要に応じてプロキシ・ポートをコマンドラインに指定します。proxy_port を省略した場合、デフォルトはポート 80 です。</p>
http.webdir.enable=true false	<p>このプロパティは、OC4J インスタンス内のすべてのサーブレットについて、サーブレット・クラス名の起動を有効または無効にします。</p> <p>true に設定した場合、OC4J インスタンスで稼働しているサーブレットはデフォルトでクラス名により起動できます。false に設定した場合、サーブレットはクラス名により起動できません。デフォルトは false です。</p> <p>Web アプリケーションごとにこの機能を無効にするには、このプロパティを true に設定して、サーブレット・クラス名の起動を許可しない各 Web アプリケーションの orion-web.xml ディスクリプタに <orion-web-app servlet-webdir="" ... /> と設定します。</p> <p>orion-web.xml で servlet-webdir に設定された値は、ORACLE_HOME/j2ee/instance/config/global-web-application でこの属性に設定されたデフォルト値、すなわち servlet-webdir="/servlet" に優先します。</p>
HTTPClient.log.level	<p>このプロパティを使用すると、次のいずれかの java.util.logging.Level 値に設定した場合に、Oracle HTTPClient パッケージを記録できます。</p> <p>CONFIG FINE FINER FINEST ALL</p> <p>HTTPClient ログを無効化するには、このプロパティを OFF に設定します。</p> <p>HTTPClient ログは、トレース・レベル (CONFIG 以下) でのみメッセージ通知され、アプリケーション・ログ・レベル (SEVERE、WARNING および INFO) では使用されません。</p> <p>ログ・レベルの設定および HTTPClient ログの有効化の詳細は、『Oracle Containers for J2EE 開発者ガイド』の第 4 章「ロギング実装のガイドライン」を参照してください。</p>

デバッグ・プロパティの概要

注意：この項で示すデバッグ・プロパティは、OC4J 10g (10.1.3.4.0) では推奨されません。

OC4J に付属するコンポーネント・ログ出力の使用方法は、11-8 ページの「OC4J コンポーネント・ログ出力の使用方法和構成」を参照してください。

次のプロパティを使用して、OC4J 内で稼働しているアプリケーションをデバッグできます。デバッグ・メッセージはコンソールに出力されます。すべてのプロパティがブール値をとります。

すべてのプロパティは、-D で始めます。

表 4-4 OC4J のデバッグ・プロパティ

デバッグ・プロパティ	説明
<code>http.session.debug=true false</code>	HTTP セッションのイベントに関する情報をコンソールに示します。
<code>http.request.debug=true false</code>	各 HTTP リクエストに関する情報をコンソールに示します。
<code>http.cluster.debug=true false</code>	HTTP クラスタリングのイベントに関する情報をコンソールに示します。
<code>http.error.debug=true false</code>	すべての HTTP エラーをコンソールに出力します。
<code>http.method.trace.allow=true false</code>	trace HTTP メソッドを有効にします。
<code>datasource.verbose=true false</code>	データソースの作成、データソースを使用する接続およびプールに解放された接続に関する詳細情報を示します。
<code>jdbc.debug=true false</code>	JDBC コールが行われるときに詳細情報を示します。
<code>ejb.cluster.debug=true false</code>	EJB クラスタリングのデバッグ・メッセージを有効にします。
<code>rmi.debug=true false</code>	RMI デバッグ情報をコンソールに出力します。
<code>rmi.verbose=true false</code>	RMI コールに関する詳細情報を示します。
<code>jca.connection.debug=true false</code>	J2CA 接続に関するその他の診断情報を示します。
<code>ws.debug=true false</code>	Web サービスのデバッグを有効にします。

統合開発環境からのリモート・デバッグの有効化

開始パラメータとして指定されている JVM デバッグ・コマンドを使用して OC4J インスタンスを起動すると、リモート・デバッグの接続が可能になるため、OC4J のアプリケーションを統合開発環境 (IDE) からリモートでデバッグできます。次の各項では、これらのパラメータの指定方法について説明します。

- [Application Server Control](#) を使用した OC4J インスタンスのリモート・デバッグの有効化
- `opmn.xml` ファイルへのデバッグ開始パラメータの指定
- 起動コマンドラインへのデバッグ開始パラメータの指定

Application Server Control を使用した OC4J インスタンスのリモート・デバッグの有効化

Application Server Control を使用して単一の OC4J インスタンスのリモート・デバッグを有効化するには、次のようにします。

1. OC4J ホームページに移動します。
2. 「管理」をクリックして、OC4J の「管理」ページを表示します。
3. 管理タスクの表の「プロパティ」の下で、「サーバー・プロパティ」行のタスク・アイコンをクリックします。
「サーバー・プロパティ」ページが表示されます。
4. 「コマンドライン・オプション」の下の「開始パラメータ:Java オプション」セクションで「行の追加」をクリックして、次に示す各デバッグ開始パラメータを追加します。
 - -Xdebug
 - -Xnoagent
 - -Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=4000
5. 「適用」をクリックして、変更を OC4J 構成に適用します。

サーバー・プロパティで変更を行ったら、変更内容を反映する前に OC4J インスタンスを再起動する必要があります。

opmn.xml ファイルへのデバッグ開始パラメータの指定

OPMN 管理の OC4J インスタンスでは、id 属性が start-parameters の <category> 要素内の <data> サブ要素 (id 属性は java-options) の値として、opmn.xml ファイルにデバッグ・パラメータを配置し、インスタンスを再起動できます。opmn.xml のエントリーは、次のようになります。

```
<module-data>
  <category id="start-parameters">
    <data id="java-options" value="-server -Xdebug -Xnoagent
      -Djava.compiler=NONE
      -Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=4000
      -Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy
      -Djava.awt.headless=true"/>
  </category>
</module-data>
```

デバッグが追加されるまで OC4J が起動されなくなるため、suspend=y は使用しないでください。このデバッグ・パラメータを使用すると、OPMN は問合せ ping からレスポンスを取得できなくなるため、OPMN による OC4J インスタンスの再起動が継続的に試行されます。

ポート 4000 などの address を設定するポートをサーバーに追加します。

注意： 4000 というポート値は任意の値です。ユーザーの接続に適した値を設定する必要があります。指定されたポートは、サーバーに接続するためにリモートのデバッグ・クライアントに設定する必要のあるポートです。

起動コマンドラインへのデバッグ開始パラメータの指定

スタンドアロン OC4J インスタンスでは、次のように、起動コマンドラインにデバッグ開始パラメータを指定できます。

```
java -Xdebug -Xnoagent -Djava.compiler=NONE
-Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=4000 -jar oc4j.jar
```

リモートからの Web アプリケーションのデバッグ

IDE から Web アプリケーションをデバッグするには、次に示す項で説明するように、サーブレットおよび JSP デバッグ用の設定を行う必要があります。

- サブレットのリモート・デバッグ用の設定
- JSP のリモート・デバッグ用の設定

サブレットのリモート・デバッグ用の設定

サブレットのリモート・デバッグを設定するには、次のようにします。

1. プロジェクトの実行、またはリモート・デバッグを行うためのデバッグ構成をマークします。
2. リモート・デバッグ固有のランタイム構成ノードの JPDA への追加を設定します。
3. 4-14 ページの「起動コマンドラインへのデバッグ開始パラメータの指定」の例で示されているように、デバッグ・パラメータを使用して OC4J インスタンスを起動します。
4. サブレットにブレークポイントを設定します。
5. リモート・デバッガを実行します。

Web ブラウザからサブレットを起動すると、サブレットがブレークポイントに到達します。

JSP のリモート・デバッグ用の設定

JSP の場合は、前項「サブレットのリモート・デバッグ用の設定」に説明されているように設定できますが、手順がもう 1 つ必要です。デフォルトで `ORACLE_HOME/j2ee/instance/config` にインストールされている `global-web-application.xml` ファイルを編集し、JSP 部分に少なくとも次に示すパラメータを設定する必要があります。

```
<init-param>
  <param-name>debug</param-name>
  <param-value>>true</param-value>
</init-param>
<init-param>
  <param-name>developer_mode</param-name>
  <param-value>>true</param-value>
</init-param>
<init-param>
  <param-name>encode_to_java</param-name>
  <param-value>>true</param-value>
</init-param>
<init-param>
  <param-name>reduce_tag_code</param-name>
  <param-value>>false</param-value>
</init-param>
<init-param>
  <param-name>extra_imports</param-name>
  <param-value></param-value>
</init-param>
<init-param>
  <param-name>main_mode</param-name>
  <param-value>recompile</param-value>
</init-param>
<init-param>
  <param-name>debug_mode</param-name>
  <param-value>>true</param-value>
</init-param>
```

OC4J の起動および停止

この章では、OC4J のインストールと、OC4J インスタンスの起動、停止および再起動の手順について説明します。この章の内容は次のとおりです。

- スタンドアロン環境での OC4J の起動
- Oracle Application Server 環境での OC4J の起動
- スタンドアロン環境での OC4J の停止
- Oracle Application Server 環境での OC4J の停止
- スタンドアロン環境での OC4J インスタンスの再起動
- Oracle Application Server 環境での OC4J インスタンスの再起動

スタンドアロン環境での OC4J の起動

スタンドアロン環境の OC4J サーバー・インスタンスは、oc4j コマンド・スクリプトまたは実行可能な oc4j.jar アーカイブのいずれかを用いて、デフォルト構成を使用して起動できます。

oc4j スクリプトによる OC4J の起動

oc4j スクリプトを使用して OC4J を起動するには、次のコマンドを `ORACLE_HOME/bin` ディレクトリから発行します。

```
oc4j -start
```

このコマンドを使用するには、`ORACLE_HOME` および `JAVA_HOME` 環境変数を設定する必要があります。詳細は、2-2 ページの「[スタンドアロン OC4J サーバーのインストール前提条件への準拠](#)」を参照してください。

oc4j.jar による OC4J の起動

oc4j.jar を起動して OC4J を起動するには、次のコマンドを `ORACLE_HOME/j2ee/home` ディレクトリから発行します。

```
java -jar oc4j.jar [args]
```

このコマンドで示したように oc4j.jar を起動すると、OC4J はデフォルトの `server.xml` 構成ファイルを使用して起動されます。このファイルは `j2ee/home/config` ディレクトリにあります。デフォルト以外の `server.xml` ファイルを使用して OC4J を起動するには、次のコマンドを発行します。変更済の構成ファイルへのパスを指定する必要があります。

```
java -jar oc4j.jar -config /yourpath/server.xml [args]
```

ランタイム・オプションを OC4J に設定するために、起動時に引数を渡すことができます。有効な引数の概要は、4-3 ページの「[起動時の OC4J ランタイム・オプションの設定](#)」を参照してください。また、次のコマンドを `ORACLE_HOME/j2ee/home` ディレクトリから発行して、コンソール・ヘルプを表示することもできます。

```
java -jar oc4j.jar -help
```

OC4J の起動時に oc4j.jar コマンドラインを使用して、システム・プロパティを JVM に設定することもできます。システム・プロパティの設定の詳細は、4-5 ページの「[起動時のシステム・プロパティの設定](#)」を参照してください。

Oracle Application Server 環境での OC4J の起動

管理構成では、すべての Oracle Application Server コンポーネント (OC4J および Oracle HTTP Server を含む) は、Application Server Control の「[クラスタ・トポロジ](#)」ページから、または OPMN コマンドライン・ツール `opmnctl` を用いて、OPMN を使用して起動する必要があります。このツールは、`ORACLE_HOME/opmn/bin` ディレクトリにインストールされています。

ローカルの Oracle Application Server インスタンス上のすべての OPMN 管理プロセス (OC4J を含む) を起動するには、次のコマンドを使用します。

```
opmnctl startall
```

ローカルの Oracle Application Server インスタンス上の特定の管理プロセス (この場合は OC4J) を起動するには、次のコマンドを使用します。

```
opmnctl startproc ias-component=default_group
```

複数の OC4J インスタンスを含むクラスタ・トポロジでは、OC4J インスタンスによって使用される EAR が単一の場所にある共有ディレクトリにある場合は、`-sequential` フラグを使用してインスタンスを起動する必要があります。

```
opmnctl startproc ias-component=default_group -sequential
```

このオプションにより、すべての OC4J インスタンスを同時に起動した場合に発生する可能性のあるリソースの競合を防ぐことができます。

あるいは、OC4J インスタンスを順番に起動するには、次に示すように、クラスタの OPMN 構成ファイル opmn.xml に sequential オプションを指定できます。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="oc4j-options" value="-sequential" />
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

opmnctl コマンドの詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

注意： Windows 環境で Oracle Application Server への接続を試行する前に、サーバーの IP アドレスで C:\WINDOWS\system32\drivers\etc\hosts ファイルを更新する必要があります。

スタンドアロン環境での OC4J の停止

スタンドアロン OC4J サーバーは、admin_client.jar または admin.jar コマンドラインユーティリティで -shutdown コマンドを起動するか、oc4j.cmd または oc4j 実行可能スクリプトを起動することで停止できます。

注意： Windows 環境の Control-C などのオペレーティング・システム・コマンド、または Linux や UNIX 環境の kill は、OC4J の停止に使用しないでください。

EJB モジュールを使用するアプリケーションが OC4J 内でアクティブに稼働している場合は、特にこれに該当します。これらのコマンドにより、サーバーが停止する前に EJB メソッド・コールやタイマー操作を完了することができなくなるためです。

admin_client.jar によるスタンドアロン OC4J の停止

admin_client.jar を使用して OC4J を停止するには、次のコマンドを発行します。

```
java -jar admin_client.jar uri adminId adminPassword -shutdown
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin password -shutdown
```

このコマンドは、ホスト・マシンが接続されていないかのように、すべてのスレッドを即座に終了させて OC4J サーバー全体を停止します。このコマンドを使用すると、クラスタ化されたアプリケーションの現在の状態はレプリケートされません。

uri、adminId および adminPassword 変数の詳細は、6-2 ページの「[admin_client.jar の構文および URI の指定の概要](#)」を参照してください。

スタンドアロン OC4J インスタンスでは、admin_client.jar の -shutdown オプションは、5-4 ページの「[admin.jar による OC4J の停止](#)」で説明されている admin.jar ユーティリティの -shutdown force オプションと同等です。

admin.jar による OC4J の停止

admin.jar を使用して OC4J を停止するには、次のコマンドを発行します。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-shutdown [ordinary|force] [reason]
```

次のオプションを指定できます。

- ordinary|force

停止タイプ。デフォルトは ordinary で、各スレッドを正常終了させます。

force オプションは、すべてのスレッドを即時終了させます。実質的にはホスト・マシンのプラグを抜く場合と同じです。このオプションを使用すると、クラスタ化されたアプリケーションの現在の状態はレプリケートされません。

- reason

停止の理由を文字列として指定します。この文字列は

ORACLE_HOME/j2ee/home/log/server.log ファイルに書き込まれます。文字列には空白を使用できません。

次の例では、admin.jar を使用して OC4J サーバーの停止を強制し、すべてのスレッドを即時終了させています。停止の reason として入力された文字列は、ORACLE_HOME/j2ee/home/config/server.log ファイルに書き込まれます。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown force
need_to_reboot_host_machine
```

oc4j スクリプトによる OC4J の停止

oc4j スクリプトの 1 つを使用して OC4J を停止するには、次のコマンドを ORACLE_HOME/bin ディレクトリから発行します。OC4J で使用される ORMI ポート（デフォルトでは 23791）と oc4jadmin アカウントのパスワードを指定する必要があります。

```
oc4j -shutdown -port oc4jOrmiPort -password adminPassword
```

次に例を示します。

```
oc4j -shutdown -port 23791 -password adminpwd
```

このコマンドを使用するには、ORACLE_HOME および JAVA_HOME 環境変数を設定する必要があります。詳細は、2-2 ページの「スタンドアロン OC4J サーバーのインストール前提条件への準拠」を参照してください。

Oracle Application Server 環境での OC4J の停止

管理構成では、Application Server Control の「クラスタ・トポロジ」ページから OC4J インスタンスを停止できます。また、OPMN コマンドライン・ツールの `opmnctl` や、OPMN にインスタンスが停止されたことを通知する `admin_client.jar` コマンドライン・ユーティリティを使用して停止することも可能です。この OPMN ツールは、`ORACLE_HOME/opmn/bin` ディレクトリにインストールされています。`admin_client.jar` ユーティリティは、`ORACLE_HOME/j2ee/instance` ディレクトリにデフォルトでインストールされます。

ローカルの Oracle Application Server インスタンス上のすべての OPMN 管理プロセス (OC4J を含む) を停止するには、次のコマンドを使用します。

```
opmnctl stopall
```

ローカルの Oracle Application Server インスタンス上の特定の OPMN 管理コンポーネント (この場合は OC4J) を停止するには、次のコマンドを使用できます。

```
opmnctl stopproc ias-component=default_group
```

`opmnctl` コマンドの詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

また、次のコマンドで、`admin_client.jar` を使用して OC4J インスタンスを停止することも可能です。

```
java -jar admin_client.jar uri adminId adminPassword -shutdown
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:opmn://localhost/home oc4jadmin password -shutdown
```

このコマンドは、すべてのスレッドを即座に終了させて OC4J インスタンス全体を停止します。OPMN 管理の OC4J インスタンスでは、OPMN が再起動を試行しないように、サーバーを故意に停止中であることが `admin_client.jar` により OPMN に通知されます。このコマンドを使用すると、クラスタ化されたアプリケーションの現在の状態はレプリケートされません。

`uri`、`adminId` および `adminPassword` 変数の詳細は、6-2 ページの「[admin_client.jar の構文および URI の指定の概要](#)」を参照してください。

スタンドアロン環境での OC4J インスタンスの再起動

次のコマンドで、`admin_client.jar` ユーティリティを使用して、スタンドアロン OC4J サーバーを再起動できます。

```
java -jar admin_client.jar uri adminId adminPassword -restart
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin password -restart
```

`uri`、`adminId` および `adminPassword` 変数の詳細は、6-2 ページの「[admin_client.jar の構文および URI の指定の概要](#)」を参照してください。

また、次のコマンドで `admin.jar` コマンドライン・ユーティリティを使用して、スタンドアロン OC4J インスタンスを再起動することもできます。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword -restart [reason]
```

[reason] の値として文字列を入力できます。この文字列は、`ORACLE_HOME/j2ee/home/config/server.log` ファイルに書き込まれます。

Oracle Application Server 環境での OC4J インスタンスの再起動

管理構成では、Application Server Control の「クラスタ・トポロジ」ページから OC4J を再起動できます。また、OPMN コマンドライン・ツールの `opmnctl` や、`admin_client.jar` コマンドライン・ユーティリティを使用して再起動することも可能です。この OPMN ツールは、`ORACLE_HOME/opmn/bin` ディレクトリにインストールされています。`admin_client.jar` ユーティリティは、`ORACLE_HOME/j2ee/instance` ディレクトリにデフォルトでインストールされます。

ローカルの Oracle Application Server インスタンス上のすべての OPMN 管理プロセス (OC4J を含む) を再起動するには、次のコマンドを使用できます。

```
opmnctl startall
```

また、次のコマンドを使用して、ローカルの Oracle Application Server インスタンス上の特定の管理プロセス (この場合は OC4J) を再起動することもできます。

```
opmnctl restartproc ias-component=home
```

複数の OC4J インスタンスを含むクラスタ・トポロジでは、`opmnctl` コマンドおよび `-sequential` フラグを使用してインスタンスを再起動する必要があります。

```
opmnctl startproc ias-component=default_group -sequential
```

`-sequential` フラグを使用すると、すべての OC4J インスタンスを同時に起動した場合、特に OC4J インスタンスによって使用される EAR が単一の場所にある共有ディレクトリにある場合に発生する可能性のあるリソースの競合を防ぐことができます。5-2 ページの「[Oracle Application Server 環境での OC4J の起動](#)」で説明されているように、クラスタの `opmn.xml` 構成ファイルに `sequential` オプションが含まれる場合は、`-sequential` フラグを指定する必要はありません。

次のコマンドで、`admin_client.jar` を使用して OC4J インスタンスを再起動できます。

```
java -jar admin_client.jar uri adminId adminPassword -restart
```

`uri`、`adminId` および `adminPassword` 変数の詳細は、6-2 ページの「[admin_client.jar の構文および URI の指定の概要](#)」を参照してください。

admin_client.jar ユーティリティの使用法

OC4J には `admin_client.jar` というコマンドライン・ユーティリティが用意されています。このユーティリティを使用すると、Oracle Application Server のクラスタ化された環境およびスタンドアロン OC4J サーバーのアクティブな OC4J インスタンスで構成、管理およびデプロイ・タスクを実行できます。また、`admin_client.jar` を使用して、OC4J インスタンスやインスタンスのグループを再起動または停止することも可能です。

`admin_client.jar` ユーティリティは、リモートで操作を実行する管理クライアントの一部でもあり、Oracle Application Server 10g リリース 3 (10.1.3.4.0) の Companion CD から、または Oracle Technology Network からダウンロードして入手できます。

特定の OC4J インスタンスで操作を実行することも、グループ内のすべての OC4J インスタンスで同時に操作を実行することも可能です。Oracle Application Server 10g リリース 3 (10.1.3.4.0) では、**グループ**は同じ**クラスタ・トポロジ** (2 つ以上の粗結合の Oracle Application Server ノード) に属する OC4J インスタンスの同期セットです。

`admin_client.jar` コマンドライン・ユーティリティを使用すると、OC4J インスタンスまたは OC4J インスタンスのグループで次の操作を実行できます。

- エンタープライズ・アプリケーション・アーカイブ (EAR)、スタンドアロン Web モジュール (WAR)、Enterprise JavaBeans (EJB) モジュール (EJB JAR)、またはスタンドアロン・リソース・アダプタ (RAR) のデプロイ
- アプリケーション、Web モジュール、EJB モジュールまたはリソース・アダプタのアンデプロイ
- 変更されたクラスを保持するデプロイ済 EJB モジュールの増分的な更新
- アプリケーションの共有ライブラリの作成、変更または削除
- アプリケーションの起動、再起動または停止
- OC4J インスタンスやインスタンスのグループの再起動または停止
- データソースおよびデータソース接続プールの追加、テストおよび削除
- JMS 接続プールや接続先の追加および削除

Application Server Control または OC4J Ant タスクを使用して類似の操作を実行できます。詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』の Application Server Control を使用したデプロイに関する項または OC4J Ant タスクを使用したデプロイに関する項を参照してください。

この章の内容は次のとおりです。

- [admin_client.jar の使用の準備](#)
- [アーカイブのデプロイ](#)
- [デプロイ後の Web サイトへの Web モジュールのバインド](#)
- [アーカイブの再デプロイ](#)
- [アーカイブのアンデプロイ](#)
- [デプロイ済 EJB モジュールの変更されたクラスの更新](#)

- 共有ライブラリの作成および管理
- アプリケーションの起動、再起動および停止
- OC4J インスタンスの再起動および停止
- データソースの管理
- JMS リソースの管理
- リモート・クライアントを使用した OC4J の管理

admin_client.jar の使用の準備

admin_client.jar ユーティリティは、各 OC4J インスタンスの `ORACLE_HOME/j2ee/instance` ディレクトリにデフォルトでインストールされます。これは、OC4J で操作を実行する際に推奨されるコマンドライン・ツールです。このユーティリティはリモートで操作を実行する管理クライアント・ユーティリティの一部でもあり、Oracle Application Server 10g リリース 3 (10.1.3.4.0) の Companion CD から、または Oracle Technology Network からダウンロードして入手できます。

OC4J インスタンスでこのユーティリティを使用して操作を実行するには、インスタンスを起動する必要があります。

この項の内容は次のとおりです。

- [admin_client.jar](#) の構文および URI の指定の概要
- リモート管理クライアントのダウンロードおよび抽出
- コンソールへの使用方法の出力
- ロギングの有効化

admin_client.jar の構文および URI の指定の概要

admin_client.jar ユーティリティは、次の構文を使用します。

```
java -jar admin_client.jar uri adminId adminPassword command
```

コマンドラインで渡される主要なパラメータは、`uri` です。これは、指定された 1 つ以上のコマンドのターゲットを指定します。URI の構文は、ターゲットとなるインスタンスによって異なります。この URI の書式は、次の項を参照してください。

- クラスタ内の OC4J インスタンスのグループでの操作の実行
- 特定の OC4J インスタンスでの操作の実行
- スタンドアロン OC4J サーバーでの操作の実行
- URI の検証

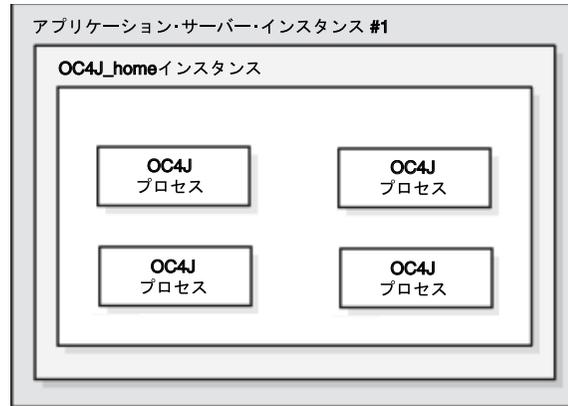
OC4J 管理ユーザー名およびパスワードも `admin_client.jar` ユーティリティに渡されます。デフォルトの管理者アカウントのユーザー名は `oc4jadmin` です。

たとえば、次のコマンドは、`petstore` アプリケーションを起動します。このアプリケーションは、クラスタのメンバーである `node1` 上の `oc4j_2` という名前の OC4J インスタンスにインストールされています。

```
java -jar admin_client.jar deployer:oc4j:opmn://node1.company.com/oc4j_2
oc4jadmin password -application petstore -start
```

図 6-1 に、クラスタ内の Oracle Application Server インスタンスの 1 つである OC4J_home という名前の OC4J インスタンスから実行するよう構成されている 4 つのプロセスを示します。

図 6-1 クラスタ内の Oracle Application Server インスタンスにおいて複数の JVM で実行される OC4J インスタンス



注意：通常、home という名前の OC4J インスタンスは、複数プロセス・モデルでの実行には適さない Application Server Control アプリケーションのホストであるため、複数のプロセスで実行するように構成することはできません。

クラスタ内の OC4J インスタンスのグループでの操作の実行

次の URI を使用して、グループ内のすべての OC4J インスタンスをターゲットとして指定します。**グループ**は、同じクラスタ・トポロジに属する OC4J インスタンスの同期セットです。グループのすべての OC4J インスタンスで、構成、管理およびデプロイ操作を同時に実行できます。たとえば、Oracle Application Server クラスタ内の oc4j_soa という名前のグループに属するすべての OC4J インスタンスを同時に停止できます。

URI は OPMN ベースのクラスタリング・フレームワークを使用します。このフレームワークでは、クラスタ・ノード同士が相互に認識しています。指定する必要があるのはホスト名と、オプションで、クラスタ内の任意の Oracle Application Server ノードの OPMN リクエスト・ポートのみです。これにより、アプリケーションが、ホスト名とクラスタ内のその他すべてのノードの OPMN ポートを取得できるようになります。

URI の構文は次のとおりです。

```
deployer:cluster:[rmis]:opmn://opmnHost[:opmnPort]/groupName
```

次に例を示します。

```
deployer:cluster:opmn://node1.company.com/oc4j_soa
```

表 6-1 グループをターゲットとするための URI パラメータ

パラメータ	説明
<i>rmis</i>	任意。ターゲットが ORMI over SSL (ORMIS) を使用する場合に指定します。
<i>opmnHost</i>	必須。クラスタ内の Oracle Application Server ノードのホスト名。どのノードでも指定できます。クラスタ内の他のノードのリストはこのノードから取得されます。
<i>opmnPort</i>	任意。opmn.xml に指定されている OPMN リクエスト・ポート。ポートを指定しない場合は、デフォルト・ポート 6003 が使用されます。
<i>groupName</i>	必須。クラスタ内で、OC4J インスタンスが属するグループの名前。

特定の OC4J インスタンスでの操作の実行

次の URI 構文を使用して、特定の OPMN 管理の OC4J インスタンス（クラスタ内のインスタンスなど）をターゲットとします。接頭辞では、`cluster` が `oc4j` に置き換えられます。

インスタンスのホストとなる Oracle Application Server ノードのホスト名を指定します。ノードのホスト名またはポートが不明な場合は、クラスタ内の別のノードのホスト名と Oracle Application Server インスタンスの名前を指定できます。アプリケーションは、OPMN クラスタリング・フレームワークを使用して Oracle Application Server インスタンスのホストとなるノードを探し出します。

URI の構文は次のとおりです。

```
deployer:oc4j:[rmis]:opmn://host[:opmnPort]/[iASInstanceName]/oc4jInstanceName
```

次に例を示します。

```
deployer:oc4j:opmn://server.company.com:6004/instance2/home
```

表 6-2 特定のインスタンスをターゲットとするための URI パラメータ

パラメータ	説明
<i>rmis</i>	任意。ターゲットが ORMI over SSL (ORMIS) を使用する場合に指定します。
<i>host</i>	必須。OPMN サーバーとして使用するための、クラスタ内でターゲットとなる Oracle Application Server ノードのホスト名。
<i>opmnPort</i>	任意。opmn.xml に指定されている OPMN リクエスト・ポート。ポートを指定しない場合は、デフォルト・ポート 6003 が使用されます。
<i>iASInstanceName</i>	任意。 <i>host</i> に指定されたノード上に存在しない場合の、ターゲットとなる Oracle Application Server インスタンスの名前。
<i>oc4jInstanceName</i>	必須。ターゲット OC4J インスタンスの名前。

スタンドアロン OC4J サーバーでの操作の実行

次の URI のいずれかを使用して、スタンドアロン OC4J サーバー・インスタンスをターゲットとします。

RMI を使用する場合は、次のようにして URI を指定します。

```
deployer:oc4j:host:rmiPort
```

次に例を示します。

```
deployer:oc4j:myserver:23791
```

ORMI over SSL (ORMIS) を使用する場合は、次のようにして URI を指定します。

```
deployer:oc4j:rmi:host:ormisPort
```

次に例を示します。

```
deployer:oc4j:rmi:myserver:23943
```

表 6-3 スタンドアロン OC4J をターゲットとするための URI パラメータ

パラメータ	説明
rmi	ターゲットが ORMI over SSL (ORMIS) を使用する場合は必須です。
host	必須。クラスタ内の Oracle Application Server ノードのホスト名。どのノードでも指定できます。クラスタ内の他のノードのリストはこのノードから取得されます。
rmiPort	RMI が使用される場合は必須です。インスタンス固有の rmi.xml ファイルに指定されている RMI ポートです。
ormisPort	ORMIS が使用される場合は必須です。インスタンス固有の rmi.xml ファイルに指定されている SSL ポートです。

URI の検証

URI は、`-validateURI` コマンドを使用して検証できます。

```
java -jar admin_client.jar uri adminId adminPassword -validateURI
```

次に例を示します。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -validateURI
```

リモート管理クライアントのダウンロードおよび抽出

管理クライアント・ユーティリティのディストリビューションには、`admin_client.jar` コマンドライン・ユーティリティが含まれています。このユーティリティは OC4J または Oracle Application Server ターゲットに接続し、一連のライフサイクル、デプロイおよびリソース構成の操作を実行できます。

リモート・システムで Oracle Application Server インスタンスに対して通常の操作を実行する必要があります。たとえばリモート・システムに、デプロイ操作、またはアプリケーション固有またはサーバーの JMX MBeans の管理目的での問合せや操作など、自動化された作成またはテスト・プロセスが複数あるとします。または、リモート・システムで、構成およびデプロイ操作の本番セットに対して定期的にスケジュールされたテストを実行するとします。管理クライアント・ユーティリティを使用してこれを実行すれば、リモート・システムに完全な OC4J または Oracle Application Server インストールを用意する必要がなくなります。

Oracle Application Server 10g リリース 3 (10.1.3.4.0) の別のディストリビューションである管理クライアント・ユーティリティは、Oracle Technology Network からダウンロードできます。また、Oracle Application Server の Companion CD にも含まれています。ディストリビューション・ファイル oc4j_admin_client_101340.zip には、OC4J インスタンスのリモート管理に必要なものがすべて含まれています。

- OC4J または Oracle Application Server ターゲットへの ORMI プロトコルを使用したリモート JMX 接続の確立に必要な Java ライブラリ
- 機能するために必要なライブラリ付きの実行可能な admin_client.jar ユーティリティ
- リモート・クライアント・ロールに関連する標準の J2EE ライブラリ

管理クライアント・ユーティリティをダウンロードおよび抽出する手順：

1. Oracle Technology Network から oc4j_admin_client_101340.zip をダウンロードします。

```
http://download.oracle.com/otn/java/oc4j/10131/oc4j_admin_client_101340.zip
```

2. oc4j_admin_client_101340.zip のコンテンツをローカル・ディレクトリに解凍します。次に例を示します。

```
>mkdir oc4j_admin_client
>cd oc4j_admin_client
>jar xvf d:¥software¥oc4j_admin_client_101340.zip
```

その結果、ディレクトリ構造は次のようになります。

```
¥j2ee
  ¥home
    oc4jclient.jar
    admin_client.jar
  ¥lib
    ejb.jar
    mail.jar
    adminclient.jar
    javax88.jar
    javax77.jar
    jmx_remote_api.jar
    jmxri.jar
  ¥lib
    xmlparserv2.jar
    dms.jar
  ¥opmn
    ¥lib
      optic.jar
  ¥jlib
    oraclepki.jar
    ojpse.jar
```

次の URI では、個々の OC4J ターゲットに異なるパターンが使用されます。

- スタンドアロン OC4J サーバー：


```
deployer:oc4j:test-cycle.oracle.com:23791
```
- Oracle Application Server の特定の OC4J インスタンス：


```
deployer:oc4j:opmn://test-cycle.oracle.com/testunit
```
- クラスタ内の OC4J インスタンスのグループ：


```
deployer:cluster:opmn://test-cycle.oracle.com/[groupName]
```

3. admin_client.jar を1つ以上のターゲット OC4J インスタンスに接続し、接続をテストします。次に例を示します。

```
>cd j2ee¥home
>java -jar admin_client.jar
  deployer:oc4j:opmn://test-cycle.oracle.com/testunit
  oc4jadmin welcome1
  -validateURI
```

```
URI deployer:oc4j:opmn://test-cycle.oracle.com/testunit is valid and connected
```

コンソールへの使用方法の出力

admin_client.jar コマンドのオンライン・ヘルプ・テキストをコンソールに出力するには、-help とコマンドラインに入力するだけです。次に例を示します。

```
java -jar admin_client.jar -help
```

特定のコマンドの詳細なヘルプを表示するには、-usage の後に続けてコマンド識別子を入力します。次に例を示します。

```
java -jar admin_client.jar -usage [command]
```

ロギングの有効化

admin_client.jar の実行中に発生するエラーのトラブルシューティングを支援するために、このツールを実行する際に Java ロギングを有効にできます。ログ・メッセージはコンソールに出力されます。

ロギングを有効にするには、次のようにします。

1. 次の1行を記述した logging.properties ファイルを作成します。

```
oracle.oc4j.admin.jmx.client.CoreRemoteMBeanServer.level=INFO
```

このファイルを `ORACLE_HOME/j2ee/instance` 以外の場所に作成する場合は、後述のコマンドでファイルへのパスを指定する必要があります。

2. 次のように `-Djava.util.logging.config.file=logging.properties` を admin_client.jar コマンドラインに設定します。

```
java -Djava.util.logging.config.file=logging.properties -jar admin_client.jar uri
adminId adminPassword command
```

logging.properties ファイルの値には、表 6-4 の Java ログ・レベル値のいずれかを設定できます。

表 6-4 Java ログ・レベル

Java ログ・レベル	説明
SEVERE	システム管理者が注意する必要があるシステム・エラーを記録します。
WARNING	エラーが発生する前に確認して処理する必要があるとして検出されたアクションまたは条件を記録します。
INFO	標準のアクションまたはイベントを記録します。ユーザー操作（「ログイン完了」など）や自動操作（ログイン・ファイルの循環など）が該当します。
CONFIG	ログ構成に関連するログ・メッセージまたは問題。
FINE	デバッグやパフォーマンスの監視に使用されるトレースまたはデバッグのメッセージを記録します。通常、詳細なイベント・データが含まれます。
FINER	非常に詳細なトレースまたはデバッグのメッセージを記録します。
FINEST	きわめて詳細なトレースまたはデバッグのメッセージを記録します。

次に例を示します。

```
oracle.oc4j.admin.jmx.client.CoreRemoteMBeanServer.level=FINE
```

アーカイブのデプロイ

admin_client.jar ユーティリティを使用すると、アプリケーション (EAR)、スタンドアロン Web モジュール (WAR) またはスタンドアロン・リソース・アダプタ (RAR) を特定の OC4J インスタンスまたは OC4J インスタンスのグループにデプロイできます。

この項の内容は次のとおりです。

- [J2EE アプリケーション \(EAR\) のデプロイ](#)
- [リモート・クライアントからの J2EE アプリケーションのデプロイ](#)
- [スタンドアロン Web モジュール \(WAR\) のデプロイ](#)
- [スタンドアロン・リソース・アダプタ \(RAR\) のデプロイ](#)
- [バッチ・デプロイでのスクリプト・ファイルの使用](#)

注意： グループにアーカイブをデプロイするには、すべてのインスタンスが同じ oc4jadmin アカウントのパスワードを保持する必要があります。

J2EE アプリケーション (EAR) のデプロイ

-deploy コマンドを使用して、EAR ファイルとしてパッケージされた J2EE アプリケーションをデプロイまたは再デプロイします。EAR 固有の構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file path/filename
-deploymentName appName [-bindAllWebApps [webSiteName]] [-targetPath path]
[-parent appName] [-deploymentDirectory path] [sequential [waitsec]] [-enableIIOP]
[-iiopClientJar path/filename] [-deploymentPlan path/filename] [-removeArchive]
```

-bindAllWebApps パラメータを指定して、EAR 内のすべての Web モジュールをそのアクセスに使用される Web サイトにバインドできます。Web サイトを指定しない場合、モジュールはデフォルト Web サイトにバインドされます。

たとえば、次のコマンドは、クラスタ内のグループ default_group に属するすべての OC4J インスタンスに、utility アプリケーションをデプロイします。アプリケーション内のすべての Web モジュールは、デフォルト Web サイトにバインドされます。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -deploy -file C:/dev/utility.ear -deploymentName utility
-bindAllWebApps
```

表 6-5 EAR デプロイの -deploy コマンドのパラメータ

パラメータ	説明
-file	必須。デプロイする EAR ファイルのパスおよびファイル名。
-deploymentName	必須。ユーザー定義のアプリケーションのデプロイ名。OC4J 内でアプリケーションを識別する際に使用されます。
-bindAllWebApps	任意。EAR 内のすべての Web モジュールを指定された Web サイトにバインドします。何も指定されない場合は、デフォルト Web サイトにバインドします。 webSiteName の値を指定できます。この値は、Web サイト構成を含む name_web-site.xml ファイルの name 部分です。 このパラメータが指定されていない場合は、デプロイ後に -bindAllWebApps コマンドを使用できます。このコマンドの詳細は、6-14 ページの「 単一の Web サイトへの全 Web モジュールのバインド 」を参照してください。
-targetPath	任意。EAR のデプロイ先となるディレクトリ。ディレクトリが指定されていない場合、EAR は ORACLE_HOME/j2ee/instance/applications ディレクトリにデフォルトでデプロイされます。 また、デプロイ済 EAR ファイルもこのディレクトリにコピーされます。後続のデプロイのたびに、この EAR ファイルは上書きされます。
-parent	任意。このアプリケーションの親アプリケーション。デフォルトは、default アプリケーションまたはグローバル Web アプリケーションです。
-deploymentPlan	任意。アプリケーションに適用するデプロイ・プランのパスおよびファイル名。プランは、前回のデプロイ時に XML ファイルとして保存されています。ファイルは、ローカル・ホスト上に存在する必要があります。
-deploymentDirectory	任意。OC4J 固有のデプロイメント・ディスクリプタおよび生成済ファイル（コンパイル済 JSP クラスや EJB ラッパー・クラスなど）が格納されるディレクトリ。 デフォルトのディレクトリは、ORACLE_HOME/j2ee/instance/applications です。

表 6-5 EAR デプロイの -deploy コマンドのパラメータ (続き)

パラメータ	説明
-sequential [waitsec]	<p>任意。グループ内の各 OC4J インスタンスにアーカイブをデプロイする場合に指定します。各ターゲット OC4J インスタンスへのデプロイは、次のターゲット・インスタンスでデプロイが始まる前に完了する必要があります。リクエストは、EAR をデプロイ中の OC4J インスタンスにはルーティングされません。</p> <p>次のように、waitsec オプションを使用して、デプロイ間に待機する秒数を指定できます。</p> <pre>-sequential 15</pre> <p>waitsec オプションの詳細は、6-16 ページの「クラスタ内の順次再デプロイ間の遅延の指定」を参照してください。</p> <p>このパラメータが指定されていない場合、アーカイブはデフォルトで、グループ内のすべての OC4J インスタンスに同時にデプロイされます。</p> <p>このオプションは、クラスタ化された環境でのみ有効です。スタンドアロン OC4J では無効です。</p>
-enableIIOP	<p>任意。OC4J サーバーで IIOP クライアント・スタブを生成する場合にこのパラメータを指定します。</p> <p>すべての EJB モジュールに対して生成されるアプリケーション・レベルのスタブは、<code>ORACLE_HOME/j2ee/instance/application-deployments/appName</code> ディレクトリの <code>_iiopClient.jar</code> というアーカイブに出力されます。また、各 EJB モジュールに対するスタブは、<code>ORACLE_HOME/j2ee/instance/application-deployments/appName/ejbModuleName</code> ディレクトリに同じ名前のアーカイブで生成されます。</p> <p>この機能を使用するには、OC4J の起動時に GenerateIIOP システム・プロパティを有効にする必要があります。このプロパティは、スタンドアロン OC4J の OC4J コマンドラインでは <code>-DGenerateIIOP=true</code>、<code>opmn.xml</code> では <code>oc4j-options</code> 値として設定されます。</p>
-iiopClientJar	<p>任意。IIOP クライアント・スタブの出力先となる JAR のパスおよびファイル名。</p> <p>すべての EJB モジュールに対して生成されるアプリケーション・レベルのスタブは、<code>ORACLE_HOME/j2ee/instance/application-deployments/appName</code> ディレクトリの <code>_iiopClient.jar</code> というアーカイブに出力されます。パスを指定する場合、アーカイブはそのパスにも設定されます。</p> <p>また、各 EJB モジュールに対するスタブは、<code>ORACLE_HOME/j2ee/instance/application-deployments/appName/ejbModuleName</code> ディレクトリに同じ名前のアーカイブで生成されます。</p> <p>この機能を使用するには、OC4J の起動時に GenerateIIOP システム・プロパティを有効にする必要があります。このプロパティは、スタンドアロン OC4J の OC4J コマンドラインでは <code>-DGenerateIIOP=true</code>、<code>opmn.xml</code> では <code>oc4j-options</code> 値として設定されます。</p>
-removeArchive	<p>任意。デプロイ後にサーバーのファイル・システムから削除する EAR ファイルを指定します。</p>

リモート・クライアントからの J2EE アプリケーションのデプロイ

次の例では、リモート・クライアントから Oracle Application Server の特定の OC4J インスタンスに EAR をデプロイする方法を示します。

```
cd j2ee/home
>java -jar admin_client.jar
  deployer:oc4j:opmn://test-cycle.oracle.com/testunit
  oc4jadmin welcome1
  -deploy
  -file d:¥temp¥rupg¥testru.ear
  -deploymentName testru -bindAllWebApps

06/06/20 17:00:16 Notification ==>Uploading file testru.ear ...
06/06/20 17:00:18 Notification ==>Application Deployer for testru STARTS.
06/06/20 17:00:19 Notification ==>Copy the archive to /scratch/sbutton/ml_
060618/j2ee/admin/applications/testru.ear
06/06/20 17:00:19 Notification ==>Initialize /scratch/sbutton/ml_
060618/j2ee/admin/applications/testru.ear begins...
06/06/20 17:00:19 Notification ==>Unpacking testru.ear
06/06/20 17:00:20 Notification ==>Done unpacking testru.ear
06/06/20 17:00:20 Notification ==>Unpacking testru-web.war
06/06/20 17:00:20 Notification ==>Done unpacking testru-web.war
06/06/20 17:00:20 Notification ==>Initialize /scratch/sbutton/ml_
060618/j2ee/admin/applications/testru.ear ends...
06/06/20 17:00:20 Notification ==>Starting application : testru
06/06/20 17:00:20 Notification ==>Initializing ClassLoader(s)
06/06/20 17:00:20 Notification ==>Initializing EJB container
06/06/20 17:00:20 Notification ==>Loading connector(s)
06/06/20 17:00:20 Notification ==>Starting up resource adapters
06/06/20 17:00:20 Notification ==>Initializing EJB sessions
06/06/20 17:00:20 Notification ==>Committing ClassLoader(s)
06/06/20 17:00:20 Notification ==>Initialize testru-web begins...
06/06/20 17:00:20 Notification ==>Initialize testru-web ends...
06/06/20 17:00:21 Notification ==>Started application : testru
06/06/20 17:00:21 Notification ==>Binding web application(s) to site default-web-site
begins...
06/06/20 17:00:21 Notification ==>Binding testru-web web-module for application testru
to site default-web-site under context root /testru
06/06/20 17:00:22 Notification ==>Binding web application(s) to site default-web-site
ends...
06/06/20 17:00:22 Notification ==>Application Deployer for testru COMPLETES. Operation
time: 3785 msecs
```

スタンドアロン Web モジュール (WAR) のデプロイ

-deploy コマンドを使用して、WAR ファイルとしてパッケージされたスタンドアロン Web モジュールをデプロイまたは再デプロイします。

WAR 固有の構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file
path/filename -deploymentName appName [-bindAllWebApps [webSiteName]]
[-targetPath path] [-parent appName] [-deploymentDirectory path]
[-contextRoot context]
[-removeArchive]
```

WAR は、まだ Web モジュール・コンポーネントがない別のデプロイ済アプリケーションの子として指定できます。指定しない場合、WAR は default アプリケーションにデプロイされます。

WAR は、すでに Web モジュールが存在するアプリケーションの子としてはデプロイできません。つまり、acme アプリケーションにすでに acme-web.war がある場合、そのアプリケーションには追加の WAR ファイルをデプロイできません。かわりに、アプリケーションの EAR ファイルに WAR を再パッケージし、そのアプリケーションを再デプロイします。

次のコマンドは、node1 がメンバーであるクラスタ内の default_group に属するすべての OC4J インスタンスの default アプリケーションに、スタンドアロン acme-web.war Web モジュールをデプロイします。-bindAllWebApps パラメータが指定されていますが、バインド先の Web サイトが指定されていないため、モジュールはデフォルト Web サイトにバインドされます。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -deploy -file C:/dev/acme-web.war -deploymentName utility
-bindAllWebApps -parent default
```

表 6-6 WAR デプロイの -deploy コマンドのパラメータ

パラメータ	説明
-file	必須。デプロイするアーカイブのパスおよびファイル名。
-deploymentName	必須。Web モジュールのユーザー定義の名前。OC4J 内で Web モジュールを識別する際に使用されます。
-bindAllWebApps	任意。Web モジュールを指定された Web サイトにバインドします。何も指定されない場合は、デフォルト Web サイトにバインドします。 webSiteName の値を指定できます。この値は、Web サイト構成を含む name_web-site.xml ファイルの name 部分です。
-targetPath	任意。アーカイブのデプロイ先となるディレクトリ。ディレクトリが指定されていない場合、アーカイブは ORACLE_HOME/j2ee/instance/applications ディレクトリにデフォルトでデプロイされます。 また、スタンドアロン WAR ファイルが含まれる生成済 EAR ファイルもこのディレクトリにコピーされます。後続のデプロイのたびに、このアーカイブは上書きされます。
-parent	任意。モジュールのデプロイ先となる親アプリケーション。デフォルトは default アプリケーションです。
-deploymentDirectory	任意。OC4J 固有のデプロイメント・ディスクリプタおよび生成済ファイル（コンパイル済 JSP クラスなど）が格納されるディレクトリ。 デフォルトのディレクトリは、ORACLE_HOME/j2ee/instance/application-deployments です。

表 6-6 WAR デプロイの -deploy コマンドのパラメータ (続き)

パラメータ	説明
-contextRoot	任意。Web モジュール・コンテキスト・ルート。Web ブラウザを介してアプリケーションにアクセスするために使用される URL の最後に追加されます。コンテキスト・ルートを指定しない場合は、-deploymentName に渡される値が使用されます。 たとえば、コンテキスト・ルートとして /petstore を指定すると、モジュールには次の URL からアクセスできます。 http://node1.company.com:7777/petstore
-removeArchive	任意。デプロイ後にサーバーのファイル・システムから削除する WAR ファイルを指定します。

スタンドアロン・リソース・アダプタ (RAR) のデプロイ

-deploy コマンドを使用して、RAR ファイルとしてパッケージされた Java コネクタ・アーキテクチャに準拠するリソース・アダプタをデプロイまたは再デプロイします。デフォルトでは、リソース・アダプタは `ORACLE_HOME/j2ee/instance/connectors` ディレクトリにデプロイされます。

スタンドアロン RAR の再デプロイまたはアンデプロイでは、default アプリケーションを再起動する必要はありません。

RAR 固有の構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -deploy -file path/filename
-deploymentName connectorName [-nativePathLib path] [-grantAllPermissions]
[-removeArchive]
```

次のコマンドは、クラスタ内の default_group に属するすべての OC4J インスタンスに acme-rar.rar モジュールをデプロイします。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -deploy -file /dev/acme-rar.rar -deploymentName acme-rar
-grantAllPermissions -removeArchive
```

表 6-7 RAR デプロイの -deploy コマンドのパラメータ

パラメータ	説明
-file	必須。デプロイする RAR ファイルのパスおよびファイル名。
-deploymentName	必須。ユーザー定義のコネクタ名。OC4J 内でコネクタを識別する際に使用されます。
-nativeLibPath	任意。RAR ファイル内の、ネイティブ・ライブラリ (DLL など) を格納するディレクトリへのパス。
-grantAllPermissions	任意。必要に応じて、リソース・アダプタによってリクエストされるすべてのランタイム権限を付与する場合にこのパラメータを指定します。
-removeArchive	任意。デプロイ後にサーバーのファイル・システムから RAR ファイルを削除する場合にこのパラメータを指定します。

詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』の「リソース・アダプタのデプロイ」を参照してください。

バッチ・デプロイでのスクリプト・ファイルの使用

`admin_client.jar` コマンドラインのデプロイ・コマンドを含むスクリプト・ファイルを指定できます。`-script` コマンドでファイルを指定すると、`admin_client.jar` によりデプロイメント・マネージャへの 1 つの接続のみでコマンドのリストを実行できます。バッチ・デプロイの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword
-script filename
```

スクリプト・ファイル `filename` には、この例のように複数の行が含まれます。

```
-deploy -file /scratch/rpan/apps/hello-planet.ear -deploymentName hello-planet
-bindWebApp -appName hello-planet -webModuleName hello-planet-web
-stop hello-planet
-start hello-planet
-redeploy -file /scratch/rpan/apps/hello-planet.ear
-deploymentName hello-planet -bindAllWebApps
-undeploy hello-planet
-validateURI
```

インストールのスクリプトまたはログを参照し、既存のコンフィギュレーション・アシスタントによって使用される関連行を抽出することで、バッチ・モードに変換できます。

デプロイ後の Web サイトへの Web モジュールのバインド

OC4J にデプロイされた各 Web モジュールは、そのアクセスに使用される Web サイトにバインドする必要があります。

通常、EAR 内の WAR ファイルとしてパッケージされた Web モジュールは、`-deploy` コマンドで `-bindAllWebApps` パラメータを使用して、EAR のデプロイ時にバインドします。しかし、EAR のデプロイ時に `-bindAllWebApps` パラメータを指定しなかった場合は、次の各項で説明するように、デプロイ後にモジュールを Web サイトにバインドできます。

- 単一の Web サイトへの全 Web モジュールのバインド
- 特定 Web モジュールの Web サイトへのバインドおよびコンテキスト・ルートの設定

単一の Web サイトへの全 Web モジュールのバインド

`-bindAllWebApps` コマンドを使用して、J2EE アプリケーション内のすべての Web モジュールを同じ Web サイトにバインドするか、デフォルトで `default-web-site` にバインドします。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -bindAllWebApps
-appName appName -webSiteName siteName
```

表 6-8 -bindAllWebApps コマンドのパラメータ

パラメータ	説明
<code>-appName</code>	必須。デプロイ時に指定された親アプリケーションの名前。
<code>-webSiteName</code>	任意。Web サイト構成を含む <code>name_web-site.xml</code> ファイルの <code>name</code> 部分。このパラメータを省略する場合、すべての Web モジュールはデフォルト Web サイトにバインドされます。

特定 Web モジュールの Web サイトへのバインドおよびコンテキスト・ルートの設定

-bindWebApp コマンドを使用して、J2EE アプリケーション内の特定 Web モジュールを、指定する Web サイトまたはデフォルトの Web サイトにバインドします。また、Web モジュールへのアクセスに使用されるコンテキスト・ルートを設定することもできます。

このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -bindWebApp
-appName appName -webModuleName moduleName -webSiteName siteName
-contextRoot contextRoot
```

表 6-9 -bindWebApp コマンドのパラメータ

パラメータ	説明
-appName	必須。デプロイ時に指定された親アプリケーションの名前。
-webModuleName	必須。バインドする Web モジュールの名前。この値は、EAR ファイル内に含まれる WAR ファイルの名前（.war 拡張子を除く）である必要があります。
-webSiteName	任意。Web サイト構成を含む name_web-site.xml ファイルの name 部分。このパラメータを省略する場合、Web モジュールはデフォルト Web サイトにバインドされます。
-contextRoot	任意。Web モジュールのコンテキスト・ルート。この値は、Web ブラウザを介してアプリケーションにアクセスするために使用される URL の最後に追加されます。次に例を示します。 http://localhost:8888/petstore コンテキスト・ルートを指定しない場合は、親アプリケーションの application.xml デプロイメント・ディスクリプタに指定されるコンテキスト・ルートが使用されます。

アーカイブの再デプロイ

-redeploy コマンドを使用して、すでにデプロイされたアーカイブを再デプロイできます。

この操作は、アプリケーションが稼働中の場合はそのアプリケーションを停止した後にアーカイブをアンデプロイして、正常な再デプロイを実行します。その後、アプリケーションのデプロイおよび再起動を実行します。一方、-deploy コマンドによるアーカイブの再デプロイは、アプリケーションを停止せず、単にアンデプロイおよび再デプロイした後に再起動します。

このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -redeploy -file path/filename
-deploymentName appName [-bindAllWebApps] [-isConnector] [-keepSettings]
[-sequential [waitsec]] [-removeArchive]
```

表 6-10 -redeploy コマンドのパラメータ

パラメータ	説明
-file	必須。再デプロイする EAR、WAR または RAR ファイルのパスおよびファイル名。
-deploymentName	必須。ユーザー定義のアプリケーションのデプロイ名。OC4J 内でアプリケーションを識別する際に使用されます。この値は、サーバーの既存アプリケーションの名前と一致する必要があります。
-isConnector	スタンドアロン RAR の再デプロイに必要。

表 6-10 -redeploy コマンドのパラメータ (続き)

パラメータ	説明
-bindAllWebApps	<p>任意。EAR 内のすべての Web モジュールを指定された Web サイトにバインドします。何も指定されない場合は、デフォルト Web サイトにバインドします。</p> <p>webSiteName の値を指定できます。この値は、Web サイト構成を含む name_web-site.xml ファイルの name 部分です。</p> <p>また、6-14 ページの「デプロイ後の Web サイトへの Web モジュールのバインド」に説明されているように、後からすべての Web モジュールを Web サイトにバインドできます。</p>
-keepSettings	<p>任意。このパラメータを指定する場合、再デプロイされるアプリケーションは、前回のデプロイからデプロイ・プランを取得して使用します。アーカイブにパッケージされたデプロイメント・ディスクリプタに設定される値は、無視されます。</p> <p>このパラメータを指定しない場合、アーカイブにパッケージされたデプロイメント・ディスクリプタの値に設定されます。</p>
-sequential [waitsec]	<p>任意。グループ内の各 OC4J インスタンスに順番にアーカイブを再デプロイする場合に指定します。各ターゲット OC4J インスタンスへの再デプロイは、次のターゲット・インスタンスで再デプロイが始まる前に完了する必要があります。リクエストは、アーカイブを再デプロイ中の OC4J インスタンスにはルーティングされません。</p> <p>次のように、waitsec オプションを使用して、デプロイ間に待機する秒数を指定できます。</p> <pre>-sequential 15</pre> <p>このパラメータが指定されていない場合、アーカイブはデフォルトで、グループ内のすべての OC4J インスタンスに同時にデプロイされます。</p> <p>このオプションは、クラスタ化された環境でのみ有効です。スタンドアロン OC4J では無効です。</p>
-removeArchive	<p>任意。デプロイ後にサーバーのファイル・システムから削除する EAR、WAR または RAR ファイルを指定します。</p>

クラスタ内の順次再デプロイ間の遅延の指定

admin_client.jar -redeploy コマンドの -sequential パラメータを指定してアプリケーションをグループに再デプロイすると、再デプロイ操作がシリアル化されます。再デプロイは一度に 1 つの OC4J インスタンスに対して行われるため、ターゲット・アプリケーションが完全に停止状態になることはありません。順次再デプロイでは、現在の OC4J インスタンスでの再デプロイ操作が完了するとすぐに、デプロイメント・マネージャにより、アプリケーション・クラスタのメンバーを実行している次の OC4J インスタンスでの再デプロイが開始されます。その結果、システムが安定した状態にならず、次の再デプロイが始まる前に新しいアプリケーション・インスタンスが完全にアクティブになり、次のような問題の原因となります。

- アプリケーションが 1 つの OC4J インスタンスで停止し、別のインスタンスで使用可能であることが mod_oc4j に通知されるまでの間、アプリケーションにアクセスできなくなる場合があります。
- セッション・レプリケーション・アクティビティを実行する機会がない場合があります。

admin_client.jar -redeploy コマンドを使用してクラスタにアプリケーションを再デプロイすると、-sequential および -keepsettings パラメータを指定しても、アプリケーションのセッション状態が失われる場合があります。

OC4J 10g (10.1.3.4.0) では、-sequential パラメータの waitsec オプションを使用して、アプリケーション・クラスタを実行している別の OC4J インスタンスへの再デプロイ間の秒数を指定できます。この遅延により、セッション状態のレプリケーションに十分な時間を設けられます。

オプションの `waitsec` 値を指定すると、デプロイメント・マネージャは、グループ内の OC4J インスタンスにおける再デプロイ操作の間に、指定された秒数待機します。グループ全体に再デプロイ操作が発生した場合、この遅延によりシステムが安定し、アプリケーションにアクセスできなくなることや、セッション状態が失われることが少なくなります。

たとえば、次の `admin_client.jar -redeploy` コマンドでは、別の OC4J インスタンスへの再デプロイ間に 15 秒の遅延が指定されています。

```
java -jar admin_client.jar deployer:cluster:opmn://host:port/home oc4jadmin
password -redeploy -file "myapp.ear" -deploymentName rolling -sequential 15
-keepsettings
```

新しい `waitsec` オプションも、`admin_client.jar -deploy` コマンドの `-sequential` パラメータに適用できます。

スケジュール済ジョブを含むアプリケーションの再デプロイ

スケジュール済ジョブを含むアプリケーションを再デプロイする場合、再デプロイ前にすべてのジョブを削除し、再デプロイ後に再発行しないと、ジョブはスケジュールどおりに実行されません。

スケジュール済ジョブを含むアプリケーションを再デプロイするには、次のようにします。

1. スケジュール済ジョブをすべて削除します。
2. アプリケーションを再デプロイします。
3. すべてのジョブを再発行します。

アーカイブのアンデプロイ

`-undeploy` コマンドは、次の各項で説明するように、アプリケーションまたはスタンドアロン Web モジュール、スタンドアロン・コネクタ・モジュールをターゲット OC4J インスタンスから削除します。

- [EAR またはスタンドアロン WAR のアンデプロイ](#)
- [スタンドアロン RAR のアンデプロイ](#)

EAR またはスタンドアロン WAR のアンデプロイ

EAR またはスタンドアロン Web モジュールをアンデプロイすると、OC4J ランタイムから削除されます。既存の Web サイトのバインディングも同様に削除されます。

EAR またはスタンドアロン WAR のアンデプロイの構文は次のとおりです。アプリケーションまたはモジュールの名前を指定する必要があります。

```
java -jar admin_client.jar uri adminId adminPassword -undeploy appName
```

スタンドアロン RAR のアンデプロイ

スタンドアロン RAR のアンデプロイの構文は次のとおりです。 `-isConnector` パラメータをコネクタ名とともに指定する必要があります。

```
java -jar admin_client.jar uri adminId adminPassword -undeploy connectorName
-isConnector
```

スタンドアロン RAR のアンデプロイでは、`default` アプリケーションを再起動する必要はありません。

デプロイ済 EJB モジュールの変更されたクラスの更新

-updateEJBModule コマンドは、OC4J インスタンスまたは OC4J インスタンスのグループで稼働中のアプリケーションにおいて、EJB モジュールの増分的または部分的な再デプロイを実行します。この機能により、EJB JAR 内の変更された Bean のみを再デプロイできます。

注意： 増分的な再デプロイは、CMP または BMP エンティティ Bean にアプリケーション全体を再デプロイするよりも効率的ですが、セッション Bean、メッセージドリブン Bean または EJB 3.0 JPA エンティティでは当てはまりません。この機能の使用に関する詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』の「更新された EJB モジュールの増分再デプロイ」を参照してください。

デプロイ済の EJB モジュールで変更されたクラスを更新する構文は、次のとおりです。EJB JAR が構成するアプリケーションの名前を指定する必要があります。スタンドアロン EJB モジュールを更新する場合は、default アプリケーションを指定します。

```
java -jar admin_client.jar uri adminId adminPassword -updateEJBModule
-appName appName -ejbModuleName ejbJarName -file path/ejbJarName
```

次に例を示します。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -updateEJBModule -appName petstore
-ejbModuleName customerEjb.jar -file build/customerEjb.jar
```

表 6-11 -updateEJBModule の構文

オプション	説明
-appName	必須。EJB モジュールが構成するアプリケーションの名前。スタンドアロン EJB モジュールを更新する場合は、default アプリケーションを指定します。
-ejbModuleName	必須。application.xml に定義されている、更新対象の EJB JAR ファイルの名前。
-file	必須。更新された EJB JAR のパスおよびファイル名。

共有ライブラリの作成および管理

admin_client.jar ユーティリティを使用すると、次の項で説明するように、OC4J インスタンスまたは OC4J インスタンスのグループで共有ライブラリの作成および管理を実行できます。

- [共有ライブラリのインストール](#)
- [既存の共有ライブラリの変更](#)
- [共有ライブラリの内容の表示](#)
- [すべての共有ライブラリのリスト出力](#)
- [共有ライブラリの削除](#)

共有ライブラリのインストール

`-publishSharedLibrary` コマンドを使用すると、特定の OC4J インスタンスまたは OC4J インスタンスのグループに共有ライブラリのディレクトリ構造を作成し、ライブラリを構成するバイナリをその中にインストールできます。共有ライブラリは、各 OC4J インスタンスの `ORACLE_HOME/j2ee/instance/shared-lib` ディレクトリに作成されます。

また、このコマンドは、各 OC4J インスタンスの `server.xml` ファイルの `<shared-library>` 要素内に共有ライブラリを宣言し、アプリケーションで使用できるようにします。

共有ライブラリのインストールの構文は次のとおりです。複数のコード・ソース（共有ライブラリを構成するバイナリ）のパスおよびファイル名は、それぞれを空白で区切って指定できます。

```
java -jar admin_client.jar uri adminId adminPassword -publishSharedLibrary
-name libName -version libVersion [-parentName parentLibName]
[-parentVersion parentLibVersion] [-installCodeSources path [path ...]]
[-addCodeSources path [path ...]] [-imports sharedLibName
[:min-version] [,max-version] [sharedLibName ...]]
```

次のコマンドは、クラスタ内の OC4J インスタンス (`default_group` のすべてのメンバー) のグループに `acme.common:2.5` 共有ライブラリをデプロイします。

```
java -jar admin_client.jar deployer:cluster:opmn://server.company.com:6004/
default_group
oc4jadmin password -publishSharedLibrary -name acme.common -version 2.5
-installCodeSources /myserver/tmp/acme-apis.jar /myserver/tmp/acmeImpl.jar
```

ターゲット OC4J サーバー内の結果のディレクトリ構造は、次のとおりです。

```
ORACLE_HOME/j2ee/home/shared-lib
  /acme.common
    /2.5
      acme-apis.jar
      acmeImpl.jar
```

表 6-12 -publishSharedLibrary コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。共有ライブラリの名前。 共通 API が複数のベンダーによって実装される場合は、 <code>oracle.jdbc</code> や <code>xerces.xml</code> などのように、名前にベンダー名とテクノロジー名の両方を含めることをお勧めします。
<code>-version</code>	必須。共有ライブラリのバージョン番号。この値はコードの実装バージョンを反映することをお勧めします。
<code>-parentName</code>	任意。該当する場合は、親共有ライブラリの名前。
<code>-parentVersion</code>	任意。該当する場合は、親共有ライブラリのバージョン番号。
<code>-installCodeSources</code>	OC4J インスタンスにアップロードし、共有ライブラリの一部としてインストールする 1 つ以上の JAR または ZIP ファイルのパスおよびファイル名。パスまたはファイル名の各文字列は、空白で区切ります。
<code>-addCodeSources</code>	任意。共有ライブラリに追加するためにすでに 1 つ以上の OC4J インスタンスにアップロードされている JAR または ZIP ファイルのパスおよびファイル名。パスまたはファイル名の各文字列は、空白で区切ります。
<code>-imports</code>	任意。この共有ライブラリにインポートする 1 つ以上の既存の共有ライブラリの名前。各名前前の文字列は空白で区切ります。 インポートするライブラリの最高または最低、あるいはその両方のバージョンを指定できます。

既存の共有ライブラリの変更

`-modifySharedLibrary` コマンドを使用すると、既存の共有ライブラリの内容を変更できます。また、このコマンドは、各 OC4J インスタンスの `server.xml` ファイル内にある共有ライブラリ定義を更新します。

既存の共有ライブラリを変更する構文は次のとおりです。複数のコード・ソース（共有ライブラリを構成するバイナリ）のパスおよびファイル名は、それぞれを空白で区切って指定できます。

```
java -jar admin_client.jar uri adminId adminPassword -modifySharedLibrary
-name libName -version libVersion [-installCodeSources path [path ...]]
[-addCodeSources path [path ...]] [-removeCodeSources path [path ...]]
[-addImports sharedLibName[:min-version] [,max-version] [sharedLibName ...]]
[-removeImports sharedLibName[:min-version] [,max-version] [sharedLibName ...]]
```

次のコマンドは、`acme.common:2.5` 共有ライブラリを更新します。

```
java -jar admin_client.jar
deployer:cluster:opmn://server.company.com:6004/default_group
oc4jadmin password -modifySharedLibrary -name acme.common -version 2.5
-addCodeSources /myserver/tmp/acme-helpers.jar
```

表 6-13 -modifySharedLibrary コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。更新する共有ライブラリの名前。
<code>-version</code>	必須。更新する共有ライブラリのバージョン番号。
<code>-installCodeSources</code>	任意。1つ以上の OC4J インスタンスにアップロードし、共有ライブラリの一部としてインストールする JAR または ZIP ファイルのパスおよびファイル名。パスまたはファイル名の各文字列は、空白で区切ります。
<code>-addCodeSources</code>	任意。共有ライブラリに追加するためにすでに OC4J インスタンスにアップロードされている 1つ以上の JAR または ZIP ファイルのパスおよびファイル名。パスまたはファイル名の各文字列は、空白で区切ります。
<code>-removeCodeSources</code>	任意。共有ライブラリから削除する 1つ以上の JAR または ZIP ファイルのパスおよびファイル名。パスまたはファイル名の各文字列は、空白で区切ります。
<code>-addImports</code>	任意。この共有ライブラリにインポートする 1つ以上の既存の共有ライブラリの名前。各名前の文字列は空白で区切ります。 インポートするライブラリの最高または最低、あるいはその両方のバージョンを指定できます。
<code>-removeImports</code>	任意。この共有ライブラリから削除する 1つ以上の既存の共有ライブラリの名前。 削除するライブラリの最高または最低、あるいはその両方のバージョンを指定できます。

共有ライブラリの内容の表示

-describeSharedLibrary コマンドを使用して、指定された共有ライブラリを構成するコード・ソースとインポート済共有ライブラリを表示します。構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -describeSharedLibrary
-name libName -version libVersion
```

表 6-14 -describeSharedLibrary コマンドのパラメータ

パラメータ	説明
-name	必須。共有ライブラリの名前。
-version	必須。共有ライブラリのバージョン番号。

すべての共有ライブラリのリスト出力

-listSharedLibraries コマンドを使用して、1 つ以上のターゲット OC4J インスタンスに定義されているすべての共有ライブラリのリストを出力します。構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -listSharedLibraries
```

注意： JDK 1.4 を使用している場合、Oracle Application Server 10g リリース 3 (10.1.3.4.0) では JDK に同梱されている Xalan ライブラリを共有ライブラリとして使用することはできません。Xalan ライブラリを使用するには、次の 2 つの方法があります。

- 同梱されている Xalan ライブラリが共有ライブラリとしてサポートされている JDK 5.0 (JDK 1.5) または JDK 6 を使用します。
- JDK1.4 の場合は、同梱されているバージョンではなく、Xalan ライブラリのスタンドアロン・ディストリビューションを使用します。

共有ライブラリの削除

-removeSharedLibrary コマンドを使用して、OC4J ターゲットから共有ライブラリを削除します。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -removeSharedLibrary
-name libName -version libVersion
```

表 6-15 -removeSharedLibrary コマンドのパラメータ

パラメータ	説明
-name	必須。削除する共有ライブラリの名前。
-version	必須。削除する共有ライブラリのバージョン番号。

アプリケーションの起動、再起動および停止

`admin_client.jar` ユーティリティを使用すると、特定の OC4J インスタンスで、または OC4J インスタンスのグループで、アプリケーションおよびその子アプリケーションを起動、再起動または停止できます。アプリケーション内のファイルが変更されている場合、アプリケーションは起動時に自動的に再デプロイされます。

これらのコマンドを使用して Application Server Control (`ascontrol`) を停止および起動することもできます。

このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -start|-stop appName
```

次の例では、クラスタ内の `node2` で `ascontrol` アプリケーションを停止しています。

```
java -jar admin_client.jar deployer:oc4j:opmn://node2.company.com:6004/home oc4jadmin password -stop ascontrol
```

OC4J インスタンスの再起動および停止

`admin_client.jar` ユーティリティを使用すると、スタンドアロン OC4J サーバー、管理環境下にある特定の OC4J インスタンス、または OC4J インスタンスのグループを停止できます。`-shutdown` コマンドは、指定された 1 つ以上の OC4J インスタンスを停止します。OPMN 管理のインスタンスの場合には、停止されることが OPMN に通知されます。`-restart` コマンドは、指定された 1 つ以上のインスタンスを再起動します。

次の各項では、これらのコマンドの構文および例を示します。

- OC4J インスタンスまたはインスタンスのグループの再起動
- 1 つ以上の OC4J インスタンスの停止

OC4J インスタンスまたはインスタンスのグループの再起動

次に示すように、`admin_client.jar` の `-restart` コマンドラインを使用すると、OC4J インスタンスまたは OC4J インスタンスのグループを再起動できます。

```
java -jar admin_client.jar uri adminId adminPassword -restart
```

たとえば、次のコマンドは、スタンドアロン OC4J サーバーを再起動します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin password -restart
```

次のコマンドは、クラスタ・トポロジ内の各 Oracle Application Server の `default_group` のメンバーであるすべての OC4J インスタンスを再起動します。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group oc4jadmin password -restart
```

1 つ以上の OC4J インスタンスの停止

次に示すように、`admin_client.jar` `-shutdown` コマンドラインを使用すると、OC4J インスタンスまたは OC4J インスタンスのグループを停止できます。

```
java -jar admin_client.jar uri adminId adminPassword -shutdown
```

たとえば、次のコマンドは、スタンドアロン OC4J サーバーを停止します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin password -shutdown
```

このコマンドは、ホスト・マシンが接続されていないかのように、すべてのスレッドを即座に終了させて OC4J サーバー全体を停止します。このコマンドを使用すると、クラスタ化されたアプリケーションの現在の状態はレプリケートされません。

次のコマンドは、OPMN 管理の Oracle Application Server 環境の指定された OC4J インスタンスを停止します。

```
java -jar admin_client.jar deployer:oc4j:opmn://localhost/home oc4jadmin password
-shutdown
```

次のコマンドは、クラスタ・トポロジ内の各 Oracle Application Server の default_group のメンバーであるすべての OC4J インスタンスを停止します。

```
java -jar admin_client.jar deployer:cluster:opmn://node1.company.com/default_group
oc4jadmin password -shutdown
```

これらのコマンドは指定された 1 つ以上のインスタンスを停止し、すべてのスレッドを即時に終了します。-shutdown コマンドを使用すると、クラスタ化されたアプリケーションの現在の状態はレプリケートされません。OPMN 管理の各 OC4J インスタンスでは、OPMN が再起動を試行しないように、サーバーを故意に停止中であることが admin_client.jar により OPMN に通知されます。

データソースの管理

admin_client.jar ユーティリティを使用すると、次の各項で説明するように、OC4J インスタンスまたは OC4J インスタンスのグループでデータソースを管理できます。

- データソース接続プールの追加、テストおよび削除
- データソースの追加、テストおよび削除

データソース接続プールの追加、テストおよび削除

admin_client.jar ユーティリティを使用すると、次の各項で説明するように、OC4J インスタンスまたは OC4J インスタンスのグループでデータソース接続プールを追加、テストおよび削除できます。

- データソース接続プールの追加
- データソース接続プールのテスト
- データソース接続プールの削除

データソース接続プールの追加

-addDataSourceConnectionPool コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションのデータソース接続プールを追加できます。

データソース接続プールを追加する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -addDataSourceConnectionPool
-applicationName applicationName -name name -factoryClass factoryClass
-dbUser dbUser -dbPassword dbPassword -url url
[-factoryProperties name1 value1 [name2 value2 [...]]]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-addDataSourceConnectionPool -applicationName default -name ScottConnectionPool
-factoryClass oracle.jdbc.pool.OracleDataSource
-dbUser scott -dbPassword tiger -url jdbc:oracle:thin:@localhost:1521:xe
```

表 6-16 -addDataSourceConnectionPool コマンドのパラメータ

パラメータ	説明
-applicationName	必須。データソース接続プールを追加するアプリケーションの名前。
-name	必須。接続プールの名前。
-factoryClass	必須。コネクション・ファクトリ実装の完全修飾パス。
-dbUser	必須。接続を取得するために使用するデフォルト・ユーザーの名前。
-dbPassword	必須。接続を取得するために使用するデフォルトのパスワード。
-url	必須。接続を取得するために使用するコネクション・ファクトリの URL。
-factoryProperties	任意。コネクション・ファクトリの定義に設定する 1 つ以上のプロパティ名および値のペア。

データソース接続プールのテスト

-testDataSourceConnectionPool コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでデータソース接続プールへのアプリケーションの接続をテストできます。

データソース接続プールへの接続をテストする構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -testDataSourceConnectionPool
-name name -sqlStatement sqlStatement [-applicationName applicationName]
[-dbUser dbUser] [-dbPassword dbPassword]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-testDataSourceConnectionPool -sqlStatement "select * from dual"
-applicationName default -name ScottConnectionPool
```

表 6-17 -testDataSourceConnectionPool コマンドのパラメータ

パラメータ	説明
-name	必須。接続プールの名前。
-sqlStatement	必須。接続をテストするために使用する SQL 文。
-applicationName	任意。データソース接続プールをテストするアプリケーションの名前。
-dbUser	任意。接続を取得するために使用するデフォルト・ユーザーの名前。
-dbPassword	任意。接続を取得するために使用するデフォルトのパスワード。

データソース接続プールの削除

-removeDataSourceConnectionPool コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションからデータソース接続プールを削除できます。データソース接続プールを削除する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword
-removeDataSourceConnectionPool -name name [-applicationName applicationName]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-removeDataSourceConnectionPool -name ScottConnectionPool -applicationName default
```

表 6-18 -removeDataSourceConnectionPool コマンドのパラメータ

パラメータ	説明
-name	必須。接続プールの名前。
-applicationName	任意。データソース接続プールを削除するアプリケーションの名前。

データソースの追加、テストおよび削除

admin_client.jar ユーティリティを使用すると、次の各項で説明するように、OC4J インスタンスまたは OC4J インスタンスのグループでデータソースを追加、テストおよび削除できます。

- マネージド・データソースの追加
- マネージド・データソースの削除
- ネイティブ・データソースの追加
- ネイティブ・データソースの削除
- データベース接続のテスト
- データソースのテスト
- アプリケーションのデータソース・ディスクリプタの取得

マネージド・データソースの追加

-addManagedDataSource コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションのマネージド・データソースを追加できます。マネージド・データソースを追加する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -addManagedDataSource
-applicationName applicationName -name name
-jndiLocation jndiLocation -connectionPoolName connectionPoolName
[-dbUser dbUser] [-dbPassword dbPassword] [-loginTimeout loginTimeout]
[-txLevel txLevel] [-dbSchema dbSchema] [-manageLocalTransactions true|false]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-addManagedDataSource -applicationName default -name ScottDataSource
-jndiLocation jdbc/ScottDataSource -connectionPoolName ScottConnectionPool
```

表 6-19 -addManagedDataSource コマンドのパラメータ

パラメータ	説明
-applicationName	必須。データソースを追加するアプリケーションの名前。
-name	必須。データソースの名前。
-jndiLocation	必須。新規データソースを JNDI にバインドするために使用する場所。
-connectionPoolName	必須。データソースが通信するコネクション・プールの名前。
-dbUser	任意。新規データソースのデフォルト・ユーザー。
-dbPassword	任意。新規データソースのデフォルト・パスワード。
-loginTimeout	任意。新規データソースのログイン・タイムアウト。
-txLevel	任意。トランザクション・レベル (local または global)。
-dbSchema	任意。使用中の EJB CMP 実装が Orion CMP の場合に使用するデータベース・スキーマ。(TopLink CMP はデフォルトです。)
-manageLocalTransactions	任意。OC4J でローカル・トランザクションを管理するかどうかを指定します。デフォルト値は true です。

マネージド・データソースの削除

`-removeManagedDataSource` コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションからマネージド・データソースを削除できます。マネージド・データソースを削除する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -removeManagedDataSource
-name name [-applicationName applicationName]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-removeManagedDataSource -name ScottDataSource -applicationName default
```

表 6-20 -removeManagedDataSource コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。削除するデータソースの名前。
<code>-applicationName</code>	任意。データソースを削除するアプリケーションの名前。

ネイティブ・データソースの追加

`-addNativeDataSource` コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションのネイティブ・データソースを追加できます。ネイティブ・データソースを追加する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -addNativeDataSource
-name name -dbUser dbUser -dbPassword dbPassword
-jndiLocation jndiLocation -loginTimeout loginTimeout
-dataSourceClass dataSourceClass -url url [-applicationName applicationName]
[-properties name1 value1 [name2 value2 ] [...]]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-addNativeDataSource -name ScottDataSource
-dbUser scott -dbPassword tiger -jndiLocation jdbc/ScottNativeDataSource
-loginTimeout 5 -dataSourceClass com.acme.DataSourceImpl
-url jdbc:oracle:thin:@localhost:1521:xe
```

表 6-21 -addNativeDataSource コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。新規データソースの名前。
<code>-dbUser</code>	必須。新規データソースのデフォルト・ユーザー。
<code>-dbPassword</code>	必須。新規データソースのデフォルト・パスワード。
<code>-jndiLocation</code>	必須。新規データソースを JNDI にバインドするために使用する場所。
<code>-loginTimeout</code>	必須。新規データソースのログイン・タイムアウト。
<code>-dataSourceClass</code>	必須。新規データソースの完全修飾クラス。
<code>-url</code>	必須。データベースに接続するために新規データソースによって使用される URL。
<code>-applicationName</code>	任意。データソースを追加するアプリケーションの名前。
<code>-properties</code>	任意。新規データソースの 1 つ以上のプロパティ。

ネイティブ・データソースの削除

`-removeNativeDataSource` コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションからネイティブ・データソースを削除できます。ネイティブ・データソースを削除する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -removeNativeDataSource
-name name [-applicationName applicationName]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-removeNativeDataSource -name ScottDataSource
```

表 6-22 -removeNativeDataSource コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。削除するデータソースの名前。
<code>-applicationName</code>	任意。データソースを削除するアプリケーションの名前。

データベース接続のテスト

`-testDatabaseConnection` コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションのデータベースへの接続をテストできます。

データベース接続をテストする構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -testDatabaseConnection
-sqlStatement sqlStatement -factoryClass factoryClass -dbUser dbUser
-dbPassword dbPassword -url url [-applicationName applicationName]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-testDatabaseConnection -sqlStatement "select * from dual"
-factoryClass oracle.jdbc.pool.OracleDataSource -dbUser scott
-dbPassword tiger -url jdbc:oracle:thin:@localhost:1521:xe
-applicationName default
```

表 6-23 -testDatabaseConnection コマンドのパラメータ

パラメータ	説明
<code>-sqlStatement</code>	必須。接続をテストするために使用する SQL 文。
<code>-factoryClass</code>	必須。テストする JDBC ファクトリ (Driver、DataSource、ConnectionPoolDataSource または XDataSource のインスタンス)。
<code>-dbUser</code>	必須。接続をテストするために使用するユーザー名。
<code>-dbPassword</code>	必須。接続をテストするために使用するパスワード。
<code>-url</code>	必須。JDBC ファクトリに設定する URL。
<code>-applicationName</code>	任意。アプリケーションの名前。

データソースのテスト

`-testDataSource` コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各 OC4J インスタンスでアプリケーションのデータソースへの接続をテストできます。

データソースをテストする構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -testDataSource
-name name -sqlStatement sqlStatement [-applicationName applicationName]
[-dbUser dbUser] [-dbPassword dbPassword]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-testDataSource -name ScottDataSource -sqlStatement "select * from dual"
-applicationName default -dbUser scott -dbPassword tiger
```

表 6-24 -testDataSource コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。テストするデータソース。
<code>-sqlStatement</code>	必須。接続をテストするために使用する SQL 文。
<code>-applicationName</code>	任意。アプリケーションの名前。
<code>-dbUser</code>	任意。接続をテストするために使用するユーザー。
<code>-dbPassword</code>	任意。接続をテストするために使用するパスワード。

アプリケーションのデータソース・ディスクリプタの取得

`-getDataSourcesDescriptor` コマンドを使用すると、アプリケーションのデータソース・ディスクリプタを取得できます。データソース・ディスクリプタを取得する構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -getDataSourcesDescriptor
[-applicationName applicationName]
```

表 6-25 -getDataSourcesDescriptor コマンドのパラメータ

パラメータ	説明
<code>-applicationName</code>	任意。データソース・ディスクリプタが属するアプリケーションの名前。

JMS リソースの管理

`admin_client.jar` ユーティリティを使用すると、次の各項で説明するように、OC4J インスタンスまたは OC4J インスタンスのグループで JMS リソースを管理できます。

- [JMS コネクション・ファクトリの管理](#)
- [JMS 宛先の管理](#)

JMS コネクション・ファクトリの管理

`admin_client.jar` ユーティリティを使用すると、次の各項で説明するように、OC4J JMS コネクション・ファクトリを管理できます。

- [JMS コネクション・ファクトリの追加](#)
- [JMS コネクション・ファクトリの削除](#)
- [JMS コネクション・ファクトリの情報の取得](#)

JMS コネクション・ファクトリの追加

-addJMSConnectionFactory コマンドを使用すると、OC4J インスタンスまたはクラスタ内のグループの各インスタンスに JMS コネクション・ファクトリを追加できます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -addJMSConnectionFactory
-domain domain -jndiLocation jndiLocation [-host host] [-port port]
[-username username] [-password password] [-clientID clientID] [-isXA true|false]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-addJMSConnectionFactory -domain Queue -jndiLocation jms/ExampleQueueCF
```

表 6-26 -addJMSConnectionFactory コマンドのパラメータ

パラメータ	説明
-domain	必須。コネクション・ファクトリの JMS ドメイン ('QUEUE'、'TOPIC' または 'UNIFIED')。
-jndiLocation	必須。コネクション・ファクトリをバインドする JNDI ロケーション。
-host	任意。コネクション・ファクトリに関連付けるホスト名 (デフォルトで、包含する OC4J JMS サーバー・ホストに設定されます)。
-port	任意。コネクション・ファクトリに関連付けるポート番号 (デフォルトで、包含する OC4J JMS サーバー・ポートに設定されます)。
-username	任意。コネクション・ファクトリに関連付けるユーザー名 (デフォルトで anonymous に設定されます)。
-password	任意。コネクション・ファクトリに関連付けるパスワード (デフォルトで Null に設定されます)。
-clientID	任意。コネクション・ファクトリに関連付ける JMS クライアント ID (デフォルトで Null に設定されます)。
-isXA	任意。XA コネクション・ファクトリであるかどうかを指定します (デフォルトで false に設定されます)。

JMS コネクション・ファクトリの削除

-removeJMSConnectionFactory コマンドを使用すると、1 つ以上の OC4J インスタンスから JMS コネクション・ファクトリを削除できます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -removeJMSConnectionFactory
-jndiLocation jndiLocation
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-removeJMSConnectionFactory -jndiLocation jms/ExampleQueueCF
```

表 6-27 -removeJMSConnectionFactory コマンドのパラメータ

パラメータ	説明
-jndiLocation	必須。削除するコネクション・ファクトリの JNDI ロケーション。

JMS コネクション・ファクトリの情報の取得

`-getJMSConnectionFactories` コマンドを使用すると、OC4J インスタンスまたはクラスタ内の OC4J インスタンスのグループで各 JMS コネクション・ファクトリの属性を戻すことができます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -getJMSConnectionFactories
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-getJMSConnectionFactories
```

JMS 宛先の管理

`admin_client.jar` ユーティリティを使用すると、次の各項で説明するように、OC4J JMS 宛先を管理できます。

- [JMS 宛先の追加](#)
- [JMS 宛先の削除](#)
- [JMS 宛先の情報の取得](#)

JMS 宛先の追加

`-addDestination` コマンドを使用すると、JMS 宛先を追加できます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -addDestination
-domain domain -name name -jndiLocation jndiLocation [-persistenceFile persistenceFile]
[-description description]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-addDestination -domain Queue -name ExampleQueue -jndiLocation jms/ExampleQueue
```

表 6-28 -addDestination コマンドのパラメータ

パラメータ	説明
<code>-domain</code>	必須。宛先の JMS ドメイン ('QUEUE' または 'TOPIC')。
<code>-name</code>	必須。宛先の OC4J JMS プロバイダ固有の名前。
<code>-jndiLocation</code>	必須。宛先をバインドする JNDI ロケーション。
<code>-persistenceFile</code>	任意。宛先に関連付ける永続性ファイル (デフォルトで Null に設定されます)。
<code>-description</code>	任意。宛先に関するテキストの説明 (デフォルトで Null に設定されます)。

JMS 宛先の削除

`-removeDestination` コマンドを使用すると、OC4J インスタンスまたはグループの各 OC4J インスタンスから JMS 宛先を削除できます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -removeDestination
-name name [-force true|false] [-removePFile true|false]
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-removeDestination -name ExampleQueue -removePFile true
```

表 6-29 -removeDestination コマンドのパラメータ

パラメータ	説明
<code>-name</code>	必須。削除する宛先の OC4J JMS プロバイダ固有の名前。
<code>-force</code>	任意。メッセージまたはコンシューマが存在するかどうかにかかわらず宛先が削除されます（デフォルトで <code>false</code> に設定されます）。
<code>-removePFile</code>	任意。ファイル・システムから永続性ファイルを削除します（デフォルトで <code>false</code> に設定されます）。

JMS 宛先の情報の取得

`-getDestinations` コマンドを使用すると、OC4J インスタンスまたはグループ内の各 OC4J インスタンスから各 OC4J JMS 宛先の属性を戻すことができます。このコマンドの構文は次のとおりです。

```
java -jar admin_client.jar uri adminId adminPassword -getDestinations
```

次に例を示します。

```
java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1
-getDestinations
```

リモート・クライアントを使用した OC4J の管理

6-5 ページの「[リモート管理クライアントのダウンロードおよび抽出](#)」で説明されているように、リモート管理クライアント・ユーティリティからファイルをインストールすると、リモート・クライアントを使用して OC4J を管理できます。その後、コマンドライン・ツールまたは JMX リモート API を介して `admin_client.jar` を使用できます。

リモートからの admin_client.jar コマンドの使用方法

6-5 ページの「[リモート管理クライアントのダウンロードおよび抽出](#)」で説明されているように、OC4J アプリケーション・サーバー・ターゲットに接続すると、リモート・クライアントから `admin_client.jar` コマンドを発行できます。OC4J インスタンス内で使用するのと同じ構文を使用します。

JConsole を使用したリモートの Oracle Application Server インスタンスへの接続

JConsole は、JDK 5.0 に含まれる JMX GUI コンソールです。JConsole は任意の JVM に接続し、稼働中の MBeanServer にフックできます。これにより、JVM のスレッドやメモリー使用量など、システムの様々な詳細が示された一連のページが表示されます。JConsole はローカル JVM に接続することも、JMX リモート API を使用してリモート JVM に接続することもできます。

管理クライアント・ユーティリティ・ディストリビューションには、JConsole によるリモートの OC4J または Oracle Application Server インスタンスへの接続に必要なライブラリが含まれています。ターゲット・インスタンスに接続するには、(Windows 環境でネイティブの実行可能ファイルとして提供されている) JConsole ユーティリティが、管理クライアント・ユーティリティ・ディストリビューションの関連する詳細とともに構成されている必要があります。

Oracle Application Server インスタンスに接続するには、次のようにします。

1. CLASSPATH 環境変数に /j2ee/instance/admin_client.jar を追加します。

```
set CLASSPATH=j2ee/home/admin_client.jar
```

2. CLASSPATH 環境変数に JConsole ライブラリを追加します。

```
set CLASSPATH=%CLASSPATH%;%JAVA_HOME%\lib\jconsole.jar
set CLASSPATH=%CLASSPATH%;%JAVA_HOME%\lib\tools.jar
```

3. OC4J ORMI プロトコルを使用できるように JMX コネクタを構成します。

```
set PROPS=jmx.remote.protocol.provider.pkgs=oracle.oc4j.admin.jmx.remote
```

4. JConsole を実行します。

```
%JAVA_HOME%\bin\jconsole
-J-Djava.class.path=%CLASSPATH%
-J-D%PROPS%
```

これにより JConsole が起動されます。

5. 「Connect to Agent」画面の「Advanced」タブで、OC4J または Oracle Application Server ターゲットの接続文字列、およびターゲットの管理ユーザー名とパスワードを入力します。

OC4J ターゲットでは、JMX URL のパターンは、Oracle Application Server ターゲットのパターンとは異なります。表 6-30 に、これらの URL パターンの例を示します。

表 6-30 OC4J および Oracle Application Server ターゲットの JMX URL

ターゲット	JMX URL
スタンドアロン OC4J サーバー	service:jmx:rmi://test-cycle.oracle.com:23791
Oracle Application Server の OC4J インスタンス	service:jmx:ormi:///opmn://test-cycle.oracle.com:6010/test1
Oracle Application Server クラスタ	service:jmx:rmi:///opmn://stadv69:6003/cluster/as101/admin

6. JConsole ユーティリティにより、ターゲット・インスタンスの OC4J MBeans が表示されます。これらの MBeans は、OC4J インスタンスの構成の表示および管理に使用できます。

Windows 環境では、JConsole によって使用される環境は、次に示す特別なシステム・プロパティ・フォームを使用して変更できます。

```
-J-Dname=value
```

次に、サンプル・コマンドのスク립トを示します。

```
setlocal

set URL=service:jmx:rmi:///opmn://test-cycle.oracle.com:6010/testunit

set JAVA_HOME=C:\java\jdk150_07

set JCONSOLE_CP
set JCONSOLE_CP=%JCONSOLE_CP%;%JAVA_HOME%\lib\jconsole.jar
set JCONSOLE_CP=%JCONSOLE_CP%;%JAVA_HOME%\lib\tools.jar

set ORACLE_HOME=D:\oc4j_admin_client
set ORACLE_CP=
set ORACLE_CP=%ORACLE_CP%;%ORACLE_HOME%\j2ee\home\admin_client.jar;

set CLASSPATH=%JCONSOLE_CP%;%ORACLE_CP%
set PROPS=
set PROPS=%PROPS%
-J-Djmx.remote.protocol.provider.pkgs=oracle.oc4j.admin.jmx.remote

set PROPS=%PROPS% -J-Djava.class.path=%CLASSPATH%

jconsole %PROPS% %URL%

endlocal
```

JMX プログラム・クライアントを使用したリモートからの OC4J の管理

管理クライアント・ユーティリティ・ディストリビューションは、JMX クライアント・アプリケーションがリモートの OC4J インスタンスに接続するための完全なクライアント環境を提供します。JMX プログラム・クライアントを使用し、MBeanServer への接続を確立できる JMX リモート API (JSR160) を介してリモートから OC4J を管理できます。JDK 5.0 を使用して実行するために必要な JAR ファイルは、管理クライアント・ユーティリティ・ディストリビューションによって提供される oc4jclient.jar および admin_client.jar のみです。

次の例では、JMX API とともにこれらの JAR ファイルが使用されています。

```
// A URL is of the form "service:jmx:rmi://127.0.0.1:23791"
JMXServiceURL serviceURL = new JMXServiceURL(_url);

Hashtable credentials = new Hashtable();
credentials.put("login", _username);
credentials.put("password", _password);

// Properties required to use the OC4J ORMI protocol
Hashtable env = new Hashtable();
env.put(JMXConnectorFactory.PROTOCOL_PROVIDER_PACKAGES,
"oracle.oc4j.admin.jmx.remote");
env.put(JMXConnector.CREDENTIALS, credentials);
JMXConnector jmxCon =
JMXConnectorFactory.newJMXConnector(serviceURL, env);
jmxCon.connect();

MBeanServerConnection mbeanServer =
jmxCon.getMBeanServerConnection();
```

JDK 5.0 では、このコードのコンパイルには Oracle ライブラリは不要で、JDK に用意されているライブラリのみが使用されます。

```
clear
@echo off
@setlocal
```

```

set J2EE_HOME=c:\java\oc4j-1013-prod\j2ee\home
set JAVA_HOME=c:\java\jdk50
set CLASSPATH=.

rem
rem Uncomment below if using JDK14
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\jmxri.jar
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\jmx_remote_api.jar
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\javax77.jar
rem

%JAVA_HOME%\bin\javac -classpath %CLASSPATH% -d . *.java
@endlocal

```

oc4j_admin_client_101340.zip ディストリビューションを使用してコードを実行するには、次のようにします。

1. 実行可能な JAR ファイルを作成します。
2. JAR ファイルを管理クライアント・ユーティリティ・ディストリビューションの j2ee/home ディレクトリに移動します。
3. リモートの OC4J インスタンスに接続します。

コードは、\$ORACLE_HOME/j2ee/home/oc4jclient.jar および \$ORACLE_HOME/j2ee/home/admin_client.jar を使用して JDK 5.0 で実行されます。

```

@echo off
@setlocal
clear
set J2EE_HOME=c:\java\oc4j-1013-prod\j2ee\home
set JAVA_HOME=c:\java\jdk50

rem Runtime classpath
set CLASSPATH=.
set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\oc4jclient.jar;
set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\admin_client.jar;

rem
rem Uncomment if using JDK14
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\jmxri.jar
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\jmx_remote_api.jar
rem set CLASSPATH=%CLASSPATH%;%J2EE_HOME%\lib\javax77.jar
@endlocal

```

この例のメイン・メソッドの接続 URL は、ローカルの OC4J インスタンスに接続するよう設定されています。ORMI ポートを介して Oracle Application Server に接続する必要がある場合には、次の形式のサービス URL を使用します。

```
service:jmx:mi|ormi://opmn://stadb57.us.oracle.com:6003/home
```

サービス URL は、OPMN デーモンから ORMI ポートを取得します。ORMI ポートは実行時に割り当てられます。OPMN 接続文字列パスを使用すると、指定された OC4J インスタンスに接続できます。

JMX クライアントを使用してリモートから OC4J インスタンスを管理する方法の詳細は、『Oracle Containers for J2EE 開発者ガイド』の「JMX リモート API (JSR-160) を使用するリモート管理」を参照してください。

admin.jar ユーティリティの使用方法

OC4J には `admin.jar` というコマンドライン・ユーティリティが用意されています。このユーティリティを使用すると、スタンドアロン OC4J インストールのアクティブな OC4J インスタンスで操作を実行できます。特に、このユーティリティを使用すると、OC4J の停止と再起動、アプリケーションのデプロイ、および現在のリソース使用状況に関する情報の収集ができます。

この章の内容は次のとおりです。

- [admin.jar の使用方法の概要](#)
- [スタンドアロン OC4J インスタンスの管理](#)
- [アプリケーションのデプロイまたはアンデプロイ](#)
- [アプリケーションの管理](#)
- [データソースの管理](#)
- [コネクタのデプロイまたはアンデプロイ](#)

注意：旧リリースで `admin.jar` ユーティリティに用意されていた OC4J の `web-site-related` オプション (`-site` コマンドを使用してアクセス可能) は、利用できなくなりました。OC4J の Web サイト構成の作成および管理方法の詳細は、[第 13 章「OC4J での Web サイトの管理」](#) を参照してください。

admin.jar の使用方法の概要

admin.jar ユーティリティは、スタンドアロン OC4J インスタンスの `ORACLE_HOME/j2ee/home` にデフォルトでインストールされます。

7-10 ページの「新しい構成への既存のデータソースの変換」で説明されているように、データソースを変換する場合を除き、このユーティリティを使用するには OC4J を起動しておく必要があります。また、このユーティリティは OC4J の起動には使用できません。ただし、7-3 ページの「スタンドアロン環境での OC4J の停止および再起動」で説明されているように、このユーティリティを使用してインスタンスを停止した後に再起動することはできます。

この項の内容は次のとおりです。

- [admin.jar の構文の概要](#)
- [コンソールへのヘルプ・テキストの出力](#)

注意: admin.jar ユーティリティは、スタンドアロン OC4J インストールの単一の OC4J インスタンスを管理するためだけに使用できます。

Oracle Application Server のコンポーネントとして稼働している OC4J インスタンスを管理するには、Oracle Process Manager and Notification Server (OPMN) を使用します。

より高度な機能を装備しているため、admin.jar のかわりに `admin_client.jar` ユーティリティを使用することをお勧めします。このユーティリティの使用方法は、第 6 章「[admin_client.jar ユーティリティの使用法](#)」を参照してください。

admin.jar の構文の概要

admin.jar ユーティリティは、次の構文を使用します。パラメータは、[表 7-1](#) で説明します。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId
      adminPassword options
```

たとえば、次のコマンドは OC4J サーバーの正常な停止を強制します。oc4jOrmiPort に指定された値は、デフォルトの 23791 です。adminId に指定されたユーザー名は、デフォルトの管理者アカウントのユーザー名の oc4jadmin です。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown
```

これらのコマンドの一部には、対象となるアプリケーションの名前をとる `-application` パラメータがあります。この値は、`server.xml` 構成ファイルの該当する `<application>` 要素内に定義されている、対象となる特定のアプリケーションの名前です。

表 7-1 ホストおよびログイン情報の設定

パラメータ	説明
<code>oc4jHost:oc4jOrmiPort</code>	admin.jar を起動する OC4J サーバーのホスト名およびポート番号。 admin.jar ツールでは、OC4J Remote Method Invocation (ORMI) プロトコルを使用して OC4J サーバーと通信します。そのため、これらの変数によって特定されるホストおよびポートは、リクエストのダイレクト先となる OC4J サーバーの <code>rmi.xml</code> ファイルに定義されます。 ORMI プロトコル用の OC4J のデフォルト・ポートは、23791 です。この値は、変更がない場合は省略できます。デフォルトを使用しない場合は、次のように <code>rmi.xml</code> ファイルの <code><rmi-server></code> 要素でホスト名とポート番号の両方を構成します。 <code><rmi-server port="oc4jOrmiPort" host="oc4jHost" /></code>

表 7-1 ホストおよびログイン情報の設定 (続き)

パラメータ	説明
<code>adminId adminPassword</code>	OC4J 管理ユーザー名およびパスワード。デフォルトの管理者アカウントのユーザー名は <code>oc4jadmin</code> です。

コンソールへのヘルプ・テキストの出力

`admin.jar` コマンドのオンライン・ヘルプ・テキストをコンソールに出力するには、`oc4jHost:oc4jOrmiPort adminId adminPassword` の後に `-help` と入力するだけです。次に例を示します。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -help
```

スタンドアロン OC4J インスタンスの管理

この項では、OC4J サーバーを管理するために `admin.jar` によって提供される機能の概要を説明します。この項の内容は次のとおりです。

- [スタンドアロン環境での OC4J の停止および再起動](#)
- [OC4J による変更済ファイルのチェック](#)

スタンドアロン環境での OC4J の停止および再起動

`admin.jar` を使用すると、OC4J サーバーのスタンドアロン・インスタンスを停止した後、再起動できます。

次のコマンドは、OC4J サーバーの停止を強制し、すべてのスレッドを即時終了させます。停止の `reason` として入力された文字列は、サーバー・ログ・ファイル `ORACLE_HOME/j2ee/home/log/server.log` に書き込まれます。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -shutdown force
      need_to_reboot_host_machine
```

表 7-2 OC4J サーバーの停止および再起動用オプション

オプション	説明
<code>-shutdown</code>	OC4J サーバーを停止します。 [ordinary force]: 停止タイプ。デフォルトは <code>ordinary</code> で、各スレッドを正常終了させます。 <code>force</code> オプションは、すべてのスレッドを即時終了させます。 [reason]: 必要に応じて、停止の理由を文字列として指定します。この文字列は <code>ORACLE_HOME/j2ee/home/log/server.log</code> ファイルに書き込まれます。文字列には空白を使用できません。
<code>-restart</code>	OC4J サーバーを再起動します。コンテナは、 <code>oc4j.jar</code> により起動されている必要があります。 [reason]: 必要に応じて、再起動の理由を文字列として指定します。この文字列は <code>ORACLE_HOME/j2ee/home/log/server.log</code> ファイルに書き込まれます。文字列には空白を使用できません。
<code>-version</code>	インストールされた OC4J のバージョンをコンソールに出力した後、終了します。

OC4J による変更済ファイルのチェック

-updateConfig オプションを使用して、サーバー・ディレクトリ構造で変更済ファイルがあるかどうかをチェックし、変更があったファイルをリロードするように OC4J に強制できます。

注意： この機能を使用するには、checkForUpdates フラグの値を all または adminClientOnly (デフォルト設定) のいずれかに設定する必要があります。checkForUpdates フラグの詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

表 7-3 更新済ファイルのチェック用オプション

オプション	説明
-updateConfig	ファイルが変更されていないかを確認し、変更があったファイルをリロードするように OC4J に強制します。

アプリケーションのデプロイまたはアンデプロイ

admin.jar を使用すると、スタンドアロン OC4J インスタンスに対して J2EE アプリケーションをデプロイまたはアンデプロイできます。

注意：

- admin.jar は、OPMN 管理 OC4J インスタンスへのアプリケーションのデプロイには使用できません。
- admin.jar では、EAR ファイルのデプロイのみをサポートします。WAR ファイルにパッケージされた Web モジュールなど、スタンドアロン・モジュールのデプロイはできません。
- admin.jar では、デプロイ・プランを使用できません。このユーティリティを使用してデプロイされたアーカイブには、OC4J 固有の必須デプロイメント・ディスク립タ・ファイル (orion-application.xml や orion-web.xml など) が含まれている必要があります。

アプリケーションのデプロイは 2 段階のプロセスです。最初にアーカイブを OC4J にデプロイし、次にアプリケーションへのアクセスに使用される Web サイトに Web モジュールをバインドする必要があります。

最初に、-deploy コマンドをアプリケーションのデプロイに使用します。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-deploy -file path/filename -deploymentName appName -targetPath deploy_dir
```

アーカイブをデプロイしたら、-bindWebApp コマンドを使用して、Web アプリケーションをそのアクセスに使用される Web サイトにバインドします。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-bindWebApp appName webAppName webSiteName contextRoot
```

たとえば、次のコマンドは、utility アプリケーションを OC4J にデプロイします。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -deploy
-file utility.ear -deploymentName utility
```

次の例では、その後、utility アプリケーションおよび utility-web Web モジュールをデフォルトの OC4J Web サイトにバインドしています。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password
-bindwebapp utility utility-web default-web-site /utility
```

表 7-4 アプリケーションのデプロイ用オプション

オプション	説明
-deploy	<p>アプリケーションをデプロイします。次のパラメータを使用して関連情報を指定します。</p> <p>-file filename: 必須。デプロイする EAR ファイルのパスおよびファイル名。</p> <p>-deploymentName appName: 必須。ユーザー定義のアプリケーションのデプロイ名。これと同じ名前が、OC4J 内でアプリケーションを識別する際に使用されます。また、アプリケーションをアンデプロイするときにも指定します。</p> <p>-targetPath path: 任意。EAR のデプロイ先となるサーバー・ノードのパス。指定しない場合、EAR は <code>ORACLE_HOME/j2ee/instance/applications</code> ディレクトリにデフォルトでデプロイされます。</p> <p>また、デプロイ済 EAR ファイルもこのディレクトリにコピーされます。後続のデプロイのたびに、この EAR ファイルは上書きされます。</p> <p>-parent appName: 任意。このアプリケーションの親アプリケーション。デプロイされると、子アプリケーション内のメソッドは、親アプリケーション内のメソッドを起動できます。親が指定されない場合は、default アプリケーションがデフォルトの親となります。</p> <p>-deploymentDirectory path: 任意。OC4J 固有のデプロイメント・ディスクリプタおよび生成済ファイル（コンパイル済 JSP クラスや EJB ラッパー・クラスなど）が格納されるディレクトリ。</p> <p>デフォルトのディレクトリは、<code>ORACLE_HOME/j2ee/instance/application-deployments</code> です。</p> <p>-iiopClientJar path/filename: 任意。EAR に含まれる各 EJB JAR 内にパッケージされたホーム、リモートおよびローカルのインタフェースに対して IIOP スタブを生成する場合に指定します。</p> <p>必要に応じて、生成されたスタブの出力先となる JAR のパスおよびファイル名を指定できます。指定しない場合、スタブのコピーは、<code>ORACLE_HOME/j2ee/homeapp-name/application-deployments/</code> にあるデプロイ済 EJB JAR と同じ名前の新しいサブディレクトリにある <code>_iiopClient.jar</code> というアーカイブに出力されます。</p> <p>この機能を使用するには、OC4J の起動時に GenerateIIOP システム・プロパティを有効にする必要があります。次に例を示します。</p> <pre>java -DGenerateIIOP=true -jar oc4j.jar</pre>

表 7-4 アプリケーションのデプロイ用オプション (続き)

オプション	説明
-bindWebApp	<p>Web アプリケーションを指定された Web サイトおよびコンテキスト・ルートにバインドします。</p> <ul style="list-style-type: none"> ■ appName: アプリケーション名。-deploy オプションの -deploymentName の値として設定されたのと同じ名前です。 ■ webAppName: Web モジュールの名前。この値は、EAR ファイル内に含まれる WAR ファイルの名前 (.WAR 拡張子を除く) である必要があります。 ■ webSiteName: この Web アプリケーションのバインド先となる Web サイトを示す name_web-site.xml ファイルの名前。 ■ contextRoot: Web モジュールのコンテキスト・ルート。この値は、http://localhost:8888/utility のように、Web ブラウザを介してアプリケーションにアクセスするために使用される URL の最後に追加されます。 <p>このオプションにより、web_site_name 変数に指定された name-web-site.xml 構成ファイルにエントリが作成されます。</p>
-undeploy appName	<p>デプロイ済 J2EE アプリケーションを OC4J インスタンスから削除します。appName の値は、ORACLE_HOME/j2ee/home/config/server.xml の application 要素に定義されている、OC4J 内のアプリケーションの名前です。</p> <p>アプリケーションをアンデプロイすると、次のような結果になります。</p> <ul style="list-style-type: none"> ■ アプリケーションは、OC4J ランタイムおよび server.xml ファイルから削除されます。 ■ すべてのアプリケーションの Web モジュールに対するバインドは、Web モジュールのバインド先であったすべての Web サイトから削除されます。 ■ アプリケーション・ファイルは、applications と application-deployments の両方のディレクトリから削除されます。 <p>オプションの -keepFiles パラメータは、OC4J 10g (10.1.3.4.0) では推奨されません。</p>

アプリケーションの管理

この項では、スタンドアロン OC4J インスタンスのアプリケーションを管理するために `admin.jar` によって提供される機能の概要を説明します。この項の内容は次のとおりです。

- アプリケーションの起動、停止または再起動
- アプリケーション内の EJB モジュールの更新

アプリケーションの起動、停止または再起動

`admin.jar` を使用すると、スタンドアロン OC4J インスタンスで停止していたアプリケーションの起動、停止または再起動ができます。

次の例では、OC4J で稼働している特定のアプリケーションを再起動しています。アプリケーション内のファイルが変更されている場合、アプリケーションまたはモジュールは自動的に再デプロイされます。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -application
myapplication -restart
```

表 7-5 アプリケーションの再起動用オプション

オプション	説明
<code>-application appName -start</code>	指定されたアプリケーションおよび子アプリケーションを起動します。
<code>-application appName -stop</code>	指定されたアプリケーションおよび子アプリケーションを停止します。
<code>-application appName -restart</code>	指定されたアプリケーションおよび子アプリケーションを再起動します。 OC4J のポーリングが有効であり、アプリケーション内のファイルが変更されている場合、そのアプリケーションは再デプロイされません。

アプリケーション内の EJB モジュールの更新

`admin.jar` ユーティリティには、OC4J インスタンスで稼働中のアプリケーション内で EJB モジュールの増分的または部分的な再デプロイを可能にする `-updateEJBModule` オプションがあります。このオプションは、開発環境から JAR ファイルを直接再デプロイするためにアプリケーション開発者が使用することを主な目的としています。

注意： 増分的な再デプロイは、CMP または BMP エンティティ Bean にアプリケーション全体を再デプロイするよりも効率的ですが、セッション Bean、メッセージドリブン Bean または EJB 3.0 JPA エンティティでは当てはまりません。この機能の使用に関する詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』の「更新された EJB モジュールの増分再デプロイ」を参照してください。

構文は次のとおりです。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-application appName -updateEJBModule relativePath [-file path/ejbJarName]
```

たとえば、次のコマンドを使用すると、`petstore` アプリケーションの `customerEjb.jar` モジュールを更新できます。開発者のマシン上には次のディレクトリ構造があると想定します。

```
/work
  /src    - application source code
  /build  - compiled class files
  /dist   - assembled EAR and JAR files
```

更新済 EJB JAR が /dist ディレクトリのアプリケーションの application.xml J2EE 標準デプロイメント・ディスクリプタに定義されている相対パスと一致する場所にある場合、次のコマンドを /dist ディレクトリから発行できます。

```
java -jar $ORACLE_HOME/admin.jar ormi://myoc4jserver:23791 oc4jadmin password
-updateApplication petstore -updateEJBModule customerEjb.jar
```

更新済ファイルが /build ディレクトリ内にある場合、JAR の場所を -file オプションに指定した次のコマンドを /dist ディレクトリから発行できます。

```
java -jar admin.jar ormi://myoc4jserver:23791 oc4jadmin password
-updateApplication petstore -updateEJBModule customerEjb.jar -file build/customerEjb.jar
```

表 7-6 EJB モジュールの更新用オプション

オプション	説明
-application <i>appName</i> -updateEJBModule	指定された EJB モジュールを新しい EJB モジュールで更新します。 <ul style="list-style-type: none"> ▪ <i>relativePath</i>: アプリケーションの application.xml J2EE デプロイメント・ディスクリプタに定義されている、更新済 BEAN を含む EBJ JAR への相対パス。 ▪ <i>-file path</i>: ファイルの場所が application.xml デプロイメント・ディスクリプタに指定されている相対パスと一致しない場合の更新済 EJB JAR のパスおよびファイル名。

データソースの管理

admin.jar を使用して、特定のアプリケーションのデータソースを作成、削除、リスト出力またはテストします。また、10.1.3 より前の data-sources.xml ファイルを新しいファイル形式に変換できます。

アプリケーション固有のデータソースの作成

新しいアプリケーション固有のデータソースを構成する -installDataSource オプションの構文は、次のとおりです。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-application appName -installDataSource -jar path
-url url -location jndiName [-pooledLocation jndiName]
[-xaLocation jndiName] [-ejbLocation jndiName] -username name
-password password [-connectionDriver className] -className className
[-sourceLocation jndiName] [-xaSourceLocation jndiName]
```

次に例を示します。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password -application myapp
-installDataSource -jar C:/jdbc/lib/ojdbc14dms.jar
-url jdbc:oracle:thin:@dev2:1521:main -location jdbc/OracleUddi
-username dbuser -password dbpw -className oracle.jdbc.pool.OracleDataSource
```

表 7-7 データソースの管理用オプション

オプション	説明
-application <i>appName</i>	指定されたアプリケーションの新しいデータソースをインストールします。次のパラメータを使用してデータソース情報を指定します。
-installDataSource	<p>-jar <i>path</i>: 必須。OC4J サーバーに追加される JDBC ドライバが含まれる JAR ファイルへのパス。</p> <p>-url <i>url</i>: 必須。JDBC データベースの URL。</p> <p>-location <i>jndiName</i>: 必須。RAW データソースの JNDI 名。たとえば、jdbc/DefaultPooledDS など。</p> <p>-pooledLocation <i>jndiName</i>: 任意。プールされたデータソースの JNDI 名。たとえば、jdbc/DefaultPooledDS など。</p> <p>-xaLocation <i>jndiName</i>: 任意。XA ソースの JNDI 名。たとえば、jdbc/xa/DefaultXADS など。-ejbLocation が指定されている場合は必須。</p> <p>-ejbLocation <i>jndiName</i>: 任意。コンテナ管理のトランザクション・データソースの JNDI 名。これは、グローバル JTA トランザクションを実行できるデータソースのみです。たとえば、jdbc/DefaultDS など。</p> <p>-username <i>name</i>: 必須。データベースにログインするためのユーザー名。</p> <p>-password <i>password</i>: 必須。データベースにログインするためのパスワード。</p> <p>-connectionDriver <i>className</i>: 任意。JDBC データベース・ドライバのクラス。</p> <p>-className <i>className</i> 必須。oracle.jdbc.pool.OracleDataSource などのデータソースのクラス名。</p> <p>-sourceLocation <i>jndiName</i>: 任意。この特殊なデータソースの基礎となるデータソースの JNDI 名。</p> <p>-xaSourceLocation <i>jndiName</i>: 任意。この特殊なデータソースの基礎となる XA データソースの JNDI 名。</p>

既存のデータソースのリスト出力、テストおよび削除

admin.jar を使用すると、特定のアプリケーションに関連付けられたデータソースのリスト出力やテスト、または削除も可能です。

表 7-8 アプリケーションおよびデータソースの管理用オプション

オプション	説明
-application <i>appName</i> -listDataSource	インストールされた各データソース・オブジェクトについて静的に構成された情報を取得します。
-application <i>appName</i> -testDataSource	既存のデータソースをテストします。次のパラメータを使用して情報を指定します。 -location <i>jndiName</i> : データソースのネームスペースの場所。たとえば、jdbc/DefaultDS など。必須。 -username <i>name</i> : パスワードとともにログインに使用するユーザー名。任意。 -password <i>password</i> : ログインに使用するパスワード。任意。
-application <i>appName</i> -removeDataSource	既存のデータソースを削除します。次のパラメータを使用して情報を指定します。 -location <i>jndiName</i> : データソースのネームスペースの場所。たとえば、jdbc/DefaultDS など。必須。

新しい構成への既存のデータソースの変換

OC4J 10g (10.1.3.4.0) 実装では、10.1.3 および 10.1.3 より前 (10.1.2 および 9.0.4) の形式の data-sources.xml ファイルも使用できます。10.1.3 より前の OC4J 実装で使用され、独自の data-sources.xml ファイルを含むアプリケーションでは、Application Server Control を使用して data-sources.xml ファイルを変更 (既存のデータソースの変更またはデータソースの作成や削除など) すると、OC4J 10g (10.1.3.4.0) 実装により data-sources.xml ファイルが 10.1.3 より前の形式から 10.1.3 形式に自動的に変換されます。

スタンドアロン OC4J が稼働中または停止中の data-sources.xml ファイルの変換

admin.jar コマンドの -convertDataSourceConfiguration オプションを使用すると、10.1.3 より前の data-sources.xml ファイルを新しいファイル形式に変換できます。

スタンドアロン環境で OC4J インスタンスがアクティブな状態では、次の構文で admin.jar を使用して、10.1.3 より前の data-sources.xml ファイルを 10.1.3 形式に手動で変換できます。

```
java -jar admin.jar ormi://oc4jHost:oc4jOrmiPort adminId adminPassword
-convertDataSourceConfiguration old-data-sources.xml new-data-sources.xml
```

たとえば、次のコマンドは、既存の構成を変換して新しいファイルに書き込みます。

```
java -jar admin.jar ormi://localhost:23791 oc4jadmin password
-convertDataSourceConfiguration C:\oc4j\j2ee\home\config\data-sources.xml
C:\new\data-sources.xml
```

変換後、古い data-sources.xml は参照に必要な情報が含まれているため、削除するのではなく名前を変更することをお勧めします。生成された新しいファイルは、レガシー・ファイルが格納されているディレクトリにコピーします。

構文では、ORMI URL はオプションです。ORMI URL は、OC4J が稼働している場合にのみ指定できます。

また、OC4J インスタンスが稼働していなくても、デプロイ前に `data-sources.xml` ファイルを変換できます。オフライン変換の構文は次のとおりです。

```
java -jar admin.jar -convertDataSourceConfiguration  
old-data-sources.xml new-data-sources.xml
```

注意：

- ORMI ポートを指定した場合は、OC4J は稼働している必要があります。OC4J が稼働していない場合には、`admin.jar` コマンドラインから ORMI URL を除外する必要があります。
 - ORMI ポートを指定しない場合、`admin.jar` コマンドは、OC4J が稼働している状態または稼働していない状態のいずれかで実行されます。
 - `admin.jar` ユーティリティは、スタンドアロン OC4J 環境でのみ使用できます。このユーティリティは、Oracle Application Server 環境にインストールされますが、OPMN 管理環境では使用できません。
 - 新しい `admin_client.jar` ユーティリティは、スタンドアロンおよび管理されている Oracle Application Server のどちらの環境でも使用できます。ただし、`admin_client.jar` ユーティリティでは、`data-sources.xml` ファイルの変換は行われません。
-

アプリケーションおよび新しい `data-sources.xml` ファイル間での一貫性の確認

変換後、アプリケーションと新しいファイルの間にデータソースの参照に使用される JNDI ロケーションに関する一貫性があることを確認するために、手動または自動で、新しい `data-sources.xml` ファイルを表示して調査します。

古い形式では複数のロケーション属性 (`location`、`ejb-location` および `xa-location`) などが使用されています。そのため、新しいファイルには使用されていないデータソース定義が含まれているので、この一貫性チェックを実行することをお勧めします。新しい 10.1.3 形式への変換により、古い `data-sources.xml` ファイルで指定されている各ロケーション属性に対応する別のデータソースが、新しい `data-sources.xml` ファイルに作成されます。多くの場合、クライアント・アプリケーションでは、`location` または `ejb-location` 属性で定義されているデータソースのみが使用されます。変換された `data-sources.xml` ファイルには、アプリケーションでは使用されない定義が含まれていますが、削除できます。

新しい `data-sources.xml` 形式の例は、『Oracle Containers for J2EE サービス・ガイド』の「データソース」の章を参照してください。

コネクタのデプロイまたはアンデプロイ

次のコマンドの1つを使用すると、RAR ファイルにパッケージされた Java コネクタ・アーキテクチャに準拠するリソース・アダプタをデプロイまたはアンデプロイできます。

表 7-9 アプリケーションのデプロイ用オプション

オプション	説明
-deployconnector	コネクタをデプロイします。次のパラメータにアプリケーション情報を指定します。 -file path: 必須。デプロイする RAR ファイルのパスおよびファイル名。 -name name: リソース・アダプタの名前。 -nativeLibPath path: RAR ファイル内の、ネイティブ・ライブラリ (DLL など) を格納するディレクトリへのパス。 -grantAllPermissions: 必要に応じて、リソース・アダプタによってリクエストされるすべてのランタイム権限を付与する場合に指定します。
-undeployconnector	指定されたコネクタをアンデプロイします。 name name: アンデプロイするコネクタの名前。 スタンドアロン RAR のアンデプロイでは、default アプリケーションを再起動する必要はありません。

クラスタおよび OC4J グループの構成と管理

この章では、Oracle Application Server 環境のクラスタ・トポロジと、Oracle Application Server クラスタ内の OC4J インスタンスのグループを構成および管理する方法について説明します。次の項目が含まれます。

- クラスタリングの概要
- Oracle Application Server クラスタ内の OC4J グループの作成および管理
- クラスタの構成
- クラスタのステータスの表示
- Oracle HTTP Server を使用したルーティングおよびロード・バランシングの構成
- アプリケーション・マウント・ポイントの構成
- 複数の JVM での OC4J インスタンスの稼働

アプリケーションのクラスタリング（セッションまたは状態のレプリケーションを目的として Oracle Application Server ノードにデプロイされたアプリケーションのクラスタリング）は、第 9 章「OC4J でのアプリケーションのクラスタリング」で説明します。

クラスタリングの概要

この項では、Oracle Application Server 10g リリース 3 (10.1.3.4.0) でサポートされるクラスタリング方式の概要と、10.1.3 リリースと旧リリースの機能の重要な変更点について説明します。次の項目が含まれます。

- クラスタリングの動作
- サポートされるクラスタリング・モデル
- クラスタリングにおける変更点

クラスタリングの動作

現行リリースでは、クラスタ・トポロジは、2 つ以上の緩やかに接続された Oracle Application Server ノードとして定義されます。

クラスタ内での接続は、Oracle Application Server コンポーネント (OC4J や Oracle HTTP Server など) 間の通信を管理する Oracle Notification Server (ONS) の機能により提供されます。ONS サーバーは、Oracle Process Manager and Notification Server (OPMN) のコンポーネントで、すべての Oracle Application Server ホストにデフォルトでインストールされます。クラスタ・トポロジの構成時、実際には各 Oracle Application Server ノードで稼働している ONS サーバーに接続しています。

Oracle Application Server の旧リリースでは、完全に接続されたサーバー・ノードの集合のクラスタリングのみをサポートしていました。つまり、各ノードは ONS 構成ファイル (ons.conf) に明示的に指定する必要がありました。クラスタに対してノードが追加または削除されると、各サーバー・ノードで構成を更新し、サーバーを再起動する必要がありました。

現行リリースでは、新しい動的検出方式をサポートし、基本的にクラスタが自己管理できるようになりました。このフレームワークでは、各 ONS は最新のクラスタ・トポロジのマップを保持します。新しい ONS がクラスタに追加されると、既存の ONS は、新しいノードとその接続情報をそれぞれのマップに追加します。同時に、新しい ONS はすべての既存のノードをそのマップに追加します。また、ONS がクラスタから削除されると、残りのノードのマップにこの変更が反映されます。

Oracle Application Server リリース 3 (10.1.3.0.0) では、ONS 構成ファイル (ons.conf) は使用されません。かわりに、ONS 構成データは、OPMN 構成ファイル opmn.xml の <notification-server> 要素に設定されています。このファイルは、各ノードの ORACLE_HOME/opmn/conf ディレクトリにあります。同様に、クラスタリング構成は、<topology> サブ要素内に設定されています。<notification-server> 要素で使用できる <topology> サブ要素は 1 つのみです。

次の例に、opmn.xml のクラスタ・トポロジ構成を示します。

```
<notification-server>
  <topology>
    <discover list="*225.0.0.20:8001"/>
  </topology>
  ...
</notification-server>
```

<topology> 要素に指定されたクラスタリング構成は、ノードにインストールされた Oracle Application Server コンポーネント (Oracle HTTP Server や OC4J など) のすべてのインスタンスに適用されます。クラスタ・トポロジ内のすべてのノードは、opmn.xml ファイルに指定されているのと同じように構成されている必要があります。

サポートされるクラスタリング・モデル

次のクラスタリング・モデルがサポートされます。

- 動的ノード検出

この構成では、同じサブネット内の各 ONS ノードはマルチキャスト・メッセージでその存在を通知します。各ノードのクラスタ・トポロジのマップはノードが追加または削除されると自動的に更新され、クラスタは自己管理できるようになります。

構成手順は、8-13 ページの「[マルチキャストを使用した動的ノード検出の構成](#)」を参照してください。

- 検出サーバーとしての静的ハブ

クラスタ内の特定のノードを、クラスタのトポロジ・マップを保持する検出サーバーとして機能するように構成します。他のノードは検出サーバーを介して相互に接続します。トポロジの検出サーバー・ハブは、別のトポロジのハブに接続できます。

8-17 ページの「[静的検出サーバーの構成](#)」を参照してください。

- 分離されたトポロジのゲートウェイを介した接続

この構成は、指定されたゲートウェイ・ノードを使用して、ファイアウォールによって分離されたトポロジまたは別のサブネット上にあるトポロジを接続する場合に使用します。

詳細は、8-19 ページの「[トポロジ間ゲートウェイの構成](#)」を参照してください。

- 手動によるノード構成

この構成では、クラスタ内の各ノードのホスト・アドレスおよびポートを手動で構成に指定します。これは、Oracle Application Server 10g リリース 2 (10.1.2) でサポートされていたものと同じクラスタリング方式で、主に下位互換性を提供するためにサポートされています。

手順は、8-21 ページの「[静的ノードツーノード通信の構成](#)」を参照してください。

クラスタリングにおける変更点

次に、Oracle Application Server 10g リリース 3 (10.1.3) のクラスタ構成における旧リリースからの変更点を示します。

- Oracle Application Server 10g (10.1.3.1.0) では、OC4J インスタンスはクラスタ・トポロジ内のグループに属しています。これにより、Oracle Application Server クラスタ全体でグループに対してデプロイ、構成および管理操作を実行できます。

Oracle Application Server 10g (10.1.3.4.0) では、グループは管理者が任意の名前を使用して明示的に作成します。グループを作成したら、グループには、クラスタ・トポロジに属する任意の OC4J インスタンスを移入できます。

注意： グループの作成および管理の手順は、Oracle Application Server 10g (10.1.3.0.0) から変更されました。10.1.3.0.0 リリースを使用していた場合は、8-4 ページの「[Oracle Application Server クラスタ内の OC4J グループの作成および管理](#)」で説明されている、10.1.3.4.0 リリースにおけるグループの作成および管理の新しい手順を確認してください。

- Distributed Configuration Management (DCM) フレームワークは、クラスタで共通の構成情報をレプリケートするために旧リリースの Oracle Application Server で使用されましたが、現行リリースには含まれていません。つまり、次のことを意味します。
 - dcmet1 コマンドライン・ユーティリティまたは Application Server Control を使用した構成はサポートされなくなりました。
 - 現在、クラスタ構成は、クラスタ内の各ノードにインストールされている opmn.xml ファイルに手動でレプリケートする必要があります。

- ONS 構成ファイル (ons.conf) は使用されなくなりました。現在、ONS 接続データは、OPMN 構成ファイル opmn.xml の <notification-server> 要素に設定されています。このファイルは、OC4J または Oracle HTTP Server インスタンスを含む各ノードの ORACLE_HOME/opmn/conf ディレクトリにあります。
- クラスタ内のその他すべてのノードと接続するように、各ノードを手動で構成する必要はなくなりました。

Oracle Application Server クラスタ内の OC4J グループの作成 および管理

OPMN 管理環境のすべての OC4J インスタンスは、**グループ**に属している必要があります。グループは、同じクラスタ・トポロジに属する一連の OC4J インスタンスです。グループでは、グループ内のすべての OC4J インスタンスに対して、共通の構成、管理およびデプロイ・タスクを同時に実行できます。

注意： Oracle Application Server クラスタのグループ内に存在するすべての OC4J インスタンスは、10.1.3.4.0 など、同じリリースであることが必要です。

Application Server Control を使用して、追加のグループを作成し、「グループ」ページから OC4J インスタンスのグループに対して次のタスクを実行できます。

- 起動、停止および再起動などのプロセス管理操作
- デプロイ、アンデプロイおよび再デプロイなどのデプロイ操作
- JDBC データソースやコネクション・プールの作成、変更または削除などの JDBC 管理操作
- JMS 宛先の作成および削除、JMS コネクション・ファクトリの作成、変更または削除などの JMS プロバイダ操作

「グループ」ページを表示するには、「クラスタ・トポロジ」ページの「グループ」セクションでグループ名をクリックします。

図 8-1 「クラスタ・トポロジ」ページの「グループ」セクション

The screenshot shows the Oracle Enterprise Manager 10g Application Server Control interface. The page title is "Cluster Topology" and it indicates the page was refreshed on Aug 2, 2006 at 4:37:13 AM PDT. The "Overview" section shows 1 Host, 1 Application Servers, 2 OC4J Instances, and 0 HTTP Server Instances. The "Members" section is set to "View By" Application Servers and shows a table of members. The "Groups" section is active, showing a table of groups. The "Administration" section lists various configuration options.

Overview

Hosts 1 Application Servers 1
OC4J Instances 2 HTTP Server Instances 0

Members

View By: Application Servers

Start Stop Restart

Select All | Select None | Expand All | Collapse All

Select	Focus	Name	Status	Type	Category	Host	CPU (%)	Memory (MB)
<input type="checkbox"/>		All Application Servers						
<input type="checkbox"/>	<input checked="" type="checkbox"/>	060725basic.stacle.com		Application Server		stacle		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	home (JVMs: 1)	↑	OC4J			1.84	313.28
<input type="checkbox"/>	<input checked="" type="checkbox"/>	myOC4J42 (JVMs: 1)	↑	OC4J			0.03	77.12

Indicates the active ASControl instance.

TIP If a parent topology member is selected all contained members are implicitly selected.

Groups

A group is a collection of OC4J instances. Certain common management tasks can be performed simultaneously on all OC4J instances in a group. For more information, see [About Groups](#)

Start Stop Delete Create

Select	Name	OC4J Instance	Status	Application Server
<input checked="" type="radio"/>	default_group	myOC4J42	↑	060725basic.stacle.com
		home	↑	060725basic.stacle.com

Administration

- Cluster MBean Browser
- Routing ID Configuration
- Java SSO Configuration
- Topology Network Configuration
- Runtime Ports

アプリケーション・サーバー・インスタンスをインストールすると、デフォルトの OC4J グループ (default_group) が自動的に作成されます。Oracle Application Server 10g (10.1.3.4.0) のインストール時に、デフォルト・グループに存在するデフォルトの OC4J インスタンスがインストーラにより作成されます。その後、OC4J インスタンスを追加してグループに編成できます。

たとえば、Oracle Application Server クラスタ全体のグループのすべての OC4J インスタンスに特定のアプリケーションをデプロイするための新規グループを作成できます。その後、Application Server Control の「グループ」ページを使用して、クラスタ全体で OC4J グループ内のアプリケーションの全インスタンスに対してアプリケーション固有の構成の変更を実行できます。

次の各項では、1 つ以上の Oracle Application Server クラスタ全体でレプリケートされたアプリケーションで、グループ操作を行うために OC4J インスタンスのグループを作成および管理する方法を説明します。

- OC4J インスタンスのグループの作成
- グループ内の OC4J インスタンスの管理
- クラスタでの変更のレプリケート
- 複数の JVM での OC4J インスタンスの稼働

OC4J インスタンスのグループの作成

グループを使用すると、複数の OC4J インスタンスに対して次の各タスクを一度で実行できます。

- グループ内のすべての OC4J インスタンスの OC4J サーバー・プロパティの変更
- グループ内のすべての OC4J インスタンスの起動または停止
- グループ内の全 OC4J インスタンス上のアプリケーションのデプロイ、アンデプロイまたは再デプロイ
- JDBC データソースやコネクション・プールの作成、変更または削除などの JDBC 管理操作の実行
- JMS 宛先の作成または削除、JMS コネクション・ファクトリの作成、変更または削除などの JMS プロバイダ操作の実行

Application Server Control を使用したグループの管理 :

1. 「クラスタ・トポロジ」ページの「グループ」で、グループを選択します。
2. 「管理」タブを選択します。
3. 「管理」ページには、グループ全体の管理機能が表示されています。これらの機能には、セキュリティ・プロバイダの管理は含まれません。

Application Server Control を使用した新規 OC4J グループの作成 :

1. 「クラスタ・トポロジ」ページの「グループ」で、「作成」を選択します。
2. 「グループの作成」ページで次のようにします。
 - a. グループ名を指定します。

グループ名に使用できるのは英数字およびアンダースコアのみで、カッコ、ピリオド、ドル記号 (\$)、アスタリスク (*) またはカンマなどの特殊文字は使用できません。名前は、文字またはアンダースコアで開始する必要があります。

表 8-1 に、OC4J インスタンスとグループの有効な名前および無効な名前の例を示します。

表 8-1 OC4J インスタンスおよびグループの名前

インスタンスまたはグループの有効な名前	インスタンスまたはグループの無効な名前
OC4J1	\$OC4J_2
_production_apps	32_PROD_test
test_environment_42	!deployGroup2
Deployment_Group3	deployment_(group3)

- b. グループに移動する OC4J インスタンスを選択します。

OC4J インスタンスを新しいグループに移動すると、インスタンスは前のグループから削除されます。インスタンスを移動する前に停止しておく必要があります。

- c. 「作成」を選択します。

注意： 次のようにして、グループの作成後に OC4J インスタンスをグループに移動することも可能です。

1. 「クラスタ・トポロジ」ページの「グループ」で、グループを選択します。
2. 「グループ: *groupname*」ページで、「追加」を選択します。

グループを作成すると、「クラスタ・トポロジ」ページのグループのリストに表示されます。8-7 ページの「[グループ内の OC4J インスタンスの管理](#)」で説明されているように、後からグループに OC4J インスタンスを追加することや、グループからインスタンスを削除することが可能です。

次の操作中にグループを作成することも可能です。

- 新規 OC4J インスタンスの作成中

新規 OC4J インスタンスの作成時に、新規グループの作成、またはインスタンスの既存のグループの特定を実行できます。グループを指定しない場合、新規インスタンスは `default_group` に割り当てられます。
- グループからの OC4J インスタンスの削除中

グループからの OC4J インスタンスの削除時に、新規グループの作成、またはインスタンスの既存のグループの特定を実行できます。

注意：グループ間での OC4J インスタンスの移動には、次の制限が適用されます。

- グループからまたはグループに OC4J インスタンスを移動する前に停止しておく必要があります。
 - インスタンスをグループから移動する際には、グループ内の 1 つ以上の OC4J インスタンスが稼働している必要があります。
 - グループの OC4J インスタンスが 1 つのみの場合、移動する前にそのインスタンスを停止して別のインスタンスを作成し、その新しいインスタンスを起動する必要があります。
-

次に、複数の OC4J インスタンスおよびグループを使用して、Oracle Application Server 環境を管理する例を示します。

- 特定の目的のために OC4J インスタンスを作成します。たとえば、デフォルトの OC4J インスタンスを管理 OC4J とし、Application Server Control のデプロイにのみ使用します。本番アプリケーションのデプロイ用には別の OC4J インスタンスを作成します。
- パフォーマンスを向上させるために追加の OC4J インスタンスを作成し、本番アプリケーションのロード・バランシングを実現します。
- 同じアプリケーションのデプロイ先である OC4J インスタンスをグループ化し、個々の OC4J インスタンスではなく、そのグループに対してアプリケーション固有の変更を行えるようにします。また、個々の OC4J インスタンスに何度もデプロイするのではなく、そのグループにアプリケーションを一度にデプロイできます。

グループ内の OC4J インスタンスの管理

OC4J には、グループへの追加の OC4J インスタンスの作成や、Oracle Application Server クラスタ内のグループからのインスタンスの削除を実行するためのツールがあります。新規作成された OC4J インスタンスは、Application Server Control を使用してアクセスおよび管理できます。

この項の内容は次のとおりです。

- [OC4J インスタンスの追加作成](#)
- [新規インスタンスへのアクセスおよび管理](#)
- [グループからの OC4J インスタンスの削除](#)

OC4J インスタンスの追加作成

グループに OC4J インスタンスを追加するには、次の方法があります。

- Application Server Control の「アプリケーション・サーバー」ページを使用
- `ORACLE_HOME/bin` ディレクトリにインストールされている `createinstance` ユーティリティを使用

Application Server Control を使用した OC4J インスタンスの作成 Application Server Control を使用して OC4J インスタンスを作成するには、次のようにします。

1. 「クラスタ・トポロジ」ページで、Oracle Application Server インスタンスの名前をクリックし、「アプリケーション・サーバー: instance_name」ページへ移動します。
2. 「OC4J インスタンスの作成」をクリックします。
3. 「OC4J インスタンスの作成」ページで、次の情報を入力します。
 - **OC4J インスタンス名**: インスタンス名を入力します。

注意: home はデフォルトの OC4J インスタンスの名前用に予約されているため、home をインスタンス名として入力することはできません。

- 次の項目のいずれかを選択します。
 - **名前を指定して既存のグループに追加:** 「既存のグループ名」からグループを選択します。
 - **名前を指定して新規グループに追加:** 「新規グループ名」フィールドで、新規グループの名前を入力します。
 - 「作成後にこのインスタンスを起動します。」を選択します。
4. 「作成」をクリックします。

インスタンスが作成されると確認画面が表示されます。この OC4J インスタンスのパスワードは、インストール用に `oc4jadmin` ユーザーに使用されるパスワードと同じです。

createinstance ユーティリティを使用した OC4J インスタンスの作成 `createinstance` ユーティリティを使用すると、次の構文でグループに追加の OC4J インスタンスを作成できます。

```
createinstance -instanceName instanceName [-port httpPort] [-groupName group]
```

注意: home はデフォルトの OC4J インスタンスの名前用に予約されているため、instanceName に home を指定することはできません。

スタンドアロンの OPMN 管理 OC4J インスタンス（「J2EE Server およびプロセス管理」インストール・タイプ）にインスタンスを新規作成するときは、HTTP リスナー・ポートを `httpPort` の値として指定する必要があります。この HTTP リスナー・ポートは、インスタンス用に作成された Web サイト構成ファイル `default-web-site.xml` に設定されます。

すべての新規 OC4J インスタンスがグループに割り当てられます。指定したグループが存在しない場合は、作成されます。`-groupName` パラメータが指定されていない場合、インスタンスは `default_group` グループに割り当てられます。

作成プロセスの一環として、パスワードの入力を要求されます。このパスワードは、このインスタンスの `oc4jadmin` ユーザーと関連付けられます。OPMN との問題を回避するために、`oc4jadmin` ユーザーが管理インスタンスの Application Server Control へのアクセスに使用するのと同じパスワードを入力することをお勧めします。

作成操作の一部として、新規インスタンスは既存の `opmn.xml` ファイルに追加されます。OPMN が新規インスタンスを認識していることを確認するために、作成操作の最後に OPMN のリロードが実行されます。このリロードでは、`createinstance` ユーティリティが OPMN の構成に使用された `MBeanServer` に接続している必要があります。新規 OC4J インスタンスのパスワードは、認証に使用されます。新規インスタンスのパスワードが `MBeanServer` が稼働しているインスタンスのパスワードと異なる場合は、エラーが戻されます。このためにインスタンスが作成されないことはありませんが、OPMN またはその他のコンポーネントが新規インスタンスに接続する必要がある場合に問題の原因になります。そのため、すべての OC4J インスタンスをターゲットの Oracle Application Server クラスタに同じパスワードで作成することをお勧めします。

また、ユーザーがグループ操作を実行できるように、Oracle Application Server クラスタ内に存在するグループの各 OC4J インスタンスの `oc4jadmin` ユーザーに同じパスワードを指定する必要があります。

注意:

- `createinstance` ユーティリティは、Oracle Application Server インスタンスの状態が実行中か停止中かに関係なく使用できます。
 - ORMI over SSL (ORMIS) リクエストの受入れに新規 OC4J インスタンスが必要な場合、『Oracle Containers for J2EE セキュリティ・ガイド』で説明しているように、インスタンス固有の `rmi.xml` ファイルで ORMIS を構成し、`opmn.xml` を ORMIS ポート情報で更新する必要があります。
-
-

HTTP ポートを `-port` の値に指定できます。この機能は、Oracle Application Server インスタンスに Oracle HTTP Server が含まれない場合に必要です。HTTP ポートを設定すると、OC4J インスタンスのホームページに直接アクセスできるようになります。

新規インスタンスは、デフォルトの home OC4J インスタンスと同じ場所の、新しい `ORACLE_HOME/j2ee/instance` ディレクトリ内に作成されます。インスタンス構成が指定される新しい `<process-type>` 要素も `opmn.xml` 構成ファイルに追加されます。

home インスタンスの `<process-type>` 要素は、home インスタンスと同じ設定で作成される新規 OC4J インスタンスのテンプレートとして機能します。`opmn.xml` ファイルに存在するインスタンスの `<process-type>` 要素の `heapsize`、`numprocs` および `timeout` 属性値などの設定を変更することで、新規インスタンスの構成を変更できます。OC4J インスタンスの構成を変更した場合は、変更内容が反映されるようにそのインスタンスを再起動する必要があります。

次のディレクトリおよびファイルは、新しい `ORACLE_HOME/j2ee/instance` ディレクトリ構造で生成されます。

```

applib/
applications/
config/
    contains default versions of all server-level configuration files
config/database-schemas/
    contains all database schema XML files packaged with OC4J
connectors/
    contains RAR files packaged with OC4J
log/
persistence/

```

新規インスタンスには、OC4J バイナリ・ライブラリが含まれません。かわりに、home インスタンスにインストールされたライブラリを利用します。この新規インスタンスには、default アプリケーションはデプロイされますが、他のデプロイ済アプリケーション (Application Server Control など) のバイナリおよび構成ファイルはコピーされません。

新規インスタンスへのアクセスおよび管理

OPMN によって起動された新規インスタンスは、Application Server Control の「クラスタ・トポロジ」ページからアクセスできます。

oc4jadmin ユーザーとしてログインし、createinstance ユーティリティを使用してインスタンスを作成したときに設定したパスワードを指定します。

ログインすると、アプリケーションのデプロイなどのあらゆる管理者タスクをインスタンスで実行できます。

グループからの OC4J インスタンスの削除

8-6 ページの「OC4J インスタンスのグループの作成」で説明されているように、別のグループに移動するか削除することで、グループから OC4J インスタンスを削除できます。OC4J インスタンスを削除するには、次の方法があります。

- Application Server Control で OC4J インスタンスがインストールされている Oracle Application Server の「アプリケーション・サーバー」ページを使用
- `ORACLE_HOME/bin` ディレクトリにインストールされている `removeinstance` ユーティリティを使用

どちらの方法でも、`j2ee` ディレクトリ構造からインスタンスに作成されたディレクトリが削除され、`opmn.xml` からインスタンスの構成データが削除されます。OC4J インスタンスの削除には次のガイドラインが適用されます。

- インストール中に Oracle Application Server によって作成された OC4J home インスタンスは削除できません。
- インストール後にユーザーによって作成された OC4J インスタンスは削除できます。
- 削除する OC4J インスタンスは停止状態である必要があります (Application Server Control によって実行されます)。
- `removeinstance` ツールの使用時に OPMN が稼働している場合は、`opmnctl reload` を起動して更新済の `opmn.xml` をランタイムにリロードする必要があります。

Application Server Control を使用した OC4J インスタンスの削除 Application Server Control を使用して OC4J インスタンスを削除するには、次のようにします。

1. 「クラスタ・トポロジ」ページで、OC4J インスタンスが稼働している Oracle Application Server インスタンスの名前をクリックし、「アプリケーション・サーバー: instance_name」ページへ移動します。
2. 削除するインスタンスの「削除」アイコンをクリックします。
3. 「確認」ページで、「はい」をクリックします。

インスタンスが削除されると確認画面が表示されます。

removeinstance ユーティリティを使用した OC4J インスタンスの削除 OC4J インスタンスは、`removeinstance` ユーティリティを使用して削除できます。このユーティリティは、`ORACLE_HOME/j2ee` ディレクトリ構造からインスタンス用に作成されたディレクトリを削除し、`opmn.xml` からインスタンスの構成データを削除します。

`removeinstance` ユーティリティは、`ORACLE_HOME/bin` ディレクトリにインストールされています。構文は次のとおりです。

```
removeinstance -instanceName instanceName
```

ユーティリティを使用してインスタンスを削除するには、次のようにします。

1. インスタンスを停止します。


```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=oc4J_instanceName
```
2. インスタンスを削除します。


```
ORACLE_HOME/bin/removeinstance -instanceName oc4J_instanceName
```

クラスタでの変更のレプリケート

Oracle Application Server リリース 3 (10.1.3) には Distributed Configuration Management (DCM) フレームワークが用意されていないため、Oracle Application Server 10.1.3 ではクラスタ内での構成ファイルの同期が変更されました。表 8-2 に、レプリケートが必要なファイルをまとめます。

リリース 10.1.3.1.0 で導入された OC4J のグループ化機能 (8-6 ページの「OC4J インスタンスのグループの作成」を参照) を使用すると、Application Server Control、admin_client.jar コマンドライン・ユーティリティまたは OC4J Ant タスクを使用して、OC4J インスタンスのグループ全体に一貫して EAR、WAR、RAR および共有ライブラリをデプロイできます。これにより、OC4J インスタンスのグループ内のモジュール・レベルで一貫性のある構成を実現できます。これらのツールを使用した OC4J インスタンスのグループへのデプロイの詳細は、第 6 章「admin_client.jar ユーティリティの使用方法」および『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

特定の構成ファイルでは、管理者はグループ化機能を使用して、Application Server Control、admin_client.jar コマンドライン・ユーティリティおよび OC4J Ant タスクから、OC4J インスタンスのグループ全体のデータソース、コネクション・プールおよび JMS リソースを構成することもできます。特に、この機能がサポートされている構成ファイルは、data-sources.xml および jms.xml です。

複数の OC4J プロセス全体で一貫性のある構成にするには、Oracle Application Server の複数の JVM 機能を使用します。この機能を使用すると、単一の OC4J 構成が同時に実行される JVM インスタンスの数 (n) を設定できます。その結果、一貫性のある単一の構成セットから、同じ OC4J インスタンスを実行している n 個の OC4J プロセスが起動されます。単一の構成セットの任意のファイルを変更すると、JVM セットの番号に対応して、起動されたすべての OC4J プロセスが更新されます。OC4J インスタンスごとに JVM の番号を構成する方法は、8-32 ページの「複数の JVM での OC4J インスタンスの稼働」で説明されています。

アプリケーションの構成を変更するためにクラスタ全体を手動で構成する必要がある場合に備え、表 8-2 に、これらの特定の機能だけでなく、すべての構成ファイルとその使用方法を説明します。

表 8-2 クラスタでレプリケートする構成ファイル

ファイル	ORACLE_HOME での場所	レプリケートまたは管理対象データ
application.xml	/j2ee/instance /config	<ul style="list-style-type: none"> ■ すべてのデプロイ済アプリケーションにデフォルトで適用される構成データに加えられた変更。 ■ データソースまたは他の共有リソースへの参照。 ■ <imported-shared-libraries> 要素内の共有ライブラリ定義。 カスタム共有ライブラリのコード・ソースは、OC4J ホストにインストールする必要があり、ライブラリは OC4J インスタンスの server.xml ファイルで参照する必要があります。
data-sources.xml	/j2ee/instance /config	<ul style="list-style-type: none"> ■ デプロイ済アプリケーションで使用できるようにする必要があるカスタム・データソースの構成データ。
default-web-site.xml	/j2ee/instance /config	<ul style="list-style-type: none"> ■ 該当する場合は、セキュアな Web サイト (HTTPS) 構成。

表 8-2 クラスタでレプリケートする構成ファイル (続き)

ファイル	ORACLE_HOME での場所	レプリケートまたは管理対象データ
*-web-site.xml	/j2ee/instance /config	<ul style="list-style-type: none"> OC4J インスタンスで使用される追加 Web サイトの構成ファイルを指定された場所にコピーします。 <p>Web サイト構成ファイルへの参照は、13-8 ページの「OC4J での Web サイトの新規作成」で説明しているように、opmn.xml または server.xml に追加する必要があります。</p>
global-web-application.xml	/j2ee/instance /config	<ul style="list-style-type: none"> 新しいサーブレット定義またはサーブレット構成の変更 (<init-param> の変更など)。 変更された JSP コンテナのプロパティ。詳細は、『Oracle Containers for J2EE JavaServer Pages 開発者ガイド』を参照してください。
j2ee-logging.xml	/j2ee/instance /config	<ul style="list-style-type: none"> ロギング構成の変更。
javacache.xml	/j2ee/instance /config	<ul style="list-style-type: none"> Java キャッシュ構成の変更。
jazn.xml	/j2ee/instance /config	<ul style="list-style-type: none"> XML ベースまたはLDAP ベースのセキュリティ・プロバイダの構成。jazn.xml ファイルの詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。
jazn-data.xml	/j2ee/instance /application-deployments/appName	<ul style="list-style-type: none"> XML ベースのプロバイダ構成を、このプロバイダを使用するすべてのアプリケーションに対して、指定された場所にレプリケートします。LDAP ベースのプロバイダを使用するアプリケーションには不要です。jazn-data.xml ファイルの詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。
jms.xml	/j2ee/instance /config	<ul style="list-style-type: none"> 接続先またはコネクション・ファクトリへの追加。
rmi.xml	/j2ee/instance /config	<ul style="list-style-type: none"> RMI 構成の変更 (ロギング構成など)。

クラスタの構成

この項では、次のクラスタリング・モデルを構成する手順について説明します。

- マルチキャストを使用した動的ノード検出の構成
- 静的検出サーバーの構成
- トポロジ間ゲートウェイの構成
- 静的ノードツーノード通信の構成

マルチキャストを使用した動的ノード検出の構成

動的ノード検出は、最も単純なクラスタリング構成です。このモデルでは、各 ONS ノードは、その存在を通知する単純なマルチキャスト・メッセージをブロードキャストし、クラスタ内のノードが相互に動的に検出できるようにします。

次のツールは、マルチキャスト検出を使用して OC4J インスタンスをクラスタに追加するために使用できます。

- `opmnctl`

このユーティリティには、マルチキャストの `port:address` で `opmn.xml` を更新するためのコマンドと、インスタンスをクラスタに追加するために必要な Web サイト構成データが含まれます。詳細は、8-15 ページの「[opmnctl によるマルチキャスト検出の構成](#)」を参照してください。

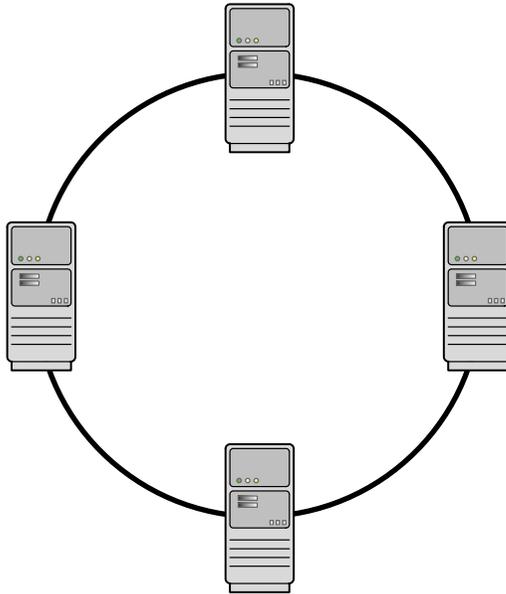
- `opmnassociate`

このユーティリティは、OC4J インスタンスをクラスタに追加するためのワンステップ・ソリューションを提供します。詳細は、8-16 ページの「[opmnassociate によるマルチキャスト検出の構成](#)」を参照してください。

注意： Oracle Application Server インスタンスは、インストール時にクラスタに追加できます。

各 ONS は、クラスタ・トポロジのマップを独自に保持します。新しい ONS がクラスタに追加されると、既存の ONS は、新しいノードとその接続情報をそれぞれのマップに追加します。同時に、新しい ONS はすべての既存のノードをそのマップに追加します。また、ONS がクラスタから削除されると、残りのノードのマップにこの変更が反映されます。

図 8-2 動的検出モデル



マルチキャスト・メッセージは異なるネットワーク構成によって制限されるため、動的ノード検出は、同じサブネット上にある ONS ノードに対してのみのオプションとなる場合があります。ただし、動的ノード検出を使用する複数のサブネットは、ゲートウェイ・サーバーを使用して接続できます。詳細は、8-19 ページの「トポロジ間ゲートウェイの構成」を参照してください。

注意：

- トポロジ内のすべてのノードは、同じマルチキャスト・アドレスおよびポートを使用するように構成する必要があります。
- マルチキャスト・アドレスは、有効なアドレス範囲（224.0.1.0 ~ 239.255.255.255）内にある必要があります。

マルチキャスト・アドレスおよびポートの割当ては、他のアプリケーションとの競合を回避するために、システム管理スタッフが管理することをお勧めします。

動的検出の構成は、トポロジ内の各 Oracle Application Server インスタンスにある opmn.xml ファイルの <topology> 要素の <discover> サブ要素内に設定されます。新しいノードをクラスタに追加するには、この要素を opmn.xml ファイルに追加するだけです。クラスタからノードを削除するには、この要素を削除します。

マルチキャスト IP アドレスおよびポートは、list 属性の値として設定します。IP アドレスの前のアスタリスク (*) は、指定された値がマルチキャスト・アドレスであることを OPMN に伝えるため、重要です。複数の値は、それぞれをカンマで区切って指定できます。

```
<opmn>
  <notification-server>
    <port ... />
    <ssl ... />
    <topology>
      <discover list="*225.0.0.20:8001"/>
    </topology>
  </notification-server>
  ...
</opmn>
```

注意: opmn.xml ファイルの変更を有効にするには、リロードする必要があります。対象となるノードで次のコマンドを実行し、opmn.xml をリロードします。

```
opmnctl reload
```

このコマンドは、OPMN 管理コンポーネント (Oracle HTTP Server、OC4J およびデプロイ済アプリケーションなど) には影響を与えません。

opmnctl によるマルチキャスト検出の構成

OPMN コマンドライン・ツール opmnctl では、新しい config topology コマンドをサポートします。このコマンドを使用すると、opmn.xml 内でマルチキャストの <discovery> エントリを指定、更新または削除できます。

opmnctl ツールは、各ノードの ORACLE_HOME/opmn/bin ディレクトリにインストールされています。このツールは各ノードで個別に実行する必要があり、そのノードの opmn.xml ファイルのみが更新されます。

OPMN 管理スタンドアロン OC4J インスタンスの追加に関する注意:

OPMN 管理の OC4J インスタンスには、Oracle HTTP Server (「J2EE Server およびプロセス管理」インストール・タイプ) は含まれません。デフォルト Web サイトは、デフォルトで HTTP リクエストをリスニングするように構成されます。

インスタンスをクラスタに追加する際に、Apache JServ Protocol (AJP) を使用するように Web サイトを構成する必要があります。この変更は、OC4J インスタンスが Oracle HTTP Server からのリクエストを受信し応答できるようにするために必要です。

デフォルトの Web サイトで使用されるプロトコルおよびポートは、Application Server Control の「ランタイム・ポート」ページを使用して構成できます。opmnctl config port update コマンドは、opmn.xml に定義されたデフォルト Web サイト構成の変更にも使用できます。詳細は、13-5 ページの「[opmnctl による Web サイトの構成](#)」を参照してください。

検出データの挿入または更新

update コマンドは、<discover> 要素を指定された値で挿入または更新します。構文は次のとおりです。

```
opmnctl config topology update discover="*multicastAddress:multicastPort"
```

次に例を示します。

```
opmnctl config topology update discover="*225.0.0.20:8001"
```

```
opmnctl reload
```

検出データの削除

delete コマンドは、<discover> 要素を opmn.xml から削除し、事実上ノードをクラスタから削除します。<topology> 要素に他のサブ要素がない場合は、同様に削除されます。

```
opmnctl config topology delete discover
```

```
opmnctl reload
```

opmnassociate によるマルチキャスト検出の構成

opmnassociate ユーティリティは、マルチキャスト検出を使用してデフォルトの home OC4J インスタンスをクラスタに追加します。このユーティリティは、次の手順を実行します。

1. opmn.xml の <discover> 要素に、指定されたマルチキャスト・アドレスおよびポートを挿入または更新します。
2. opmn.xml の対応する <port> 要素を変更して、Apache JServ Protocol (AJP) を使用して Oracle HTTP Server からのリクエストを受信し応答するようにデフォルト Web サイトを構成します。
3. OPMN を再起動し、新しい構成をランタイム環境にロードします。

opmnassociate ツールは、各 OC4J インスタンスの `ORACLE_HOME/bin` ディレクトリにインストールされています。このツールは各インスタンスで個別に実行する必要があり、そのインスタンスの `opmn.xml` ファイルのみが更新されます。

Linux または UNIX 環境での構文は次のとおりです。

```
opmnassociate.sh "*multicastAddress:multicastPort" [-restart]
```

次に例を示します。

```
opmnassociate.sh "*225.0.0.20:8001" -restart
```

Windows 環境での構文は次のとおりです。

```
opmnassociate "*multicastAddress:multicastPort" [-restart]
```

次に例を示します。

```
opmnassociate "*225.0.0.20:8001" -restart
```

IP アドレスの前にアスタリスク (*) を付ける必要があります。

注意： opmnassociate ユーティリティは、デフォルトの home OC4J インスタンスをクラスタに追加する場合にのみ使用できます。home2 などの別の OC4J インスタンスを追加する場合は、8-15 ページの「[opmnctl によるマルチキャスト検出の構成](#)」に説明されているように、opmnctl ユーティリティを使用します。一般的に、opmnassociate は、マルチキャスト検出を構成するために設定されている、より完全な opmnctl コマンドを簡略化したものです。マルチキャスト検出の構成には、opmnctl を使用することをお勧めします。

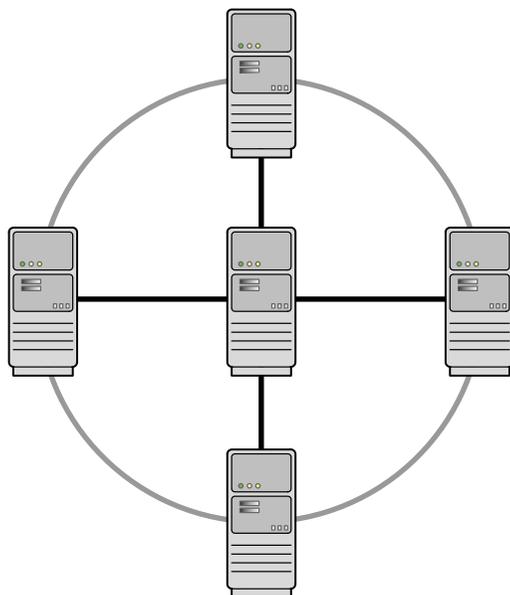
静的検出サーバーの構成

この構成は peer-to-peer クラスタリング・モデルに似ており、同じクラスタ内に、静的ハブまたは検出サーバーとして機能するように構成された 1 つ以上の ONS ノードがあります。

クラスタ内の各 ONS ノードは、クラスタ・トポロジのマップを保持する検出サーバーと接続を確立します。検出サーバーは、接続ノードに最新のトポロジ・マップを提供し、接続ノードがクラスタ内の他の ONS ノードと通信できるようにします。

opmnctl を使用すると、静的検出サーバーへの接続を構成できます。詳細は、8-18 ページの「[opmnctl による静的検出サーバー接続の構成](#)」を参照してください。

図 8-3 静的検出サーバー・モデル



クラスタ内の各静的ハブ・ノードにある opmn.xml ファイルの <discover> 要素内に、検出サーバーの TCP/IP 接続情報を設定します。次に例を示します。

```
<opmn>
  <notification-server>
    <port ... />
    <ssl ... />
    <topology>
      <discover list="node1.company.com:6200"/>
    </topology>
  </notification-server>
  ...
</opmn>
```

必須情報は次のとおりです。

- 静的検出サーバーのホスト名または IP アドレス
- OPMN の remote ポート。次に示すように、静的サーバーにインストールされている opmn.xml ファイル内の <port> 要素に定義されています。

```
<port local="6100" remote="6200" request="6003"/>
```

注意: opmn.xml ファイルは、OPMN ランタイムで変更を有効にするためにリロードする必要があります。対象となるノードで次のコマンドを実行し、opmn.xml をリロードします。

```
opmnctl reload
```

このコマンドは、OPMN 管理コンポーネント (Oracle HTTP Server、OC4J およびデプロイ済アプリケーションなど) には影響を与えません。

opmnctl による静的検出サーバー接続の構成

OPMN コマンドライン・ツール opmnctl では、新しい config topology コマンドをサポートします。このコマンドを使用すると、opmn.xml 内で <discover> エントリを指定、更新または削除できます。

opmnctl ツールは、各ノードの `ORACLE_HOME/opmn/bin` ディレクトリにインストールされています。このツールは各ノードで個別に実行する必要があり、そのノードの opmn.xml ファイルのみが更新されます。

検出データの挿入または更新

update コマンドは、<discover> 要素を指定された値で挿入または更新します。構文は次のとおりです。

```
opmnctl config topology update discover="serverHost:opmnRemotePort"
```

次に例を示します。

```
opmnctl config topology update discover="node.company.com:6200"
```

```
opmnctl reload
```

検出データの削除

delete コマンドは、<discover> 要素を opmn.xml から削除し、事実上ノードをクラスタから削除します。<topology> 要素に他のサブ要素がない場合は、同様に削除されます。

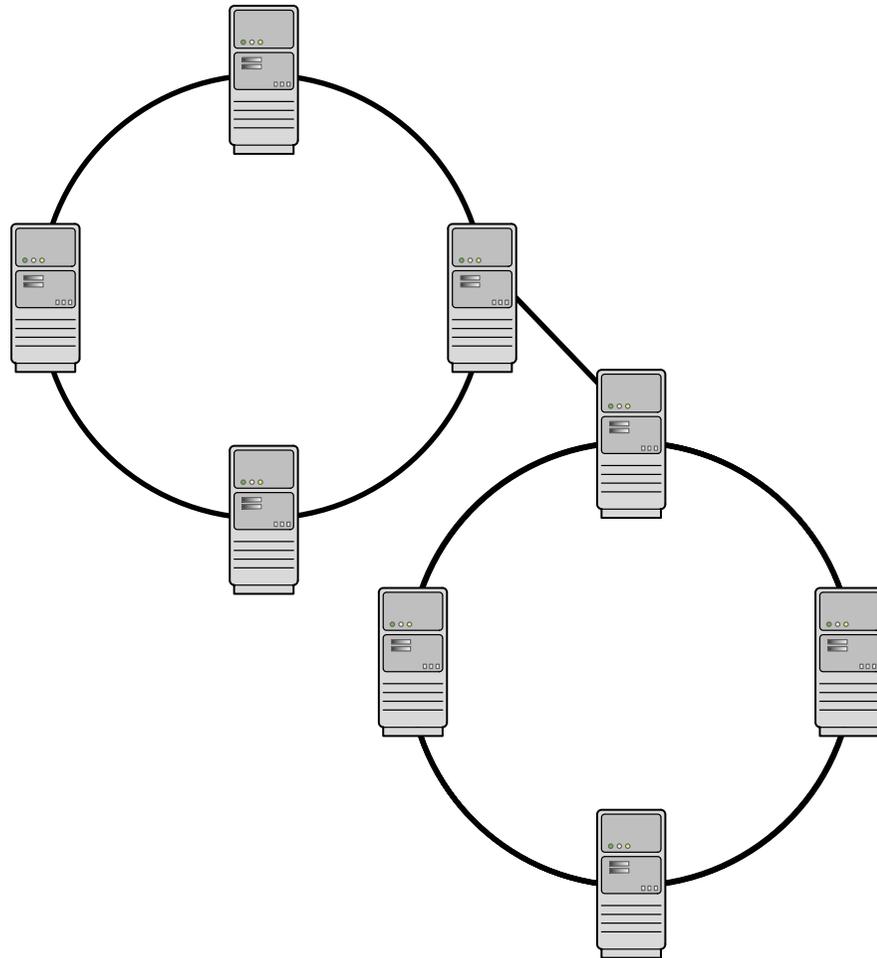
```
opmnctl config topology delete discover
```

```
opmnctl reload
```

トポロジ間ゲートウェイの構成

クラスタ・トポロジが異なるサブネット上にある、またはファイアウォールや物理的な場所によって分離されている状況では、特定の ONS ノードをゲートウェイとして構成し、ONS 通知を異なるトポロジ間で送信できるようにすることができます。

図 8-4 ゲートウェイ・サーバーを使用したトポロジの接続



このモデルでは、それぞれの分離されたトポロジ内の ONS ノードはゲートウェイ・サーバーとして構成され、クラスタへのエントリ・ポイントとなります。ゲートウェイ構成は、`<topology>` 要素の `<gateway>` サブ要素内に指定されます。

ソース・ゲートウェイ・ノードとその接続先の各ターゲット・ノードのホストおよびポートは、`list` 属性の値として設定します。ノードがリストされる順序は重要ではありません。

- ノードごとに、サーバーのホスト名または IP アドレスと、OPMN の `remote` ポートを指定します。このポートは、次に示すように、静的サーバーにインストールされた `opmn.xml` ファイルの `<port>` 要素に定義されています。

```
<port local="6100" remote="6200" request="6003"/>
```

- 各ノードのデータをアンパサンド (&) で区切ります。
- ノードのリストの最後に / を 1 つ指定します。

次の例に、ゲートウェイ・ノード node2 および node3 と接続する node1 に対する opmn.xml の構成を示します。同じ構成をこれらのゲートウェイ・ノードのそれぞれに設定できます。リストの最後の / に注意してください。

```
<opmn>
  <notification-server>
    <port ... />
    <ssl ... />
    <topology>
      <discover list="*224.0.0.37:8205"/>
      <gateway list="node1.com:6201&node2.com:6202&node3.com:6203"/>
    </topology>
  </notification-server>
  ...
</opmn>
```

<gateway> 要素の他に、<topology> 要素に <discover> 要素があることに注意してください。この要素には、ノード独自のクラスタ内で動的検出に使用されるマルチキャスト・アドレスおよびポートが指定されています。

あるいは、前述の例の <topology> 要素全体を、クラスタ・トポロジ内のすべてのノードにある opmn.xml ファイルにコピーできます。node1 のみが <gateway> 構成を利用します。つまり、他のノードによって無視されます。

構成を単純化するには、すべてのゲートウェイ・ノード（ソースおよびターゲット）の接続データを <gateway> サブ要素に設定した後、この要素を各ゲートウェイ・ノードにある opmn.xml ファイルにコピーします。ここでも、ノードの順序は重要ではありません。各ノードはリストの自身のエントリを無視するだけです。

注意： opmn.xml ファイルは、OPMN ランタイムで変更を有効にするためにリロードする必要があります。対象となるノードで次のコマンドを実行し、opmn.xml をリロードします。

```
opmnctl reload
```

このコマンドは、OPMN 管理コンポーネント（Oracle HTTP Server、OC4J およびデプロイ済アプリケーションなど）には影響を与えません。

ネットワーク接続の有無にかかわらず動作するマシンの構成

localhost を使用する単一のマシンで作業している場合は、<notification-server> 要素の <ipaddr> サブ要素に IP アドレスを追加し、クラスタの <port> 要素に定義されているように localhost の OPMN リモート・ポートを参照するよう、<discover> 要素に検出リストを明示的に設定します。次にこの構成の例を示します。

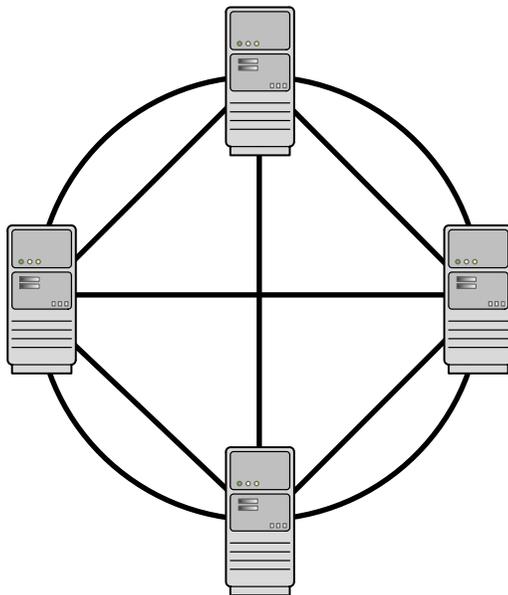
```
<notification-server>
  <ipaddr remote="127.0.0.1" request="127.0.0.1"/>
  <port local="6101" remote="6201" request="6004"/>
  <ssl enabled="true"
wallet-file="$ORACLE_HOME¥opmn¥conf¥ssl.wlt¥default"/>
  <topology>
    <discover list="localhost:6201"/>
  </topology>
</notification-server>
```

localhost IP アドレス 127.0.0.1 を指定すると、マシンはネットワークに接続していても接続してなくても動作します。

静的ノードツリーノード通信の構成

この静的構成モデルは、基本的には Oracle Application Server 10g リリース 1 (10.1.2) および 9i リリース 1 (9.0.4) で使用されたものと同じ方式です。このモデルは、主にこれらの旧リリースとの下位互換性を提供するためにサポートされています。

図 8-5 静的ノードツリーノード・モデル



この構成では、クラスタ内の各ノードのホスト・アドレスおよび ONS リモート・リスナー・ポートを含むノード・リストを指定します。Oracle Application Server リリース 3 (10.1.3.0.0) より前のリリースでは、ONS 構成データが `opmn.xml` に統合される場合、この構成は `ons.conf` 構成ファイルに設定されていました。

クラスタ内の各ノードのホスト・アドレスおよび ONS リモート・リスナー・ポート (`<notification-server>` の `<port>` サブ要素内に指定) を `<nodes>` サブ要素内に定義します。各ノードはカンマで区切ります。

次に例を示します。

```
<opmn>
  <notification-server>
    <port local="6101" remote="6202" request="6004"/>
    <ssl ... />
    <topology>
      <nodes list="node1-sun:6201,node2-sun:6202"/>
    </topology>
  </notification-server>
  ...
</opmn>
```

クラスタ内の各ノードに同じリストを指定します。各 ONS インスタンスは、リストで自身を識別し、そのエントリを無視します。

注意: opmn.xml ファイルは、OPMN ランタイムで変更を有効にするためにリロードする必要があります。対象となるノードで次のコマンドを実行し、opmn.xml をリロードします。

```
opmnctl reload
```

このコマンドは、OPMN 管理コンポーネント (Oracle HTTP Server、OC4J およびデプロイ済アプリケーションなど) には影響を与えません。

クラスタのステータスの表示

opmnctl または Application Server Control を使用して、クラスタ内の Oracle Application Server コンポーネントの現行のステータスを表示できます。

- [opmnctl によるクラスタのステータスの表示](#)
- [Application Server Control でのクラスタのステータスの表示](#)

opmnctl によるクラスタのステータスの表示

クラスタ内の Oracle Application Server ノードで opmnctl を使用して、クラスタのステータスをチェックできます。

```
opmnctl @cluster status
```

出力には、クラスタ内のアクティブな各 Oracle Application Server インスタンスにインストールされたコンポーネントのステータスが示されます。

```
Processes in Instance: AppSrv1.comp1.yourcompany.com
```

ias-component	process-type	pid	status
OC4JGroup:COLORS	OC4J:home	26880	Alive
OC4JGroup:COLORS	OC4J:oc4j_soa	26256	Alive
HTTP_Server	HTTP_Server	26879	Alive

```
Processes in Instance: AppSrv2.comp2.yourcompany.com
```

ias-component	process-type	pid	status
OC4JGroup:COLORS	OC4J:home	26094	Alive
OC4JGroup:COLORS	OC4J:oc4j_soa	N/A	Down
HTTP_Server	HTTP_Server	26093	Alive

Application Server Control でのクラスタのステータスの表示

Application Server Control のホームページの左上隅にある「[クラスタ・トポロジ](#)」リンクをクリックします。

結果ページには、クラスタ内のアクティブな各 Oracle Application Server インスタンスとともに、各インスタンス上のアクティブなアプリケーションが表示されます。このページから、クラスタ内のインスタンスまたはデプロイ済アプリケーションにアクセスできます。

Oracle HTTP Server を使用したルーティングおよびロード・バランシングの構成

ロード・バランシングという用語は、クラスタ内のサーバー・インスタンスに受信サービス・リクエストを分散するプロセスを指します。Oracle Application Server クラスタでのロード・バランシングは、Oracle HTTP Server の `mod_oc4j` モジュールによって管理されます。この構成では、Oracle HTTP Server インスタンスは受信 HTTP および HTTPS リクエストに対するフロントエンド・リスナーとして機能します。`mod_oc4j` はリクエストされたアプリケーションを提供する OC4J インスタンスに各リクエストをルーティングします。

Oracle Application Server リリース 3 (10.1.3) では、ロード・バランシングは、同じクラスタに属する Oracle Application Server インスタンスに対しては完全に動的です。Oracle HTTP Server または `mod_oc4j` をさらに構成する必要はありません。

- 動的 OC4J インスタンス検出

Oracle HTTP Server インスタンスではクラスタ内の各 OC4J インスタンスとそこにデプロイされたアプリケーションに関する情報が動的に更新され、Oracle HTTP Server が適切なインスタンスにリクエストをルーティングできるようになります。

詳細は、8-29 ページの「[アプリケーション・マウント・ポイントの動的構成の有効化](#)」を参照してください。

- 動的ルーティング

新しいリリースでは、**ルーティング ID** 方式をサポートしています。この方式を使用すると、Oracle HTTP Server インスタンスがリクエストを転送する OC4J インスタンスを必要に応じて制御できます。基本的には、特定の Oracle HTTP Server インスタンスからのリクエストを処理する一連の OC4J インスタンスを制御できるようになります。すべての Oracle HTTP Server および OC4J インスタンスは、インストール時にデフォルトのルーティング ID を使用するように構成されるため、構成は不要です。

詳細は、8-23 ページの「[Web サーバーのルーティング ID を使用した OC4J リクエスト・ルーティングの制御](#)」を参照してください。

クラスタ内の様々な Oracle HTTP Server および OC4J インスタンスの ONS サーバーが、この章で説明しているクラスタリング構成方式のいずれかを使用して接続されることが唯一の要件です。詳細は、8-13 ページの「[クラスタの構成](#)」を参照してください。

Web サーバーのルーティング ID を使用した OC4J リクエスト・ルーティングの制御

OPMN 管理のインストールでは、すべての Oracle HTTP Server および OC4J インスタンスにルーティング ID が割り当てられます。この ID は、`opmn.xml` に定義されています。Oracle HTTP Server インスタンスは、ルーティング ID を共有する OC4J インスタンスにのみ受信 Web リクエストをルーティングします。つまり、特定の Oracle HTTP Server インスタンスがリクエストをルーティングする一連の OC4J インスタンスを事実上定義できます。

デフォルトのルーティング ID がすべてのコンポーネント・インスタンスに割り当てられるため、インストールと同時に、クラスタ内のすべての Oracle HTTP Server インスタンスはリクエストをクラスタ内の OC4J インスタンスにルーティングできます。

一般的な Oracle Application Server クラスタでは、ユーザーからのリクエストは 1 つ以上の Oracle HTTP Server インスタンスによって受信され、それらのリクエストはクラスタ内にデプロイされたアプリケーションにルーティングされます。各アプリケーション・サーバー、OC4J インスタンス、グループおよびデプロイ済アプリケーションのルーティング ID により、Oracle HTTP Server がそれぞれのリクエストをルーティングする宛先が決定されます。

注意：アプリケーション・サーバー、コンポーネントまたは個々のアプリケーションのルーティング ID を変更すると、HTTP リクエストがデプロイ済アプリケーションに送信されなくなる場合があります。クラスタ内でそのアプリケーションの別のインスタンスが使用可能で、ルーティング ID が同一である場合以外は、この変更によってユーザーがアプリケーションを使用できなくなる可能性があります。

これ以降では、次の各項でルーティング ID の変更方法を説明します。

- [Application Server Control を使用したルーティング ID の変更](#)
- [opmn.xml ファイルのルーティング ID の変更](#)

AJP リクエストをリスニングする Web サイトの構成方法の詳細は、13-3 ページの「[Web サイト接続データの構成](#)」を参照してください。

Application Server Control を使用したルーティング ID の変更

Application Server Control を使用して、クラスタの各アプリケーションおよびコンポーネントに割り当てられたルーティング ID を変更または表示するには、次のようにします。

1. 「クラスタ・トポロジ」ページに移動します。
2. ページの「管理」セクションまでスクロールし、「**ルーティング ID 構成**」をクリックします。

「ルーティング ID 構成」ページには、クラスタ・トポロジ内のコンポーネントおよびアプリケーションの階層が表示されます。たとえば、アプリケーション・サーバーの「**拡張**」アイコンをクリックすると、そのアプリケーション・サーバー・インスタンス内のグループが表示されます。各グループ内には、グループを構成する OC4J インスタンスが表示されます。また、特定の OC4J インスタンスを開くと、その OC4J インスタンスにデプロイされているアプリケーションが表示されます。

デフォルトで、アプリケーション・サーバー・インスタンスにルーティング ID が割り当てられ、グループ、OC4J インスタンスおよびアプリケーションは、アプリケーション・サーバーのルーティング ID を継承します。特定のグループ、OC4J インスタンスまたはアプリケーションに別のルーティング ID を入力すると、アプリケーション・サーバーから継承されたルーティング ID は新しいルーティング ID で上書きされます。

クラスタ内で複数のアプリケーション・サーバー・インスタンスを管理している場合、ある OC4J インスタンスをメンバーとして含むグループがアプリケーション・サーバーごとに 1 つずつ表示されるため、同じグループが階層に複数表示されます。これは、「ルーティング ID 構成」ページの階層が、各アプリケーション・サーバーの Oracle ホーム・ディレクトリに格納されている Oracle Process Management and Notification (OPMN) ソフトウェアの構成ファイル (opmn.xml) に基づいているためです。そのため、グループのルーティング ID を変更する際には注意してください。特定の Oracle HTTP Server リクエストをグループ内の一部の OC4J インスタンスにのみルーティングする必要がある場合を除き、「ルーティング ID 構成」ページでは、グループのすべてのインスタンスに同じルーティング ID を割り当ててください。

opmn.xml ファイルのルーティング ID の変更

ルーティング ID は、opmn.xml の、id 属性が routing-id である <data> 要素に定義されます。<data> 要素エントリは、起動時にインスタンスに渡されるパラメータを指定する <category id="start-parameters"> のサブ要素です。インスタンスごとに設定されるデフォルトの routing-id 値は、g_rt_id です。

```
<category id="start-parameters">
  <data id="routing-id" value="g_rt_id"/>
</category>
```

デフォルトのルーティング ID が指定されている <data> 要素は、Oracle Application Server インスタンスの OPMN 構成データが指定されている <ias-instance> 要素内に設定されます。ルーティング ID はこのレベルに設定されるため、この <data> 要素に設定される routing-id 値は、Oracle Application Server インスタンス内にインストールされる Oracle HTTP Server および OC4J コンポーネントのすべてのインスタンスに適用されます。

```
<opmn>
  <process-manager>
    ...
    <ias-instance id="instance1" name="instance1">
      ...
      <environment>
        ...
      </environment>
      <module-data>
        <category id="start-parameters">
          <data id="routing-id" value="g_rt_id"/>
        </category>
      </module-data>
    </environment>
    <ias-component id="HTTP_Server">
      ...
    </ias-component>
    <ias-component id="default_group">
      ...
    </ias-component>
  </ias-instance>
</process-manager>
</opmn>
```

しかし、ルーティング ID は、コンポーネントの <category id="start-parameters"> 要素内に <data> 要素を追加して、個々の Oracle HTTP Server または OC4J インスタンス・レベルに設定できます。この値は、Oracle Application Server インスタンス・レベルに割り当てられたルーティング ID に優先します。

routing-id 属性の値には任意の文字列を指定できます。この識別子には特定の書式はありません。opmn.xml の次のエントリは、Oracle HTTP Server インスタンスのルーティング ID を設定します。

```
<opmn>
  <process-manager>
    ...
    <ias-instance id="instance1" name="instance1">
      ...
      <ias-component id="HTTP_Server">
        <environment>
          ...
        </environment>
        <process-type id="HTTP_Server" module-id="OHS">
          <module-data>
            <category id="start-parameters">
              <data id="start-mode" value="ssl-enabled"/>
              <data id="routing-id" value="group_b_id"/>
            </category>
          </module-data>
        </process-type>
      </ias-component>
    </ias-instance>
  </process-manager>
</opmn>
```

```

        <process-set id="HTTP_Server" numprocs="1"/>
    </process-type>
</ias-component>
</ias-instance>
</process-manager>
</opmn>
opmn.xml の次のエントリは、OC4J の home インスタンスのルーティング ID を設定します。
<opmn>
  <process-manager>
    ...
    <ias-instance id="instance1" name="instance1">
      ...
      <ias-component id="default_group">
        <environment>
        </environment>
        <process-type id="home" module-id="OC4J" status="enabled">
          <module-data>
            <category id="start-parameters">
              <data id="java-options" ... />
              <data id="routing-id" value="group_b_id"/>
            </category>
          </module-data>
          <process-set id="HTTP_Server" numprocs="1"/>
          <port id="default-web-site" range="12501-12600" protocol="ajp" />
          <port id="rmi" range="12401-12500"/>
          <port id="jms" range="12601-12700"/>
          <port id="rmis" range="12701-12800"/>
          <process-set id="default_group" numprocs="1"/>
        </process-type>
      </ias-component>
    </ias-instance>
  </process-manager>
</opmn>

```

mod_oc4j ロード・バランシング・オプションの設定

Oracle HTTP Server 内の mod_oc4j モジュールにより、OC4J プロセスにリクエストが委任されます。Oracle HTTP Server が OC4J を対象とした URL のリクエストを受信すると、Oracle HTTP Server により mod_oc4j モジュールにリクエストがルーティングされます。その後、このモジュールによりそのリクエストは OC4J プロセスにルーティングされます。OC4J プロセスが失敗すると、OPMN により障害が検出され、mod_oc4j は OC4J プロセスが再起動されるまで失敗した OC4J プロセスにリクエストを送信しません。

mod_oc4j を構成して、OC4J プロセスへのリクエストをロード・バランシングできます。Oracle HTTP Server では、mod_oc4j を介して、様々なロード・バランシング・ポリシーをサポートしています。ロード・バランシング・ポリシーには、フェイルオーバーおよび高可用性に加え、ネットワーク・トポロジおよびホスト・マシンの性能に応じてパフォーマンス上の利点もあります。

必要とするルーティングのタイプや複雑さに応じて、mod_oc4j に異なるロード・バランシング・ルーティング・アルゴリズムを指定できます。ステートレス・リクエストは、mod_oc4j.conf に指定されているアルゴリズムに基づいて使用可能な任意の宛先にルーティングされます。ステートフル HTTP リクエストは、セッション識別子を使用して前のリクエストを処理した OC4J プロセスに転送されます。ただし、mod_oc4j が OPMN との通信を通じてこのプロセスは使用できないと判断した場合は除きます。この場合、mod_oc4j は指定されたロード・バランシング・プロトコルに従って、使用可能な OC4J プロセスにリクエストを転送します。

デフォルトでは、すべての OC4J インスタンスの重み付けは同一であり（すべてのインスタンスの重み付けは 1）、`mod_oc4j` はラウンド・ロビン方式を使用してリクエストを転送する OC4J インスタンスを選択します。OC4J インスタンスの重み付けは、インスタンスが処理するリクエストの数を決定するために、トポロジ内のその他の使用可能な OC4J インスタンスの重み付けに対する比率として認識されます。リクエストが確立されたセッションに属する場合、リクエストは `mod_oc4j` により同じ OC4J インスタンスおよびセッションを開始したのと同じ OC4J プロセスに転送されます。

`mod_oc4j` のロード・バランシング・オプションでは、リクエストを送信する OC4J インスタンスを決定する際に、OC4J インスタンスで稼働している OC4J プロセスの数は考慮されません。インスタンスに構成された重み付けおよびその可用性に基づいて OC4J インスタンスが選択されます。

`mod_oc4j` のロード・バランシング・ポリシーを変更するには、`ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf` ファイルに `Oc4jSelectMethod` および `Oc4jRoutingWeight` ディレクティブを設定します。

1. 各 Oracle Application Server インスタンスの `mod_oc4j.conf` ファイルにある `<IfModule mod_oc4j.c>` セクションで、`Oc4jSelectMethod` ディレクティブに表 8-3 に示す値のいずれかを設定します。

`Oc4jSelectMethod` ディレクティブを `roundrobin:weighted` または `random:weighted` に設定した場合は、`Oc4jRoutingWeight` ディレクティブに重み付けを指定する必要もあります（次の手順を参照）。

ルーティング・アルゴリズムの選択の詳細は、8-28 ページの「[mod_oc4j のロード・バランシング・オプションの選択](#)」を参照してください。

表 8-3 Oc4jSelectMethod の値

値	説明
<code>roundrobin</code> (デフォルト)	<code>mod_oc4j</code> によりリストにトポロジのすべての OC4J プロセスが表示され、プロセスがリストから順番に選択されます。
<code>roundrobin:local</code>	<code>roundrobin</code> と似ていますが、リストにはローカル OC4J プロセスのみが含まれます。使用可能なローカル OC4J プロセスがない場合には、リモート OC4J プロセスが選択されます。
<code>roundrobin:weighted</code>	各インスタンスに構成されているルーティングの重み付けに基づいて、 <code>mod_oc4j</code> により合計のリクエスト負荷が各 OC4J インスタンスに分散されます。その後、ラウンド・ロビン方式でローカル・インスタンスから OC4J プロセスが選択されます。 重み付けは、 <code>Oc4jRoutingWeight</code> ディレクティブを使用して構成します。
<code>random</code>	<code>mod_oc4j</code> によりトポロジのすべての OC4J プロセスのリストから、OC4J プロセスがランダムに選択されます。
<code>random:local</code>	<code>random</code> と似ていますが、ローカル OC4J プロセスが優先されます。使用可能なローカル OC4J プロセスがない場合には、リモート OC4J プロセスが選択されます。
<code>random:weighted</code>	トポロジの各インスタンスに構成されている重み付けに基づいて、 <code>mod_oc4j</code> により OC4J プロセスが選択されます。 重み付けは、 <code>Oc4jRoutingWeight</code> ディレクティブを使用して構成します。
<code>metric</code>	プロセスがどの程度ビジーであるかを示すランタイム・メトリックに基づいて、 <code>mod_oc4j</code> によりリクエストがルーティングされます。
<code>metric:local</code>	<code>metric</code> と似ていますが、ローカル OC4J プロセスが優先されます。使用可能なローカル OC4J プロセスがない場合には、リモート OC4J プロセスにルーティングされます。

例:

```
Oc4jSelectMethod random:local
```

メトリックベースのロード・バランシングの設定方法の詳細は、『Oracle HTTP Server 管理者ガイド』を参照してください。

2. Oc4jSelectMethod ディレクティブに重み付けベースのメソッド (roundrobin:weighted または random:weighted) を設定した場合は、Oc4jRoutingWeight ディレクティブに重み付けを指定する必要もあります。

Oc4jRoutingWeight の構文は次のとおりです。

```
Oc4jRoutingWeight hostname weight
```

Oc4jRoutingWeight ディレクティブを設定しない場合は、デフォルトの重み付けである 1 が使用されます。

例: 3つのインスタンス (A、B および C) で構成されるトポロジがあり、B と C では A の 2 倍のリクエストを取得する必要がある場合には、B および C に次のディレクティブを設定します。

```
Oc4jSelectMethod roundrobin:weighted
Oc4jRoutingMethod hostB 2
Oc4jRoutingMethod hostC 2
```

デフォルト値は 1 であるため、hostA には Oc4jRoutingMethod を設定しなくてもかまいません。

3. 変更内容が反映されるよう、トポロジのすべてのインスタンス上の Oracle HTTP Server を再起動します。

```
> opmnctl @cluster restartproc ias-component=HTTP_Server
```

mod_oc4j のロード・バランシング・オプションの選択

mod_oc4j の使用するロード・バランシング・オプションを決定する際には、次のガイドラインに従います。

- 同じ Oracle ホームで Oracle HTTP Server および OC4J が稼働している同一のマシンのあるトポロジには、ローカル・アフィニティ・アルゴリズムを使用するラウンド・ロビンが適しています。この場合、同じマシン上のすべての OC4J プロセスが使用できないという極端な場合を除き、mod_oc4j を使用して別のマシンにリクエストをルーティングしても、Oracle HTTP Server にはほとんどメリットはありません。
- あるマシンのセットで Oracle HTTP Server が稼働し、別のセットでリクエストを処理する OC4J インスタンスが稼働している分散デプロイには、単純なラウンド・ロビンおよびメトリックベースのアルゴリズムが適しています。特定の設定でこの 2 つのいずれが適しているかを判断するには、それぞれ実験を行って結果を比較する必要があります。これが必要なのは、システムの動作および受信リクエストの配分によって結果が異なるためです。
- 異なる特徴を持った複数のノードで異なる Oracle Application Server インスタンスが稼働している異機種間のデプロイには、重み付けが設定されたラウンド・ロビン・アルゴリズムが適しています。各インスタンスに重み付けを設定するだけでなく、最大限の効果を得るために、各 Oracle Application Server インスタンスで稼働している OC4J プロセスの数もチューニングしてください。たとえば、重み付けが 4 のマシンでは重み付けが 1 のマシンの 4 倍のリクエストが取得されますが、重み付けが 4 のシステムで 4 倍の OC4J プロセスが稼働していることを確認する必要があります。
- メトリックベースのロード・バランシングは、CPU またはデータベース接続の数など、アプリケーションのパフォーマンスを占有するメトリックがほとんどない場合に便利です。

アプリケーション・マウント・ポイントの構成

受信リクエストをルーティングするために、Oracle HTTP Server では、リクエストに指定された URL とリクエストを処理する OC4J インスタンスをマップする、アプリケーション固有のマウント・ポイントのリストを使用します。この項では、マウント・ポイント作成に関する次の項目について説明します。

- [アプリケーション・マウント・ポイントの動的構成の有効化](#)
- [マウント・ポイント構成アルゴリズムの変更点](#)
- [マウント・ポイントの構成データの表示](#)

マウント・ポイントの構成の詳細は、『Oracle HTTP Server 管理者ガイド』を参照してください。

アプリケーション・マウント・ポイントの動的構成の有効化

旧リリースの Oracle Application Server では、アプリケーション・マウント・ポイントのリストは `mod_oc4j` 構成ファイルの `mod_oc4j.conf` で手動で管理する必要がありました。

現行リリースでは、新しいノードおよびアプリケーションがクラスタに対して追加または削除されると、マウント・ポイントのリストは動的に更新されます。ONS 通知を使用して、クラスタ内のすべての OC4J インスタンスは、それぞれのデプロイ済アプリケーションのマウント・ポイント・データを `mod_oc4j` に送信し、`mod_oc4j` によってこの情報は内部ルーティング表に追加されます。

この動的検出方式は、クラスタ化された Oracle Application Server インスタンスではデフォルトで有効になっているため、追加構成は不要です。

各 OC4J インスタンスによって Oracle HTTP Server に送信されるマウント・ポイント情報には次のものがあります。

- OC4J ホスト・アドレス
- OC4J ポート情報 (Apache JServ Protocol (AJP) リスナー・ポートなど)

この値は、ノードにある `opmn.xml` ファイルで AJP に割り当てられる最も低い値の使用可能なポートです。
- Web モジュール名

この値は、バインド先の `*-web-site.xml` 構成ファイルでモジュールに定義される `<web-app>` 要素の `name` 属性の値として定義されます。
- アプリケーションに定義されている Web コンテキストまたはルート・コンテキスト

この値は、`*-web-site.xml` 構成ファイルでモジュールに定義される `<web-app>` 要素の `root` 属性に設定されます。

注意： 動的に構成されるマウント・ポイントは、`mod_oc4j` 構成ファイル (`mod_oc4j.conf`) には書き込まれません。

新しいアプリケーションが OC4J インスタンスにデプロイされると、マウント・ポイント情報が Oracle HTTP Server に送信され、`mod_oc4j` は動的にアプリケーションを検出し、リクエストをルーティングできるようになります。

反対に、アプリケーションが OC4J インスタンスから停止または削除されると、`mod_oc4j` ルーティング表はアプリケーションが存在しないことを反映するように更新され、`mod_oc4j` はアプリケーション・インスタンスへのリクエストのルーティングを停止します。

8-30 ページの「[マウント・ポイント構成アルゴリズムの変更点](#)」で説明されているように、アプリケーションのマウント・ポイントは手動で構成することも可能です。マウント・ポイント・リストの表示の詳細は、8-31 ページの「[マウント・ポイントの構成データの表示](#)」を参照してください。マウント・ポイントの構成の詳細は、『Oracle HTTP Server 管理者ガイド』を参照してください。

マウント・ポイント構成アルゴリズムの変更点

動的マウント・ポイントの作成はデフォルトで有効になっていますが、手動で構成されたマウント・ポイントを引き続き使用することもできます。これは、旧リリースの Oracle Application Server でサポートされたデフォルトの方式です。

静的マウント・ポイントは、mod_oc4j 構成ファイルの mod_oc4j.conf に定義されます。このファイルは、ORACLE_HOME/Apache/Apache/conf ディレクトリにインストールされています。デフォルトでは、Oracle HTTP Server はアプリケーションのデプロイ時に動的マウント・ポイントを作成しますが、mod_oc4j.conf に定義される静的マウント・ポイントも有効とみなされます。

使用するマウント・ポイントの構成方式は、mod_oc4j.conf の Oc4jRoutingMode パラメータに指定されます。表 8-4 に、この変数の値を示します。マウント・ポイントの構成および mod_oc4j.conf の使用方法は、『Oracle HTTP Server 管理者ガイド』を参照してください。

注意： mod_oc4j 構成ファイルの Oc4jRoutingMode を Static に変更する場合には、Application Server Control へのアクセス権を失わないように、mod_oc4j.conf に次の構成を追加する必要があります。

```
Oc4jMount /em/* home
Oc4jMount /em home
```

表 8-4 Oc4jRoutingMode の値

値	説明
Dynamic	動的に構成されたマウント・ポイントのみが使用されます。静的マウント・ポイントは無視されます。
Static	mod_oc4j.conf に定義される、手動で構成された静的マウント・ポイントのみが使用されます。動的マウント・ポイントは新しいアプリケーションに対して作成されません。
DynamicOverride	動的マウント・ポイントと静的マウント・ポイントの両方が使用されます。競合する場合は、動的に構成されたマウント・ポイントが使用されます。
StaticOverride	動的マウント・ポイントと静的マウント・ポイントの両方が使用されますが、競合する場合は、手動で構成されたマウント・ポイントが使用されます。 デフォルトで mod_oc4j.conf に定義されていなくても、この値がデフォルト・モードです。

次に示す mod_oc4j.conf の例では、DynamicOverride モードを有効にしています。競合が発生した場合、このモードでは、指定された動的マウント・ポイントが静的マウント・ポイントに優先します。

```
#####
# Oracle iAS mod_oc4j configuration file: mod_oc4j.conf #
#####

LoadModule oc4j_module libexec/mod_oc4j.so
Oc4jRoutingMode DynamicOverride
<IfModule mod_oc4j.c>
  <Location /oc4j-service>
    SetHandler oc4j-service-handler
  </Location>
  Oc4jMount /j2ee/*
  Oc4jMount /webapp home
  Oc4jMount /webapp/* home
  Oc4jMount /cabo home
  Oc4jMount /cabo/* home
  Oc4jMount /stressH home
```

```
Oc4jMount /stressH/* home
</IfModule>
```

マウント・ポイントの構成データの表示

マウント・ポイントの構成データを Oracle HTTP Server ホストで生成される Web ページに出力するように、Oracle HTTP Server を構成できます。

次のエントリを、Oracle HTTP Server ホスト・マシンにある Oracle HTTP Server 構成ファイル `httpd.conf` に追加します。このファイルは、`ORACLE_HOME/Apache/Apache/conf` にインストールされています。

```
<IfModule mod_oc4j.c>
    Oc4jSet StatusUri /oc4j-status
</IfModule>
```

ここで、`/oc4j-status` コンテキスト URI を Oracle HTTP Server サーバーの URL の最後に追加すると、マウント・ポイント・データを表示できます。

```
http://ohsHost:ajpPort/oc4j-status
```

次に例を示します。

```
http://node1.company.com:7777/oc4j-status
```

次に、結果の Web ページにコメントとともに表示されるサンプルの出力を示します。

```
hostname          : node1.company.com
local instance    : node1.company.com
select method     : Round-Robin
select affinity   : None
# OHS routing configuration
routing mode      : Static-Dynamic
routing ID        : g_rt_id

OC4J Dynamic routing
# Applications using dynamic routing

# 'ascontrol' application
application       : ascontrol
context           : /em
process (Jgroup) : 0

# 'demos' application
application       : demos
context           : /ojspdemos/jst1, /ojspdemos
process (Jgroup) : 0 (demos)

OC4J Process List

process,ias instance,host,port,status
0 : home.node1.company.com, node1.company.com, 12502, ALIVE
1 : home.node1.company.com, node1.company.com, 12501, ALIVE
2 : home.node1.company.com, node1.company.com, 12503, ALIVE
```

複数の JVM での OC4J インスタンスの稼働

OC4J は、標準 Java Development Kit (JDK) の Java 仮想マシン (JVM) で稼働します。各 OC4J インスタンスはデフォルトで 1 つの JVM を使用しますが、図 8-6 に示すように、複数の JVM で稼働するように構成できます。OC4J インスタンスを複数の JVM で稼働するように構成すると、インスタンスは基本的に複数のプロセスで稼働します。これにより、パフォーマンスが向上し、デプロイ済のアプリケーションに一定レベルのフォルト・トレランスが提供されます。

図 8-6 複数の JVM で稼働する OC4J インスタンス



この図には、クラスタ内の Oracle Application Server インスタンスの 1 つである OC4J_home という名前の OC4J インスタンスから実行するよう構成されている 4 つのプロセスを示します。

注意：通常、home という名前の OC4J インスタンスは、複数プロセス・モデルでの実行には適さない Application Server Control アプリケーション ascontrol のホストであるため、複数のプロセスで実行するように構成することはできません。

複数のプロセスで実行される OC4J インスタンスで、EJB タイマーを使用するアプリケーションは実行できません。EJB タイマーがサポートされているのは、(opmn.xml 構成ファイルの <process-set> 要素が numprocs=1 の) 単一の JVM で実行される OC4J インスタンスのみです。

ただし、複数の JVM が効率的に稼働するには、追加のハードウェア・リソースが必要です。また、複数のプロセスが同じホストで稼働していて、そのホストが停止した場合には、すべての JVM プロセスが停止します。

複数のアプリケーション・サーバー・インスタンスをインストールおよび管理する場合には、それらのアプリケーション・サーバー・インスタンスを複数のホストにインストールできます。アプリケーション・サーバーをクラスタ化し、「クラスタ・トポロジ」ページ (またはコマンドラインや API) から OC4J グループを作成することで、アプリケーションのクラスタリングおよびロード・バランシングを利用することもできます。第 9 章「OC4J でのアプリケーションのクラスタリング」で説明されているように、アプリケーションのクラスタリングにより、各 JVM で稼働しているアプリケーションの異なるインスタンスに状態情報がレプリケートされます。

また、Oracle Application Server クラスタおよび OC4J グループにより、ハードウェアまたはネットワークの停止からさらに強固に保護されます。1 つのホストが停止しても、別のホストにデプロイされているアプリケーションは使用可能です。

注意： (ascontrol アプリケーションとして表される) Application Server Control は、複数の JVM で稼働している OC4J インスタンスでは稼働できません。アクティブな Application Server Control インスタンスのホストである OC4J インスタンスには、複数の JVM を構成しないでください。

OC4J インスタンス用の追加の JVM の作成

デフォルトで、各 OC4J インスタンスは 1 つの JVM を使用します。ただし、各 JVM にインスタンスのコピーが存在する状態で、OC4J インスタンスが複数の JVM で稼働するように構成できます。Application Server Control の「サーバー・プロパティ」ページを使用するか、opmn.xml ファイルで OC4J インスタンスの numprocs 属性を直接設定することにより、OC4J インスタンスに追加の JVM を作成できます。

注意： opmn.xml ファイルで OC4J インスタンスの numprocs 属性が 1 (n) より大きい場合、その属性を Application Server Control を使用して設定するかファイルを編集して設定するかにかかわらず、Oracle Application Server は n 個の別々の物理 JVM プロセスを開始します。各プロセスは関連する OC4J インスタンス (すべてのデプロイ済アプリケーションを含む) のコピーの実行専用です。この値を設定する際には、物理ハードウェア・リソースも考慮に入れてください。

Application Server Control を使用して OC4J インスタンスに追加の JVM を作成する方法

Application Server Control の「サーバー・プロパティ」ページで、OC4J インスタンスに 1 つ以上の JVM を追加できます。

Application Server Control を使用して OC4J インスタンスに追加の JVM を作成する手順：

1. OC4J ホームページに移動して「管理」をクリックし、OC4J の「管理」ページを表示します。このページには、OC4J インスタンスに対して実行可能な様々な管理タスクがリストされた表が表示されています。
2. 「管理」ページで、「サーバー・プロパティ」を選択し、OC4J の「サーバー・プロパティ」ページを表示します。
3. 「VM のプロセス数」フィールドに、OC4J インスタンスに構成する JVM の数を入力します。
4. 「適用」をクリックします。
5. 「クラスタ・トポロジ」ページまたは OC4J ホームページから OC4J インスタンスを再起動します。

opmn.xml ファイルの OC4J インスタンスに追加の JVM を作成する方法

デフォルトで、各 OC4J インスタンスは 1 つの JVM を使用します。ただし、複数の JVM で稼働するように OC4J インスタンスを構成できます。opmn.xml ファイルの OC4J インスタンスに関する構成で <process-set> 要素の numprocs 属性を設定することで、1 つ以上の JVM を追加できます。

opmn.xml ファイルの OC4J インスタンスに追加の JVM を作成する手順：

1. opmn.xml ファイルを編集します。
2. OC4J 構成の <process-set> 要素で、numprocs 属性の値を OC4J を実行する JVM の数に設定します。

例 8-1 に、opmn.xml の numprocs の設定を示します。

例 8-1 opmn.xml の OC4J の numprocs 属性

```
<opmn>
. . .
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value=" -Djava.awt.headless=true"/>
        <data id="java-bin" value="/jdk/bin"/>
        <data id="oc4j-options" value="-validateXML -verbosity 10"/>
      </category>
      <category id="stop-parameters">
        <data id="java-options" value="-Djava.awt.headless=true"/>
      </category>
    </module-data>
    <start timeout="600" retry="2"/>
    <stop timeout="120"/>
    <restart timeout="720" retry="2"/>
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="rmi" range="12401-12500"/>
    <port id="jms" range="12601-12700"/>
    <port id="rmis" range="12701-12800"/>
    <process-set id="default_group" numprocs="2"/>
  </process-type>
</ias-component>
</opmn>
```

3. opmn.xml ファイルを保存します。
4. OC4J インスタンスを再起動します。

複数の JVM の監視

複数の JVM を使用する場合には、現在のハードウェア・リソースで構成に対応できることを確認するために JVM のパフォーマンスを監視することが重要です。Application Server Control から、OC4J インスタンスに関連付けられている JVM のパフォーマンスを監視および比較できます。

次の各項では、Application Server Control を使用して JVM メトリックを監視する方法を説明します。

- [ダイナミック・モニタリング・サービス JVM メトリックの監視](#)
- [J2SE JVM 5.0 メトリックを監視するための jmxremote システム・プロパティの設定](#)
- [Oracle Application Server 環境の J2SE 5.0 JVM メトリックの監視](#)
- [スタンドアロン OC4J 環境の J2SE 5.0 JVM メトリックの監視](#)

Application Server Control から J2SE 5.0 の JVM メトリックを監視するには、JDK 5.0 (1.5) で OC4J を実行し、各 OC4J インスタンスに jmxremote システム・プロパティを設定して監視を有効にする必要があります。

ダイナミック・モニタリング・サービス JVM メトリックの監視

Oracle Application Server 環境で OC4J を実行している場合は、各 JVM の一連のダイナミック・モニタリング・サービス (DMS) メトリックを監視できます。これらのメトリックは、スタンドアロン OC4J 環境では使用できません。

Application Server Control を使用して Oracle Application Server 環境の DMS JVM メトリックを表示するには、次のようにします。

1. OC4J ホームページに移動します。
2. OC4J ホームページの「一般」セクションで、「仮想マシン」フィールドを探します。
3. OC4J インスタンスに JVM がいくつ構成されているかを示す数をクリックします。

「JVM メトリック」ページには、選択した OC4J インスタンスに構成されているすべての JVM の主要なメトリックのサマリーが表示されます。この表を使用して、複数の JVM のパフォーマンスを比較できます。

4. 詳細は、JVM の名前をクリックしてください。

OC4J の「JVM」ページには、JVM の状態を詳細に示す一連のグラフおよび数値メトリックが表示されます。「データの表示」リストからリフレッシュ間隔を選択します。その後、パフォーマンス・グラフで一定期間の変化を参照できます。

J2SE JVM 5.0 メトリックを監視するための jmxremote システム・プロパティの設定

J2SE JVM 5.0 メトリックを監視するために、Application Server Control、OC4J 起動オプション、また、OPMN 管理環境の場合は `opmn.xml` ファイルを使用して、jmxremote システム・プロパティを設定できます。

Application Server Control コンソールを使用した jmxremote システム・プロパティの設定

Application Server Control を使用して、各 OC4J インスタンスの JVM J2SE 5.0 メトリックの監視を有効化するには、次のようにします。

1. OC4J ホームページに移動して「管理」をクリックし、OC4J の「管理」ページを表示します。このページには、OC4J インスタンスに対して実行可能な様々な管理タスクがリストされた表が表示されています。
2. OC4J の「管理」ページで、「サーバー・プロパティ」を選択し、OC4J の「サーバー・プロパティ」ページを表示します。
3. ページの「コマンドライン・オプション」セクションまでスクロールして、「J2SE 5.0 プラットフォーム MBean の有効化」を選択します。
4. 「適用」をクリックして、変更内容を適用します。
5. 「クラスタ・トポロジ」ページに移動して OC4J インスタンスを選択し、「再起動」をクリックします。

OC4J 起動オプションでの jmxremote システム・プロパティの設定

OC4J のランタイム起動オプションとして次の文字列を指定することにより、JVM J2SE 5.0 メトリックの監視を有効化することもできます。

```
com.sun.management.jmxremote
```

OC4J のランタイム起動オプションの指定方法の詳細は、4-3 ページの「[起動時の OC4J ランタイム・オプションの設定](#)」を参照してください。

スタンドアロン環境で OC4J を稼働している場合は、OC4J java コマンドに次の引数を指定します。

```
java -Dcom.sun.management.jmxremote -jar oc4j.jar
```

opmn.xml ファイルへの jmxremote システム・プロパティの設定

OPMN 管理の Oracle Application Server 環境で OC4J を実行している場合には、次に示すように、opmn.xml ファイルに `-Dcom.sun.management.jmxremote` を指定します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-Dcom.sun.management.jmxremote"/>
        ...
      </category>
      ...
    </module-data>
    ...
  </process-type>
  ...
</ias-component>
```

Application Server Control を使用して J2SE 5.0 プラットフォーム MBeans を有効化すると、opmn.xml ファイルに `jmxremote` システム・プロパティが設定されます。この方法を使用した場合、opmn.xml ファイルに手動でプロパティを設定する必要はありません。

Oracle Application Server 環境の J2SE 5.0 JVM メトリックの監視

Application Server Control を使用して Oracle Application Server 環境の J2SE 5.0 JVM メトリックを表示するには、次のようにします。

1. OC4J ホームページの「一般」セクションで、「仮想マシン」フィールドを探します。
2. OC4J インスタンスに JVM がいくつ構成されているかを示す数をクリックします。
「JVM メトリック」ページが表示されます。
3. JVM の名前をクリックします。
OC4J の「JVM」ページが表示されます。
4. ページの「関連リンク」セクションまでスクロールし、「J2SE 5.0 メトリック」をクリックします。

スタンドアロン OC4J 環境の J2SE 5.0 JVM メトリックの監視

Application Server Control を使用してスタンドアロン OC4J 環境の J2SE 5.0 JVM メトリックを表示するには、次のようにします。

1. OC4J ホームページで、「パフォーマンス」をクリックして「OC4J のパフォーマンス」ページを表示します。
2. ページの「関連リンク」セクションまでスクロールし、「J2SE 5.0 メトリック」をクリックします。

OC4J でのアプリケーションのクラスタリング

この章では、OC4J 10g (10.1.3.4.0) に付属するアプリケーションのクラスタリング・フレームワークについて説明します。この章の内容は次のとおりです。

- [OC4J におけるアプリケーションのクラスタリングの概要](#)
- [アプリケーションのクラスタリングにおける旧リリースの OC4J との相違点](#)
- [アプリケーションのクラスタリングの構成](#)

OC4J におけるアプリケーションのクラスタリングの概要

OC4J には、開発および本番用にクラスタ化された環境を作成するための柔軟性のあるフレームワークが用意されています。**アプリケーション・クラスタ**は、2つ以上の OC4J インスタンスがホストとなる同じアプリケーション・セットです。OC4J アプリケーションのクラスタリング・フレームワークでは、次のことをサポートします。

- HTTP セッションまたはステートフル・セッション Enterprise JavaBean (SFSB) インスタンスに含まれるオブジェクトおよび値のレプリケーション
- マルチキャストまたは **peer-to-peer** 通信を使用したメモリー内レプリケーション、あるいはデータベースに対する状態データの永続性
- OC4J インスタンス間での受信リクエストのロード・バランシング
- アプリケーション・クラスタ内のアプリケーション間での透過的フェイルオーバー
- グローバル・サーバー・レベルまたはアプリケーション・レベルでの OC4J インスタンス内の構成

様々な新しいサブ要素を持つ新しい `<cluster>` 要素が、アプリケーションのクラスタリングを管理するための単一の方法となるように、XML スキーマ定義に追加されています。この要素とそのサブ要素の説明は、9-12 ページの「[<cluster> 要素の指定](#)」を参照してください。

アプリケーションのクラスタリングにおける旧リリースの OC4J との相違点

OC4J 10g (10.1.3) のアプリケーションのクラスタリング・フレームワークに含まれなくなった機能を次に示します。

サポートされなくなったアイランド

アイランドという概念は、旧リリースの OC4J におけるクラスタリング・フレームワークの一部でしたが、現行 OC4J ではサポートされていません。

旧リリースでは、アイランドとは基本的に、HTTP セッション・データがレプリケートされる Oracle Application Server クラスタ内の OC4J インスタンスのグループでした。アイランドは、クラスタ全体でデータをレプリケートしないことでオーバーヘッドを削減しましたが、構成および管理のオーバーヘッドを増加させました。また、アイランドは Web アプリケーションにのみ適用でき、EJB アプリケーションではアイランド構成を使用できませんでした。

OC4J 10g (10.1.3) では、`<cluster>` 要素の `write-quota` 属性を使用して、データをレプリケートするノードの数を事実上制限できます。この属性を使用すると、状態レプリケーションの範囲を制御できるようになります。

`write-quota` 属性の詳細は、9-6 ページの「[アプリケーションの状態データをレプリケートする JVM の数の管理](#)」および 9-12 ページの「[<cluster> 要素の指定](#)」を参照してください。

使用されなくなった loadbalancer.jar

`loadbalancer.jar` アーカイブは、旧リリースの OC4J ではロード・バランシング機能を提供していましたが推奨されず、現行リリースからは削除されています。

アプリケーションのクラスタリング固有の非推奨の XML 要素

次の XML 要素は、OC4J 10g (10.1.3.4.0) では推奨されていません。クラスタリングの構成に使用しないでください。

- OC4J 構成ファイル `server.xml` の `<cluster-config>` 要素
- Web サイト構成ファイル `*-web-site.xml` の `<web-site>` 要素の `cluster-island` 属性

新しい `<cluster>` 要素がすべてのアプリケーション・クラスタ管理に使用されるようになりました。

アプリケーションのクラスタリングの構成

アプリケーションのクラスタリングは、OC4J インスタンス内でクラスタ化される各アプリケーションの `orion-application.xml` ファイルに `<cluster>` 要素を追加すると有効になります。デプロイ済アプリケーションの場合、このファイルは、`ORACLE_HOME/j2ee/instance/application-deployments/applicationName` ディレクトリにあります。この要素とそのサブ要素の説明は、9-12 ページの「[<cluster> 要素の指定](#)」を参照してください。

この項の内容は次のとおりです。

- [アプリケーションのクラスタリングの有効化](#)
- [レプリケーション・ポリシーの設定](#)
- [アプリケーションの状態データをレプリケートする JVM の数の管理](#)
- [同期レプリケーションまたは非同期レプリケーションの使用](#)
- [マルチキャスト・レプリケーションの構成](#)
- [peer-to-peer レプリケーションの構成](#)
- [OPMN のステート・レプリケーション用ポートの指定](#)
- [データベース・レプリケーションの構成](#)
- [アプリケーションの JVM、OC4J インスタンスおよび Application Server インスタンスの確認](#)
- [クラスタリングの無効化](#)
- [<cluster> 要素の指定](#)

アプリケーションのクラスタリングの有効化

アプリケーションのクラスタリングは、OC4J インスタンス内で稼働しているすべてのアプリケーションに対してグローバルにもアプリケーション単位でも有効にできます。

- **すべてのアプリケーションに対するクラスタリングの有効化**

アプリケーションのクラスタリングは、default アプリケーションの構成ファイルである `ORACLE_HOME/j2ee/instance/config/application.xml` を使用して、OC4J インスタンスにデプロイされたすべてのアプリケーションに対してデフォルトで有効にできます。OC4J インスタンスにデプロイされたその他のすべてのアプリケーションは、アプリケーションのクラスタリングの構成を含めデフォルトのプロパティをこのアプリケーションから継承します。

- **特定のアプリケーションに対するクラスタリングの有効化**

アプリケーションのクラスタリングは、アプリケーション固有の `ORACLE_HOME/j2ee/instance/application-deployments/app_name/orion-application.xml` ファイルに定義されます。このファイルの設定は、グローバル構成にも親アプリケーションから継承される構成にも優先します。

注意：アプリケーションのクラスタリングは、Oracle Enterprise Manager 10g Application Server Control を使用してアプリケーションをデプロイする際に、デプロイ・タスクまたはデプロイ・プラン・エディタのいずれかを使用して構成することもできます。

詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

単一の OC4J インスタンスの特定アプリケーションの `orion-application.xml` ファイルへの変更内容は、Oracle Application Server クラスタ内のすべてのアプリケーションに対するその他の OC4J インスタンスの対応する XML ファイルにレプリケートする必要があります。詳細は、8-11 ページの「[クラスタでの変更のレプリケート](#)」を参照してください。

アプリケーション・レベルでは、OC4J インスタンスにアプリケーションがデプロイされる際に、デプロイ・プラン・エディタを使用してアプリケーションのクラスタリングを構成できます。デプロイ・プラン・エディタにより、各アプリケーションの `orion-application.xml` ファイルに値が設定されます。デプロイ・プラン・エディタの使用方法は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

重要：空の `<distributed />` タグを、アプリケーションのクラスタリングを使用するように構成されたアプリケーションの一部であるすべての Web モジュールの `web.xml` ファイルに追加する必要があります。デプロイ後、この J2EE 標準 Web モジュールのディスクリプタは、OC4J 内の `ORACLE_HOME/j2ee/instance/applications/app_name/web_module/WEB-INF` ディレクトリにあります。

レプリケーション・ポリシーの設定

レプリケーション・ポリシーは、HttpSession またはステートフル・セッション Bean の状態をレプリケートするタイミングと、レプリケート対象がすべての属性および変数の値か、変更された値のみかを定義します。レプリケーションは負荷の高いプロセスとなる可能性があり、データを頻繁にレプリケートしすぎると、サーバーのパフォーマンスに影響を与えることがあります。ただし、データをレプリケートする頻度が低すぎると、サーバーが停止した場合にデータが失われる可能性があります。

アプリケーション内のすべての Web モジュールおよび EJB モジュールに適用されるレプリケーション・ポリシーは、アプリケーションの `orion-application.xml` 構成ファイルの `<replication-policy>` 要素に指定されます。この要素の構文は次のとおりです。

```
<replication-policy trigger="onSetAttribute|onRequestEnd|onShutdown"
  scope="modifiedAttributes|allAttributes" />
```

- `trigger` 属性は、レプリケートするタイミングを指定します。JVM が突然終了した場合にデータが失われないようにデータを頻繁にレプリケートするため、デフォルトでは `onRequestEnd` ポリシーが適用されます。

`trigger` 属性の値の概要は、[表 9-1](#) を参照してください。

- `scope` 属性はレプリケート対象のデータ（すべての属性や変数の値、または変更された値のみのいずれか）を定義します。デフォルトでは、変更された HTTP セッション属性のみがレプリケートされます。ステートフル・セッション Bean の場合は、すべてのメンバー変数がレプリケートされます。

`scope` 属性の値の概要は、[表 9-2](#) を参照してください。

表 9-1 <replication-policy> の trigger 属性の値

trigger の値	HttpSession	ステートフル・セッション Bean
onSetAttribute	値の変更時に、HTTP セッション属性に加えられた各変更をレプリケートします。プログラムのな面から見ると、HttpSession オブジェクトで setAttribute () がコールされるたびにレプリケーションは発生します。 このオプションは、セッションが頻繁に変更されている場合、リソースが集中する可能性があります。	該当なし。
onRequestEnd (デフォルト)	HTTP セッション属性に加えられたすべての変更をキューに入れ、HTTP レスポンスが送信される直前にすべての変更をレプリケートします。	EJB メソッド・コールのたびに、Bean の現在の状態をレプリケートします。状態は頻繁にレプリケートされますが、信頼性は向上します。
onShutdown	JVM が ([Ctrl]+[C] など) で正常に終了するたびに、HTTP セッションの現在の状態をレプリケートします。状態は、ホストが突然終了した場合 (システム・クラッシュなど) にはレプリケートされません。 セッションの状態はまだレプリケートされていないため、JVM の終了と同時にすべてのセッション・データが一度にネットワーク上に送信され、ネットワーク・パフォーマンスに影響を与える可能性があります。このオプションは、JVM の停止に必要な時間を著しく増加させる可能性もあります。	JVM が正常に終了するたびに、Bean の現在の状態をレプリケートします。状態は、ホストが突然終了した場合 (システム・クラッシュなど) にはレプリケートされません。 Bean の状態はまだレプリケートされていないため、JVM の終了と同時にすべての状態データが一度にネットワーク上に送信され、ネットワーク・パフォーマンスに影響を与える可能性があります。このオプションは、JVM の停止に必要な時間を著しく増加させる可能性もあります。

表 9-2 <replication-policy> の scope 属性の値

scope の値	HttpSession	ステートフル・セッション Bean
modifiedAttributes (デフォルト)	変更された HTTP セッション属性、すなわち HttpSession オブジェクトで setAttribute () をコールして変更された値のみをレプリケートします。	該当なし。
allAttributes	HTTP セッションに設定されたすべての属性の値をレプリケートします。	ステートフル・セッション Bean に設定されたすべてのメンバー変数の値をレプリケートします。

orion-application.xml の <replication-policy> 要素は、アプリケーション内の Web モジュールと EJB モジュールを区別できません。しかし、EJB モジュール用に別のレプリケーション・ポリシーをコンポーネント固有の orion-ejb-jar.xml 構成ファイルの <session-deployment> 要素の replication 属性に指定できます。

replication 属性の有効な値は、表 9-3 を参照してください。次に例を示します。

```
<session-deployment name="MyStatefulVM" replication="onShutdown" />
<session-deployment name="MyEntity2" replication="onRequestEnd" />
```

このファイルの値は、orion-application.xml の対応する設定に優先するため、EJB モジュール用のレプリケーション・ポリシーを orion-ejb-jar.xml に、Web コンポーネント用のポリシーを orion-application.xml に事実上設定できます。

表 9-3 ステートフル・セッション EJB のレプリケーション・ポリシーの構成

replication の値	説明
onRequestEnd (デフォルト)	EJB メソッド・コールのたびに、Bean の現在の状態をレプリケートします。状態のレプリケーション頻度は高くなりますが、ホスト障害時における信頼性は向上します。これはデフォルト値です。
onShutdown	JVM が正常に終了するたびに、Bean の現在の状態をレプリケートします。状態は、ホストが突然終了した場合（システム・クラッシュや Linux または UNIX 環境での kill -9 起動など）にはレプリケートされません。
none	データをレプリケートしません。

アプリケーションの状態データをレプリケートする JVM の数の管理

<cluster> 要素の write-quota 属性を使用すると、状態データがレプリケートされる JVM の数を事実上制限できます。この機能を使用すると、状態レプリケーションの範囲を制御して、ネットワーク・トラフィックおよび関連するオーバーヘッドを削減できます。

write-quota のデフォルト値は 1 で、状態を Oracle Application Server クラスタ内のもう 1 つの JVM にレプリケートします。

アプリケーション・グループのメンバーは、実際は Oracle Application Server ノードではなく JVM で実行されます。クラスタのコンポーネントとして、複数の JVM がノードごとに稼働するアーキテクチャおよび構成を作成できます。

ハードウェアが停止した場合はフェイルオーバーによって保護される別の物理ノードに状態レプリケーションを格納するには、allow-colocation 属性を false に設定します。これを実行するには、状態レプリケーション・マネージャで、状態レプリケーションを格納するために別の物理ノードで稼働されているピア（write-quota が 1 よりも大きい場合は複数のピア）を選択する必要があります。

Oracle Application Server クラスタ内のすべての JVM に状態をレプリケートするには、クラスタ内の JVM の合計数を write-quota の値として指定する必要があります。

同期レプリケーションまたは非同期レプリケーションの使用

デフォルトでは、OC4J インスタンスはデータを他のインスタンスに非同期的にレプリケートします。しかし、<cluster> 要素内に <synchronous-replication> サブ要素を指定して、同期レプリケーションを有効にできます。これにより、レプリケーションを続行する前に、レプリケートする OC4J インスタンスは、データを受信したという他の 1 つ以上のピア・インスタンスからの確認応答を待機するように強制されます。

マルチキャスト・レプリケーションの構成

マルチキャスト IP レプリケーションは、スタンドアロン OC4J インストールで使用されるデフォルトのレプリケーション・プロトコルです。このモードでは、OC4J はマルチキャスト・パッケージを使用して、HTTP セッションおよびステートフル・セッション Bean の状態の変更を送受信します。これらのパッケージは、同じマルチキャスト・アドレスおよびポートを使用して他の OC4J プロセスによって取得されるネットワーク上で送信されます。失われたメッセージは特定されて再送信されるため、信頼性のある送信サービスが提供されます。

この構成は、同じマルチキャスト・アドレスおよびポートをすべての OC4J インスタンスに指定する必要があります。OC4J マルチキャストで使用されるデフォルト値は、アドレスは 230.230.0.1、ポートは 45566 です。これらの値は、必要に応じて、該当する XML 構成ファイルで変更できます。

空の `<cluster>` 要素を各インスタンスの `orion-application.xml` ファイルに追加するだけで、複数のアプリケーション・インスタンス間でマルチキャスト・レプリケーションを有効にできます。

```
<orion-application ...>
  ...
  <cluster/>
</orion-application>
```

次の例では、`ip` および `port` 属性を使用して新しいマルチキャスト・アドレスおよびポートを指定しています。オプションの `bind_addr` 属性は、バインドするネットワーク・インタフェース・カード (NIC) の指定に使用できます。この属性は、OC4J ホスト・マシンに複数のネットワーク・カードが装備されており、それぞれに固有の IP アドレスを保持し、マルチキャスト・メッセージの送受信に使用する NIC を定義する場合に使用すると便利です。

```
<orion-application ...>
  ...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes" />
    <protocol>
      <multicast ip="225.130.0.0" port="45577" bind_addr="226.83.24.10" />
    </protocol>
  </cluster>
</orion-application>
```

マルチキャスト・レプリケーションに対する既存の JavaGroups 構成の使用

OC4J によって提供されるマルチキャスト・ベースおよび `peer-to-peer` ベースのレプリケーション方式は、JavaGroups 通信プロトコル・スタックに基づいて構築されています。OC4J のこれらの方式は OC4J 固有の構成を使用するため、そのいずれかを使用して、状態データのメモリー内レプリケーションを行うことをお勧めします。

しかし、OC4J のクラスタリング・フレームワーク内で独自の JavaGroups 構成を使用することもできます。この機能は、`<cluster>` 要素内の `<property-config>` サブ要素に次のいずれかを指定すると有効になります。

- JavaGroups 構成プロパティが含まれる文字列
- その情報が含まれる XML 構成ファイルへの URL

詳細は、9-12 ページの「[<cluster> 要素の指定](#)」を参照してください。

peer-to-peer レプリケーションの構成

OC4J では、TCP を使用して Oracle Application Server クラスタ内のインスタンス間の接続を確立する peer-to-peer (P2P) トポロジでのレプリケーションをサポートします。各アプリケーション・インスタンスに保持される状態データは、各 OC4J インスタンスにユニキャストされます。

次の 2 つの peer-to-peer レプリケーションがサポートされます。

- 動的 peer-to-peer。Oracle Process Manager and Notification Server (OPMN) を使用して、ピア・ノードが相互に動的に検出して通信できるようにします。この構成は、各種コンポーネント (OC4J を含む) の管理に OPMN を使用する Oracle Application Server 環境で、デフォルトで使用されます。

詳細は、「[動的 OPMN 管理の peer-to-peer レプリケーションの構成](#)」を参照してください。

- 静的 peer-to-peer。クラスタ内の各ノードは、他の 1 つ以上のピア・ノードを認識するように明示的に構成されます。この構成は、比較的少ない数のスタンドアロン OC4J インスタンスがまとめてクラスタ化されているスタンドアロン OC4J 環境でのみサポートされます。

詳細は、「[静的 peer-to-peer レプリケーションの構成](#)」を参照してください。

動的 OPMN 管理の peer-to-peer レプリケーションの構成

Oracle Application Server 環境では、Oracle Process Manager and Notification Server (OPMN) を使用して動的 peer-to-peer レプリケーションを提供します。このレプリケーション・モデルでは、各 Oracle Application Server ノードは OPMN に自己登録します。次に、ノードは OPMN に使用可能なノードのリストを問い合せて、クラスタ内の他のノードを動的に検出して通信できるようにします。

注意： この機能を使用するには、まず、OPMN の動的マルチキャスト検出方式または静的検出サーバー方式のいずれかを使用して、アプリケーションのホストとなるすべてのノードをクラスタのメンバーにする必要があります。

詳細は、8-3 ページの「[サポートされるクラスタリング・モデル](#)」を参照してください。

各ノードは、OPMN に ONS (ハートビート) メッセージを定期的を送信して現行のステータスを OPMN に通知し、OPMN が使用可能なピア・ノードのリアルタイムなリストを保持し、障害が発生したときにノードに通知できるようにします。ノードが失われた場合は、別のノードがリクエストを処理できます。

```
<orion-application ...>
...
<cluster>
  <protocol>
    <peer>
      <opmn-discovery />
    </peer>
  </protocol>
</cluster>
</orion-application>
```

静的 peer-to-peer レプリケーションの構成

この構成では、他の1つ以上のピア・ノードのホスト・アドレスおよびポートを指定して、peer-to-peer 通信に使用できるようにします。ノードは各ピアを認識するようになるため、各ピアのピアも認識するようになり、最終的には、クラスタ内のすべてのノードが相互に認識するようになります。

この構成の主な課題は、ホストおよびポート定義が最新の状態で維持されるようにすることです。これには、多大な管理労力が必要となる可能性があります。次の要素および属性は、この構成に影響を及ぼします。

- <peer> 要素の start-port 属性: ローカルの OC4J プロセスがピア通信のためにバインドしようとするホストに開始ポートを指定します。このポートが使用できない場合、OC4J は、このポートの値を増やながら使用可能なポートを見つけます。
- <node> 要素: ピア・ノードを指定します。この要素の host および port 属性は、ピア通信に使用されるノード・アドレスおよびポートの名前を定義します。
- <peer> 要素の range 属性: start-port 属性の値ではなく、各 <node> 要素に指定されたポートに適用されます。range 属性は、指定されたポートがノードで使用できない場合に、port の値を増やす回数を定義します。

次の例は、sample アプリケーションとともに3つのクラスタ・ノードにデプロイされたアプリケーション・デプロイメント・ディスクリプタ orion-application.xml に指定されている静的 peer-to-peer 構成を示しています。

この構成では、各ノードはもう1つのノードをそのピアとして指定しています。その結果、クラスタ内のすべてのノードは相互に接続を確立できます。この例は、各ノードが連続して起動された場合にのみ機能します。つまり、www3.company.com は www2.company.com より前に起動する必要があります。起動しないと、www2.company.com は www3.company.com を認識できません。

1. 最初に、www1.company.com がそのピアとして www2.company.com を指定します。

```
<orion-application ...>
...
<cluster>
  <protocol>
    <peer start-port="7900" range="10" timeout="6000">
      <node host="www2.company.com" port="7900" />
    </peer>
  </protocol>
</cluster>
</orion-application>
```

2. 次に、www2.company.com がそのピアとして www3.company.com を指定します。

```
<orion-application ...>
...
<cluster>
  <protocol>
    <peer start-port="7900" range="10" timeout="6000">
      <node host="www3.company.com" port="7900" />
    </peer>
  </protocol>
</cluster>
</orion-application>
```

3. 最後に、www3.company.com がそのピアとして www1.company.com を指定します。

```
<orion-application ...>
...
<cluster>
  <protocol>
    <peer start-port="7900" range="10" timeout="6000">
      <node host="www1.company.com" port="7900" />
    </peer>
```

```

        </protocol>
    </cluster>
</orion-application>

```

代替構成では、すべてのノードが同じノードをピアとして指定します。たとえば、www1.company.com ノードと www3.company.com ノードはどちらも、www2.company.com をピアとして指定します。この構成では、www2.company.com は最初に起動されるノードである必要があり、他のノードはこのノードに接続して相互に接続を確立します。

OPMN のステート・レプリケーション用ポートの指定

OPMN 管理の Oracle Application Server 環境でステート・レプリケーションを使用してアプリケーションをデプロイすると、OPMN により、クラスタ全体への状態の伝播に使用されるポートが動的に割り当てられます。この割当ては、peer-to-peer レプリケーションが有効化されているアプリケーションのポート範囲に制限できます。明確に定義されたポート範囲を使用するファイアウォールまたはネットワークでのインストールには、ステート・レプリケーション用のポートを指定する必要があります。

peer-to-peer のステート・レプリケーションのポート範囲を指定するには、次のようにします。

1. opmn.xml ファイルの OC4J インスタンス構成に <port> 要素を追加します。
2. <port> 要素の id 属性の値として、peer-to-peer レプリケーションが有効化されているアプリケーションの名前を指定します。
3. <port> 要素の range 属性にポート範囲を指定します。

たとえば、peer-to-peer レプリケーション用に設定された rac-web という名前のアプリケーションのデプロイの場合、次の OC4J インスタンス構成で <port id=rac-web .../> というラベルの行は、ステート・レプリケーションに 15213 ~ 15214 のポートを使用するよう OPMN に指示しています。

```

<port id="default-web-site" range="80-100" protocol="http"/>
<port id="rmi" range="12401-12500"/>
<port id="rmis" range="12701-12800"/>
<port id="jms" range="12601-12700"/>
<port id="rac-web" range="15213-15214"/>

```

データベース・レプリケーションの構成

新しいクラスタリング・フレームワークでは、HTTP セッションおよびステートフル・セッション Bean の状態をデータベースにレプリケートできます。クラスタ化された OC4J フレームワークの外側にデータを保存して、クラスタ内のすべての OC4J インスタンスで致命的な障害が発生した場合にセッション全体をリカバリできるようにします。完全 HTTP セッションまたはステートフル・セッション Bean のオブジェクトは、データベースにレプリケートされます。

データベースへの接続は、<protocol> の <database> サブ要素の data-source 属性に指定されるデータソースを使用して作成されます。data-source 属性の値を、data-sources.xml に指定されているデータソースの jndi-name に設定します。

指定するデータソースは、あらかじめ OC4J インスタンス内に存在する必要があります。データソースを作成および使用方法は、『Oracle Containers for J2EE サービス・ガイド』を参照してください。

次の例では、MyOracleDS データソースを介してアクセスされるデータベースにデータをレプリケートするようにアプリケーションを構成しています。

```

<orion-application ...>
    ...
    <cluster>
        <protocol>
            <database data-source="jdbc/MyOracleDS"/>
        </protocol>
    </cluster>
</orion-application>

```

セッション・データは、データベースの次の表に保存されます。

- OC4J_HTTP_SESSION: HTTP セッションのメタデータが格納されます。
- OC4J_HTTP_SESSION_VALUE: HTTP セッションでアプリケーション・ユーザーによって設定される値が格納されます。
- OC4J_EJB_SESSION: ステートフル・セッション Bean の現在の状態が格納されます。

これらの表は、データベース・レプリケーションの初回起動時に、OC4J によって作成されます。表スキーマの詳細は、付録 C「セッション状態表の概要」を参照してください。

セッション・データがデータベースに格納される時間の長さは、セッションの有効時間 (TTL) に基づいています。現在のデータベースの時間とセッションが最後にアクセスされた時間の差がセッション・タイムアウト値より大きくなると、セッションは期限切れになったとみなされます。セッションの TTL を決定する実際の式は次のとおりです。

$$(\text{Current Database Time} - \text{Last Accessed Time}) > \text{Max Inactive Time}$$

期限切れになったセッションは、OC4J タスク・マネージャの次回実行時にデータベースから削除されます。タスク・マネージャの間隔の設定手順は、10-2 ページの「OC4J タスク・マネージャの構成」を参照してください。

セッションが正常終了せずに OC4J サーバーが終了した場合は、孤立したレコードがデータベースに作成されます。これらのレコードも、タスク・マネージャの次回実行時に削除されます。

アプリケーションの JVM、OC4J インスタンスおよび Application Server インスタンスの確認

システム・プロパティを使用して、クラスタ内でアプリケーション・インスタンスが稼働している JVM、OC4J インスタンスおよび Application Server インスタンスを確認できます。この情報は、Oracle Application Server 上に管理ユーティリティを作成する場合のみでなく、デバッグを実行する場合にも便利です。

クラスタ内で稼働しているアプリケーション・インスタンスの環境に関する情報を取得するには、`System.getProperty()` コールを使用して複数のプロパティにアクセスします。OC4J で実行されている Java プロセスでは、例 9-1 に示すように、システム・プロパティを使用して、JVM、OC4J インスタンスおよび Application Server インスタンスの情報をシステム・コンソールに出力できます。

例 9-1 システム・プロパティ値を出力するための `System.getProperty()` コール

```
System.out.println("Oracle home name: " + System.getProperty("oracle.home"));
System.out.println("OC4J instance name: " +
    System.getProperty("oracle.oc4j.instanceName"));
System.out.println("AS instance name: " +
    System.getProperty("oracle.ons.instanceName"));
System.out.println("Instance:Group:JVM PID: " +
    System.getProperty("oracle.ons.indexid"));
```

表 9-4 に、アプリケーションの JVM、グループ、OC4J インスタンス、Oracle ホームおよび Oracle Application Server インスタンスを指定するシステム・プロパティを説明します。

表 9-4 JVM、OC4J インスタンスおよび Application Server インスタンスのシステム・プロパティ

プロパティ	値
oracle.home	Oracle Application Server がインストールされている物理ディレクトリの名前を含む文字列
oracle.ons.instancename	Oracle Application Server インスタンスの名前を含む文字列
oracle.oc4j.instancename	OC4J インスタンスの名前を含む文字列
oracle.ons.indexid	OC4J インスタンス名、インスタンスが属するグループ、およびアプリケーション・インスタンスを実行している JVM の組合せを含む、次の書式の文字列 <i>OC4J_INSTANCE.oc4j_groupname.jvm_number</i> 例: java_ee1.javaee_group.2

クラスタリングの無効化

クラスタリングは、<cluster> 要素のブールの enabled 属性を使用して、グローバルにまたは特定のアプリケーションに対して無効にできます。アプリケーションの orion-application.xml ファイルでこの属性を false に設定すると、そのアプリケーションはクラスタから事実上削除されます。

<cluster> 要素の指定

<cluster> 要素は、アプリケーションのクラスタリングを管理するための単一の方法です。この要素は、グローバル・レベルのアプリケーションのクラスタリング構成では ORACLE_HOME/j2ee/instance/config/application.xml ファイルでのみ使用されます。また、アプリケーション・レベルのクラスタリング構成ではアプリケーション固有の orion-application.xml ファイルで使用されます。

<cluster>

OC4J インスタンス内で稼働しているエンタープライズ・アプリケーションのアプリケーション・クラスタリング構成が含まれます。

<cluster> のサブ要素:

```
<property-config>
<flow-control-policy>
<replication-policy>
<protocol>
<synchronous-replication>
```

属性:

- enabled: アプリケーションに対してクラスタリングが有効かどうかを指定します。デフォルトは true です。この値をアプリケーション・レベルで設定すると、親アプリケーション (default アプリケーションを含む) から継承された値は上書きされます。
- group-name: レプリケーション・グループ・チャネルを確立する際に使用する名前。指定しないと、OC4J サーバー構成ファイル server.xml に定義されているアプリケーション名がデフォルトで使用され、新しいグループ・チャネルがエンタープライズ・アプリケーションごとに作成されます。

値を指定すると、アプリケーションとすべての子アプリケーションは、このグループ名に関連付けられたチャネルを使用します。

<database> タグが指定されている場合、この属性は無視されます。

- **allow-colocation**: アプリケーションの状態を同じホスト・マシンに存在するノードにレプリケートできるかどうかを指定します。デフォルトは `true` です。しかし、複数のホストが使用可能である場合は、この属性を `false` に設定する必要があります。

複数の OC4J インスタンスが同じマシン上でインスタンス化されている場合、インスタンスごとに異なるリスナー・ポートを `default-web-site.xml`、`jms.xml` および `rmi.xml` の各構成ファイルに指定する必要があります。

- **write-quota**: アプリケーションの状態をレプリケートする必要がある他のアプリケーション・グループ・メンバー (JVM) の数。この属性は、旧 OC4J リリースで使用されたアイランドの概念と同様に、状態が書き込まれる JVM の数を制限することによってオーバーヘッドの削減を可能にします。

デフォルトの JVM の数は 1 です。

`<database>` タグが指定されている場合、この属性は無視されます。

- **cache-miss-delay**: セッションがローカルに見つからない場合に、別のグループ・メンバーがセッションで応答するのをプロセス内で待機する時間 (ミリ秒単位)。セッションが見つからない場合、リクエストは指定された時間にわたって一時停止します。

デフォルトは 1000 ミリ秒です。リクエストの負荷が重いと予想されるインストールでは、この値を 5000 などに増やす必要があります。この値を高くすると、`allow-colocation` が `true` に設定されている場合、OC4J インスタンスの内部でセッション・データのレプリカが作成されることも防ぎます。

`<database>` タグが指定されている場合、この属性は無視されます。

<property-config>

JavaGroups グループ通信プロトコルを使用して、クラスタ内のノード間でセッションの状態をレプリケートするために必要なデータが含まれます。

属性:

- **url**: JavaGroups の XML 構成ファイルへのリンク
- **property-string**: JavaGroups JChannel の作成方法を定義するプロパティが含まれる文字列

<replication-policy>

適用するレプリケーション・ポリシー。データをレプリケートするタイミングとレプリケート対象のデータを定義します。

属性:

- **trigger**: レプリケーションが発生する頻度。この属性の値は、9-5 ページの表 9-1 を参照してください。
- **scope**: レプリケート対象のデータ。この属性の値は、9-5 ページの表 9-2 を参照してください。

<protocol>

データ・レプリケーションに使用する方式を定義します。指定できる方式は 1 つのみです。

サブ要素:

```
<multicast>
<peer>
<database>
```

<multicast>

レプリケーションにマルチキャスト通信を使用するために必要な構成が含まれます。これはデフォルトで使用されるプロトコルです。

属性:

- `ip`: 使用するマルチキャスト・アドレス。OC4J のデフォルトは 230.230.0.1 です。
- `port`: 使用するマルチキャスト・ポート。OC4J のデフォルトはポート 45566 です。
- `bind_addr`: バインドするネットワーク・インタフェース・カード (NIC)。この属性は、OC4J ホスト・マシンに複数のネットワーク・カードが装備されており、それぞれに固有の IP アドレスを保持している場合に使用すると便利です。

<peer>

レプリケーションに `peer-to-peer` (P2P) 通信を使用するために必要な構成が含まれます。

サブ要素:

<opmn-discovery>

<node>

属性:

- `start-port`: ピア通信用に割当てを試みるためのノード上の内部ポート。OC4J は、この値を増やしながら使用可能なポートが見つけます。デフォルトはポート 7800 です。スタンドアロン OC4J インストールでの静的 `peer-to-peer` レプリケーションの構成に対してのみ有効です。
- `range`: 使用可能なピア・ノードの検索中に、各 <node> サブ要素に指定された `port` の値を増やす回数。デフォルトは 5 回の増分です。スタンドアロン OC4J インストールでの静的 `peer-to-peer` レプリケーションの構成に対してのみ有効です。
- `timeout`: 使用可能なピア・ノードの検索中に、ピアからのレスポンスを待機する時間 (ミリ秒単位)。デフォルトは 3000 ミリ秒です。スタンドアロン OC4J インストールでの静的 `peer-to-peer` レプリケーションの構成に対してのみ有効です。
- `bind_addr`: バインドするネットワーク・インタフェース・カード (NIC)。この属性は、OC4J ホスト・マシンに複数のネットワーク・カードが装備されており、それぞれに固有の IP アドレスを保持している場合に使用すると便利です。

<opmn-discovery>

Oracle Application Server 環境で動的 `peer-to-peer` レプリケーションを使用するように OC4J を構成します。

<node>

静的 `peer-to-peer` 通信を使用する場合にポーリングするノードのホスト名およびポートが含まれます。この要素の 1 つ以上のインスタンスを <peer> 要素内に指定できます。

属性:

- `host`: ピア・ノードのホスト名 (URL)。
- `port`: `peer-to-peer` 通信に使用されるノード上のポート。デフォルトはポート 7800 です。

<database>

状態データをデータベースに保存するために必要な接続情報が含まれます。

属性:

- `data-source`: データベース接続情報を格納するデータソースの名前。この値は、`data-sources.xml` に指定されているデータソースの `jndi-name` 属性の値である必要があります。

<flow-control-policy>

レプリケーション時にクラスタリング・メッセージの処理に割り当てるメモリー量を制御します。この要素は、レプリケーション時にノード間で送信されるデータ量（バイト単位）をゲートして、メモリー不足エラーを防止することを目的としています。

属性:

- **enabled:** フロー制御が有効かどうかを指定します。デフォルトは true です。
- **max-bytes:** 受信ノードが受け入れることができる最大バイト数。この値に達すると、追加メッセージを送信できるようになるまで、送信ノードは受信側からの確認応答を待機する必要があります。デフォルト値は、500000 です。
- **min-bytes:** 送信するバイト数を多くする必要があるという確認応答をトリガーせずに、受信ノードが受け入れることができる最小バイト数。受信済バイト数がこの値を下回る場合、送信側から受け入れることができるバイト数がさらに多いことを受信側は確認応答します。デフォルトは 0 です。
- **threshold:** min-bytes が指定されない場合に、その属性値を決定するために、この係数値が受信リクエストに適用されます。デフォルト値は、0.25 です。

<synchronous-replication>

指定した場合、アプリケーション・データをレプリケートするノードは、データ更新を受信したという他の 1 つ以上のピア・ノードからの確認応答を待機してから、レプリケーションを続行します。この要素は任意です。デフォルトの動作では、ノードは他のノードにデータを非同期的にレプリケートし続けます。

属性:

- **timeout:** ピア・ノードからのレスポンスを待機する時間（ミリ秒単位）。この値を超えると、確認応答を受信しなくてもレプリケーションは続行されます。デフォルト値は、10000 ミリ秒（10 秒）です。

タスク・マネージャとスレッド・プールの構成

この章では、OC4J インスタンスに対するタスク・マネージャの構成、および OC4J インスタンスおよび Web サイト・アプリケーションに対するスレッド・プールの構成に関するガイドラインを示します。この章の内容は次のとおりです。

- [OC4J タスク・マネージャの構成](#)
- [OC4J スレッド・プールの構成](#)

OC4J タスク・マネージャの構成

タスク・マネージャは、HTTPセッションのタイムアウトや変更済構成ファイルのチェックなど、保留タスクをすべて実行するバックグラウンド・プロセスです。デフォルトでは、1秒（1000ミリ秒）間隔で実行します。

タスク・マネージャの実行間隔は、server.xml 構成ファイルの <application-server> 要素の taskmanager-granularity 属性にミリ秒単位で指定されます。これは、OC4J コンテナ・レベルのパラメータです。デフォルトは 1000 ミリ秒です。

たとえば、server.xml の次のエントリは、1分（60000ミリ秒）間隔で実行するようにタスク・マネージャを構成します。

```
<application-server ... taskmanager-granularity="60000" ...>
```

server.xml の変更後に、OC4J を再起動する必要があります。

注意：このパラメータは、TaskManager MBean の granularity 属性を使用して設定することもできます。この属性は、Application Server Control の JMX ブラウザからアクセスできます。

MBean へのアクセスおよび使用による OC4J プロセスの管理の詳細は、[第12章「OC4JでのMBeanの使用法」](#)を参照してください。

OC4J スレッド・プールの構成

スレッド・プールは、OC4J プロセスおよび OC4J インスタンスにデプロイされているアプリケーションで利用および再利用するためのスレッドを作成して格納します。要求に応じて新規スレッドを作成するのではなく、既存のスレッドを再利用することで、パフォーマンスを向上させ、JVM や基礎となるオペレーティング・システムに対する負荷を軽減します。

[表 10-1](#) に、OC4J 10g (10.1.3.4.0) で使用可能なスレッド・プールをリストします。

表 10-1 OC4J スレッド・プール

スレッド・プール	説明
system	OC4J ランタイムで使用します。 このプールのスレッドでは、OC4J インスタンスにデプロイされているアプリケーションを実行できません。 このスレッド・プールの構成は変更しないでください。
http	HTTP および AJP リクエストを処理します。 rmi request スレッド・プールが存在しない場合は、http スレッド・プールが RMI リクエストを処理します。 rmi connection スレッド・プールが存在しない場合は、http スレッド・プールが RMI 接続を処理します。
jca	リソース・アダプタからのワーク管理リクエストを処理します。 OC4J インスタンスにデプロイされるリソース・アダプタが必要な場合、リソース・アダプタ (JMS コネクタなど) で使用されるワーカー・スレッドを格納するワーク管理スレッド・プールは、OC4J プロセス内に作成されます。
rmi request	RMI リクエストを処理します。 このオプションのスレッド・プールを使用すると、スレッド・リソースの割当てをより強力に管理できます。

表 10-1 OC4J スレッド・プール (続き)

スレッド・プール	説明
rmi connection	RMI 接続を処理します。 このオプションのスレッド・プールには、RMI 接続でブロック読取りを実行するスレッドが用意されています。 rmi connection プールは、RMI 接続のみでなく、RMI スレッド・リスナーおよび JMS サーバー・スレッドにも使用されます。
カスタム	1 つ以上のアプリケーションで使用します。 アプリケーションに個別のカスタム・スレッド・プールを使用すると、スレッド・リソースの競合を減らすことができます。カスタム・スレッド・プールは、クラスタ内の一連のアプリケーションで共有できます。

デフォルトでは、OC4J の起動時にこれらのスレッド・プールが 3 つ作成されます。

- system
- http
- jca

各スレッド・プールでは、アイドル・スレッドは、リクエスト数が使用可能なスレッド数を超えないかぎり、新規スレッドが生成される前に再利用されます。非アクティブ状態で 10 分経過すると、アイドル・スレッドは自動的に破棄されます。

各スレッド・プールは、server.xml ファイルの <thread-pool> 要素または <custom-thread-pool> 要素によって定義されます。表 10-2 にこれらの要素の属性をまとめ、デフォルトの属性値を示します。

デフォルトのスレッド・プール構成を使用することも、変更することも可能です。各 OC4J インスタンスでは、system を除く任意のスレッド・プールの属性値を変更できます。また、rmi request、rmi connection および 1 つ以上のカスタム・スレッド・プールを追加できます。次の各項では、スレッド・プールの構成方法について説明します。

- [スレッド・プール構成の変更](#)
- [アプリケーションのカスタム・スレッド・プールの構成](#)

表 10-2 <thread-pool> および <custom-thread-pool> の属性

属性	説明
name	<p>name 属性にはスレッド・プールの名前を指定します。デフォルト値はありません。</p> <p>カスタム・スレッド・プールの場合、名前には任意の文字列値を指定できます。</p> <p><thread-pool> 要素では、名前は次に示す値のいずれかである必要があります。</p> <ul style="list-style-type: none"> ■ system OC4J ランタイムで使用するスレッド・プール ■ http HTTP リクエスト、AJP リクエスト、RMI リクエスト (rmi request スレッド・プールが構成されていない場合) および RMI 接続 (rmi connection スレッド・プールが構成されていない場合) を処理するスレッド・プール ■ jca J2CA ワーク・マネージャがリソース・アダプタ・リクエストを処理するためのワーク管理スレッド・プール ■ rmi request RMI リクエストを処理するスレッド・プール ■ rmi connection RMI 接続でスレッドによりブロック読取りが実行されるスレッド・プール <p>これらの名前付きプールのスレッドの名前にはそれぞれ、SystemThreadPool_、HTTPThreadPool_、WorkManager_、RMIRequestThreadPool_ および RMIConnectionThreadPool_ という接頭辞が付いており、接尾辞は増加するカウンタです。</p>
min	<p>プールに作成するスレッドの最小数。デフォルト値は 0 です。</p> <p>jca スレッド・プールのスレッドの最小数は、マシンにインストールされている CPU 数の倍数としてください。しかし、この数値は小さく設定する必要があります。スレッドの数が多ほど、オペレーティング・システムおよびガベージ・コレクタの負荷が大きくなります。</p> <p>rmi connection スレッド・プールの min の値は、任意の時点で保持する物理的な接続数に比例します。queue 値は、接続トラフィックの急増に対処します。</p>
max	<p>プールに作成できるスレッドの最大数。最大サイズ未満でかつアイドル・スレッドがない場合に、新規スレッドが生成されます。新規スレッドが生成される前に、アイドル・スレッドが使用されます。デフォルト値は、1024 です。</p> <p>通常、rmi connection スレッド・プールでは、内部で使用するために RMI および JMS リスナーとして 3 つのスレッドが作成されるため、スレッドの必要な最大数に 3 を足して max の値を設定する必要があります。たとえば、max="16" と指定すると、リクエストの処理に使用可能なスレッドは 13 のみです。同様に、max 値が 20 の場合には、使用可能なスレッドは 17 のみです。</p> <p>rmi connection スレッド・プールの max の値も、任意の時点で保持する物理的な接続数に比例します。queue 値は、接続トラフィックの急増に対処します。</p> <p>jca スレッド・プールのスレッドの最大数は、マシンにインストールされている CPU 数の倍数としてください。しかし、この数値は小さく設定する必要があります。スレッドの数が多ほど、オペレーティング・システムおよびガベージ・コレクタの負荷が大きくなります。</p>

表 10-2 <thread-pool> および <custom-thread-pool> の属性 (続き)

属性	説明
queue	キュー内に保持できるリクエストの最大数。デフォルト値は 0 です。 queue 値は、スレッドの最大数の 2 倍以上のサイズに設定してください。
keepAlive	新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。タイムアウトに達するとスレッドは破棄されます。デフォルト値は、600000 です。 スレッドを破棄しないようにするには、-1 に設定します。 デフォルト値の 600000 ミリ秒 (10 分) は、値が -1 ではない場合に設定できる最小値でもあります。 keepAlive を 0 (ゼロ) に設定すると、アクティブなポーリングが原因で CPU 使用率が高くなります。
stackSize	スレッド・プール・スタックのサイズ。デフォルト値は 0 です。
debug	値を true に設定すると、起動時にアプリケーション・サーバーのスレッド・プール情報がコンソールに出力されます。デフォルトは false です。debug が false の場合、スレッド・プール情報は出力されません。

スレッド・プール構成の変更

次に示す方法で、Application Server Control を使用するか、server.xml ファイルを編集することで、OC4J インスタンスのスレッド・プール構成を変更できます。

- Application Server Control の「スレッド・プール構成」ページでスレッド・プールの属性値を変更
- Application Server Control のシステム MBean ブラウザを使用してスレッド・プール MBeans の属性を変更

MBean へのアクセスおよび MBean を使用した OC4J の管理の詳細は、12-5 ページの「[システム MBean ブラウザの使用方法](#)」を参照してください。

- server.xml に <thread-pool> 要素を追加して、rmi request または rmi connection スレッド・プール、あるいはその両方を構成
server.xml の変更後に、OC4J を再起動する必要があります。

注意： スレッド・プールの構成またはデフォルト構成の変更は、エキスパート・モードのタスクです。変更の必要がない場合には、デフォルトのスレッド・プール構成を使用することをお勧めします。

Application Server Control を使用したスレッド・プール構成の変更

Application Server Control を使用して OC4J インスタンスのスレッド・プール構成を変更するには、次のようにします。

1. OC4J ホームページに移動して「**管理**」をクリックし、OC4J の「**管理**」ページを表示します。このページには、OC4J インスタンスに対して実行可能な様々な管理タスクがリストされた表が表示されています。
2. 「**管理**」ページの「**プロパティ**」で、「**スレッド・プール構成**」を選択して OC4J の「**スレッド・プール構成**」ページを表示します。
3. このページに表示されている任意のスレッド・プールの 1 つ以上の属性値を変更します。
属性値の詳細は、10-4 ページの表 10-2 または B-19 ページの「**<thread-pool>**」を参照してください。
4. 「**適用**」をクリックします。
5. 「**クラスタ・トポロジ**」ページまたは OC4J ホームページから OC4J インスタンスを再起動します。

次の「**MBeans を使用したスレッド・プール構成の変更**」で説明するように、Application Server Control の MBeans を使用してスレッド・プールを構成することもできます。

MBeans を使用したスレッド・プール構成の変更

Application Server Control の MBeans を使用してスレッド・プールの属性を変更するには、次のようにします。

1. OC4J ホームページに移動して「**管理**」をクリックし、OC4J の「**管理**」ページを表示します。このページには、OC4J インスタンスに対して実行可能な様々な管理タスクがリストされた表が表示されています。
2. 「**管理**」ページの「**JMX**」で、「**システム MBean ブラウザ**」を選択して OC4J の「**システム MBean ブラウザ**」ページを表示します。このページには、OC4J インスタンスによって公開されたシステム MBean が表示されています。
3. ページの左側にあるナビゲーション・ツリーを開いて、ThreadPool の下にある OC4J インスタンスのスレッド・プールを選択します。
4. 編集ボックスのあるスレッド・プールの任意の属性を変更します。
属性値の詳細は、10-4 ページの表 10-2 または B-19 ページの「**<thread-pool>**」を参照してください。
5. 「**適用**」をクリックします。

server.xml への <thread-pool> 要素の追加

次の例では、<thread-pool> 要素を使用して、server.xml ファイルに rmi request スレッド・プールを構成します。

```
<thread-pool
  name="rmi request"
  min="50"
  max="50"
  queue="2560"
  keepAlive="-1"
  stackSize="0"
  debug="true"/>
```

この構成では、OC4J により RMI リクエストを処理する別のスレッド・プールが作成されます。スレッド・プールの属性は次のようになります。

- 最低のスレッド数は 5
- 最高のスレッド数は 50
- キューの最大リクエスト数は 2560
- keepAlive 値は -1 (タイムアウトなし)
- stackSize 値は 0 (JVM が決定)
- debug 属性の設定は true

次の例では、server.xml に別のスレッド・プールを構成する <thread-pool> 要素を示します。1 つは RMI リクエストを、もう 1 つは RMI 接続を、最後の 1 つは HTTP および AJP リクエストを処理します。

```
<thread-pool
  name="rmi request"
  min="50"
  max="50"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

```
<thread-pool
  name="rmi connection"
  min="44"
  max="44"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

```
<thread-pool
  name="http"
  min="40"
  max="40"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

OC4J インスタンスの起動時に、10-4 ページの表 10-2 のデフォルトの属性値が設定された状態で、http スレッド・プールがデフォルトで作成されます。その他の rmi-* スレッド・プールが存在しない場合、このスレッド・プールでは、HTTP および AJP リクエストだけでなく、RMI リクエストおよび RMI 接続も処理されます。

注意： server.xml の変更後に、OC4J を再起動する必要があります。

アプリケーションのカスタム・スレッド・プールの構成

server.xml ファイルに <custom-thread-pool> 要素を追加することで、OC4J インスタンスで使用する別のカスタム・スレッド・プールをアプリケーションに作成できます。その後、アプリケーションの *-web-site.xml ファイルにある <web-site> 要素の custom-thread-pool 属性のスレッド・プールを参照することにより、アプリケーションがそのカスタム・スレッド・プールを使用できるようにできます。server.xml ファイルには、複数の <custom-thread-pool> 要素を指定でき、それぞれのカスタム・スレッド・プールを使用するアプリケーションを 2 つ以上構成できます。

server.xml では、<custom-thread-pool> 要素は <application-server> 要素のサブ要素で、name の値が制限されていないことを除き、<thread-pool> 要素と同じ属性が指定されています。次に例を示します。

```
<custom-thread-pool name="mypool" min="3" />
```

name 属性は必須で、その他すべての属性はオプションです。この要素の詳細は、B-9 ページの「<custom-thread-pool>」を参照してください。

<thread-pool> 要素の詳細は、10-5 ページの「スレッド・プール構成の変更」および B-19 ページの「<thread-pool>」を参照してください。

-web-site.xml ファイルの詳細は、B-22 ページの「Web サイト構成ファイル (-web-site.xml) の概要」を参照してください。custom-thread-pool 属性は、B-22 ページの表 B-24 で説明されています。

次の例では、HTTP Web サイトの default-web-site.xml ファイルの <web-site> 要素に custom-thread-pool 属性を追加して、デフォルト以外のスレッド・プールを使用する HTTP Web サイトを構成しています。

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
web-site-10_0.xsd"
  protocol="http"
  port="8888"
  custom-thread-pool="mypool1"
  display-name="OC4J 10g (10.1.3) Default Web Site"
  schema-major-version="10"
  schema-minor-version="0"
/>
```

古いスレッド・プール形式からの変換

server.xml の <global-thread-pool> および <work-manager-thread-pool> 要素は、スレッド・プールが古い形式で構成されています。これらの要素は、OC4J 10g (10.1.3.4.0) では推奨されていません。これらの要素の属性は、B-10 ページの表 B-7 および B-21 ページの表 B-23 で説明されています。

server.xml ファイルに <global-thread-pool> または <work-manager-thread-pool> 要素が含まれている場合、OC4J 10g (10.1.3.4.0) により古い形式の要素が server.xml の新しい形式に更新されます。たとえば、server.xml ファイルに次の要素が含まれているとします。

```
<global-thread-pool
  min="60"
  max="60"
  queue="20000"
  keepAlive="-1" />

<work-manager-thread-pool
  min="23"
  max="24"
  queue="5000"
  keepAlive="-1" />
```

OC4J の起動後、server.xml ファイルには、<global-thread-pool> および <work-manager-thread-pool> 要素ではなく、次の <thread-pool> 要素が含まれています。

```
<thread-pool
  name="http"
  min="60"
  max="60"
  queue="20000"
  keepAlive="-1"
  stackSize="0" />
```

```
<thread-pool
  name="jca"
  min="23"
  max="24"
  queue="5000"
  keepAlive="-1" />
```

表 10-3 に、<global-thread-pool> および <work-manager-thread-pool> の属性が、OC4J 10g (10.1.3.1.0) で導入された新しいスレッド・プールにどのようにマップされているかを示します。

表 10-3 古いスレッド・プール構成の新しいスレッド・プールへのマッピング

古いスレッド・プール属性	<thread-pool> の name 属性の値	新しいスレッド・プール属性
<global-thread-pool> の min、max、queue、keepAlive および debug 属性	http	<thread-pool> の min、max、queue、keepAlive および debug 属性
<work-manager-thread-pool> の min、max、queue、keepAlive および debug 属性	jca	<thread-pool> の min、max、queue、keepAlive および debug 属性
<global-thread-pool> の cx-min、cx-max、cx-queue、cx-keepAlive および cx-debug 属性	rmi request	<thread-pool> の min、max、queue、keepAlive および debug 属性
<global-thread-pool> の rmiRequest-min、rmiRequest-max、rmiRequest-queue、rmiRequest-keepAlive および rmiRequest-debug 属性	rmi connection	<thread-pool> の min、max、queue、keepAlive および debug 属性

たとえば、OC4J により、次の <global-thread-pool> 要素から新しい <thread-pool> 要素が生成されます。

```
<global-thread-pool
  keepAlive="-1"
  debug="false"
  cx-keepAlive="-1"
  cx-debug="false"
  rmiRequest-keepAlive="-1"
  rmiRequest-debug="false"
  min="40"
  max="40"
  queue="2560"
  cx-min="44"
  cx-max="44"
  cx-queue="2560"
  rmiRequest-min="50"
  rmiRequest-max="50"
  rmiRequest-queue="2560"/>
```

<thread-pool> 要素の等価は次のとおりです。

```
<thread-pool
  name="rmi request"
  min="50"
  max="50"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

```
<thread-pool
  name="rmi connection"
  min="44"
  max="44"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

```
<thread-pool
  name="http"
  min="40"
  max="40"
  queue="2560"
  keepAlive="-1"
  stackSize="0"/>
```

OC4J でのロギング

この章では、OC4J で使用できるシステムおよびアプリケーションのロギング機能の使用方法について説明します。この章の内容は次のとおりです。

- OC4J により生成されるログ・ファイルの概要
- プレーン・テキスト・ファイル・ロギングの使用法
- Oracle Diagnostic Logging (ODL) の使用法
- OC4J ロギングの構成
- LogViewer を使用した OC4J ログのアプリケーション・メッセージの表示
- Application Server Control を使用したアプリケーション・ログ出力の構成
- アプリケーションの log4j メッセージの OC4J ログへのリダイレクト

OC4J により生成されるログ・ファイルの概要

各 OC4J プロセスでは、トラブルシューティングを支援するために多くのログ・ファイルが生成されます。1つの OC4J インスタンスに対して複数のプロセスが稼働している場合、複数セットのログ・ファイルが生成されます。

OC4J では、次に示す 2 種類のログ・ファイルが生成されます。

■ プレーン・テキストのログ・ファイル

プレーン・テキストのログは、OC4J コンポーネントに使用されるデフォルトのログ・ファイルで、開発環境での使用に最適です。これらのテキスト・ファイルに記録されたメッセージは、任意のエディタまたは Oracle Enterprise Manager 10g Application Server Control で読み取ることができます。

■ Oracle Diagnostic Logging (ODL) のログ・ファイル

これらのファイルに記録されたメッセージは、Application Server Control を使用して表示できる XML 形式を使用します。ODL ではログ・ファイルの循環がサポートされています。

ログ・ファイルは、データの記録対象となるコンポーネントまたはアプリケーションに応じて異なる場所に生成されます。各コンポーネントまたはアプリケーションのロギング構成は、コンポーネント固有の XML 構成ファイルに定義されます。

表 11-1 に、生成される各種ログ・ファイルの名前および場所とともに、各コンポーネントのロギング構成が含まれる XML 構成ファイルを示します。特に明記されていないかぎり、示されるパスはすべて、スタンドアロン OC4J の場合は `ORACLE_HOME/j2ee/home` 内、OPMN 管理の OC4J インスタンスの場合は `ORACLE_HOME/j2ee/instance` 内です。

表 11-1 OC4J で生成されるログ・ファイルのリスト

コンポーネント	構成ファイル	デフォルトのログ・ファイル名および場所
Java ロギングを使用する OC4J コンポーネント	この ODL 形式のログ・ファイルの構成方法は、11-8 ページの「OC4J ロギングの構成」を参照してください。	スタンドアロン OC4J: /log/oc4j/log.xml OPMN 管理 OC4J: /log/instance_default_group_1/oc4j/log.xml
Application Server Control	/application-deployments/ascontrol/orion-application.xml	スタンドアロン OC4J: /log/ascontrol-application.log OPMN 管理 OC4J: /log/instance_default_group_1/ascontrol-application.log
OC4J にデプロイされたアプリケーション	/application-deployments/app_name/orion-application.xml	スタンドアロン OC4J: /application-deployments/app_name/application.log OPMN 管理 OC4J: /application-deployments/app_name/instance_default_group_1/application.log
グローバル (default) アプリケーション	/config/application.xml	スタンドアロン OC4J: /log/global-application.log OPMN 管理 OC4J: /log/instance_default_group_1/global-application.log

表 11-1 OC4J で生成されるログ・ファイルのリスト (続き)

コンポーネント	構成ファイル	デフォルトのログ・ファイル名および場所
デフォルト Web サイトのアクセス・ロギング	/config/default-web-site.xml	スタンドアロン OC4J: /log/default-web-access.log OPMN 管理 OC4J: /log/instance_default_group_1/default-web-access.log
OC4J サーバー	/config/server.xml	スタンドアロン OC4J: /log/server.log OPMN 管理 OC4J: /log/instance_default_group_1/server.log
JMS	/config/jms.xml	スタンドアロン OC4J: /log/jms.log OPMN 管理 OC4J: /log/instance_default_group_1/jms.log
RMI	/config/rmi.xml	スタンドアロン OC4J: /log/rmi.log OPMN 管理 OC4J: /log/instance_default_group_1/rmi.log
OPMN	ORACLE_HOME/opmn/conf/opmn.xml	ORACLE_HOME/opmn/logs

oc4j.properties ファイルを介して OC4J インスタンスを構成し、デフォルトの宛先ではなく特定のデバッグ先にトレース・ファイルを生成できます。Oracle Application Server では、インスタンスが同じグループの場合でも、異なる 2 つの OC4J インスタンスが同じ宛先にトレース出力を生成する構成はサポートされていません。各 OC4J インスタンスでは、それぞれのトレース・ファイルが管理されます。

プレーン・テキスト・ファイル・ロギングの使用法

プレーン・テキスト・ロギングは、OC4J で使用されるデフォルトの形式です。

この方式は、XML ファイルに合せてメッセージを振り分けます。しかし、同じサイズの複数のログ・ファイルに書き込むのではなく、そのコンポーネントに対するすべてのメッセージを単一のログ・ファイルに書き込みます。次の各項では、テキスト・ロギングの使用法について説明します。

- [テキスト・ファイル・ロギングの有効化または無効化](#)
- [テキスト・ログ・ファイルの管理](#)
- [テキスト・ログ・ファイルの表示](#)

テキスト・ファイル・ロギングの有効化または無効化

テキスト・ロギングは、表 11-1 に示した XML 構成ファイル (default-web-site.xml ファイルを除く) の要素により、有効または無効になります。(Web サイトのアクセス・ロギングの構成方法は、13-14 ページの「Web サイトのアクセス・ロギングの構成」を参照してください。)

ロギングは、各コンポーネントの XML 構成ファイルの <log> 要素の <file> サブ要素によって有効になります。この要素には、単一の path 属性が含まれ、生成されるログ・ファイルの名前と、必要に応じて場所を指定します。

```
<log>
  <file path="application.log" />
</log>
```

コンポーネントのテキスト・ロギングを無効にするには、対応する構成ファイルの <file> 要素を削除するか、コメント・アウトします。この行を削除しないで ODL を有効にすると、両方のロギング・オプションが有効になります。

たとえば、アプリケーションのテキスト・ロギングを無効にするには、そのアプリケーションの orion-application.xml ファイルで次の <file> 要素をコメント・アウトします。

```
<log>
!-- <file path="application.log" /> -->
</log>
```

ODL とテキスト・ロギングの両方を同時に有効にできますが、ディスク領域を節約するには、これらのオプションのいずれかを無効にする必要があります。

テキスト・ログ・ファイルの管理

テキスト・ロギングにはサイズ制限やログ循環機能がないため、ログ・ファイルの監視は重要です。チェックしないままにしておくと、ログ・ファイルのサイズは増大し続け、ディスク超過が発生する可能性があります。

これらのファイルを管理する唯一の方法は、OC4J を停止し、ファイルを削除した後、OC4J を再起動してログ・ファイルを新しく始めることです。

テキスト・ログ・ファイルの表示

すべてのテキスト・ログ・ファイルは、11-2 ページの表 11-1 に示した場所にデフォルトで生成されます。テキスト・ログ・ファイルは、log 拡張子によって識別できます。

OC4J コンポーネントに対して生成されるテキスト・ログ・ファイルは、次のように Application Server Control を使用して表示できます。

1. Application Server Control のページの一番下にある「ログ」リンクをクリックします。
2. 「OC4J」を開きます。
3. 「<instanceName>」を開きます。デフォルトのインスタンス名は home です。

デプロイ済 J2EE アプリケーションのテキスト・ログ・ファイルは、Application Server Control を使用して表示することはできません。

Oracle Diagnostic Logging (ODL) の使用方法

Oracle Diagnostic Logging フレームワーク (ODL) には、標準 Java フレームワークを補完して、ログ・データと Oracle ログ分析ツールを自動的に統合するプラグイン・コンポーネントが用意されています。

ODL フレームワークでは、ログ・ファイルは XML 文書形式で作成され、他の Oracle Application Server やカスタム開発されたコンポーネント (Application Server Control など) によるログの分析および再利用を可能にします。ODL では、テキストベースのロギングとは異なり、ログ・ファイルの循環がサポートされています。

- [ODL の有効化または無効化](#)
- [ODL ログ・ファイルの管理](#)
- [ODL ログ・ファイルの表示](#)

ODL の有効化または無効化

ODL は、<odl> 要素を [表 11-1](#) に示した XML ファイルの <log> 要素内に追加すると有効になります。

注意:

- ODL は、アプリケーションがデプロイされる際に、デプロイ・プラン・エディタを使用して log プロパティの odlis に値を設定すると、アプリケーションに対して有効にできます。

デプロイ・プラン・エディタを使用したアプリケーションの構成方法は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。
 - Web サイトに対する ODL は、異なる構成を使用します。この構成の詳細は、13-14 ページの「[Web サイトのアクセス・ロギングの構成](#)」を参照してください。
 - ODL とテキスト・ファイル・ロギングの両方を同時に有効にできます。しかし、ディスク領域を節約するには、これらのオプションのいずれかを無効にする必要があります。
-
-

<odl> 要素には次の属性があります。すべて必須です。

- path: このコンポーネントの log.xml ファイルが生成されるディレクトリへのパス。

重要:

次の例に示すように、パスを ../log/appName と指定します。このパスは、Application Server Control を使用してログ・ファイルを表示するために必要です。

- max-file-size: 個々のログ・ファイルに対して許容される最大サイズ (KB 単位)。この制限に達すると、新しいログ・ファイルが生成されます。
- max-directory-size: ログ・ファイル・ディレクトリに対して許容される最大サイズ (KB 単位) を設定します。この制限を超えると、最も古いログ・ファイルから消去されます。

たとえば、petstore アプリケーションの orion-application.xml ファイルの次のエントリは、このアプリケーションに対して log.xml ファイルを生成します。また、ログ・ファイルの最大値を 1,000KB、ディレクトリの最大値を 10,000KB に設定します。

```
<log>
  <odl path="../log/petstore/" max-file-size="1000" max-directory-size="10000" />
</log>
```

この構成を使用して、petstore ログ・ファイルは、使用している OC4J インストールに応じて、次の場所に生成されます。

- スタンドアロン OC4J:
ログ・ファイルは、`ORACLE_HOME/j2ee/home/application-deployments/log/petstore` に生成されます。
- OPMN 管理 OC4J:
ファイルは、`ORACLE_HOME/j2ee/instance/application-deployments/log/instance_default_group_1/petstore` という OC4J インスタンス固有のディレクトリに生成されます。

ODL ログ・ファイルの管理

ODL フレームワークでは、ログ・ファイルの循環などログ・ファイルの管理をサポートしています。ログ・ファイルの最大サイズおよびログ・ディレクトリの最大サイズも定義できます。また、OLD を使用すると次のような利点があります。

- 保存する診断情報の合計容量を制限できます。
- 作成された順序に従って、古いセグメント・ファイルは削除され、新しいセグメント・ファイルは保存されます。
- コンポーネントをアクティブなままにできるため、診断ログ・ファイルがクリーンアップされる際にも停止する必要がありません。

ODL ログは、現行のログ・ファイル `log.xml`、および古いメッセージが含まれるゼロ以上の OLD アーカイブ (セグメント・ファイル) を含む一連のログ・ファイルです。ODL を有効にすると、新しいメッセージはそれぞれ、`log.xml` の最後に追加されます。このログ・ファイルは循環ポイントに達すると名前が変更され、新しい `log.xml` ファイルが作成されます。

現行のログ・ファイルが循環ポイントに達するとセグメント・ファイルが作成されます。この循環ポイントは最大 ODL セグメント・サイズとして、また、一部の OC4J ログでは循環時間および循環頻度として指定されています。`log.xml` ファイルは `logn.xml` という名前に変更されます。ここで `n` は 1 から始まる整数です。新しい `log.xml` ファイルは、コンポーネントにより新しい診断メッセージが生成されると作成されます。

最後のログ・ファイルがいっぱいになると、次の手順で処理されます。

1. ディレクトリの領域を空けるため、最も古いログ・ファイルが消去されます。
2. `log.xml` ファイルが最新の `logn.xml` ファイルに書き込まれます。`n` は、最新のログ・ファイルを 1 だけ増やした値です。

サイズベースのログ循環

アプリケーションまたはコンポーネントの ODL ログのサイズを制限するには、ログ・ディレクトリの最大サイズを指定する構成オプションを使用します。ディレクトリ内の全ファイルの合計サイズが最大サイズに達すると、合計サイズを指定された制限内に保つために最も古いアーカイブが削除されます。

注意：最新のセグメント・ファイルは削除されません。

たとえば、log9872 という名前のセグメント・ファイルで開始し、最大ディレクトリ・サイズに達したとき、ログ・ファイル・ディレクトリに次のファイルが存在するとします。

File	Size
log.xml	10002
log9872.xml	15000
log9873.xml	15000
log9874.xml	15000
log9875.xml	15000
log9876.xml	15000

この場合、log.xml がいっぱいになると log9872.xml が削除されて、log.xml の内容が新しいファイル log9877.xml に移動されます。その後、新しい診断メッセージが新しい log.xml ファイルに書き込まれます。

たとえば、petstore という名前の OC4J アプリケーションに最大の ODL セグメント・サイズおよび最大ディレクトリ・サイズを指定するには、ファイル `ORACLE_HOME/j2ee/instance_name/application-deployments/petstore/orion-application.xml` に次のエントリを追加します。

```
<log>
<odl path="../../log/petstore/" max-file-size="1000" max-directory-size="10000" />
</log>
```

j2ee-logging.xml ファイルに構成されている OC4J コンポーネントの場合は、最大セグメント・サイズおよびディレクトリ・サイズに加えて、循環時間と循環頻度も指定できます。

時間ベースのログ循環

時間ベースのログ循環では、<logging-configuration> ルート要素の <log-handlers> 要素の <log_handler> サブ要素に次に示すプロパティを指定できます。

- **baseRotationTime**: 循環のベース時間 (オプション)。ベース時間の書式は次のいずれかです。
 - hh:mm。04:20 など。この書式ではローカル・タイムゾーンを使用します。
 - yyyy-MM-dd。2006-08-01 など。この書式ではローカル・タイムゾーンを使用します。
 - yyyy-MM-ddThh:mm。2006-08-01T04:20 など。この書式ではローカル・タイムゾーンを使用します。
 - yyyy-MM-ddThh:mm:ss.sTZD。TZD はタイムゾーン・インジケータです。TZD は、UTC を示す Z または {+|-}hh:mm です。たとえば、2006-03-01T04:20:00-08:00 は米国太平洋標準時タイムゾーンの 2006 年 3 月 1 日 4:20:00 を表します。

baseRotationTime を指定しない場合、デフォルト値は UTC の 1970 年 1 月 1 日 00:00 です。

- **rotationFrequency**: 分単位の循環頻度。その他に、hourly、daily、weekly のいずれかの値を指定できます。

これらのプロパティは次のファイルに指定します。

```
ORACLE_HOME/j2ee/instance_name/config/j2ee-logging.xml
```

たとえば、毎日ローカル時間の午前 4 時、またはサイズが 2000000 バイトに達した時にログ・ファイルを循環すると指定するには、次のようにします。

```
<log_handler name="h1" class="oracle.core.ojdl.logging.ODLHandlerFactory">
  <property name="path" value="log"/>
  <property name="baseRotationTime" value="04:00"/>
  <property name="rotationFrequency" value="daily"/>
  <property name="maxFileSize" value=" 2000000"/>
</log_handler>
```

ODL ログ・ファイルの表示

ODL 形式のログ・ファイルは、Application Server Control で「ログ」リンクをクリックすると表示でき、管理者は、一元管理された場所から OC4J 内で稼働しているすべてのコンポーネントおよびアプリケーションによって生成されたログ出力を集計および表示できます。

ODL ログ・ファイルは、「ログ・ファイル」ページでは .xml 拡張子で識別されます。

1. Application Server Control のページの一番下にある「ログ」リンクをクリックします。
2. 「OC4J」を開きます。
3. 「<instanceName>」を開きます。スタンドアロン OC4J と OAS のいずれでも、デフォルト・インスタンス名は home です。
 - OC4J ログ・ファイルを表示するには、**診断メッセージ・ログ**を開いた後、log.xml を開きます。
 - 特定の J2EE アプリケーションの ODL ログを表示するには、次のようにします。
 - 「アプリケーション <applicationName>」を開きます。
 - **診断メッセージ・ログ**を開きます。このディレクトリ内に生成された log.xml ファイルを開いて表示します。

OC4J ログの構成

OC4J の各種コンポーネントでは、OC4J ログ・ファイルに書き込む Java ログ出力を利用します。OC4J ログ・ファイルは、Oracle Diagnostic Logging フレームワークを使用して XML 形式で生成され、Application Server Control を使用して表示できます。

この項の内容は次のとおりです。

- [OC4J コンポーネント・ログ出力の使用手法と構成](#)
- [OC4J ログ・ファイルの表示](#)
- [oracle ログ出力の構成](#)

OC4J コンポーネント・ログ出力の使用手法と構成

OC4J には、OC4J ログ・ファイル (log.xml) に書き込む多くのコンポーネント・ログ出力が用意されています。使用可能なコンポーネント・ログ出力は、Application Server Control の「ログ出力の構成」ページで表示および構成できます。

Java のログ・レベルは、個々のコンポーネント・ログ出力ごとに設定できます。NULL に設定すると、ログ出力は、親に設定されたログ・レベルを継承します。

したがって、すべてのログ出力のデフォルト・レベルは、oracle ログ出力から継承されるデフォルト値である INFO です。この値は、Java ログ・レベル NOTIFICATION にマップされます。このデフォルト値の変更方法は、11-10 ページの「[oracle ログ出力の構成](#)」を参照してください。

「ログ出力の構成」ページからログ出力に設定されたログ・レベルは保存されず、OC4J ランタイムにのみ適用されます。OC4J が再起動されると、ログ・レベルは親ログ出力から継承されるデフォルト設定に戻ります。

表 11-2 に、Application Server Control を使用して設定できるログ・レベルと、各 Java ログ・レベルのマップ先となる ODL メッセージ・タイプ:ログ・レベルを示します。

表 11-2 OC4J コンポーネント・ログ出力のログ・レベル

Java ログ・レベル	ODL メッセージ・タイプ: ログ・レベル	ODL の説明
NULL		ログ出力は、親に設定されたログ・レベルを継承します。
SEVERE	ERROR:1	システム管理者が注意する必要があるシステム・エラーを記録します。
WARNING	WARNING:1	エラーが発生する前に確認して処理する必要があるとして検出されたアクションまたは条件を記録します。
INFO	NOTIFICATION:1	標準のアクションまたはイベントを記録します。ユーザー操作（「ログイン完了」など）や自動操作（ログイン・ファイルの循環など）が該当します。
CONFIG	NOTIFICATION:16	構成関連のメッセージまたは問題を記録します。
FINE	TRACE:1	デバッグやパフォーマンスの監視に使用されるトレースまたはデバッグのメッセージを記録します。通常、詳細なイベント・データが含まれます。
FINER	TRACE:16	非常に詳細なトレースまたはデバッグのメッセージを記録します。
FINEST	TRACE:32	きわめて詳細なトレースまたはデバッグのメッセージを記録します。

Application Server Control を使用して OC4J コンポーネント・ログ出力を構成する手順：

1. 「管理」リンクをクリックします。
2. 「ログ出力の構成」をクリックします。
3. ログ・レベルを表 11-2 の左列に示された値に設定します。
4. 「適用」をクリックして、変更を OC4J ランタイムに適用します。

OC4J ログ・ファイルの表示

OC4J ログ・ファイルは、Application Server Control を使用して表示できます。ファイルを表示するには、次のようにします。

1. Application Server Control のページの一番下にある「ログ」リンクをクリックします。
2. 「OC4J」を開きます。
3. 「<instanceName>」を開きます。
4. 診断メッセージ・ログを開きます。

すべての ODL ログ・ファイルと同様に、新しいメッセージはそれぞれ、log.xml という最新のログ・ファイルに格納されます。最大サイズに達すると、ログは logn.xml というアーカイブ・ログ・ファイルにコピーされます。n は、1 から始まる整数です。

oracle ログ出力の構成

oracle ログ出力の構成は、j2ee-logging.xml に定義されます。このファイルは、`ORACLE_HOME/j2ee/instance/config` ディレクトリにインストールされています。

Application Server Control を使用して oracle ログ出力を構成する手順：

1. OC4J ホームページで「管理」をクリックします。
2. 管理タスクから、「ログ出力の構成」を選択して「ログ出力の構成」ページを表示します。
3. 「すべてを開く」をクリックして、現在 OC4J インスタンスにロードされているログ出力の全リストを表示します。
4. ページに表示されている任意のログ出力のログ・レベルを選択します。

j2ee-logging.xml で oracle ログ出力を構成する手順：

また、j2ee-logging.xml 構成ファイルは手動で編集することもできます。このファイルの変更後には、OC4J を再起動してください。

構成ファイルには、<logging-configuration> ルート要素内に次の 2 つの要素があります。

■ <log_handlers>

この要素には、異なる 3 つのログ・ハンドラを定義する <log_handler> 要素が含まれます。

- oc4j-handler
oracle ログ出力用のログ・ハンドラです。
- oracle-webservices-management-auditing-handler
oracle.webservices.management.auditing ログ出力用のログ・ハンドラです。
- oracle-webservices-management-logging-handler
oracle.webservices.management.logging ログ出力用のログ・ハンドラです。

次のプロパティは、各ログ・ハンドラの <property> サブ要素に指定します。

- path: ハンドラがログ・ファイルを生成するディレクトリを指定します。この値は変更しないでください。
- maxFileSize: ディレクトリ内のログ・ファイルに対して許容される最大サイズ（バイト単位）を設定します。この制限を超えると、新しいファイルが生成されます。
- maxLogSize: ログ・ファイル・ディレクトリに対して許容される最大サイズ（バイト単位）を設定します。この制限を超えると、最も古いログ・ファイルから消去されます。

■ <loggers>

この要素には、次の情報を定義する <logger> 要素が含まれます。

- name: ログ出力名。この値は変更しないでください。
- level: このログ出力が機能する最低ログ・レベル。このレベルは、デフォルトで ODL の値 NOTIFICATION:1 に設定されます。この値は Application Server Control の「ログ出力の構成」ページに表示される Java ログ・レベル INFO にマップされず。
この値は、Java ロギング・レベル (FINE) または ODL メッセージ・タイプ: ログ・レベル (TRACE:1) のいずれかに設定できます。
- useParentHandlers: ログ出力で親ハンドラを使用する必要があるかどうかを示します。この値はデフォルトで false に設定されるため、oracle ログ出力は親 (root ログ出力) に設定されたログ・レベルを継承しません。
- <handler>: 使用するハンドラの名前。この値は変更しないでください。

次の例では、ODL メッセージ・タイプ: ログ・レベルとして TRACE:32 を指定し、デフォルトのログ・レベルを FINEST に設定しています。

```
<logging_configuration>
  <log_handlers>
    <log_handler name='oc4j-handler'
      class='oracle.core.ojdl.logging.ODLHandlerFactory'>
      <property name='path' value='%ORACLE_HOME%/j2ee/%OPMN_PROC_TYPE%/log/
        %OPMN_PROC_TYPE% %OPMN_PROC_SET% %OPMN_PROC_INDEX%/oc4j' />
      <property name='maxFileSize' value='10485760' />
      <property name='maxLogSize' value='104857600' />
    </log_handler>
  </log_handlers>
  <loggers>
    <logger name='oracle' level='TRACE:32' useParentHandlers='false'>
      <handler name='oc4j-handler' />
    </logger>
  </loggers>
</logging_configuration>
```

LogViewer を使用した OC4J ログのアプリケーション・メッセージの表示

アプリケーション・ログ・メッセージを OC4J ロギング・システムに送信し、LogViewer を使用して Application Server Control で表示できます。標準の JDK ロギング API を使用する場合、ログ・ハンドラとログ・レベルを構成して、アプリケーションからのログ・エントリを OC4J のメイン・ログである `ORACLE_HOME/j2ee/instance/log/oc4j/log.xml` に送信できます。このログは、LogViewer により、**診断ログ**として Application Server Control にリストされます。

LogViewer を使用して OC4J ログのアプリケーション・メッセージを表示する手順:

1. アプリケーションで、次に示すように、ログ出力ネーミング階層を使用し、異なるレベルでログ・メッセージを発行するようログ出力を設定します。

例 11-1 異なるログ・レベルを設定するためのアプリケーション・ログ出力

```
Logger logger = Logger.get("emp.app.web.EmployeeFrontEnd");
public void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException {response.setContentType(CONTENT_TYPE);
logger.fine(
String.format("Handling web request for %s", request.getRequestURL()));
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>EmployeeFrontEnd</title></head>");
out.println("<body>");
Employee test = Employee.getTestInstance();
logger.finest(
String.format("Test Employee Instance: %s", test));
logger.finest(
String.format("Calling %s to locate office for %s",
    employeeManager.toString(),
    test.identifier(test.ID_SHORT)));
String location = employeeManager.locateEmployeeOffice(test);
logger.finest(
String.format("bean returned %s for %s ",
    location, test.identifier(test.ID_SHORT)));
out.printf("<p>Employee: %s<br>Office: %s</p>", test.identifier(test.ID_SHORT),
location);
logger.fine(String.format("Employee currently earns $%s", test.getSalary()));
test.raiseSalary(15D);
```

```
out.printf("<p>Give employee 15percent raise, now earns %s", test.getSalary());
out.println("</body></html>");
out.close();
}
```

2. 次の例に示すように、j2ee-config.xml ファイルの <logging_configuration> 要素に、アプリケーション・ログ出力名とログ・レベルを指定し、ログ出力で oc4j-handler を使用するよう宣言する新しい <logger> 要素を追加します。

```
<logging_configuration>
  <log_handlers>
    <log_handler name="oc4j-handler"
      class="oracle.core.ojdl.logging.ODLHandlerFactory">
      <property name="path" value="./log/oc4j"/>
      <property name="maxFileSize" value="10485760"/>
      <property name="maxLogSize" value="104857600"/>
      <property name="encoding" value="UTF-8"/>
      <property name="supplementalAttributes" value="J2EE_APP.name,
        J2EE_MODULE.name,WEBSERVICE.name,WEBSERVICE_PORT.name"/>
    </log_handler>
  </log_handlers>
  <loggers>
    <logger name="oracle" level="NOTIFICATION:1" useParentHandlers="false">
      <handler name="oc4j-handler"/>
      <handler name="console-handler"/>
    </logger>
    <logger name="emp" level="FINEST"
      <handler name="oc4j-handler"/>
    </logger>
  </loggers>
</logging_configuration>
```

oracle ルート・ログ出力に書き込まれたすべてのメッセージは、oc4j-handler に送信されます。このハンドラは、ORACLE_HOME/j2ee/instance/log/oc4j/log.xml ファイルに XML 形式でメッセージを書き出します。

3. Application Server Control のページの一番下にある「ログ」リンクをクリックし、[図 11-1](#) に示すように、使用可能なすべてのログをリストします。

図 11-1 Application Server Control のログ出カリスト

ORACLE Enterprise Manager 10g
Application Server Control Setup Logs Help Logout

OC4J: home >

Log Files

View: All Logs Page Refreshed Sep 5, 2007 5:35:05 AM GMT

Select log files and...

Select All | Select None | Expand All | Collapse All

Select Item	Log Type	Modified	Size (bytes)	View	Search
<input type="checkbox"/> Components					
<input checked="" type="checkbox"/> Audit					
<input checked="" type="checkbox"/> Enterprise Manager					
<input type="checkbox"/> OC4J					
<input type="checkbox"/> home					
<input checked="" type="checkbox"/> Application EJB3-ExternalInterceptor-EAR					
<input checked="" type="checkbox"/> Application HRApp					
<input checked="" type="checkbox"/> Application ascontrol					
<input checked="" type="checkbox"/> Application aussietripper					
<input checked="" type="checkbox"/> Application default					
<input checked="" type="checkbox"/> Application fat					
<input checked="" type="checkbox"/> Application formauthfilter					
<input checked="" type="checkbox"/> Application h2c					
<input checked="" type="checkbox"/> Application how-to-cluster					
<input checked="" type="checkbox"/> Application javasso					
<input checked="" type="checkbox"/> Application system					
<input checked="" type="checkbox"/> Application test1-web					
<input checked="" type="checkbox"/> Application test2-web					
<input checked="" type="checkbox"/> Application test3-web					
<input checked="" type="checkbox"/> Application test4-app					
<input checked="" type="checkbox"/> Diagnostic Logs	Server	September 5, 2007 5:12:06 AM GMT	7,196,800		
<input checked="" type="checkbox"/> Web Services					
<input type="checkbox"/> jms.log	Server	September 5, 2007 3:39:57 AM GMT	8,162		
<input type="checkbox"/> rmi.log	Server	September 5, 2007 3:39:56 AM GMT	8,162		
<input type="checkbox"/> server.log	Server	September 5, 2007 3:39:57 AM GMT	11,432		

Select log files and...

- 「ログ・ファイル」ページで、「**診断ログ**」をクリックし、LogViewer により OC4J ログのメッセージが表示されるようにします。

図 11-2 に示すように、「診断ログ」ページには、OC4J サーバーのログ・エントリおよびアプリケーションのログ・エントリが表示されます。各ログ・メッセージには、web_EmployeeFrontEnd または ejb_EmployeeManagerBean など、ログ・エントリが生成されたコンポーネントが表示されています。

図 11-2 LogViewer の「診断ログ」ページ

ORACLE Enterprise Manager 10g
Application Server Control

OC4J: home > Log Files >

View Log: Diagnostic Logs

Page Refreshed Sep 5, 2007 5:42:11 AM GMT • View Data Manual Refresh

Log File [D:\java\oc4j-10133-prod\i2ee\home\log\oc4j\log.xml](#)
(Click on the log file link to download the log or view it as plain text)

Component Name **home** Component Type **OC4J**
Modified **September 5, 2007 5:12:06 AM GMT** Size (bytes) **7,281,411**
Log Type **Server**

Log File Contents

Log Entries Displayed Last 100

Log Message: September 5, 2007 5:40:26 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	1
Module ID	home web.EmployeeFrontEnd	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970826578:340	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Handling web request for http://localhost/hrapp/employeeemanager

Log Message: September 5, 2007 5:40:26 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	1
Module ID	home ejb.EmployeeManagerBean	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970826578:340	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Locating office for Fred Bloggs

Log Message: September 5, 2007 5:40:26 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	1
Module ID	home web.EmployeeFrontEnd	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970826578:340	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Employee currently earns \$55000.0

- 個々のリクエストのすべてのログ・エントリを順番に表示するには、ログ・エントリの「実行コンテキスト ID」(ECID) リンクをクリックします。

Oracle ロギング・フレームワークは、各ログ・メッセージに割り当てられた ECID に基づいて、同じ実行パスにある異なるコンポーネントからの様々なログ・エントリを関連付けることができます。図 11-3 に示すように、LogViewer により、選択した ECID のすべてのログ・エントリがタイムスタンプ順に表示されます。

図 11-3 順番に表示された実行コンテキストのログ・メッセージ

ORACLE Enterprise Manager 10g
Application Server Control

OC4J: home > Log Files > View Log: Diagnostic Logs >

Log Message Details for Execution Context: 10.189.38.107:90556:1188970927250:375

Page Refreshed Sep 5, 2007 5:45:46 AM GMT

Log Message: September 5, 2007 5:42:07 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	1
Module ID	home_web.EmployeeFrontEnd	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970927250:375	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Handling web request for http://localhost/hrapp/employeeemanager

Log Message: September 5, 2007 5:42:07 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	32
Module ID	home_web.EmployeeFrontEnd	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970927250:375	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Test Employee Instance: 1 Fred Bloggs fred.bloggs@acme.com 5/09/2007 \$55000

Log Message: September 5, 2007 5:42:07 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	32
Module ID	home_web.EmployeeFrontEnd	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970927250:375	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Calling EmployeeManager stateless session to locate office for Fred Bloggs

Log Message: September 5, 2007 5:42:07 AM GMT

Component Name	bar	Component ID	bar
Host Network Address	10.189.38.107	Message Level	1
Module ID	home_ejb.EmployeeManagerBean	User ID	sbutton
Execution Context ID	10.189.38.107:90556:1188970927250:375	Execution Context Sequence	0
Host Name	sbutton-m6	Thread ID	19
Message Type	Trace		

Message Text
Locating office for Fred Bloggs

- LogViewer の検索機能を使用して、OC4J ログの目的の項目を検索します。

Application Server Control を使用したアプリケーション・ログ出力の構成

アプリケーションが実行されてそのログ出力が登録された後、または `j2ee-config.xml` ファイルにそれらを構成した後は、Application Server Control を使用してアプリケーション・ログ出力を構成できます。「ログ出力の構成」ページを使用すると、アプリケーション・ロギングを簡単にカスタマイズできます。

Application Server Control を使用してアプリケーション・ログ出力を構成する手順：

1. OC4J ホームページで「管理」をクリックします。
2. 管理タスクから、「ログ出力の構成」を選択して「ログ出力の構成」ページを表示します。
3. 「すべてを開く」をクリックして、現在 OC4J インスタンスにロードされているログ出力の全リストを表示します。

OC4J ログ出力およびアプリケーション・ログ出力が、各ログ出力のレベルを指定できる選択リストとともに表示されます。

4. ページにリストされている任意のログ出力のログ・レベルを選択します。

アプリケーションの log4j メッセージの OC4J ログへのリダイレクト

Oracle Application Server ディストリビューションには、OracleAppender クラスを含む JAR ファイル (`ORACLE_HOME/diagnostics/lib/ojdl-log4j.jar`) が同梱されています。このクラスは Oracle log4j アペンダで、log4j メッセージを Oracle Diagnostic Logging (ODL) の XML 形式に変換します。アプリケーションでこのアペンダを使用するには、アペンダを構成し、アプリケーションに対して log4j クラス・ライブラリを使用可能にする必要があります。

`log4j.properties` ファイルには、適切な log4j 設定と有効化する Java ログ・レベルを構成できます。例 11-2 に示すように、OracleAppender を使用して、log4j エントリを取得し、Oracle ログ・システムにルーティングできます。

例 11-2 Oracle log4j アペンダの構成ファイル

```
log4j.rootLogger=TRACE,OJDL
log4j.appender.OJDL=oracle.core.ojdl.log4j.OracleAppender
log4j.appender.OJDL.LogDirectory=${oracle.j2ee.home}/log/oc4j
#log4j.appender.APP1.MaxSize=1000000
#log4j.appender.APP1.MaxSegmentSize=200000
#log4j.appender.APP1.Encoding=iso-8859-1
log4j.appender.OJDL.ComponentId=OracleProd
```

この構成では、log4j メッセージが `ORACLE_HOME/j2ee/home/log/oc4j/log.xml` ファイルに送信されます。LogViewer によりこのファイルが読み取られ、Application Server Control に診断ログとして表示されます。

Oracle log4j アペンダを構成したら、OC4J 共有ライブラリ・メカニズムを使用して、log4j クラス・ライブラリを OC4J に公開し、アプリケーションにライブラリをインポートできます。

アプリケーションへの Oracle log4j アペンダの構成

log4j.properties ファイルの OracleAppender クラスを使用して、アプリケーションに Oracle log4j アペンダを構成できます。その後、デプロイ中にアプリケーションに構成をインポートできます。

デプロイ時にアプリケーションに log4j.properties ファイルをインポートするには、OC4J 共有ライブラリ・メカニズムを使用します。OC4J により、クラス・パスから properties ファイルが読み取られます。log4j.properties ファイルを含む共有ライブラリをアプリケーションにインポートすると、ファイルへのアクセスが可能になり、アプリケーションに log4j を構成できます。

異なるログ・レベルを有効化し、アプリケーションのデプロイ時にファイルを選択するための、異なる log4j.properties ファイルを含む一連の共有ライブラリを OC4J に公開することもできます。デプロイ後、異なるログ設定の切替えを行えるように、アプリケーションが使用する log4j.properties ファイルを変更できます。

アプリケーションに Oracle log4j アペンダを構成する手順：

1. log4j.config.info.jar など、構成済ログ・レベルを示すファイル名を使用して、JAR ファイルに log4j.properties ファイルを配置します。
2. JAR ファイルを含む OC4J 共有ライブラリを作成します。

図 11-4 に、Application Server Control を使用して OC4J インスタンスに共有ライブラリを作成する方法を示します。

図 11-4 log4j.properties ファイルを含む OC4J 共有ライブラリ

The image shows two screenshots of the Oracle Enterprise Manager 10g Application Server Control interface. The top screenshot shows the 'Create Shared Library: Attributes' step, where the 'Shared Library Name' is 'log4j.config.info' and the 'Shared Library Version' is '1.0'. The bottom screenshot shows the 'Add Archives: Add Archive' step, where the 'File is present on local host' option is selected, and the 'Location on Local Host' is 'D:\temp\log4j\log4j.config.info.jar'. The 'Add Archives' step also shows options for 'File is present on the server where Application Server Control is running' and 'File is already present on the server where the target OC4J instance is running'.

ORACLE Enterprise Manager 10g
Application Server Control

Setup Logs Help Logout

Attributes Add Archives Import Shared Libraries

Create Shared Library: Attributes

Cancel Step 1 of 3 Next

* Shared Library Name log4j.config.info

* Shared Library Version 1.0

Cancel Step 1 of 3 Next

Setup | Logs | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
[About Oracle Enterprise Manager 10g Application Server Control](#)

ORACLE Enterprise Manager 10g
Application Server Control

Setup Logs Help Logout

Attributes Add Archives Import Shared Libraries

Add Archives: Add Archive

Shared Library Name log4j.config.info Shared Library Version 1.0

Cancel Continue

File is present on local host. Upload file to this shared library's directory on the target OC4J instance.
Location on Local Host D:\temp\log4j\log4j.config.info.jar Browse...

File is present on the server where Application Server Control is running. Upload file to this shared library's directory on the target OC4J instance.
Location on AS Control Server

The location on the server must be an absolute path or a path relative to j2ee/home

File is already present on the server where the target OC4J instance is running.
Location on Target Server

The location on the server must be an absolute path

Cancel Continue

Setup | Logs | Help | Logout

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
[About Oracle Enterprise Manager 10g Application Server Control](#)

図 11-5 には、アプリケーションにインポートできる、log4j.properties ファイルを含む 2 つの異なる共有ライブラリがあることが示されています。

図 11-5 log4j 構成の共有ライブラリ

ORACLE Enterprise Manager 10g
Application Server Control

OC4J: home >

Confirmation
Created the shared library log4j.config.info-1.0

Shared Libraries
A shared library is used to share a set of code sources across applications. You define an application's dependency on a shared library when deploying the application. This page allows you to manage the shared libraries in this OC4J instance.

Shared Library	Version	Archives	Shared Library Imports	Applications Importing Library	Shared Libraries Importing Library	System Shared Library	Delete
aaa	1.0	1	0	0	0		
apache.commons.logging	1.0.4	1	0	0	2	✓	
AuditInterceptor	1.0	1	0	0	0		
AuditInterceptor	1.1	1	0	0	0		
global.libraries	1.0	1	0	5	0		
global.taq.libraries	1.0	3	4	6	0		
log4j.config.info	1.0	1	0	0	0		
log4j.config.trace	1.0	1	0	1	0		

3. log4j.config.info など、log4j.properties ファイルを含む共有ライブラリを、デプロイ中にアプリケーションにインポートします。
 - a. 図 11-6 に示すように、「デプロイ・タスク」ページで「クラスのロードの構成」を選択します。

図 11-6 「デプロイ・タスク」の「クラスのロードの構成」

Deployment Tasks
The table below provides a set of common deployment tasks you might want to perform for this application. Only those tasks that apply to the current application are enabled.

Task Name	Go To Task	Description
Map Environment References		Map any environment references in your application (for example, data sources) to physical entities currently present on the operational environment.
Select Security Provider		A security provider acts as the source for available users and groups when mapping security roles.
Map Security Roles		Map any security roles exposed by your application to existing users and groups. The list of users and groups is obtained from the security provider you selected for this application.
Configure EJBs		Configure the Enterprise JavaBeans in your application.
Configure Clustering		Configure clustering of your application.
Configure Class Loading		Manipulate the classpath of your application.

- b. 図 11-7 に示すように、「クラスのロードの構成」ページの「共有ライブラリのインポート」領域で共有ライブラリを選択します。

図 11-7 インポートするアプリケーションの共有ライブラリ

ORACLE Enterprise Manager 10g
Application Server Control

Select Archive Application Attributes **Deployment Settings** Help Logout

Deployment Settings: Configure Class Loading Cancel OK

Archive Type J2EE Application (EAR file) Application Name HRApp
 Archive Location EAR.ear Parent Application default
 Deployment Plan Creating a new plan Bind Web Module to Site default-web-site
 Context Root /hrapp

Import Shared Libraries
 The following table lists the shared libraries installed in this OC4J instance. Select Import to declare your application's dependency on a shared library. Optionally specify a minimum or maximum version to import.
 Inherit parent application's shared library imports
 TIP When checked, future changes to the parent application's shared library imports will be effective to this application.

Shared Library	Available Versions	Minimum Version To Use	Maximum Version To Use	Imported By Parent Application	Import
aaa	1.0				<input type="checkbox"/>
apache.commons.logging	1.0.4				<input type="checkbox"/>
AuditInterceptor	1.0, 1.1				<input type="checkbox"/>
global.libraries	1.0			✓	<input checked="" type="checkbox"/>
global.tag.libraries	1.0			✓	<input checked="" type="checkbox"/>
log4j_config.info	1.0				<input checked="" type="checkbox"/>
log4j_config.trace	1.0				<input type="checkbox"/>
oracle.cache	10.1.3			✓	<input checked="" type="checkbox"/>
oracle.dms	3.0			✓	<input checked="" type="checkbox"/>
oracle.gdk	10.1.0_2			✓	<input checked="" type="checkbox"/>

log4j.info 共有ライブラリにより、アプリケーションの log4j ログ・エントリの処理方法が決定されます。この場合、ルート・ログ出力は INFO レベルに設定され、OC4J ログ・システムへのログ・エントリの送信に OracleAppender クラスが使用されます。

最終的に、アプリケーションのカスタマイズ済の共有ライブラリ設定は、[図 11-8](#) に示すように、`orion-application.xml` 構成ファイルに書き込まれます。

図 11-8 orion-application.xml の共有ライブラリ設定

```
<orion-application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  schema-major-version="10" schema-minor-version="0" >
  <ejb-module remote="false" path="EJB.jar" />
  <web-module id="WEB" path="WEB.war" />
  <persistence path="persistence" />
  <imported-shared-libraries>
    <import-shared-library name="log4j.config.info"/>
  </imported-shared-libraries>
  <principals path="principals.xml" />
  <jazn provider="XML" />
  <log>
    <file path="application.log" />
  </log>
</orion-application>
```

アプリケーションで DEBUG や TRACE などのさらに低いログ・レベルを使用するよう変更する場合は、`orion-application.xml` の `<import-shared-library>` 要素を変更し、アプリケーションを再起動して、関連する `log4j.properties` ファイルが含まれる共有ライブラリをインポートします。OC4J 共有ライブラリ・メカニズムと一貫性のあるネーミング規則を使用すると、再利用可能な log4j 構成をアプリケーションに必要なだけ作成できます。

Oracle log4j アペンダの構成後には、単一の `log.xml` ファイルに、log4j からのログ・メッセージと OC4J からのログ・エントリが含まれます。

アプリケーションによる log4j クラス・ライブラリの使用

アプリケーションで Oracle log4j アペンダを使用するには、アプリケーションに対して log4j クラスを使用可能にする必要があります。これらのクラスは、log4j および `ojd1-log4j` ライブラリにあります。次のいずれかの方法で、アプリケーションに対してクラスを使用可能にできます。

- アプリケーションにライブラリを含めます。

Java EE 5 アプリケーションの場合は、`<library-directory>` 機能を使用して、共有ライブラリを格納する EAR ファイル内のディレクトリを指定し、そのディレクトリに log4j および `ojd1-log4j` ライブラリを配置します。`application.xml` ファイルの `<library-directory>` 要素で、アプリケーションの共有ライブラリを指定します。これにより、起動時に含めるアーカイブ用に指定したディレクトリが OC4J によりスキャンされます。
- log4j および `ojd1-log4j` ライブラリを含む共有ライブラリを OC4J に作成し、そのデプロイ中にアプリケーションに共有ライブラリをインポートします。デプロイすると、アプリケーションでライブラリを使用できるようになります。
- Web アプリケーションがある場合は、log4j および `ojd1-log4j` ライブラリを `WEB-INF/lib` ディレクトリに配置すると、アプリケーションでそれらを使用できます。

`ojd1-log4j` ライブラリは log4j ライブラリに依存しているため、同じクラス・ローダー・レベルでその両方がアクセス可能であることが必要です。

OC4J での MBean の使用方法

この章では、OC4J に付属するシステム MBean を使用して、OC4J インスタンス内のデプロイ済アプリケーション、サービスおよびその他のリソースを管理する方法について説明します。次の項目が含まれます。

- [OC4J での MBean および Java Management Extensions \(JMX\) のサポート](#)
- [システム MBean ブラウザの使用方法](#)
- [JMX 通知のサブスクリプション](#)
- [アプリケーション固有の MBean の使用方法](#)

OC4J での MBean および Java Management Extensions (JMX) のサポート

OC4J では、Java Management Extensions (JMX) 1.2 仕様をサポートします。この仕様は、J2EE 環境でリソース（サービス、アプリケーションなど）を管理するための標準インタフェースを作成できるようにします。

Oracle Enterprise Manager 10g Application Server Control のユーザー・インタフェースは、OC4J インスタンスの完全管理および監視に使用できる JMX 準拠のクライアントに基づいて構築されています。Application Server Control に提供されている JMX 機能は、MBean として知られる Java コンポーネントを使用して有効化されます。MBean については、次の項で説明します。

OC4J 内の JMX 管理が可能なリソースには、次のものがあります。

- OC4J サーバー
- OC4J インスタンス内で稼働するアプリケーションおよび Web モジュール
- J2EE サービス (JTA や JMS など)
- OC4J プロセス (タスク・マネージャなど)
- データソースおよびセキュリティ構成

この項の内容は次のとおりです。

- [MBeans の概要](#)
- [トップレベルの OC4J システム MBean の概要](#)
- [MBean を介して加えた変更を有効にする時期](#)
- [MBean データの永続性](#)

MBeans の概要

MBean (マネージド Bean) は、JMX 管理が可能なリソースを表す Java オブジェクトです。MBean は、Sun 社が公開している J2EE 1.4 仕様の一部である J2EE Management 仕様 (JSR-77) に定義されています。

OC4J 内の管理可能なリソースは、それぞれ対応する MBean のインスタンスによって管理されます。たとえば、OC4J の起動時に、J2EESWebSite MBean のインスタンスは、サーバー内で構成された各 Web サイトを表すために作成されます。

OC4J に付属する各システム MBean は、システム MBean ブラウザからアクセスできる管理インタフェースを公開します。MBean のインタフェースは、次のもので構成されます。

- 属性: JMX クライアントでリモートに取得または設定できるタイプの名前 / 値ペア。属性は、Enterprise JavaBeans (EJB) モジュールに設定されるプロパティに似ています。たとえば、J2EEApplication:petstore MBean の state 属性は、アプリケーションが現在稼働しているかどうかを示します。
- 操作: JMX クライアントが MBean で起動できるメソッド。たとえば、stop 操作は、petstore アプリケーションとすべての子アプリケーションを停止するために使用できます。
- 通知: 新規アカウントの作成時などに、生成されるブロードキャスト・エラーまたは特定のイベントです。たとえば、通知を送信して、petstore アプリケーションが停止していることを警告できます。

前述したように、Application Server Control のアプリケーションは、システム MBean に基づいて構築されています。ユーザー・インタフェースでプロパティを設定したりタスクを実行する場合、実際には基礎となる MBean で属性を設定したり操作を起動します。

柔軟性を高めるために、Application Server Control では、システム MBean ブラウザのコンポーネントを使用して、OC4J に付属するシステム MBean に直接アクセスできるようにもしています。この管理ツールの使用法は、12-5 ページの「システム MBean ブラウザの使用法」を参照してください。

トップレベルの OC4J システム MBean の概要

次の表に、システム MBean ブラウザのインタフェースを介して公開されるトップレベルの OC4J システム MBean の概要を示します。

表 12-1 トップレベルの OC4J システム MBean

MBean	説明
J2EEDomain	管理ドメインを表します。これは、トップレベルの管理オブジェクトです。このドメインにバインドされた他のすべての MBean は、システム MBean ブラウザでこのノードの下に表示されます。
J2EEServer	単一の OC4J インスタンスを表します。
ClassLoading	OC4J インスタンス内のクラスのロードに関連するすべての状態にアクセスできるようにします。稼働している OC4J インスタンスでのクラスのロードに関する問題のトラブルシューティングを支援するために用意された 15 を超える組込み問合せを実行する操作が含まれます。 この MBean は、それぞれインスタンス化されたクラス・ローダーを表す ClassLoader MBean のインスタンスを遅れて作成します。
EJBCompiler	EJB のデプロイ時に、クライアント・サイドの IIOP スタブを生成するように、OC4J インスタンスを構成します。また、EJB モジュールのコンパイルに使用するコンパイラの指定にも使用されます。
J2EEApplication	OC4J インスタンスにデプロイされる J2EE アプリケーションを表します。 追加の MBean インスタンスは、アプリケーションの様々なコンポーネントを表す子ノードとして表示されます。 <ul style="list-style-type: none"> OC4JWebModule: J2EE アプリケーションの一部としてデプロイされた Web モジュール用に生成される OC4J 固有の orion-web.xml デプロイメント・ディスクリプタを使用して設定されるプロパティを表します。 WebModule: WAR ファイルにパッケージされる J2EE web.xml デプロイメント・ディスクリプタを使用して設定されるプロパティを表します。JSP MBean および Servlet MBean のインスタンスは、Web モジュール内のアクティブな JSP およびサーブレット用に作成されます。
J2EELogging	j2ee-logging.xml ファイルに定義される Java ログ出力コンポーネントを表します。Java ロギング・フレームワークの概要 (ログ・レベルを含む) は、Sun 社のサイト http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/overview.html を参照してください。
J2EEWebSite	OC4J サーバー内で定義された Web サイトを表します。Web サイト構成の詳細は、第 13 章「OC4J での Web サイトの管理」を参照してください。
JDBCdriver	特定の JDBC ドライバを表します。
JMSAdministratorResource	OC4J インスタンスで使用される OC4J JMS サーバーを表します。OC4J JMS サーバーおよび JMS コネクション・ファクトリの管理の他、接続先の追加 / 削除のための操作が含まれます。
JMSResource	JMS サーバーのメッセージ (タイプ別)、アクティブなハンドラ、アクティブな接続に関する統計を表示します。子 MBean には、接続、接続先および永続サブスクライバ・リソースに関する統計が含まれます。
JNDINamespace	OC4J インスタンスにデプロイされたすべてのアプリケーションに対する JNDI バインディングがすべて格納される XML 文書を返します。
JNDIResource	特定のアプリケーションに対する JNDI バインディングをすべて返します。

表 12-1 トップレベルの OC4J システム MBean (続き)

MBean	説明
JSPConfig	OC4J JSP コンテナを構成します。様々な構成値の詳細は、『Oracle Containers for J2EE JavaServer Pages 開発者ガイド』を参照してください。MBean 属性に加えられた変更を有効にするには、OC4J サーバーを再起動する必要があります。
JTAResource	トランザクション・マネージャのインスタンスを表します。この MBean で configureCoordinator 操作を起動するには、OC4J サーバーを再起動して新しい 2 フェーズ・コミット・コーディネータの構成を有効にする必要があります。
JVM	OC4J インスタンスが稼働する Java 仮想マシンを記述します。システム・プロパティを取得 / 設定し、ガベージ・コレクションを強制起動する操作が含まれます。
SecurityProvider	特定のアプリケーションのセキュリティ管理に使用されます。一部の属性および操作を有効にするために、対応するアプリケーションまたは OC4J サーバーを再起動する必要があります。
TaskManager	OC4J タスク・マネージャのインスタンスを記述します。この MBean は、タスク・マネージャの精度の設定に使用されます。
ThreadPool	単一のインスタンス化されたスレッド・プールを表します。プール内のスレッドの最大数および最小数の設定に使用します。
TimerService	EJB タイマーのインスタンスを表します。詳細は、『Oracle Containers for J2EE Enterprise JavaBeans 開発者ガイド』を参照してください。

MBean を介して加えた変更を有効にする時期

コンポーネントの停止中または稼働中に、MBean を介して管理コンポーネントを変更できます。

一般に、管理コンポーネントに加えた変更 (属性に設定される値または操作の結果) は、OC4J ランタイムですぐに使用できます。

しかし、場合によっては、新しい属性値または操作結果を OC4J ランタイムで使用できるようにするために、OC4J サーバー、影響を受けるアプリケーション、あるいは MBean まで再起動する必要があります。このような場合、MBean および Application Server Control には新しい値が表示されますが、必要な再起動が完了するまで、古い値は OC4J ランタイムで引き続き使用されます。

たとえば、JSPConfig MBean の timeout 属性の値を 30 から 15 に変更するとします。新しい値の 15 は、MBean と Application Server Control の「JSP コンテナのプロパティ」ページの両方に表示されます。しかし、JSPConfig 属性に対するすべての変更には OC4J サーバーの再起動が必要であるため、サーバーが再起動されるまで古い値の 30 が引き続き使用されます。

再起動が必要な場合、システム MBean ブラウザには必須アクションを示す必要な再起動プロパティが表示されます。次の表 12-2 に、このプロパティの値を示します。

表 12-2 必要な再起動プロパティの値

値	影響
OC4J の再起動	OC4J インスタンスを再起動する必要があることを示します。
アプリケーションの再起動	MBean が登録されている J2EE アプリケーションを再起動する必要があることを示します。このカテゴリに属する MBean は、コンソールの左側のナビゲーション・ペインで J2EEApplication ノードの下に表示されます。
MBean の再起動	影響を受ける MBean を再起動する必要があることを示します。

変更は、MBean レベルではなく、個々の属性 / 操作レベルで管理されます。つまり、MBean には、新しい値をランタイムで使用できるようにするために再起動が必要な属性と、すぐに変更できるようになるその他の属性を含めることができます。

MBean データの永続性

MBean を介して設定される永続データは、対応する XML 構成ファイルに書き込まれます。たとえば、JSPConfig MBean の属性に設定された新しい値は、global-web-application.xml 構成ファイルに書き込まれます。

MBean がデータを保持するかどうかは、システム MBean ブラウザに表示される「永続ポリシー」プロパティによって示されます。

表 12-3 「永続ポリシー」プロパティの値

値	影響
更新時	MBean に設定された永続データは、属性の変更が適用される時や操作が起動される時に、対応する構成ファイルに即時書き込まれます。
行わない	MBean に設定されたデータは保持されず、ランタイム・メモリーにのみ存在します。

システム MBean ブラウザの使用方法

システム MBean ブラウザは、Web ベースの Application Server Control のユーザー・インタフェースのコンポーネントで、比較的簡単に使用できます。この機能を使用するには、次のようになります。

1. Application Server Control を起動します。
2. 「管理」リンクをクリックします。
3. 「システム MBean ブラウザ」をクリックします。
4. 特定の MBean インスタンスには、コンソールの左側のナビゲーション・ペインからアクセスします。ナビゲーション・ペインでノードを開き、アクセスする MBean までドリルダウンします。
5. 右側のペインで、「属性」タブをクリックして選択した MBean の属性にアクセスします。属性値を変更した場合は、**変更の適用**ボタンをクリックして OC4J ランタイムに変更を適用します。

注意： **変更の適用**ボタンは、ブラウザのページに変更可能な値を持つ属性が 1 つ以上表示される場合にのみ表示されます。

6. 「操作」タブをクリックして MBean の操作にアクセスします。特定の操作を選択した後、「起動」ボタンをクリックしてコールします。

JMX 通知のサブスクリプション

OC4J に付属する多くのシステム MBean には、MBean によって登録された状態の変更によってトリガーされる通知を生成する機能があります。この項では、MBean 生成の通知をサブスクリプションおよび表示する方法について説明します。

すべての MBean が通知を生成するわけではありません。

システム MBean ブラウザまたは「通知サブスクリプション」ページから、通知をサブスクリプションできます。

システム MBean ブラウザから 1 つ以上の MBean の通知をサブスクリプションするには、次のようにします。

1. Application Server Control で「**管理**」リンクをクリックします。
2. 「**システム MBean ブラウザ**」をクリックします。
3. 特定の MBean インスタンスには、コンソールの左側のナビゲーション・ペインからアクセスします。ナビゲーション・ペインでノードを開き、アクセスする MBean までドリルダウンします。
4. 右側のペインで、「**通知**」タブをクリックして選択した MBean の通知にアクセスします。このタブが表示されない場合は、この MBean は通知を生成しません。
5. 「**サブスクリプション**」ボックスを選択します。
6. 「**適用**」ボタンをクリックします。

「通知サブスクリプション」ページから複数の MBean によって生成される通知をサブスクリプションするには、次のようにします。

1. Application Server Control で「**管理**」リンクをクリックします。
2. 「**通知サブスクリプション**」アイコンをクリックします。通知を生成する MBean がすべて表示されます。
3. サブスクリプションする通知ごとに「**サブスクリプション**」を選択します。
4. 「**適用**」ボタンをクリックします。

アプリケーション固有の MBean の使用方法

J2EE アプリケーションとともに OC4J にデプロイされるベンダー提供の MBean には、Application Server Control のユーザー・インタフェースのアプリケーションのホームページからアクセスできます。OC4J システム MBean と同様に、ユーザー・インタフェースを介して、アプリケーション固有の MBean で属性の表示および設定と、操作の起動ができます。

1. Application Server Control で「**アプリケーション**」リンクをクリックします。
2. MBean が属するアプリケーションの名前をクリックします。これにより、アプリケーションのホームページが開きます。
3. 「**アプリケーション定義の MBeans**」リンクをクリックします。アプリケーションによって定義された MBean がページにリスト表示されます。
4. 右側のペインで、「**属性**」タブをクリックして選択した MBean の属性にアクセスします。属性値を変更した場合は、**変更の適用** ボタンをクリックして OC4J ランタイムに変更を適用します。

注意： **変更の適用** ボタンは、ブラウザのページに変更可能な値を持つ属性が 1 つ以上表示される場合にのみ表示されます。

5. 「**操作**」タブをクリックして MBean の操作にアクセスします。特定の操作を選択した後、「**起動**」ボタンをクリックして実行します。

OC4J での Web サイトの管理

この章では、OC4J インスタンスにデプロイされた Web アプリケーションにアクセスできるように、追加 Web サイトを構成する方法について説明します。また、HTTPS を使用して、クライアントと OC4J の間で Secure Socket Layer (SSL) 通信を利用するセキュアな Web サイトを構成および有効化する方法についても説明します。

この章の内容は次のとおりです。

- [OC4J の Web サイトの概要](#)
- [Web サイト接続データの構成](#)
- [OC4J での Web サイトの新規作成](#)
- [OC4J でのセキュアな Web サイトの構成](#)
- [Web サイトの起動と停止](#)
- [Web サイトのアクセス・ロギングの構成](#)

OC4J の Web サイトの概要

OC4J では、OC4J インスタンスにデプロイされたアプリケーションに送信される Web リクエストは、Web サイトで受信されます。Web サイトとは、特定のプロトコルおよびポート（またはポートの範囲）でリクエストを受け入れるように構成されたリスナーです。OC4J インスタンスにデプロイされた各 Web モジュールは、そのアクセスに使用される Web サイトにバインドする必要があります。このバインディングは通常、アプリケーション・デプロイ・プロセスの一環として実行されます。

デフォルト Web サイトは、インストール時に各 OC4J インスタンス内に作成されます。デフォルト Web サイトの構成は、構成ファイル `default-web-site.xml` に定義されます。このファイルは `ORACLE_HOME/j2ee/instance/config` ディレクトリにデフォルトでインストールされます。Web サイトの構成の概要は、13-3 ページの「[Web サイト接続データの構成](#)」を参照してください。

■ スタンドアロン OC4J

スタンドアロン OC4J 構成では、デフォルト Web サイトは、HTTP リクエストを特定のポート（デフォルトでは 8888）で直接受信するように構成されます。あるいは、セキュアな HTTPS リクエストを受信するようにこのサイトを構成できます。

■ 単一の OPMN 管理 OC4J インスタンス

単一の OPMN 管理 OC4J インストールでは、デフォルト Web サイトは、同様に HTTP または HTTPS リクエストを直接受信するように構成できます。特定のリスナー・ポートを `default-web-site.xml` に指定するか、ポートの範囲を OPMN 構成ファイル (`opmn.xml`) に設定できます。詳細は、13-3 ページの「[OPMN 管理 OC4J インスタンスでの Web サイト・データの構成](#)」を参照してください。

■ 複数の OPMN 管理 OC4J インスタンス

2 つ以上の OPMN 管理 OC4J インスタンスのクラスタでは、デフォルト Web サイトは、Apache JServ Protocol (AJP) を介して Oracle HTTP Server から転送されたリクエストを受信するように構成されます。

あるいは、セキュアな AJP のリクエストを受信するようにこのサイトを構成できます。特定のリスナー・ポートを指定するか、ポートの範囲を OPMN 構成ファイルに設定できます。OPMN 構成の詳細は、13-3 ページの「[OPMN 管理 OC4J インスタンスでの Web サイト・データの構成](#)」を参照してください。

注意： 現行リリースでは、OC4J インスタンスでサポートされているのは一度に 1 つの AJP Web サイトのみです。

Oracle Application Server で AJP および AJP の両方を使用する場合は、それらを異なる OC4J インスタンスに構成し、2 つの Oracle HTTP Server (OHS) インスタンスを使用する必要があります。1 つは AJP リクエストのルーティング用で、もう 1 つは AJP のリクエストのルーティング用です。

デフォルト・サイトの他に、必要に応じて、新しい Web サイトを各 OC4J インスタンスで構成できます。(1 つの Web サイトは一度に複数のプロトコルでリスニングできません。) 次のいずれかの理由で、新しい Web サイトの作成が必要になる場合があります。

■ 管理アクセスと一般的な Web アクセスの分離

デフォルトでは、Application Server Control のアプリケーションは、デフォルト Web サイトから `/em` コンテキストを介してアクセスされます。しかし、必要に応じて Application Server Control のアプリケーション専用新しい Web サイトを作成し、管理アクセスと一般的なアプリケーション・アクセスを切り分けることができます。

- セキュアな Web サイトと非セキュアな Web サイトの利用
SSL を利用してセキュアな接続を作成するようにデフォルト Web サイトを構成できます。あるいは、追加サイトを作成し、セキュアな接続を必要とする Web アプリケーションにバインドすることもできます。

追加 Web サイトの作成および構成の詳細は、13-8 ページの「[OC4J での Web サイトの新規作成](#)」を参照してください。

Web サイト接続データの構成

次の各項で説明するように、Web サイトで使用されるプロトコルおよびリスナー・ポートは、スタンドアロン OC4J 環境と Oracle Application Server 環境では構成方法が異なります。

- [スタンドアロン OC4J インストールでの Web サイト・データの構成](#)
- [OPMN 管理 OC4J インスタンスでの Web サイト・データの構成](#)

スタンドアロン OC4J インストールでの Web サイト・データの構成

スタンドアロン OC4J インストールでは、Web サイトで使用されるプロトコルおよびリスナー・ポートを対応する *-web-site.xml 構成ファイルに明示的に定義する必要があります。これらのファイルの概要は、13-8 ページの「[Web サイト構成ファイルの作成](#)」を参照してください。

デフォルト Web サイトは、デフォルトでポート 8888 で HTTP プロトコルを介して受信されるリクエストをリスニングするように構成されます。

OPMN 管理 OC4J インスタンスでの Web サイト・データの構成

Oracle Application Server インストールでは、Oracle Process Manager and Notification Server (OPMN) を使用して OC4J インスタンスを管理しますが、OPMN を使用すると Web サイトのプロトコルおよびポートの構成を効率的に管理できます。OPMN は、起動時に指定された範囲の最小値から始まるポート値を選択し、この値を 1 ずつ増やしながら空きポートを見つけます。この方法で OPMN がポートの範囲から選択できるようにすると、OC4J プロセス間で発生する可能性がある競合を回避できます。

Application Server Control や opmnctl コマンドを使用して、または opmn.xml ファイルで、OC4J ポートの範囲を変更できます。

Application Server Control でのポート範囲の変更

Application Server Control を使用してポート範囲を変更するには、次のようにします。

1. 「クラスタ・トポロジ」ページで、「ランタイム・ポート」をクリックします。
2. 変更するポートの「ポートの構成」アイコンをクリックします。
3. 「サーバー・プロパティ」ページの「ポート」セクションで、変更するポートのポート範囲を変更します。
4. 「適用」をクリックします。
5. 「クラスタ・トポロジ」ページに移動して、変更した OC4J インスタンスを選択し、「再起動」をクリックします。
6. 「確認」ページで、「はい」をクリックします。

opmn.xml におけるプロトコルおよびポート範囲の変更

このモデルでは、Web サイトが使用するプロトコルは、OPMN 構成ファイル opmn.xml で Web サイトに対して定義された <port> 要素内に指定されます。この要素内には、Web サイトが使用するリスナー・ポートの範囲を指定することもできます。

注意： opmnctl コマンドライン・ツールには、OC4J インスタンスの opmn.xml ファイルに定義された特定の Web サイトに対する <port> 要素の更新に使用できるコマンドが用意されています。

使用方法は、13-5 ページの「[opmnctl による Web サイトの構成](#)」を参照してください。

opmn.xml に指定されたプロトコルおよびポートの値は、対応する Web サイト構成ファイルに設定された対応する値に優先します。Oracle Application Server 環境では、OPMN を使用して Web サイト・プロトコルおよびポート設定を管理する必要はありません。これらの値を opmn.xml に設定せず、かわりに適切な Web サイト構成ファイルに直接値を設定できます。

<port> 要素は opmn.xml 構成ファイルに定義されます。このファイルは、`ORACLE_HOME/opmn/conf` ディレクトリにあります。この要素の構文は次のとおりです。

```
<port id="webSiteName" protocol="http|https|ajp|ajps"
  range="startPort-endPort"/>
```

表 13-1 に、<port> 要素の属性を示します。

表 13-1 <port> 要素の属性

属性	説明
id	必須。Web サイトの名前を定義します。これは、Web サイト構成ファイルの名前から .xml 拡張子を除いたものです。
protocol	任意。Web サイトがリクエストの受信に使用するプロトコルを指定します。有効値は次のとおりです。 <ul style="list-style-type: none"> ■ http ■ https ■ ajp ■ ajps <p>https または ajps のいずれかを指定すると、Web サイトに対して定義された *-web-site.xml 構成ファイルの <web-site> ルート要素のセキュアな属性の値は上書きされます。</p> <p>protocol 属性の値を変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、range 属性を使用します。</p> <p>Oracle Application Server で ajp および ajps の両方を使用する場合は、それらを異なる OC4J インスタンスに構成し、2 つの Oracle HTTP Server (OHS) インスタンスを使用する必要があります。1 つは AJP リクエストのルーティング用で、もう 1 つは AJPS リクエストのルーティング用です。</p>

表 13-1 <port> 要素の属性 (続き)

属性	説明
range	<p>任意。OPMN による割当てに使用できるポートの範囲の開始ポートと終了ポートを指定します。</p> <p>使用されるデフォルトのリッスナー・ポートの範囲は次のとおりです。</p> <ul style="list-style-type: none"> ■ HTTP: 8888-8987 ■ AJP: 12501-12600 <p>開始ポートと終了ポートを同じポート番号に設定することで、範囲のかわりに単一のポートを指定できます。</p> <p>protocol 属性の値を変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、range 属性を使用します。</p> <p>Oracle Application Server で ajp および ajps の両方を使用する場合は、それらを異なる OC4J インスタンスに構成し、2 つの Oracle HTTP Server (OHS) インスタンスを使用する必要があります。1 つは AJP リクエストのルーティング用で、もう 1 つは AJPS リクエストのルーティング用です。</p>

通信プロトコルを定義する <port> 要素はすべて、OC4J インスタンスを定義する <process-type> 要素内に設定します。<process-type> 要素は、<ias-component> 要素のサブ要素 (id 属性は default_group) です。

たとえば、次の例の <port> 要素は、ポート 12501 ~ 12600 で AJP リクエストをリスニングするように、OC4J home インスタンスのデフォルト Web サイトを構成します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    ...
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="rmi" range="12401-12500">
    <port id="jms" range="12601-12700">
    <process-set id="default" numprocs="1"/>
  </process-type>
</ias-component>
```

注意: opmn.xml ファイルの変更を有効にするには、リロードする必要があります。対象となるノードで次のコマンドを実行し、opmn.xml をリロードします。

```
opmnctl reload
```

このコマンドは、OPMN 管理コンポーネント (Oracle HTTP Server、OC4J およびデプロイ済アプリケーションなど) には影響を与えません。

opmnctl による Web サイトの構成

OPMN コマンドライン・ツール opmnctl には、config port コマンドが用意されています。このコマンドを使用すると、opmn.xml に定義される Web サイトの構成を指定、更新または削除できます。

opmnctl ツールは、各ノードの ORACLE_HOME/opmn/bin ディレクトリにインストールされています。このツールは各ノードで個別に実行する必要があり、そのノードの opmn.xml ファイルのみが更新されます。

opnm.xml での Web サイト構成データの挿入または更新

config port update コマンドは、指定されたデータを新規または既存の <port> 要素に設定します。このコマンドの構文は次のとおりです。

```
opmnctl config port update ias-component=componentName
  process-type=instanceName portid=webSiteName [range=startPort-endPort]
  [protocol=http|https|ajp|ajps]
```

たとえば、OC4J インスタンスのデフォルト Web サイトは、現在、HTTP リクエストをリスニングするように構成されています。次のコマンドは、デフォルト Web サイトの構成を、Oracle HTTP Server からの Apache JServ Protocol (AJP) リクエストを受信して応答するように変更します。

```
opmnctl config port update ias-component=default_group process-type=home
  portid=default-web-site protocol=ajp
```

```
opmnctl reload
```

opmnctl reload コマンドは、OC4J ランタイムに更新された opmn.xml ファイルをリロードするために起動されます。

Web サイトのプロトコルを変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、config port update コマンドで range パラメータを使用します。

Oracle Application Server で AJP および AJPS の両方を使用する場合は、それらを異なる OC4J インスタンスに構成し、2 つの Oracle HTTP Server (OHS) インスタンスを使用する必要があります。1 つは AJP リクエストのルーティング用で、もう 1 つは AJPS リクエストのルーティング用です。

opnm.xml からの Web サイト構成データの削除

delete コマンドは、指定された Web サイトに対して定義された <port> 要素を削除します。構文は次のとおりです。

```
opmnctl config port delete ias-component=componentName
  process-type=instanceName portid=webSiteName
```

たとえば、次の例ではデフォルト Web サイトに対して定義された <port> 要素を opmn.xml から削除しています。

```
opmnctl config port update ias-component=default_group process-type=home
  portid=default-web-site
```

```
opmnctl reload
```

次の表 13-2 に、`opmnctl config port` コマンドラインで設定できるオプションを示します。

表 13-2 opmnctl config port のオプション

オプション	説明
<code>ias-component</code>	opmn.xml の OC4J 構成を更新するには、 <code>default_group</code> と設定します。
<code>process-type</code>	更新する OC4J インスタンスの識別子（home など）に設定します。この値は、opmn.xml の <code><ias-component></code> の <code><process-type></code> サブ要素の <code>id</code> 属性の値と一致します。
<code>portid</code>	Web サイトの名前に設定します。これは、Web サイト構成ファイルの名前から .xml 拡張子を除いたものです。
<code>protocol</code>	<p>Web サイトがリクエストの受信に使用するプロトコルを指定します。update 操作に対してのみ有効です。有効値は次のとおりです。</p> <ul style="list-style-type: none"> ■ http ■ https ■ ajp ■ ajps <p>https または ajps のいずれかを指定すると、Web サイトに対して定義された <code>*-web-site.xml</code> 構成ファイルの <code><web-site></code> ルート要素のセキュアな属性の値は上書きされます。</p> <p>Web サイトのプロトコルを変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、<code>range</code> パラメータを使用します。</p>
<code>range</code>	<p>OPMN による割当てに使用できるポートの範囲の開始ポートと終了ポートを設定します。update 操作に対してのみ有効です。</p> <p>デフォルトのポートの範囲は次のとおりです。</p> <ul style="list-style-type: none"> ■ HTTP: 8888-8987 ■ AJP: 12501-12600 <p>開始ポートと終了ポートを同じポート番号に設定することで、範囲のかわりに単一のポートを指定できます。</p> <p>Web サイトのプロトコルを変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、<code>range</code> パラメータを使用します。</p>

OC4J での Web サイトの新規作成

OC4J インスタンス内での新しい Web サイトの有効化は、基本的に 2 段階（必要に応じて 3 段階）プロセスです。

1. OC4J がインストールされたディレクトリ構造内で Web サイトの XML 構成ファイルを作成します。
2. 新しい Web サイト構成ファイルへの参照を OC4J 構成ファイル `server.xml` に追加します。
3. OPMN 管理 OC4J インスタンスの場合は、Web サイトのプロトコルおよびポートの範囲を定義する `<port>` 要素を `opmn.xml` に追加します。

これらの手順を完了すると、Web サイトはアプリケーションとバインドできるようになります。次の各項では、Web サイトの構成について詳しく説明します。

- [Web サイト構成ファイルの作成](#)
- [server.xml での Web サイト構成ファイルの参照](#)
- [opmn.xml での Web サイトの接続データの定義](#)
- [Web サイト間での Web アプリケーションの共有](#)
- [Cookie ドメインの指定](#)

Web サイト構成ファイルの作成

Web サイト構成ファイルに定義される主な情報には、次のものがあります。

- サイトへのアクセスに使用される URL の最後に追加される、サイトにバインドされる各アプリケーションの Web コンテキスト (`/em` など)。
- サイトで使用されるプロトコル。OPMN 管理の環境では、この値は `opmn.xml` に指定されたプロトコルによって上書きされます。
- サイトがリスニングするポート。OPMN 管理の環境では、この値は `opmn.xml` に指定されたポートの範囲によって上書きされます。
- サイトへのユーザー・アクセスを追跡する、アクセス・ログ・ファイルの場所。

新しい構成ファイルを作成する最も簡単な方法は、デフォルト Web サイトの構成ファイル `default-web-site.xml` のコピーを作成することです。このファイルは、`ORACLE_HOME/j2ee/instance/config` ディレクトリにあります。ファイルには、次の規則に従って名前を付けます。

```
webSiteName-web-site.xml
```

一般的な構成ファイルには、次の情報を指定する属性が含まれる `<web-site>` ルート要素を指定します。

- `host`: この Web サイトのホストを DNS ホスト名または IP アドレスのいずれかで指定します。サーバーが複数の IP アドレスを保持する場合、ALL 設定を使用してすべての IP アドレスをリスニングできます。
- `port`: Web サイトのリスナー・ポート。
- `display-name`: Web サイトの表示用の名前。
- `virtual-hosts`: この Web サイトにバインドされる追加ドメイン。

また、`<web-site>` 要素には通常、次のサブ要素が含まれます。

- `<default-web-app>` 要素。Web サイトからデフォルトでアクセスされる Web アプリケーションを定義します。単一のアプリケーションが Web サイトにバインドされる場合 (Application Server Control など)、そのアプリケーションをこの要素内に指定します。

- Web サイトにバインドされる各 Web モジュールに対する 1 つ以上の <web-app> サブ要素。これらの要素は、各アプリケーションが Web サイトにバインドされるときに OC4J によって追加されますが、必要に応じて手動でファイルに追加することができます。少なくとも、各 <web-app> 要素には次の属性があります。
 - application 属性。Web モジュールが属する J2EE アプリケーションの名前を指定します (.ear 拡張子を除いた EAR ファイル名と同じです)。
 - name 属性。Web モジュールの名前を指定します (.war 拡張子を除いた WAR ファイル名と同じです)。
 - root 属性。Web モジュールのバインド先となる、この Web サイトのコンテキスト・パスまたはコンテキスト・ルートを指定します。
- <access-log> 要素。サイトに送信されるリクエストが記録されるログ・ファイルを指定します。

たとえば、Application Server Control への管理アクセスを提供するためにのみ使用される Web サイトを定義する、ascontrol-web-site.xml という構成ファイルを作成するとします。このファイルの <web-site> ルート要素には、次に示すように、必須構成データがすべて含まれます。

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  web-site-10_0.xsd" port="1810"
  display-name="Application Server Control Web Site">
  <default-web-app application="ascontrol" name="ascontrol" access-log="true" />
  <access-log path="../log/ascontrol-web-access.log" />
</web-site>
```

この要素の構造の詳細は、B-22 ページの <web-site> 要素の説明を参照してください。

注意： この例で示すように、Application Server Control のみが使用する Web サイトを作成する場合、ORACLE_HOME/j2ee/ からアクセスされる OC4J ホームページの Application Server Control の起動リンクを正しい URL で更新することも必要です。

server.xml での Web サイト構成ファイルの参照

すべての Web サイト構成ファイルの場所は、J2EE_HOME/config ディレクトリにある OC4J 構成ファイル server.xml の <web-site> 要素で参照する必要があります。この宣言が server.xml にないと、アプリケーションは Web サイトにバインドできません。

各 <web-site> 要素は、サンプルの server.xml の次のエントリに示すように、対応する Web サイトの XML ファイルのパスおよびファイル名を指定します。

```
<application-server ... >
  <web-site path="./default-web-site.xml" />
  <web-site path="./ascontrol-web-site.xml" />
</application-server>
```

この例では、すべての Web サイト構成ファイルの場所は、server.xml の場所に対して相対的です。

注意： OC4J ポーリングが無効になっている場合、server.xml に対する変更を有効にするには、OC4J を再起動する必要があります。

opmn.xml での Web サイトの接続データの定義

Oracle Application Server インストールでは、Oracle Process Manager and Notification Server (OPMN) を使用して OC4J インスタンスを管理しますが、OPMN を使用すると Web サイトの プロトコルおよびポートの構成を効率的に管理できます。opmnctl config port コマンドを使用して、Web サイトの新しい <port> 要素を opmn.xml の OC4J インスタンス定義に追加します。

次の例では、ascontrol Web サイトのプロトコル (HTTP) およびポート (1810) を設定しています。

```
opmnctl config port update ias-component="default_group" id="ascontrol-web-site"
protocol="http" range="1810"
```

次の例のコマンドは、新しい <port> 要素をホスト・マシンにある opmn.xml ファイルの home OC4J インスタンス定義に追加します。これで、この OC4J インスタンスは 2 つの Web サイト、すなわちデフォルト・サイトと新しい ascontrol サイトを保持するように構成されます。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    ...
    <port id="default-web-site" protocol="ajp" range="12501-12600"/>
    <port id="ascontrol-web-site" protocol="http" range="1810"/>
    <port id="rmi" range="12401-12500">
    <port id="jms" range="12601-12700">
    <process-set id="default" numprocs="1"/>
  </process-type>
</ias-component>
```

Web サイト間での Web アプリケーションの共有

Web アプリケーションを共有するという事は、セッション、サーブレット・インスタンス、コンテキスト値を含め、アプリケーションを構成するすべてのものを共有するという事を暗に示しています。

このモードの一般的な用途は、基本的に Web アプリケーションを 2 つの異なる Web サイトにバインドして、同じコンテキスト・パス上の HTTP サイトと HTTPS サイト間でその Web アプリケーションを共有することです。これにより、リクエストのすべての情報を暗号化する必要がなく、機密情報のみを必要に応じて暗号化するため、パフォーマンスは向上します。

もう 1 つの利点は、セッションの追跡に SSL 証明書ではなく Cookie が使用されることです。SSL 証明書は、追跡する際に各証明書を格納するために 50KB を使用するので、セッションがタイムアウトする前に、セッションに対してメモリー不足の問題が発生する場合があります。Cookie の使用により、Web アプリケーションの安全性が低下する可能性はありますが、一部のブラウザでは適切にサポートされていない、SSL セッション・タイムアウトなどの問題に対処するためには必要です。

アプリケーションのバインド先となる各 Web サイトを定義する *-web-site.xml ファイルで、<web-app> 要素の shared 属性を true に設定すると、アプリケーションを共有として設定できます。この属性は、デフォルトでは false です。

たとえば、サンプルの petstore アプリケーションをデフォルトの OC4J Web サイト (ポート 8888 でリスニング) と新しいセキュアな Web サイト (ポート 4443 でリスニング) の間で共有するには、各 Web サイト構成ファイルで次の <web-app> 要素を追加または変更します。この構成により、アプリケーションは HTTP 接続と HTTPS 接続の両方を受け入れることができるようになります。

default-web-site.xml の <web-app> エントリ :

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  web-site-10_0.xsd" port="8888" display-name="OC4J 10g (10.1.3) HTTP Web Site">
  <web-app application="petstore" name="petstore" load-on-startup="true"
    root="/petstore" shared="true" access-log="true"/>
  <access-log path="../log/http-web-access.log" />
</web-site>
```

secure-web-site.xml の同様のエントリ :

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  web-site-10_0.xsd" port="4443" secure="true" display-name="My Secure Web Site">
  <web-app application="petstore" name="petstore" load-on-startup="true"
    root="/petstore" shared="true" access-log="true"/>
  <access-log path="../log/secure-web-access.log" />
  <ssl-config keystore="../../server.keystore" keystore-password="welcome"
    provider="com.sun.net.ssl.internal.ssl.Provider" />
</web-site>
```

Cookie ドメインの指定

Cookie ドメインを特定の値に設定できます。これにより、Cookie の作成時にドメインは特定の値に設定され、Cookie は Web ブラウザによってドメイン内の Web サイトに送信できるようになります。

ドメインが指定されない場合、ブラウザは、site1.acme.com などの完全修飾されたサーバー名のドメインにデフォルトで設定されます。この場合、ブラウザは site2.acme.com に Cookie を転送できません。しかし、Cookie ドメインが明示的に acme.com に設定される場合、Cookie はいずれのサーバーにも送信できます。

アプリケーションの J2EE 標準 orion-web.xml ファイルの <session-tracking> 要素に cookie-domain 属性を設定します。cookie-domain 属性には、指定したドメイン名のコンポーネントが 2 つ以上ある DNS ドメインが含まれます。次に例を示します。

```
<session-tracking cookie-domain=".oracle.com" />
```

OC4J でのセキュアな Web サイトの構成

OC4J では、HTTPS および AJP/S を使用して、クライアントと OC4J の間の Secure Socket Layer (SSL) 通信をサポートします。SSL を利用してセキュアな接続を作成するように、デフォルト Web サイトの構成ファイルを変更できます。あるいは、追加サイトを作成し、セキュアな接続を必要とする Web アプリケーションにバインドすることもできます。

SSL の鍵および証明書の詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。

この項の内容は次のとおりです。

- [セキュアな Web サイトの構成ファイルの作成](#)

セキュアな Web サイトの構成ファイルの作成

次の例に示すように、<web-site> 要素の下に適切な SSL 設定を指定します。

```
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  web-site-10_0.xsd" port="4443" secure="true" display-name="My Secure Web Site">
  <access-log path=" ../log/secure-web-access.log" />
  <ssl-config keystore=" ../server.keystore" keystore-password="welcome"
    provider="com.sun.net.ssl.internal.ssl.Provider" />
</web-site>
```

太字で示した <web-site> への追加に注意してください。

- 値を true に設定した secure 属性を追加します。設定 secure="true" は、HTTP プロトコルで SSL ソケットを使用することを指定します。
- port 属性を使用可能なポートに設定します。SSL ポートのデフォルトは 443 です。この例では、port 属性は 4443 に設定されています。
- <ssl-config> 要素を追加します。secure フラグを true に設定するときは常に、この要素が必要です。この要素は、次の属性および要素を取ります。

- 任意の factory 属性。アプリケーションで JSSE を使用しない場合に使用するサード・パーティの SSLServerSocketFactory 実装を指定するために使用します。

アプリケーションでサード・パーティの SSLServerSocketFactory 実装を使用する場合、<ssl-config> の <property> サブ要素を使用してパラメータをファクトリに送信できます。

注意： factory 属性およびそのパラメータは、OC4J 10g (10.1.3.4.0) では推奨されません。

- keystore および keystore-password 属性。キーストアのディレクトリ・パスおよびパスワードを指定します。指定されるキーストアには、HTTPS による OC4J への接続を認可されたクライアントの証明書が含まれている必要があります。keystore の値は、絶対ディレクトリ・パスまたは相対ディレクトリ・パスのいずれかを示し、ファイル名が含まれます。
- 任意の provider 属性。使用するセキュリティ・プロバイダを指定するために使用できます。

デフォルトでは、Sun 社の実装 (com.sun.net.ssl.internal.ssl.Provider) が使用されます。(この例ではデフォルトの実装が示されていますが、これは暗黙であるため指定する必要はありません。)

- 1 つ以上の <property> 要素。SSLServerSocketFactory に渡すパラメータが含まれます。各要素には、name 属性および value 属性が含まれ、名前 / 値ペアとしてパラメータを指定できます。

注意： SSLServerSocketFactory のパラメータは、OC4J 10g (10.1.3.4.0) では推奨されません。

Web サイト構成ファイルが用意できたら、J2EE_HOME/config ディレクトリにある OC4J 構成ファイル server.xml を参照する <web-site> 要素を追加します。server.xml にこの記述がないと、アプリケーションは Web サイトにバインドできません。次に例を示します。

```
<application-server ... >
  <web-site path=" ../default-web-site.xml" />
  <web-site path=" ../mycustom-web-site.xml" />
  <web-site path=" ../secure-web-site.xml" />
</application-server>
```

構成が完了すると、OC4J は一方のポートで SSL HTTP リクエストを、もう一方のポートで非 SSL HTTP リクエストをリスニングします。server.xml 構成ファイルで該当する `*-web-site.xml` をコメントアウトすると、SSL リクエストまたは非 SSL リクエストのいずれかを無効にできます。

```
<!-- <web-site path="./secure-web-site.xml" /> commented out to remove SSL -->
```

`<web-site>`、`<web-app>`、`<session-tracking>` 要素の要素および属性の詳細は、『Oracle Containers for J2EE サブレット開発者ガイド』を参照してください。

クライアント認証の要求

`<ssl-config>` 要素の `needs-client-auth` 属性を `true` に設定すると、サーバーによるクライアントの認証を要求できます。次に例を示します。

```
<web-site ... secure="true" ... >
  <ssl-config keystore="../../server.keystore" keystore-password="welcome"
    needs-client-auth="true" />
</web-site>
```

この手順では、OC4J がセキュアな通信のためにアイデンティティに応じてクライアント・エンティティを受け入れるか拒否するモードを設定します。`needs-client-auth` 属性は、接続に関するクライアントの証明連鎖を要求するように OC4J に指示します。クライアントのルート証明書が認識されると、そのクライアントは受け入れられます。

`<ssl-config>` 要素に指定されるキーストアには、HTTPS による OC4J への接続を認可されたクライアントの証明書が含まれている必要があります。

OC4J によるクライアント認証のリクエスト

OC4J では、**クライアント認証モード**をサポートします。このモードでは、サーバーはクライアントと通信する前に、クライアントからの認証を明示的に要求します。この場合、クライアントには独自の証明書が必要です。クライアントは、証明書およびルート証明書で終わる証明連鎖を送信して自己認証を行います。OC4J は、クライアントに至る信頼の連鎖を確立する際に、指定されたリストからルート証明書のみを受け入れるように構成できます。

OC4J が信頼する証明書は、**トラスト・ポイント**と呼ばれます。これは、OC4J がクライアントからの連鎖で検出する最初の証明書であり、独自のキーストア内のものと一致します。この信頼関係を構成するには、次の3つの方法があります。

- クライアント証明書がキーストアにあること
- クライアントの連鎖に含まれる中間認証局の証明書の1つがキーストアにあること
- クライアントの連鎖に含まれるルート認証局の証明書がキーストアにあること

OC4J は、偽造された証明書を防ぐために、トラスト・ポイントを含む証明連鎖全体が有効であるかどうかを検証します。

`needs-client-auth` 属性を使用してクライアント認証を要求するには、次のようにします。

1. クライアントの連鎖のうちトラスト・ポイントとする証明書を決定します。このトラスト・ポイントを使用して証明書の発行を制御すること、または認証局を発行者として信頼することを確認します。
2. クライアントの証明書の認証用に、中間証明書またはルート証明書をトラスト・ポイントとしてサーバーのキーストアにインポートします。
3. OC4J が特定のトラスト・ポイントにアクセスしないようにする場合は、これらのトラスト・ポイントがキーストアにないことを確認します。
4. 前述の手順を実行して、サーバーにインストールされた中間証明書またはルート証明書を含むクライアント証明書を作成します。別の認証局を信頼する場合は、その認証局からの証明書を取得します。
5. 証明書をクライアント上にあるファイルに保存します。
6. クライアントが HTTPS 接続を開始するときに、証明書を指定します。

- a. クライアントがブラウザの場合は、クライアントのブラウザのセキュリティ領域内で証明書を設定します。
- b. クライアントが Java クライアントの場合は、HTTPS 接続の開始時に、クライアント証明書および証明連鎖をプログラムで指定する必要があります。

Web サイトの起動と停止

Web サイトは、OC4J インスタンスで構成されると、デフォルトで使用できるようになります。しかし、Application Server Control には、「管理」→「J2EE Web サイト」ページから個々の Web サイトを停止および起動する機能があります。これらのページには、各 Web サイトの構成も表示され、各サイトにバインドされている Web モジュールにアクセスできます。

注意： `ascontrol-web-site` は Application Server Control で使用されるため、ユーザー・インタフェースを介して停止することはできません。

1. Application Server Control で「管理」リンクをクリックします。
2. 「管理タスク」→「プロパティ」の下の「J2EE Web サイト」アイコンをクリックします。OC4J インスタンスで構成された Web サイトがページにリスト表示されます。
3. 目的の Web サイトの名前をクリックします。

Web サイトのアクセス・ロギングの構成

OC4J には Web サイトごとにアクセス・ログを生成する機能があり、クライアントによって Web サイトに送信されたリクエストを記録します。

アクセス・ログは、テキストベースのログ・ファイルまたは Oracle Diagnostic Logging (ODL) ファイルとして生成できます。これらのファイルは、Application Server Control を使用して表示できる XML 形式で生成されます。Web サイトに対して構成できるアクセス・ロギングのタイプは 1 つのみです。

アクセス・ロギングは、Web サイトに対して、その Web サイト構成ファイル (`*-web-site.xml`) で `<access-log>` または `<odl-access-log>` 要素を使用して構成されます。どちらの要素も構成ファイルに指定されていない場合、アクセス・ログはその Web サイトに対して生成されません。

この項の内容は次のとおりです。

- [テキストベースのアクセス・ロギングの構成](#)
- [テキスト・アクセス・ログ・ファイルの表示](#)
- [ODL アクセス・ロギングの構成](#)
- [ODL アクセス・ログ・ファイルの表示](#)
- [Web モジュールまたはアプリケーションに対するアクセス・ロギングの有効化または無効化](#)

テキストベースのアクセス・ロギングの構成

テキストベースのアクセス・ロギングは、対応する Web サイトの構成ファイル (*-web-site.xml) の <web-site> ルート要素の <access-log> サブ要素を使用して構成されます。

注意: このロギング形式はログの循環をサポートしないため、テキストベースのアクセス・ログ・ファイルの監視は重要です。チェックしないままにしておくと、アクセス・ログ・ファイルのサイズは増大し続け、ディスク超過が発生する可能性があります。

この <access-log> 要素には次の属性があります。

- **path:** アクセス・ログのパスおよびファイル名を指定します。これは唯一の必須属性です。単独で指定するとアクセス・ログが生成されます。

default-web-site.xml の次のエントリに示すように、このパスは、Application Server Control を使用してログを表示できるように、`ORACLE_HOME/j2ee/instance/config` ディレクトリに対して相対的である必要があります。

```
<access-log path="../../log/default-web-access.log" />
```

- **format:** 複数のサポートされる変数 (ログ・エントリの先頭に付加される情報となる) を 1 つ以上を指定します。サポートされる変数は、`$time`、`$timeUsed`、`$request`、`$ip`、`$host`、`$path`、`$size`、`$method`、`$protocol`、`$user`、`$status`、`$referer`、`$agent`、`$cookie: [name]`、`$header: [name]` および `$mime` です。変数の間には、ログ・メッセージの値の間に表示するセパレータを入力できます。デフォルト設定は次のとおりです。

```
"$ip - $user - [$time] '$request' $status $size"
```

このデフォルト構成は、次のようなログ・メッセージになります (2 番目のメッセージは 2 行目に折り返されています)。

```
148.87.1.180 - - [17/Nov/2004:10:23:18 -0800] 'GET / HTTP/1.1' 200 2929
148.87.1.180 - - [17/Nov/2004:10:23:53 -0800] 'GET
/webseervices/statefulTest HTTP/1.1' 200 301
```

この例では、ユーザーは NULL、時刻は (format 設定に指定された) 大カッコで囲まれ、リクエストは (指定された) 一重引用符で囲まれ、最初のメッセージのステータスおよびサイズはそれぞれ 200 および 2929 です。

`$timeUsed` 変数により、HTTP トランザクションの処理にかかった時間が秒単位で計測されます。この変数の名前では大 / 小文字が区別されます。

- **split:** 新しいアクセス・ログを開始する頻度を指定します。サポートされる値は、`none` (デフォルト値の `never` と同じ)、`hour`、`day`、`week` および `month` です。split を指定する場合、`suffix` 属性 (次の項目で説明) を使用して、ファイル名の最後に追加するタイムスタンプ・データを指定できます。
- **suffix:** `split` 属性が指定されている場合、ログのベース・ファイル名の最後に追加するタイムスタンプ情報を指定します。

デフォルトの `suffix` 値は `-yyyy-mm-dd` です。

たとえば、デフォルトの `suffix` 値を使用する、`split` が指定された次の <access-log> 要素を考えてみます。

```
<access-log path="../../log/mysite-web-access.log" split="day" />
```

生成されるログ・ファイルには、次のような名前が付けられます。

```
mysite-web-access-2004-11-17.log
```

使用される形式は `java.text.SimpleDateFormat` の形式で、`suffix` 設定で使用される記号は、そのクラスの記号表記に従います。`SimpleDateFormat` のドキュメントで説明されているように、大 / 小文字は区別されます。`SimpleDateFormat` と、そこで使用される書式記号の詳細は、次の URL で Sun 社の最新の Javadoc を参照してください。

<http://java.sun.com/j2se/>

`default-web-site.xml` の次のエントリは、`default-web-access.log` という名前のファイルを生じます。

```
<web-site>
...
  <access-log path="../log/default-web-access.log" />
</web-site>
```

ファイルは、使用している OC4J インストールに応じて、次の場所に生成されます。

- スタンドアロン OC4J:
ログ・ファイルは、`ORACLE_HOME/j2ee/home/log/` に生成されます。
- Oracle Application Server:
ファイルは、`ORACLE_HOME/j2ee/instance/application-deployments/log/instance_default_group_1` という OC4J インスタンス固有のディレクトリに生成されます。

テキスト・アクセス・ログ・ファイルの表示

アクセス・ログのテキスト・ファイルは、Application Server Control で「ログ」リンクをクリックすると表示できます。テキストベースのログ・ファイルは、「ログ・ファイル」ページでは `.log` 拡張子によって識別されます。

1. Application Server Control のページの一番下にある「ログ」リンクをクリックします。
2. 「OC4J」を開きます。
3. 「<instanceName>」を開きます。デフォルトのインスタンス名は `home` です。

ODL アクセス・ロギングの構成

ODL フレームワークでは、ログ・ファイルは XML 文書形式で作成されます。ODL アクセス・ロギングの主な利点は、テキストベースのロギングとは異なり、ログ・ファイルの循環をサポートしていることです。

ODL アクセス・ロギングは、Web サイトの構成ファイルの `<web-site>` ルート要素の `<odl-access-log>` サブ要素を使用して構成されます。この要素には次の属性があります。これらの属性はすべて必須です。

- `path`: Web サイトの `log.xml` ファイルが生成されるディレクトリへのパス。
このパスは、Application Server Control を使用してログ・ファイルを表示できるように、`*-web-site.xml` 構成ファイルに対して相対的である必要があります。
より簡単に管理するには、Web サイトの名前をパスに含めます。
- `max-file-size`: 個々のログ・ファイルに対して許容される最大サイズ (KB 単位)。この制限に達すると、新しいログ・ファイルが生成されます。
- `max-directory-size`: ログ・ファイル・ディレクトリに対して許容される最大サイズ (KB 単位) を設定します。この制限を超えると、最も古いログ・ファイルから消去されます。

ディレクトリの最大サイズに達するまで、`log.xml` という新しいファイルが `path` 属性に指定されたディレクトリ内に生成されます。各ログ・ファイルは、属性に指定された最大サイズ以下になります。

たとえば、default-web-site.xml の次のエントリは、log.xml ファイルを生成します。また、ORACLE_HOME/j2ee/home/log/ 内の /default-web-access ディレクトリで、ログ・ファイルの最大値を 1,000KB に、ディレクトリの最大値を 10,000KB に設定します。

```
<web-site>
...
  <odl-access-log path="../log/default-web-access/" max-file-size="1000"
    max-directory-size="10000" />
</web-site>
```

ログ・ファイルは、使用している OC4J インストールに応じて、次の場所に生成されます。

- スタンドアロン OC4J:
ログ・ファイルは、ORACLE_HOME/j2ee/home/log/default-web-access/ に生成されます。
- Oracle Application Server:
ファイルは、ORACLE_HOME/j2ee/instance/application-deployments/log/instance_default_group_1/default-web-access という OC4J インスタンス固有のディレクトリに生成されます。

ODL アクセス・ログの詳細は、11-6 ページの「ODL ログ・ファイルの管理」を参照してください。

ODL アクセス・ログ・ファイルの表示

ODL 形式のログ・ファイルは、Application Server Control で「ログ」リンクをクリックすると表示でき、管理者は、一元管理された場所から OC4J 内で稼働しているすべてのコンポーネントおよびアプリケーションによって生成されたロギング出力を集計および表示できます。

ODL ログ・ファイルは、「ログ・ファイル」ページでは .xml 拡張子で識別されます。

1. Application Server Control のページの一番下にある「ログ」リンクをクリックします。
2. 「OC4J」を開きます。
3. 「<instanceName>」を開きます。スタンドアロン OC4J と OAS のいずれでも、デフォルト・インスタンス名は home です。
4. デフォルト Web サイト・ノードを開きます。
5. 診断メッセージ・ログを開きます。

Web モジュールまたはアプリケーションに対するアクセス・ロギングの有効化または無効化

<access-log> または <odl-access-log> 要素が Web サイト構成ファイルに定義されている場合、アクセス・ロギングは、Web サイトにバインドされたアプリケーション内の Web モジュールに対してデフォルトで有効になりません。OC4J 10g 10.1.3.1.0 では、構成ファイルのアプリケーション固有の <web-app> 要素の access-log 属性のデフォルト値は false です。

しかし、モジュールの access-log 属性を true に設定すると、特定のモジュールに対してアクセス・ロギングを有効にできます。

Web モジュールが大量のリクエストを送信してテキストベースのアクセス・ログ・ファイルが急速に肥大化するような状況では、アクセス・ロギングを無効にしておくことをお勧めします。Web サイト構成ファイルまたはモジュールで access-log が true に設定されている場合、access-log 属性を false に設定することでモジュールのアクセス・ロギングを無効化できます。

たとえば、`default-web-site.xml` の次のエントリは、`default` アプリケーションの DMS Web コンポーネントに対するアクセス・ロギングを無効にしますが、`admin_web` モジュールに対するテキストベースのアクセス・ロギングは有効化されます。

```
<web-site ...>
  <web-app application="default" name="dms0" root="/dmsoc4j" access-log="false" />
  <web-app application="default" name="admin_web" root="/adminoc4j" />
  <access-log path="../../log/http-web-access.log" access-log="true" />
</web-site>
```

OC4J への DTD および XSD の登録

この章では、XML ファイル検証を実行する場合に必要な、XML ファイルの検証を実行する場合に必要な新しいエンティティ（特に XML 形式のデプロイメント・ディスクリプタの定義に使用されるベンダー固有の DTD および XSD）を OC4J 内で登録するプロセスについて説明します。この項の内容は次のとおりです。

- [登録する XSD の検証](#)
- [DTD または XSD の登録](#)

登録する XSD の検証

OC4J には、ファイルを読み取る際に XSD で定義されている XML デプロイメント・ディスクリプタを検証する機能があります。この機能を有効にするには、OC4J の起動時に `oc4j.jar` コマンドラインで `-validateXML` 引数を渡します。コマンドライン・オプションの詳細は、第 4 章「OC4J ランタイムの構成」を参照してください。

検証には、XML 文書を定義する XSD が OC4J サーバーに登録されている必要があります。このエンティティが登録されていないと、XML 検証は実行できません。

XML 文書が読み取られると、パーサーは、文書に宣言されている XSD を識別する 1 つ以上のキーを、エンティティ・リゾルバとして知られる OC4J コンポーネントに渡します。エンティティ・リゾルバは、登録済エンティティの場所を解決してパーサーに返し、XML 文書を検証できるようにします。

エンティティの参照には、公開識別子とシステム識別子の 2 種類のキーが使用されます。どちらの識別子も XML 文書に宣言されます。

- 公開識別子は文字列
- システム識別子は URL

エンティティ・リゾルバでエンティティを検出できるようにするには、`entity-resolver-config.xml` ファイルのエントリを使用して、これらの識別子のいずれかまたは両方を OC4J に登録する必要があります。また、エンティティの場所もこのファイルに指定する必要があります。

デフォルトでは、`entity-resolver-config.xml` には標準 J2EE XSD の他、すべての OC4J 固有の XSD の登録エントリがすでにあります。このため、J2EE 以外または OC4J 以外のエンティティに対するエントリの追加のみが必要となります。

DTD または XSD の登録

DTD または XSD を OC4J に登録するには、`entity-resolver-config.xml` ファイルに追加する必要があります。このファイルは、OC4J ホスト・マシンの `ORACLE_HOME/j2ee/instance/config` ディレクトリにあります。

各エンティティは、`<entity>` 要素内に宣言されます。この要素には次のサブ要素があります。

- `<description>`: エンティティの任意の説明が含まれます。
- `<public-id>`: エンティティの公開識別子が含まれます。
- `<system-id>`: エンティティのシステム識別子が含まれます。
`<public-id>` または `<system-id>` のいずれかを指定する必要がありますが、両方を指定する必要はありません。
- `<location>`: エンティティの場所を指定します。この場所は、エンティティへの完全修飾パスまたはローカルに解決できる URL のいずれかです。

次の `<entity>` 要素は、`acme-web.dtd` を OC4J に登録します。XML 文書内の `<!DOCTYPE>` 要素に宣言されている公開識別子とシステム識別子はいずれも登録されます。

```
<entity>
  <description>acme-web-2_0.dtd</description>
  <public-id>-//Acme//Acme web Descriptor 2.0//EN</public-id>
  <system-id>http://xmlns.acme.com/dtd/acme-web-2_0.dtd</system-id>
  <location>META-INF/acme-web-2_0.dtd</location>
</entity>
```

次の例では、acme-application.xsd を OC4J に登録します。システム識別子は、XML 文書内のルート要素の xsi:schemaLocation または xsi:noNamespaceSchemaLocation 属性のいずれかに宣言されています。

```
<entity>
  <description>acme-application-1_0.xsd</description>
  <public-id />
  <system-id>http://xmlns.acme.com/schema/acme-application-1_0.xsd</system-id>
  <location>META-INF/acme-application-1_0.xsd</location>
</entity>
```

注意：OC4J サーバーは、entity-resolver-config.xml の変更後に再起動する必要があります。

OC4J のトラブルシューティング

この付録では、OC4J の使用時に発生する可能性のある一般的な問題と、それらの解決策について説明します。次の項目が含まれます。

- [問題と解決策](#)
- [その他のヘルプ](#)

問題と解決策

この項では、一般的な問題とその解決策について説明します。この項の内容は次のとおりです。

- [最大同時タイマーに関する警告](#)
- [java.lang.OutOfMemory エラー](#)
- [アプリケーションのパフォーマンスがガベージ・コレクションの一時停止の影響を受ける](#)
- [無効または不要なライブラリ要素によるパフォーマンスの低下](#)
- [ClassCastException および ClassNotFoundException エラー](#)
- [OC4J が起動できない:Java コンパイラを検出できない](#)
- [アプリケーションのクラスタリング時にエラーが発生する](#)
- [JDK 5.0 から JDK 1.4.2 へのダウングレード時にエラーが発生する](#)
- [JMX MBeanServer および MBeanServerConnection インタフェースでサポートされていないメソッド](#)
- [Oracle Application Server でアプリケーションを起動する際に OC4J がハングする](#)

最大同時タイマーに関する警告

問題

同時タイマーの数が最大値を超えると、次の例のような警告が表示されます。

```
WARNING J2EE OJR-10002
```

```
The number of concurrent Timers has reached the maximum limit
```

OC4J 10g (10.1.3.4.0) では、デフォルトで許可されている同時タイマーは 8 つのみです。(タイマーは、EJB タイマー、タイマー・サービスまたはスケジューラによって起動されます。) この制限は、各タイマーが短期のものであると予測されているため、デフォルトでは低い数値になっています。タイマーの数が制限値であるとき、タイマーがなんらかの理由で長く稼働している場合、タイマーはそれ以上実行されません。新しいタイマーが発生すると、OC4J により警告メッセージが表示されます。

解決策

この問題を回避するには、次に示す 2 つの OC4J システム・プロパティのいずれかを使用します。

- `timer.service.debug`

このプロパティは、現在実行中のタイマー数に関する情報など、タイマー・サービスの追加の診断情報を記録するかどうかを決定します。次に例を示します。

```
-Dtimer.service.debug=true
```

- `executor.concurrent.tasks`

このプロパティは、エグゼキュータ・サービスの同時タスク数を指定します。このプロパティを使用して、OC4J で許可されている同時タイマーの最大数を増加できます。次に例を示します。

```
-Dexecutor.concurrent.tasks=12
```

注意：各タイマーは、個別のスレッドで実行されます。タイマーの最大数を高く設定しすぎると、多数のタイマーが実行され、OC4J で多くのスレッドが使用されることとなります。実行が終了したら、スレッドを再利用することをお勧めします。

システム・プロパティの設定の詳細は、第4章「OC4J ランタイムの構成」を参照してください。

java.lang.OutOfMemory エラー

問題

「メモリ不足」エラーは、Java インスタンスのヒープ・サイズが OC4J 内で稼働しているアプリケーションに必要なメモリより少なくなっていることを意味します。

解決策

次のように、OC4J の起動時に OC4J プロセスのヒープ・サイズを必要な量のメモリにまで増やします。

```
java -Xms512m -Xmx512m -jar oc4j.jar
```

アプリケーションが OPMN 管理の環境で稼働している場合は、これらの JVM 設定は opmn.xml 構成ファイルの <data id="java-options"> 要素内に定義します。次に例を示します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-Xms512m -Xmx512m -Djava.awt.headless=true
          -Dhttp.webdir.enable=false"/>
        ...
      </category>
      ...
    </module-data>
  </process-type>
</ias-component>
```

アプリケーションが Linux または UNIX 環境で稼働している場合は、ulimit 設定で JVM プロセスによるこの量のメモリの割当てが可能になっていることを確認します。

アプリケーションのパフォーマンスがガベージ・コレクションの一時停止の影響を受ける

問題

OC4J で稼働するアプリケーションが無応答になり、単純なリクエストに著しい遅延を示します。これは、JVM がメモリ不足のしきい値を超え、メモリを解放しようとフル・ガベージ・コレクションを実行していることが原因です。

解決策

停止時間の短いインクリメンタル・コレクタの使用を検討してください。これは、メジャー・コレクションの一部をマイナー・コレクションごとに行い、メジャー・ガベージ・コレクションによる長い停止を回避します。このコレクタ（トレイン・コレクタとも呼ばれる）では、マイナー・コレクションごとに、古い世代（通常1回のメジャー・コレクションで収集されるオブジェクトを保持するメモリ・プール）の一部を収集します。その結果、より短い停止時間が多数のマイナー・コレクションに分散することになります。

全体的なスループットを考えると、インクリメンタル・コレクタはデフォルトの古い世代のコレクタより遅くなります。

インクリメンタル・コレクタを使用するには、アプリケーションの起動時に Java コマンドラインで -Xincgc オプションを渡す必要があります。XX:NewSize および -XX:MaxNewSize オプションを使用して、若い世代（オブジェクト・プール）の初期サイズと最大サイズを同じ値に設定します。-Xms および -Xmx オプションを使用して、Java の初期ヒープ・サイズと最大ヒープ・サイズを同じ値に設定します。

たとえば、1GB の物理メモリーを保持するサーバーでこのコレクタを使用するには、次のように設定します。

```
java -server -Xincgc -XX:NewSize=64m -XX:MaxNewSize=64m -Xms512m -Xmx512m
```

ガベージ・コレクションのチューニングの詳細は、
<http://java.sun.com/docs/hotspot/gc1.4.2/> で入手できる『Tuning Garbage Collection with the 1.4.2 Java Virtual Machine』を参照してください。

無効または不要なライブラリ要素によるパフォーマンスの低下

問題

OC4J プロセス・メモリーがプログラムの実行中に一貫して増加する場合、グローバルな application.xml ファイルの無効なシンボリック・リンクに対する参照があることが考えられます。一般に、この問題は、Java オブジェクト・メモリーではなく C ヒープが増加するという特徴があり、従来の Java オブジェクト・メモリー・リークとともに見られます。OC4J は、application.xml ファイルのリンクを使用して、すべてのリソースをロードします。これらのリンクが無効の場合、C ヒープは増加し続け、OC4J はメモリー不足となります。

解決策

すべてのシンボリック・リンクが有効であることを確認し、OC4J を再起動します。

さらに、OC4J でロード対象として構成される JAR ファイル数を最小限に抑えます。使用されていないすべての JAR ファイルを、構成と、OC4J で検索対象として構成されるディレクトリから削除します。OC4J では、すべての JAR ファイルでクラスとリソースを検索するため、ファイル・キャッシュで余分なメモリーとプロセッサ時間が使用される原因となります。

ClassCastExceptions および ClassNotFoundException エラー

問題

ほとんどのクラス・ロード・エラーは、クラスの可視性（過剰または不十分）に関連しています。複数の JAR にパッケージされたクラス間、あるいは親アプリケーションからデフォルトで継承されたクラス間の衝突が問題になることがあります。

解決策

『Oracle Containers for J2EE 開発者ガイド』の第 3 章「OC4J クラス・ロード・フレームワークの利用」では、クラス・ロードに関連する問題の回避およびトラブルシューティングについて詳しく説明しています。また、これらの問題の多くを回避するために、OC4J 内で共有ライブラリを使用する方法についても説明しています。

OC4J が起動できない : Java コンパイラを検出できない

問題

次のようなエラーが OC4J の起動時に表示されます。

```
05/10/28 13:58:49 Error initializing server: Error initializing ejb-modules:  
Error generating wrappers for file:/C:/oc4j/j2ee/home/applications/admin_ejb.jar:  
javac.exe not found under <directory>, please use a valid jdk or specify the  
location of your java compiler in server.xml using the <java-compiler .../> tag
```

解決策

このエラーは、OC4J が必要な JDK を検出できないことを意味します。この問題を解決するには、コマンドラインで javac.exe がある場所から OC4J を起動します。これにより、JDK の場所が設定されます。

次に例を示します。

```
C:¥ORACLE_HOME¥j2ee¥home¥C:¥jdk¥bin¥java -jar oc4j.jar
```

アプリケーションのクラスタリング時にエラーが発生する

問題

アプリケーションに対してクラスタリングを構成する際に、次のエラーがスローされます。

```
WARNING: The service implementation <classname> does not implement
java.io.Serializable. *This class is not suitable for clustered environments*
indicated by recoverable=true.
```

解決策

このエラーは、シリアライズ可能なクラスではないため、OC4J レプリケーション・フレームワークをこのクラスで利用できないことを意味します。

JDK 5.0 から JDK 1.4.2 へのダウングレード時にエラーが発生する

問題

Oracle Application Server のコンポーネントとしてインストールされた OPMN 管理の OC4J インスタンスは、デフォルトでは JDK リリース 5.0 を使用しますが、JDK リリース 1.4.2 を使用するように構成する際に次のエラーが発生します。

```
oracle.oc4j.loader.util.AnnotatedLinkageError:
MBeanServerEjbHome_StatefulSessionHomeWrapper1 (Unsupported major.minor
version 49.0)
```

解決策

Oracle Application Server のコンポーネントとしてインストールされた OPMN 管理の OC4J インスタンスは、デフォルトでは JDK リリース 5.0 を使用します。この新しいバージョンの JDK は EJB 3.0 を使用するために必要で、パフォーマンスが非常に改善されます。しかし、OC4J にデプロイされるアプリケーションに JDK リリース 1.4.2 が必要な場合は、旧バージョンへのダウングレードが必要になることもあります。

JDK 5.0 から JDK 1.4.2 に切り替えるには、すべてのコンパイル済アプリケーション・ファイルを OC4J インスタンスから削除する必要があります。

1. OC4J インスタンスを停止します。
2. `ORACLE_HOME/j2ee/instance/application-deployments` ディレクトリを削除します。

このディレクトリを削除すると、OC4J が JDK 1.4.2 で再起動される際に、アプリケーション・ファイルは再コンパイルされます。

`opmn.xml` 構成ファイルを手動で編集して、各 OC4J インスタンスに使用する JDK を指定できます。JDK とともにインストールされた、`JAVA_HOME` 環境変数に定義されている `javac` コンパイラを使用する場合は、`server.xml` から `<java-compiler>` 要素も削除し、OC4J がデフォルト設定を再検出できるようにします。

JMX MBeanServer および MBeanServerConnection インタフェースでサポートされていないメソッド

問題

JMX MBeanServer インタフェースには、次の操作で取得した MBeanServer オブジェクトが使用されている場合には、J2EE アプリケーションで使用できないメソッドが多数あります。

```
MBeanServer mbsrv = MBeanServerFactory.newMBeanServer();
```

戻された MBeanServer オブジェクトに次に示すメソッドを使用すると、`UnsupportedOperationException` 例外がスローされます。

```
public final ClassLoader getClassLoaderFor(ObjectName mbeanName)

public final ClassLoader getClassLoader(ObjectName loaderName)

public final ClassLoaderRepository getClassLoaderRepository()

public final Object instantiate(String className)

public final Object instantiate(String className, ObjectName loaderName)

public final Object instantiate(String className, Object[] params, String[] signature)

public final Object instantiate(String className, ObjectName loaderName, Object[]
params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name)

public final ObjectInstance createMBean(String className, ObjectName name, ObjectName
loaderName)

public final ObjectInstance createMBean(String className, ObjectName name, Object[]
params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name, ObjectName
loader, Object[] params, String[] signature)

public final ObjectInputStream deserialize(ObjectName name, byte[] data)

public final ObjectInputStream deserialize(String className, byte[] data)

public final ObjectInputStream deserialize(String className, ObjectName loaderName,
byte[] data)
```

アプリケーションで Oracle JMX コネクタが使用されている場合、`MBeanServerConnection` インタフェースの多くのメソッドを使用できません。作成された `MBeanServerConnection` オブジェクトに次に示すメソッドを使用すると、`UnsupportedOperationException` 例外がスローされます。

```
public final ObjectInstance createMBean(String className, ObjectName name)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loaderName)

public final ObjectInstance createMBean(String className, ObjectName name,
Object[] params, String[] signature)

public final ObjectInstance createMBean(String className, ObjectName name,
ObjectName loader, Object[] params, String[] signature)
```

解決策

アプリケーションで JMX MBeanServer または MBeanServerConnection インタフェースが使用されている場合は、アプリケーションでサポートされていないメソッドを使用しないでください。

Oracle Application Server でアプリケーションを起動する際に OC4J がハングする**問題**

OPMN 管理の環境では、OC4J を起動しようとする OPMN がハングし、その結果次のようなエラーが発生します。

```
ias-component/process-type/process-set:
  default_group/home/default_group/

Error
Process (index=1,uid=2012873812,pid=2988)
time out while waiting for a managed process to start
```

解決策

かなりの量のリソースを必要とするアプリケーション（様々なコンポーネントに対して複数のデータベース接続を取得しようとするアプリケーションなど）が原因で、OC4J が起動できない場合があります。これは、opmn.xml で OC4J インスタンスに対して定義される <start-timeout> 要素に、アプリケーションを起動できる最大時間を指定することにより管理できます。この値に達すると、アプリケーションは起動されません。この値は、インスタンスにデプロイされたすべてのアプリケーションに適用されます。

次の例では、home OC4J インスタンスにデプロイされたアプリケーションに対するタイムアウトの値を 800 秒に増やしています。

```
<ias-component id="default_group">
  ...
  <process-type id="home" module-id="OC4J" status="enabled">
    ...
    <start timeout="800" retry="2"/>
  </process-type>
</ias-component>
```

その他のヘルプ

その他の解決策は、Oracle の次のサポート用 Web サイトで検索できます。

- 次に示す Oracle Technology Network で入手できる Oracle Application Server のリリース・ノート

<http://www.oracle.com/technology/documentation/index.html>

- Oracle MetaLink

<http://metalink.oracle.com>

それでも問題の解決策が見つからない場合は、オラクル社カスタマ・サポート・センターに問い合せてください。

OC4J で使用される構成ファイル

この付録では、OC4J サーバーとそこにデプロイされる J2EE アプリケーションおよびモジュールの構成データを格納するために使用される XML ファイルについて詳しく説明します。

- OC4J で使用される XML 構成ファイルの概要
- OC4J サーバー構成ファイル (server.xml) の要素
- Web サイト構成ファイル (*-web-site.xml) の概要

OC4J で使用される XML 構成ファイルの概要

OC4J インスタンスと、そこにデプロイされるアプリケーションおよびモジュールの構成データは、様々な XML ファイルに保存されます。図 B-1 に、これらの XML ファイルとそれぞれの役割について概要を示します。

OC4J で使用される Oracle 独自の XML ファイルを定義するスキーマは、次の URL で参照できます。

<http://www.oracle.com/technology/oracleas/schema/index.html>

図 B-1 OC4J で使用される XML ファイル

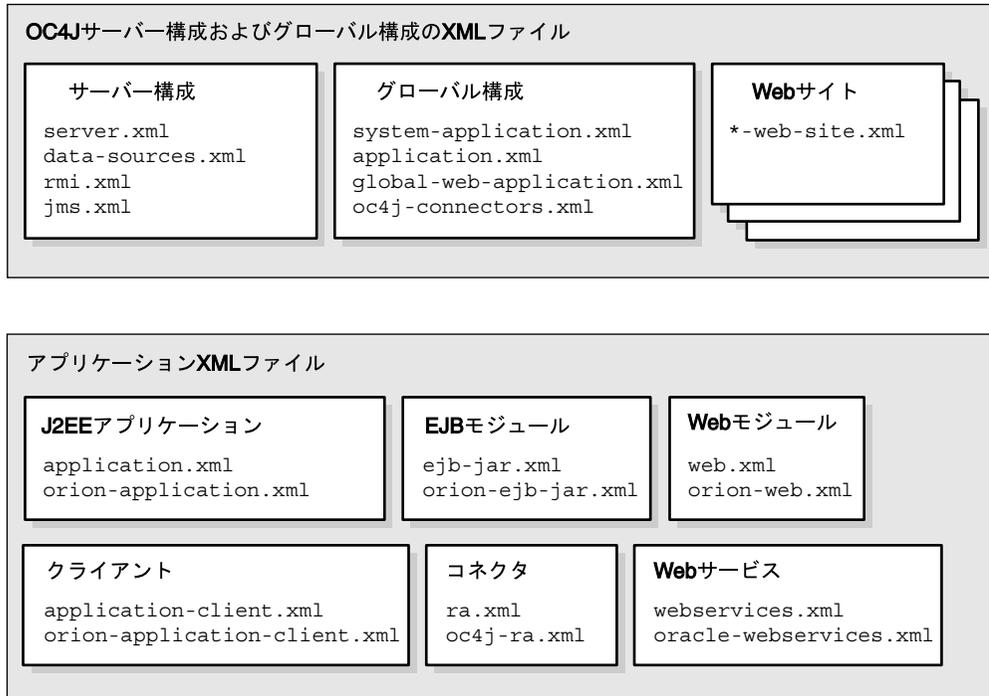


表 B-1 では、OC4J サーバー・レベルの XML ファイルごとの役割と機能の他、先の図に示したグローバル構成ファイルについても説明します。

特に記載されていない場合、ここに示すファイルはすべて `ORACLE_HOME/j2ee/instance/config` ディレクトリにデフォルトでインストールされます。

表 B-1 サーバー・レベルおよびグローバルの構成ファイル

XML 構成ファイル	機能 / コンポーネント
<code>server.xml</code>	OC4J サーバー構成ファイル。 サーバーを構成し、このファイルに追加する XML ファイル (JMS サポート用の <code>.jms.xml</code> など) を指定します。その他の XML ファイルをリストすることによって、サービスを個別のファイルに構成できますが、 <code>server.xml</code> ファイルは、それらのファイルを OC4J 構成に使用することを示します。
<code>data-sources.xml</code>	OC4J 内のアプリケーションによって使用されるすべてのデータベースの OC4J データソース構成が含まれます。
<code>rmi.xml</code>	OC4J RMI ポート構成と HTTP 上での RMI トンネリングが含まれます。
<code>.jms.xml</code>	OC4J 内で JMS および MDB によって使用される Destination のトピックおよびキューの OC4J JMS 構成が含まれます。
<code>system-application.xml</code>	OC4J インスタンスにインストールされたその他すべてのアプリケーションの親である、 <code>system</code> アプリケーションの構成が含まれます。このファイルは、必須の共有ライブラリのロードに必要なデータなど、OC4J の起動時に使用される構成データを提供します。
<code>application.xml</code>	<code>default</code> アプリケーションの構成が含まれます。指定された親がないユーザー・デプロイのアプリケーションおよびスタンドアロン・モジュールはすべて、デフォルトでこのアプリケーションにデプロイされます。 このファイルには、デプロイされたアプリケーションに適用されるデフォルトの構成値となる共通の設定が含まれます。 このファイルは、J2EE 標準デプロイメント・ディスクリプタの <code>application.xml</code> とはまったく関係ありません。
<code>global-web-application.xml</code>	OC4J 内のサーブレットおよび JSP コンテナを構成するための Oracle 固有のファイル。
<code>oc4j-connectors.xml</code>	OC4J インスタンスにインストールされたすべてのスタンドアロン・リソース・アダプタに対するグローバルな OC4J 固有の構成データが含まれます。
<code>*-web-site.xml</code>	OC4J インスタンス内に作成された Web サイトに対する構成データが含まれる OC4J 固有のファイル。通常、 <code>ORACLE_HOME/j2ee/instance/config</code> ディレクトリにインストールされますが、別の場所にインストールすることもできます。 各 OC4J インスタンス内に作成されたデフォルト Web サイトに対する構成は、 <code>default-web-site.xml</code> に定義されます。

表 B-2 では、前述の図に示した様々なアプリケーション・レベルの XML ファイルの役割と機能について説明します。

特に記載されていない場合、ここに示すファイルはすべて `ORACLE_HOME/j2ee/instance/config` ディレクトリにデフォルトでインストールされます。

表 B-2 アプリケーション・レベルの構成ファイル

XML 構成ファイル	機能 / コンポーネント
application.xml	標準の J2EE アプリケーション・ディスクリプタ・ファイル。ローカルの application.xml ファイルは、J2EE アプリケーション・モジュールを含む J2EE EAR ファイルを定義します。このファイルは、J2EE アプリケーション EAR ファイル内に存在します。
orion-application.xml	<p>OC4J 固有のデプロイメント・ディスクリプタ。特定のデプロイされたアプリケーションの構成データが含まれます。</p> <p>このファイルでは、<jazn-web-app> 要素を使用して、サーブレット実行に OracleAS JAAS Provider および Oracle Single Sign-On のプロパティを構成できます。特定のセキュリティ・サブジェクトの権限でサーブレットを起動するには、これらの機能を適切に設定する必要があります。</p> <p>Oracle Single Sign-On による認証とともに、Web アプリケーションのセキュリティ・プロバイダとして Oracle Identity Management が使用されている場合は、<jazn-web-app> を使用して Oracle Java Authentication and Authorization Service (JAAS) Provider ユーザー・コンテキストとサーブレット・セッションを同期できます。ユーザー・コンテキストとセッションを同期するには、デフォルトで <code>sso.session.synchronize</code> プロパティを <code>true</code> に設定します。これは、<jazn-web-app> の <property> サブ要素で実行できます。</p> <pre><jazn-web-app ...> <property name="sso.session.synchronize" value="true"/> </jazn-web-app></pre> <p>また、このプロパティを <code>false</code> に設定することもできます。</p> <p>orion-application.xml への変更内容を反映するには、アプリケーション (Application Server Control またはセキュリティ・プロバイダ MBean を使用して変更した場合) または OC4J (手動で変更した場合) を再起動する必要があります。</p> <p>JAAS およびこの要素の機能の詳細は、『Oracle Containers for J2EE セキュリティ・ガイド』を参照してください。また、次の URL で関連する Sun 社のドキュメントを参照できます。</p> <p>http://java.sun.com/j2se/1.4.2/docs/guide/security/jaas/JAASRefGuide.html</p>
web.xml	<p>J2EE Web アプリケーションのデプロイメント・ディスクリプタ。WAR ファイルに含まれる Web アプリケーション・デプロイ・パラメータの定義に使用されます。</p> <p>また、このファイルには、サーブレットおよび JSP の URL パターンを指定できます。たとえば、サーブレットは <servlet> 要素に、その URL パターンは <servlet-mapping> 要素に定義されます。</p>

表 B-2 アプリケーション・レベルの構成ファイル (続き)

XML 構成ファイル	機能 / コンポーネント
orion-web.xml	開発者モードおよび JSP の自動リロードなどの OC4J 機能を有効にするかどうかなど、アプリケーション・レベルの OC4J 固有の構成データを使用して、標準 J2EE ディスクリプタを拡張します。
ejb-jar.xml	Enterprise JavaBeans (EJB) JAR ファイルに含まれる J2EE EJB モジュールのデプロイメント・ディスクリプタ。JAR 内の EJB モジュールの特殊な構造的な特徴および依存性を定義し、コンテナとのやり取りを Bean で想定する方法について、EJB コンテナに指示します。
orion-ejb-jar.xml	OC4J 固有の EJB デプロイメント・ディスクリプタ。EJB プール設定、タイムアウトおよび再試行の設定、JNDI マッピングおよび finder メソッドの仕様など、アーカイブ内のすべての EJB モジュールに対する OC4J 固有の構成データを定義します。また、TopLink 永続性マネージャのプロパティも含まれます。
application-client.xml	J2EE アプリケーション・クライアント構成ファイル。アーカイブにパッケージされる J2EE アプリケーション・クライアントで使用される EJB モジュールとその他のリソースを記述します。
orion-application-client.xml	EJB モジュールのホーム・インタフェースまたは外部リソース (データソースなど) への JNDI マッピング、JMS キューあるいはメール・セッションなど、OC4J デプロイ・データが含まれます。
ra.xml	J2EE 標準デプロイメント・ディスクリプタ。RAR ファイル内にパッケージされたリソース・アダプタに対する実装コード、構成プロパティ、セキュリティ設定に関する情報が含まれます。
oc4j-ra.xml	単一のリソース・アダプタに対する OC4J 固有のデプロイ構成データが含まれます。このデータには、EIS 接続情報、使用される JNDI 名、接続プーリング・パラメータおよびリソース・プリンシパル・マッピングなどがあります。
webservices.xml	J2EE 標準 Web サービス・デプロイメント・ディスクリプタ。WSDL 情報や JAX-RPC マッピング・データなど、WAR ファイル内にパッケージされた Web サービス・アプリケーションに対する Web サービスを記述します。
oracle-webservices.xml	WSDL ファイルを公開するかどうかなど、OC4J Web サービス・コンテナで使用されるプロパティを定義します。また、Web サービスとして実装された EJB モジュール固有のエンドポイント・アドレスおよびデータを定義します。このファイルは、Web サービスを含む WAR または EJB JAR のいずれかにパッケージできます。

OC4J サーバー構成ファイル (server.xml) の要素

OC4J 構成ファイル `server.xml` は、`ORACLE_HOME/j2ee/instance/config` ディレクトリにあります。このファイルは、OC4J サーバーとすべての J2EE アプリケーション、サーバー内で有効な Web アプリケーションおよび Webs サイトの構成の起点となります。

OC4J により必要に応じて記述方法が追加および更新されるため、OC4J のドキュメントで特に指示されていないかぎり、`server.xml` を手動で編集しないでください。

`server.xml` ファイルには、OC4J インスタンス内の各アプリケーションのアプリケーション・ディスクリプタに対する直接参照または間接参照が含まれます。一般的な J2EE アプリケーションの場合、この参照は、抽出された EAR のトップレベル・ディレクトリを指します。したがって、EAR ファイルに含まれる `application.xml` ファイルを指します。OC4J グローバル・アプリケーションの場合、`server.xml` ファイルは、OC4J グローバル・アプリケーション・ディスクリプタを直接指定します。

また、`server.xml` ファイルは、他の XML 構成ファイルも指定します。各 XML ファイルの場所には、フルパスまたは `server.xml` ファイルが存在する場所に対する相対パスを指定できます。また、XML ファイルの名前には、ファイルの内容が適切な DTD に準拠しているかぎりは、任意の名前を指定できます。

- `<rmi-config>` 要素は、`rmi.xml` ファイルの名前と場所を示します。
- `<jms-config>` 要素は、`jms.xml` ファイルの名前と場所を示します。
- `<global-application>` 要素は、グローバル `application.xml` ファイルの名前と場所を示します。
- `<global-web-app-config>` 要素は、`global-web-application.xml` ファイルの名前と場所を示します。
- `<web-site>` 要素は、1つの `*-web-site.xml` ファイルの名前と場所を示します。複数の Web サイトを使用できるため、複数の `<web-site>` エントリを指定できます。

`server.xml` ファイル形式は、次の URL で参照できる `application-server-10_1.xsd` よって記述されます。

<http://www.oracle.com/technology/oracleas/schema/index.html>

server.xml ファイルの例

OC4J の `server.xml` 構成ファイルの例を次に示します。ここでは、`<!-- comments -->` を使用して、様々なセクションを説明しています。

```
<application-server xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
  application-server-10_1.xsd" application-directory="../applications"
  deployment-directory="../application-deployments"
  connector-directory="../connectors"
  schema-major-version="10" schema-minor-version="0" >
  <!-- Shared library definitions -->
  <shared-library name="global.libraries" version="1.0" library-compatible="true">
    <code-source path="../applib"/>
    <code-source path="../../..../sqlj/lib"/>
    <code-source path="../../..../lib/dsv2.jar"/>
  </shared-library>
  <shared-library name="global.tag.libraries" version="1.0"
    library-compatible="true">
    <code-source path="../jsp/lib/taglib/standard.jar"/>
  </shared-library>
  <!-- J2EE services -->
  <rmi-config path="/rmi.xml" />
  <sep-config path="/internal-settings.xml" />
  <jms-config path="/jms.xml" />
  <javacache-config path="../../..../javacache/admin/javacache.xml" />
  <!-- Logging -->
```

```

<j2ee-logging-config path="./j2ee-logging.xml" />
<log>
  <file path="./log/server.log" />
</log>
<java-compiler name="javac" in-process="false" encoding="ISO8859_1"
  extdirs="c:\sdk\jdk\jre\lib\ext" />
<!-- Default application configuration -->
<global-application name="default" path="application.xml" />
<!-- Deployed application configuration -->
<application name="petstore" path="../applications\petstore.ear" start="true" />
<application name="ascontrol" path="../applications\ascontrol.ear"
  start="true" />
<!-- Default Web application configuration file -->
<global-web-app-config path="global-web-application.xml" />
<!-- Transaction Manager configuration file -->
<transaction-manager-config path="transaction-manager.xml" />
<!-- Configuration files for enabled Web sites -->
<web-site path="./default-web-site.xml" />
</application-server>

```

<application-server>

必須 / 任意: 必須。1つのみ。

子要素:

これは、OC4J 構成ファイルのルート要素です。

表 B-3 <application-server> の属性

名前	説明
application-directory	値: 文字列 デフォルト: ../applications デプロイされるアーカイブのターゲット・ディレクトリ。
application-auto-deploy-directory	値: 文字列 デフォルト: 該当なし アプリケーションのオート・デプロイ / 再デプロイをトリガーして、EAR ファイルがコピーされるディレクトリ。
connector-directory	値: 文字列 デフォルト: ../connectors スタンドアロン・リソース・アダプタのターゲット・ディレクトリ。
deployment-directory	値: 文字列 デフォルト: ../application-deployments OC4J 固有のデプロイメント・ディスクリプタおよび生成済ファイル (コンパイル済 JSP クラスや EJB ラッパー・クラスなど) が格納されるディレクトリ。
check-for-updates	値: all adminClientOnly none デフォルト: adminClientOnly OC4J ポーリングを有効にします。OC4J ポーリングは、現在デプロイされているアプリケーションおよびモジュールに対して変更が加えられていないかを自動的にチェックし、変更されたコンポーネントを再デプロイします。 サポートされる値とそれぞれの値の影響の詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

表 B-3 <application-server> の属性 (続き)

名前	説明
localhostIsAdmin	値: ブール デフォルト: true true の場合、管理操作を起動するプロセスが OC4J ホスト・マシンに対してローカルなプロセスであると、より簡単にアクセスできます。
taskmanager-granularity	値: int 型 デフォルト: 1000 タスク・マネージャが機能する間隔 (ミリ秒単位)。デフォルトは 1 秒 (1000 ミリ秒) 間隔です。

<application>

親要素: <application-server>

必須 / 任意: 任意。複数指定可能。

子要素:

OC4J インスタンスにデプロイされる J2EE アプリケーションを定義します。アプリケーションを定義する <application> 要素は、アプリケーションのデプロイ時に OC4J により server.xml に追加されます。このため、通常、この要素を手動で変更する必要はありません。

表 B-4 <application> の属性

名前	説明
name	値: 文字列 デフォルト: 該当なし アプリケーション名。通常、.ear 拡張子を除いた EAR ファイル名と同じです。
path	値: 文字列 デフォルト: 該当なし EAR ファイルの場所または抽出された EAR のトップレベル・ディレクトリ。したがって、パスは、アプリケーションとともにパッケージされる J2EE 標準 application.xml ディスクリプタを間接的に指します。
start	値: ブール デフォルト: true true の場合、アプリケーションは OC4J とともに起動され、リクエストの処理および JMX MBean による構成に使用できます。 false の場合、アプリケーションは OC4J とともに起動されず、リクエストの処理に使用できません。しかし、JMX による構成には使用できます。

<code-source>**親要素:** <shared-library>**必須 / 任意:** 必須。複数指定可能。

共有ライブラリ定義に含まれる JAR または ZIP ファイルへのパスを指定します。

表 B-5 <code-source> の属性

名前	説明
path	<p>値: 文字列 デフォルト: 該当なし</p> <p>共有ライブラリに含まれる JAR または ZIP ファイルへのパス。</p> <p>パスは、/shared-lib ディレクトリの外の場合は絶対パス、あるいは /shared-lib/library_name ディレクトリ内で JAR ファイルを格納するサブディレクトリに対する相対パスです。相対パスの場合、path 属性の値として指定する必要があるのは、アーカイブ・ファイル名のみです。</p> <p>必要に応じて、path="*" を設定し、共有ライブラリのサブディレクトリ内にあるすべてのアーカイブを消費するように OC4J に強制することができます。</p>

<custom-thread-pool>**親要素:** <application-server>**必須 / 任意:** 任意。複数指定可能。**子要素:**

OC4J プロセス内の指定された名前の単一スレッド・プールの構成が含まれます。1 つ以上のアプリケーションを、このスレッド・プールを使用するように構成できます。詳細は、10-2 ページの「[OC4J スレッド・プールの構成](#)」を参照してください。

表 B-6 <custom-thread-pool> の属性

名前	説明
name	<p>値: 文字列 デフォルト: 必須</p> <p>スレッド・プールの名前。</p>
min	<p>値: 文字列 デフォルト: 0</p> <p>OC4J が同時に実行できる最小スレッド数。</p>
max	<p>値: 文字列 デフォルト: 1024</p> <p>OC4J が同時に実行できる最大スレッド数。</p>
queue	<p>値: 文字列 デフォルト: 0</p> <p>キュー内に保持できるリクエストの最大数。</p>

表 B-6 <custom-thread-pool> の属性 (続き)

名前	説明
keepAlive	<p>値: 文字列 デフォルト: 600000</p> <p>新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。このタイムアウトは、スレッドのアイドル状態を維持する時間を指定します。タイムアウトに達するとスレッドは破棄されます。</p> <p>スレッドを破棄しないようにするには、-1 に設定します。</p> <p>デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。</p>
stackSize	<p>値: 文字列 デフォルト: 0</p> <p>スレッド・プール・スタックのサイズ。</p>
debug	<p>値: ブール デフォルト: false</p> <p>true の場合、起動時にスレッド・プール情報がコンソールに出力されます。false の場合、スレッド・プール情報は出力されません。</p>

<execution-order>

親要素: <startup-class>、<shutdown-class>

必須 / 任意: 任意。1 つのみ。

子要素:

起動クラスごとの実行順序を指定します。クラスを実行する順序を示す数値を指定します。

<global-application>

親要素: <application-server>

必須 / 任意: 必須。1 つのみ。

子要素:

default アプリケーションとして知られる、OC4J グローバル・アプリケーションを指定します。name 属性は、名前を定義します。path 属性は、OC4J グローバル・アプリケーション・ディスクリプタとして使用するものを指定します。

表 B-7 <global-application> の属性

名前	説明
name	<p>値: 文字列 デフォルト: default</p> <p>グローバル・アプリケーション名。</p>
path	<p>値: 文字列 デフォルト: application.xml</p> <p>グローバル・アプリケーション・ディスクリプタ・ファイルのファイル名およびパス。デフォルトのディスクリプタは、ORACLE_HOME/j2ee/instance/config/application.xml です。</p>

<global-thread-pool>

親要素: <application-server>

必須/任意: 任意。1つのみ。

子要素:

OC4J プロセス内のスレッド・プールの古い構成形式が含まれます。server.xml ファイルに <global-thread-pool> 要素が含まれる場合、OC4J の起動時に作成される http スレッド・プールには min、max、keep-alive および queue 属性値が適用されます。cx-* 属性は rmi connection スレッド・プールに適用され、rmiRequest-* 属性は rmi request スレッド・プールに適用されます。詳細は、10-2 ページの「OC4J スレッド・プールの構成」を参照してください。

<global-thread-pool> 要素は、OC4J 10g (10.1.3.4.0) では推奨されません。server.xml ファイルにこの要素が含まれる場合は、OC4J により等価の <thread-pool> 要素に変更されます。この要素は、新しい構成形式のスレッド・プールを定義します。

表 B-8 <global-thread-pool> の属性

名前	説明
min	値: 文字列 デフォルト: 該当なし OC4J が同時に実行できる最小スレッド数。
max	値: 文字列 デフォルト: 該当なし OC4J が同時に実行できる最大スレッド数。
queue	値: 文字列 デフォルト: 該当なし キュー内に保持できるリクエストの最大数。
debug	値: ブール デフォルト: false true の場合、起動時にスレッド・プール情報がコンソールに出力されます。debug が false の場合、スレッド・プール情報は出力されません。
keep-alive	値: 文字列 デフォルト: 600000 新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。このタイムアウトは、スレッドのアイドル状態を維持する時間を指定します。タイムアウトに達するとスレッドは破棄されます。 値を -1 にすると、スレッドは破棄されません。 デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。
cx-min	値: 文字列 デフォルト: 該当なし OC4J が同時に実行できる最小の接続スレッド数。
cx-max	値: 文字列 デフォルト: 該当なし OC4J が同時に実行できる最大の接続スレッド数。

表 B-8 <global-thread-pool> の属性 (続き)

名前	説明
cx-queue	<p>値: 文字列 デフォルト: 該当なし</p> <p>キュー内に保持できるリクエストの最大数。</p>
cx-debug	<p>値: ブール デフォルト: false</p> <p>true の場合、起動時にスレッド・プール情報がコンソールに出力されます。cx-debug が false の場合、スレッド・プール情報は出力されません。</p>
cx-keep-alive	<p>値: 文字列 デフォルト: 600000</p> <p>新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。このタイムアウトは、スレッドのアイドル状態を維持する時間を指定します。タイムアウトに達するとスレッドは破棄されます。</p> <p>値を -1 にすると、スレッドは破棄されません。</p> <p>デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。</p>
rmiRequest-max	<p>値: 文字列 デフォルト: 該当なし</p> <p>OC4J が同時に実行できる最小の接続スレッド数。</p>
rmiRequest-min	<p>値: 文字列 デフォルト: 該当なし</p> <p>OC4J が同時に実行できる最大の接続スレッド数。</p>
rmiRequest-queue	<p>値: 文字列 デフォルト: 該当なし</p> <p>キュー内に保持できるリクエストの最大数。</p>
rmiRequest-debug	<p>値: ブール デフォルト: false</p> <p>true の場合、起動時にスレッド・プール情報がコンソールに出力されます。rmiRequest-debug が false の場合、スレッド・プール情報は出力されません。</p>
rmiRequest-keep-alive	<p>値: 文字列 デフォルト: 600000</p> <p>新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。このタイムアウトは、スレッドのアイドル状態を維持する時間を指定します。タイムアウトに達するとスレッドは破棄されます。</p> <p>値を -1 にすると、スレッドは破棄されません。</p> <p>デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。</p>

<global-web-app-config>

親要素: <application-server>

必須 / 任意: 必須。1つのみ。

子要素:

デフォルトではその他すべての Web アプリケーションの親である OC4J グローバル Web アプリケーションの構成ファイルを指定します。

デフォルトの Web アプリケーションの名前およびルート・ディレクトリ・パス、またはコンテキスト・ルートは、グローバル・アプリケーション・ディスクリプタに指定され、デフォルトの Web アプリケーションは default-web-site.xml ファイルにより Web サイトにバインドされます。スタンドアロン OC4J では、デフォルトの Web アプリケーションに対するデフォルトのコンテキスト・ルートは「/」です。

表 B-9 <global-web-app-config> の属性

名前	説明
path	値: 文字列 デフォルト: global-web-application.xml グローバル Web アプリケーション・ディスクリプタ・ファイルのファイル名およびパス。デフォルトのディスクリプタは、ORACLE_HOME/j2ee/instance/config/global-web-application.xml です。

<import-shared-library>

親要素: <shared-library>

必須 / 任意: 任意。複数指定可能。

<shared-library> 要素に囲まれて定義される共有ライブラリによってインポートされる共有ライブラリを指定します。共有ライブラリを構成および使用する方法は、『Oracle Containers for J2EE 開発者ガイド』を参照してください。

表 B-10 <import-shared-library> の属性

名前	説明
name	値: 文字列 デフォルト: 必須 インポートする共有ライブラリの名前。
version	値: 文字列 デフォルト: 必須 インポートするバージョン番号。

<init-param>

親要素: <startup-class>、<shutdown-class>

必須 / 任意: 任意。複数指定可能。

子要素: <param-name>、<param-value>

初期化パラメータを <startup-class> または <shutdown-class> 要素内に指定します。OC4J が受け取る文字列型のキー / 値ペアが含まれます。入力 Hashtable 引数内に指定されます。JNDI を使用してそれぞれの値と名前をバインドするため、キー / 値ペアの名前は一意である必要があります。

表 B-11 <init-param> の属性

名前	説明
path	値: 文字列 デフォルト: global-web-application.xml グローバル Web アプリケーション・ディスクリプタ・ファイルのファイル名およびパス。デフォルトのディスクリプタは、ORACLE_HOME/j2ee/instance/config/global-web-application.xml です。

<j2ee-logging-config>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ指定可能。

子要素:

J2EE ログイン構成ファイルとして使用するファイルを定義します。

表 B-12 <j2ee-logging-config> の属性

名前	説明
path	値: 文字列 デフォルト: ../j2ee-logging.xml ログ出力構成ファイルのファイル名およびパス。

<java-compiler>

親要素: <application-server>

必須 / 任意: 任意。1つのみ。

子要素:

EJB モジュールのコンパイルに使用する Java コンパイラの構成パラメータを指定します。デフォルトでは、JAVA_HOME 環境変数に定義された JDK とともにインストールされる javac コンパイラが使用されます。

表 B-13 <java-compiler> の属性

名前	説明
name	値: 文字列 デフォルト: javac modern classic javac ojc jikes 使用する Java コンパイラの名前。
in-process	値: ブール デフォルト: false コンパイラをプロセス内で実行するか、プロセス外で実行するかを指定します。 false に設定すると、別の JVM プロセスが生成され、その内部でコンパイラは実行されます。これは、OC4J によって使用されるデフォルトのコンパイラ実行モードで、メモリー・リソースをより適切に管理できます。 true に設定すると、OC4J と同じ JVM プロセス内でコンパイラは実行されます。
encoding	値: 文字列 デフォルト: ISO-8859-1 使用するソース・ファイルのエンコーディング。
bindir	値: 文字列 デフォルト: 該当なし コンパイラの実行可能ファイルが格納されているディレクトリへの絶対パス。デフォルトの javac コンパイラを使用するために、この属性を指定する必要はありません。
extdir	値: 文字列 デフォルト: 該当なし 該当する場合、コンパイラの拡張ライブラリの場所を指定します。
debug	値: ブール デフォルト: false コンパイル時間のデバッグ出力を生成する場合は、true に設定します。

<javacache-config>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ指定可能。

子要素: なし

Java Object Cache 構成ファイル javacache.xml へのパスを指定します。

表 B-14 <javacache-config> の属性

名前	説明
path	値: 文字列 デフォルト: ../../../../javacache/admin/javacache.xml javacache.xml ファイルへのパス。

<jms-config>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ指定可能。

子要素:

OC4J JMS 構成ファイルとして使用するファイルを指定します。

表 B-15 <jms-config> の属性

名前	説明
path	値: 文字列 デフォルト: jms.xml OC4J JMS 構成ファイルのファイル名およびパス。

<log>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ指定可能。

子要素: <file>

囲まれた <file> 要素は、OC4J サーバー・ログ・ファイルの場所を指定します。

<rmi-config>

親要素: <application-server>

必須 / 任意: 任意。1つのみ指定可能。

子要素:

OC4J RMI 構成ファイルとして使用するファイルを定義します。

表 B-16 <rmi-config> の属性

名前	説明
path	値: 文字列 デフォルト: rmi.xml OC4J RMI 構成ファイルのファイル名およびパス。

<shared-library>

親要素: <application-server>

必須 / 任意: 任意。複数指定可能。

子要素: <code-source>、<import-shared-library>

OC4J インスタンス内にインストールされる共有ライブラリを宣言します。共有ライブラリを構成および使用する方法は、『Oracle Containers for J2EE 開発者ガイド』を参照してください。

表 B-17 <shared-library> の属性

名前	説明
name	値: 文字列 デフォルト: 必須 /shared-lib ディレクトリ内に作成される共有ライブラリ・ディレクトリの名前。
version	値: 文字列 デフォルト: 必須 /shared-lib/library_name ディレクトリ内で共有ライブラリのアーカイブ・ファイルを格納するサブディレクトリの名前となるバージョン番号。
library-compatible	値: ブール デフォルト: false この属性は、内部使用専用です。

<shutdown-class>

親要素: <shutdown-classes>

必須 / 任意: 任意。複数指定可能。

子要素: <execution-order>、<init-param>

OC4J が終了する前に実行する停止クラスを <shutdown-classes> 要素内に定義します。

表 B-18 <shutdown-class> の属性

名前	説明
classname	値: 文字列 デフォルト: 必須 oracle.j2ee.server.OC4JShutdown インタフェースを実装するクラスの名前。

<startup-class>

親要素: <startup-classes>

必須 / 任意: 任意。複数指定可能。

子要素: <execution-order>、<init-param>

OC4J の初期化の際に実行する起動クラスを <startup-classes> 要素内に定義します。

表 B-19 <startup-class> の属性

名前	説明
classname	値: 文字列 デフォルト: 必須 oracle.j2ee.server.OC4JStartup インタフェースを実装するクラスの名前。
failure-is-fatal	値: ブール デフォルト: false true の場合、例外がスローされると、OC4J は例外を記録して終了します。false の場合、OC4J は例外を記録し、続行します。

<thread-pool>

親要素: <application-server>

必須 / 任意: 任意。複数指定可能。

子要素:

OC4J プロセス内の単一の system、http、jca、rmi request または rmi connection スレッド・プールの構成が含まれます。詳細は、10-2 ページの「[OC4J スレッド・プールの構成](#)」を参照してください。

表 B-20 <thread-pool> 属性

名前	説明
name	値: 文字列 デフォルト: 必須 system rmi request rmi connection http jca スレッド・プール名。次のいずれかの値である必要があります。 <ul style="list-style-type: none"> ■ system 古い形式では公開されていなかった非表示のスレッド・プール。 ■ rmi request RMI リクエストを処理するスレッド・プール。 ■ rmi connection RMI 接続でスレッドによりブロック読取りが実行されるスレッド・プール。 ■ http HTTP リクエスト、AJP リクエスト、RMI リクエスト (rmi request スレッド・プールが構成されていない場合) および RMI 接続 (rmi connection スレッド・プールが構成されていない場合) を処理するスレッド・プール。 ■ jca J2CA ワーク・マネージャ用のワーク管理スレッド・プール。 これらのプールのスレッドの名前にはそれぞれ、SystemThreadGroup_、RMIRequestThreadGroup_、RMIConnectionThreadGroup_、HTTPThreadGroup_ および WorkManager_ という接頭辞が付いており、接尾辞は増加するカウンタです。
min	値: 文字列 デフォルト: 0 OC4J が同時に実行できる最小スレッド数。
max	値: 文字列 デフォルト: 1024 OC4J が同時に実行できる最大スレッド数。
queue	値: 文字列 デフォルト: 0 キュー内に保持できるリクエストの最大数。

表 B-20 <thread-pool> 属性 (続き)

名前	説明
keepAlive	<p>値: 文字列 デフォルト: 600000</p> <p>新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。このタイムアウトは、スレッドのアイドル状態を維持する時間を指定します。タイムアウトに達するとスレッドは破棄されます。</p> <p>スレッドを破棄しないようにするには、-1 に設定します。</p> <p>デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。</p>
stackSize	<p>値: 文字列 デフォルト: 0</p> <p>スレッド・プール・スタックのサイズ。</p>
debug	<p>値: ブール デフォルト: false</p> <p>true の場合、起動時にアプリケーション・サーバーのスレッド・プール情報がコンソールに出力されます。false の場合、スレッド・プール情報は出力されません。</p>

<transaction-manager-config>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ指定可能。

子要素:

トランザクション・マネージャ構成ファイルを指定します。

表 B-21 <transaction-manager-config> の属性

名前	説明
path	<p>値: 文字列 デフォルト: transaction-manager.xml</p> <p>トランザクション・マネージャ構成ファイルのファイル名およびパス。デフォルトのファイルは、ORACLE_HOME/j2ee/instance/config/transaction-manager.xml です。</p>

<web-site>

親要素: <application-server>

必須 / 任意: 任意。複数指定可能。

子要素:

OC4J 内に定義された単一の Web サイトの構成ファイルを参照します。<web-site> 要素は、Web サイトごとに作成する必要があります。作成しないと、そのサイトは OC4J 内で有効になりません。詳細は、[第 13 章「OC4J での Web サイトの管理」](#)を参照してください。

表 B-22 <web-site> の属性

名前	説明
path	値: 文字列 デフォルト: 該当なし Web サイトを定義する *-web-site.xml 構成ファイルのファイル名およびパス。

<work-manager-thread-pool>

親要素: <application-server>

必須 / 任意: 任意。1 つのみ。

子要素:

OC4J プロセス内にあるリソース・アダプタのワーク管理スレッド・プールの構成が含まれます。詳細は、10-2 ページの「[OC4J スレッド・プールの構成](#)」を参照してください。

この要素は、OC4J 10g (10.1.3.4.0) では推奨されません。server.xml ファイルにこの要素が含まれる場合は、OC4J により等価の <thread-pool> 要素に変更されます。この要素は、jca スレッド・プールを定義します。

表 B-23 <work-manager-thread-pool> 属性

属性	説明
min	値: 文字列 デフォルト: 該当なし ワーク管理スレッド・プールに作成するスレッドの最小数。スレッド・プールを無効にするには、この値を 0 に設定します。
max	値: 文字列 デフォルト: 40 ワーク管理スレッド・プールに作成できるスレッドの最大数。 ワーク管理スレッド・プールでは、内部で 3 つのワーカー・スレッドを使用します。たとえば、max="16" と指定すると、リクエストの処理に使用可能なワーカー・スレッドは 13 のみです。同様に、max 値が 20 の場合には、使用可能なスレッドは 17 のみです。そのため、必要な最大のスレッド数に 3 を足してこの値を設定する必要があります。
queue	値: 文字列 デフォルト: 0 ワーク管理スレッド・プールでキュー内に保持できるスレッドの最大数。デフォルトの 0 を使用すると、ワーク・リクエストの突発的な急増に対処するためのキューは維持されません。

表 B-23 <work-manager-thread-pool> 属性 (続き)

属性	説明
keepAlive	値: 文字列 デフォルト: 600000 新規リクエストの待機中に、スレッドをアライブ (アイドル) にしておく時間 (ミリ秒単位)。タイムアウトに達するとスレッドは破棄されます。 スレッドを破棄しないようにするには、-1 に設定します。 デフォルト値の 600000 ミリ秒 (10 分) は、-1 ではない場合に設定できる最小値でもあります。
debug	値: ブール デフォルト: false true の場合、起動時にアプリケーション・サーバーのワーク管理スレッド・プール情報がコンソールに出力されます。false の場合、スレッド・プール情報は出力されません。

Web サイト構成ファイル (*-web-site.xml) の概要

この項での要素の説明は、default-web-site.xml などの OC4J Web サイト構成ファイルに適用されます。

<web-site>

必須 / 任意: 必須。1 つのみ。

子要素:

```
<description>
<frontend>
<web-app>
<default-web-app>
<user-web-apps>
<access-log>
<odl-access-log>
<ssl-config>
```

これは、Web サイト構成ファイルのルート要素です。

表 B-24 Web サイト構成ファイルの属性

名前	説明
custom-thread-pool	値: 文字列 デフォルト: 該当なし 必要に応じて、この構成ファイルの <web-app> 要素よってこの Web サイトにバインドされている、各アプリケーションで使用されるカスタム・スレッド・プールを指定します。
display-name	値: 文字列 デフォルト: 該当なし 必要に応じて、わかりやすいまたは非公式の Web サイト名を定義します。

表 B-24 Web サイト構成ファイルの属性 (続き)

名前	説明
host	<p>値: 文字列 デフォルト: 該当なし</p> <p>この Web サイトのホストを DNS ホスト名または IP アドレスのいずれかで指定します。サーバーが (複数の IP アドレスを保持する) マルチホーム・マシンの場合、ALL 設定を使用してすべての IP アドレスをリスニングできます。</p>
log-request-info	<p>値: ブール デフォルト: false</p> <p>エラーが発生した場合に、受信リクエストに関する情報を Web サイトのログに書き込むかどうかを指定します。Web サイトのログは、<access-log> または <odl-access-log> 要素のいずれかを使用して有効にします。これらの要素については、この項で後述します。(Web サイトのログの有効化の詳細は、13-17 ページの「Web モジュールまたはアプリケーションに対するアクセス・ロギングの有効化または無効化」を参照してください。)</p>
max-request-size	<p>値: 文字列 デフォルト: 15000</p> <p>受信 HTTP リクエストの最大サイズ (バイト単位) を設定します。この最大値を超えるリクエストをクライアントが送信すると、クライアントは「リクエスト・エンティティが大きすぎます」というエラーを受け取ります。デフォルトの最大値は 15000 です。</p>
secure	<p>値: ブール デフォルト: false</p> <p>Secure Socket Layer (SSL) 機能をサポートするかどうかを指定します。</p> <p>protocol 設定が ajp13 の場合 (Oracle Application Server 環境で使用)、true に設定すると、Oracle HTTP Server と OC4J の間のセキュアな AJP プロトコルとなります。protocol 設定が http の場合 (スタンドアロン OC4J で使用)、true に設定すると、クライアントと OC4J の間の HTTPS プロトコルとなります。</p> <p>また、secure="true" と設定した場合は、<ssl-config> 要素 (<web-site> 要素のサブ要素) を使用してキーストアのパスおよびパスワードを指定する必要があります。この要素については、この項で後述します。</p> <p>SSL および HTTPS 機能は、Oracle HTTP Server を介して Oracle HTTP Server とクライアントの間の通信にも使用できます。詳細は、『Oracle Application Server セキュリティ・ガイド』を参照してください。</p>
protocol	<p>値: 文字列 デフォルト: 該当なし</p> <p>Web サイトが使用しているプロトコルを指定します。有効な値は、http および ajp13 (AJP の場合デフォルト) です。Oracle Application Server の本番環境では、ajp13 設定のみを使用してください。AJP プロトコルは、Oracle HTTP Server および mod_oc4j とともに使用します。各プロトコルには対応するポートが、各ポートには対応するプロトコルが必要です。</p> <p>ajp13 または http 設定のいずれかをセキュア・モード (SSL) で使用するには、secure フラグを true に設定し、<ssl-config> サブ要素を使用してキーストアのパスおよびパスワードを指定する必要があります。この要素については、この項で後述します。</p> <p>protocol 属性の値を変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、range 属性を使用します。</p>

表 B-24 Web サイト構成ファイルの属性 (続き)

名前	説明
port	<p>値: 文字列 デフォルト: 該当なし</p> <p>この Web サイトのポート番号を指定します。各ポートには対応するプロトコルが、各プロトコルには対応するポートが必要です。スタンドアロン OC4J では、デフォルトで port 設定 8888 を使用して、OC4J リスナーに直接アクセスしますが、これは必要に応じて変更できます。</p> <p>Oracle Application Server 環境では、このポート設定は Oracle Process Management and Notification (OPMN) システムによって上書きされます。Oracle Application Server は、Oracle Web Cache が有効に設定された Oracle HTTP Server を介したアクセスに、ポート 7777 をデフォルトで使用します。</p> <p>UNIX 環境では、1024 未満のポート番号へのアクセスには root 権限が必要です。また、クライアント・ブラウザからポートの指定がない場合は、ポート 80 が HTTP プロトコル用、ポート 443 が HTTPS 用であると想定されます。</p> <p>protocol 属性の値を変更しても、ポート範囲は、指定したプロトコルのデフォルトの範囲にはリセットされません。ポート範囲を変更するには、range 属性を使用します。</p>
use-keep-alive	<p>値: ブール デフォルト: true</p> <p>サーブレット・コンテナの一般的な動作は、リクエストが完了した後に接続をクローズすることです。しかし、use-keep-alive 設定を true にすると、リクエスト間で接続が維持されます。AJP プロトコルの場合、常に接続が維持され、この属性は無視されます。その他のプロトコルの場合、デフォルトは true です。これを無効にすると、パフォーマンスが著しく低下する場合があります。</p>
virtual-hosts	<p>値: 文字列 デフォルト: 該当なし</p> <p>この任意の属性は、同じ IP アドレスを共有する仮想サイトに使用すると便利です。値は、この Web サイトに関連付けられるホスト名をカンマで区切ったリストです。</p>

<description>

Web サイトのオプションの簡単な説明が含まれます。

<frontend>

HTTP クライアントから見たときの、この Web サイトの認識可能なフロントエンド・ホストおよびポートを指定します。サイトがロード・バランサまたはファイアウォールの背後にある場合、URL リライティング（セッション ID を URL にエンコードする技術）などの機能に対する Web アプリケーション・コードに適切な情報を与えるには、<frontend> を指定する必要があります。

<frontend> 要素に指定されたホストおよびポートを使用して、アプリケーションを実行しているバックエンド・サーバーは、URL リライティングではサーバー自体ではなく、参照先のフロントエンドを認識します。したがって、後続のリクエストは、バックエンドに直接アクセスすることなく、フロントエンドから適切に受信されます。

表 B-25 に、<frontend> の属性を示します。

表 B-25 <frontend> の属性

名前	説明
host	値: 文字列 デフォルト: 該当なし www.acme.com など、フロントエンド・サーバーのホスト名を指定します。
port	80 など、フロントエンド・サーバーのポート番号を指定します。

<web-app>

この要素は、特定の Web モジュールをこの Web サイトにバインドします。server.xml ファイルの J2EE アプリケーション・アーカイブの名前 (.ear 拡張子を除いた EAR ファイル名) と、J2EE アプリケーション内の Web モジュールの名前を指定します。Web モジュールは、アプリケーションの EAR ファイル内にある J2EE の application.xml ファイル（または EAR ファイル内にある orion-application.xml ファイル）に定義されます。Web モジュールは、<web-app> 要素の root 属性に指定された場所にバインドされます。

注意： EAR ファイル内の WAR ファイルではなく、WAR ファイル単独でデプロイできます。スタンドアロン OC4J では、このような Web アプリケーションは OC4J のデフォルト・アプリケーションに追加されます。（OC4J では、常になんらかの親アプリケーションが存在する必要があります。）詳細は、1-10 ページの「[OC4J のアプリケーション階層の概要](#)」を参照してください。

この場合、Web サイトの XML ファイルの <web-app> 要素は、J2EE アプリケーション・アーカイブの名前ではなく、デフォルトのアプリケーションの名前を指定します。詳細は、次の属性の説明および例を参照してください。

特に application および name 属性に関する Web サイトの XML ファイルとの間のマッピングについては、このマニュアルの他の章で例を示します。6-9 ページの「[J2EE アプリケーション \(EAR\) のデプロイ](#)」（EAR ファイル内の WAR ファイルをデプロイする一般的な例）および 6-12 ページの「[スタンドアロン Web モジュール \(WAR\) のデプロイ](#)」（WAR ファイル単独で OC4J のデフォルト・アプリケーションにデプロイする例）を参照してください。

表 B-26 に、<web-app> の属性を示します。

表 B-26 <web-app> の属性

名前	説明
access-log	<p>値: 文字列 デフォルト: false</p> <p>Web サイトにリクエストを記録する OC4J アクセス・ロギングが Web モジュールに対して有効かどうかを指定します。アクセス・ロギングを有効化する場合は、true に設定します。ログ・ファイル管理が問題になる場合は、false に設定し、モジュールに対するアクセス・ロギングを無効にします。</p> <p>アクセス・ログ構成の詳細は、この項の <access-log> および <odl-access-log> 要素の説明を参照してください。</p>
application	<p>値: 文字列 デフォルト: 該当なし</p> <p>J2EE アプリケーション・アーカイブ名 (.ear 拡張子を除いた EAR ファイル名) を指定します。この名前は server.xml ファイルの <application> 要素の name 属性に対応します。</p> <p>OC4J のデフォルトのアプリケーションを親として使用し、スタンドアロン OC4J に WAR ファイルを単独でデプロイする場合、server.xml ファイルの <global-application> 要素に基づいて、application 属性はデフォルトのアプリケーションの名前をかわりに反映します。</p>
load-on-startup	<p>値: ブール デフォルト: false</p> <p>任意。アプリケーションの起動時に、Web モジュールを事前にロードするかどうかを指定します。指定しない場合は、最初のリクエスト時にロードされます。サポートされる値は、true および false です。デフォルトは false ですが、モジュールまたはアプリケーションが Oracle Enterprise Manager 10g Application Server Control を使用してデプロイされる場合は、この値を明示的に true に設定します。</p>
max-inactivity-time	<p>値: 文字列 デフォルト: 0</p> <p>任意。Web モジュールを非アクティブにしておく時間 (分単位) を指定します。この時間を経過すると、OC4J は Web モジュールを停止します。デフォルトでは、非アクティブが原因で Web モジュールが停止されることはありません。</p>
name	<p>値: ブール デフォルト: 該当なし</p> <p>特定の J2EE アプリケーション内の Web モジュールの名前を指定します。これは、J2EE の application.xml ファイルの <module> 要素の <web> サブ要素の <web-uri> 値 (.war 拡張子を除く) に対応します。J2EE の application.xml ファイルは EAR ファイル内にあります。</p>

表 B-26 <web-app> の属性 (続き)

名前	説明
root	<p>値: 文字列 デフォルト: 該当なし</p> <p>Web モジュールのバインド先のパスを指定します。この属性により、モジュールの起動に使用される URL のコンテキスト・ルート部分が定義されます。たとえば、Web サイト <code>www.example.com</code> の Web モジュール <code>CatalogApp</code> が、コンテキスト・ルート <code>/catalog</code> にバインドされる場合、モジュールは次のように起動できます。</p> <pre>http://www.example.com/catalog</pre> <p>root 属性は、J2EE の <code>application.xml</code> ファイルの対応する <code><web></code> 要素の <code><context-root></code> 値に優先します。コンテキスト・ルートとしてスラッシュ (/) を指定すると、OC4J のデフォルトの Web アプリケーションが上書きされます。</p> <p>アプリケーションのデプロイ時には、コンテキスト・ルートとして / を使用できます。次の例では、<code>admin_client.jar</code> を使用して WAR ファイルをデプロイし、/ にバインドしています。</p> <pre>% java -jar admin_client.jar deployer:oc4j:localhost oc4jadmin welcome1 \ -deploy -file d:my-web-store.war -deploymentName mws_2 \ -bindAllWebApps -contextRoot "/"</pre> <p>次の例で示すように、EAR ファイルにコンテキスト・ルートが / に設定された <code>application.xml</code> ファイルが含まれている場合、アプリケーションのデプロイ時には / がデフォルトのコンテキスト・ルートになります。</p> <pre><application> <display-name>Web-Store</display-name> <module> <web> <web-uri>my-web-store.war</web-uri> <context-root>/</context-root> </web> </module> </application></pre> <p>Oracle HTTP Server のデフォルトの ping の URL もスラッシュ (/) であるため、アプリケーションのデプロイ時に / をコンテキスト・ルートとして使用すると、次に示す問題のいずれかまたは両方が発生する可能性があります。</p> <ul style="list-style-type: none"> ■ Oracle HTTP Server 用の ping が直接 OC4J に移動します。 ■ 外部の HEAD リクエストが <code>*-web-access.log</code> ファイルに表示されます。 <p>これらの問題は、<code>ORACLE_HOME/Apache/Apache/conf/dms.conf</code> ファイルに <code>Oc4jMountCopy off</code> ディレクティブを配置することで回避できます。</p>

表 B-26 <web-app> の属性 (続き)

名前	説明
shared	<p>値: 文字列 デフォルト: false</p> <p>Web サイトがプロトコルとポートの特定のペアで定義される場合に、公開された Web モジュールを Web サイト間で共有することを許可します。サポートされる値は、true および false (デフォルト) です。shared="true" は、スタンドアロン OC4J でのみ使用してください。</p> <p>HTTPS の Web アプリケーションに共有のマークが付いている場合、セッション・トラッキングの方針は、SSL セッション・トラッキングから、Cookie や URL リライティングによるセッション・トラッキングに戻ります。Cookie の使用により、Web アプリケーションの安全性が低下する可能性はありますが、一部のブラウザでは適切にサポートされていない、SSL セッション・タイムアウトなどの問題に対処するためには必要です。</p>

<default-web-app>

この要素は、この Web サイトにバインドされるデフォルトの Web アプリケーションに対する参照を作成します。単一のアプリケーションが Web サイトにバインドされる場合 (Application Server Control など)、そのアプリケーションをこの要素内に指定します。

ユーザーにとって、この要素はスタンドアロン OC4J 環境のみに関係します。Oracle Application Server 環境では、OC4J のデフォルトの Web アプリケーションにはシステムレベルの機能がありますが、他では価値がありません。

<default-web-app> 要素は、前述の <web-app> 要素と同じ属性を使用しますが、load-on-startup のデフォルト設定が true であることに注意してください。

<user-web-apps>

この要素は、ユーザー・ディレクトリおよびアプリケーションをサポートするために使用します。各ユーザーには、Web モジュールおよび関連する web-application.xml ファイルがあります。ユーザー・アプリケーションは、サーバー・ルートの /username/ に配置されます。

表 B-27 に、<user-web-apps> の属性を示します。

表 B-27 <user-web-apps> の属性

名前	説明
max-inactivity-time	<p>値: int 型 デフォルト: 該当なし</p> <p>任意の整数属性。Web モジュールを非アクティブにしておく時間 (分単位) を指定します。この時間を経過すると、OC4J は Web モジュールを停止します。デフォルトでは、非アクティブが原因で Web モジュールが停止されることはありません。</p>
path	<p>ユーザー・アプリケーションのローカル・ディレクトリを指定するパスを指定します (ユーザー名にワイルドカードも使用可能)。たとえば、UNIX 環境でのデフォルト・パスの設定は /home/username です。username は、特定のユーザー名に置き換えられます。</p>

<access-log>

この要素は、この Web サイトに対してテキストベースのアクセス・ロギングを有効にし、アクセス・ログに関する情報（パス、ファイル名、含まれる情報など）を指定するために使用します。ログ・ファイルには、受信リクエスト（Web サイトの各アクセス）が記録されます。

構成の詳細は、13-15 ページの「[テキストベースのアクセス・ロギングの構成](#)」を参照してください。

<odl-access-log>

この要素は、Web サイトに対して ODL ベースのアクセス・ロギングを有効にし、アクセス・ログに関する情報（パス、各ファイルのサイズの最大値、ログ・ディレクトリ内のすべてのファイルの合計サイズなど）を指定するために使用します。ログ・ファイルには、受信リクエスト（Web サイトの各アクセス）が記録されます。

構成の詳細は、13-16 ページの「[ODL アクセス・ロギングの構成](#)」を参照してください。

<ssl-config>

該当する場合、この要素は SSL 構成設定を指定します。<web-site> 要素の secure 属性を true に設定する場合は、この要素を常に使用する必要があります。

<ssl-config> のサブ要素は次のとおりです。

<property>

表 B-28 に、<ssl-config> の属性を示します。

表 B-28 <ssl-config> の属性

名前	説明
keystore	<p>値: 文字列 デフォルト: 該当なし</p> <p>このインストールでユーザー・ベースの証明書および鍵を格納するためにこの Web サイトで使用されるキーストア・データベース（バイナリ・ファイル）に対する相対パスまたは絶対パス。パスの値にはファイル名が含まれます。相対パスは、Web サイトの XML ファイルの位置に対して相対的です。</p> <p>キーストアは、<code>java.security.KeyStore</code> インスタンスであり、Sun 社の JDK に付属する <code>keytool</code> ユーティリティを使用して作成およびメンテナンスできます。</p>
keystore-password	<p>値: 文字列 デフォルト: 該当なし</p> <p>キーストアのオープンに必要なパスワード。</p>
needs-client-auth	<p>値: 文字列 デフォルト: false</p> <p>OC4J に対するクライアント（Oracle HTTP Server など）であるエンティティが、OC4J と通信するための認可に証明書を発行する必要があるかどうかを示します。サポートされる値は、true（クライアント認証、証明書は必須の場合）および false（デフォルト、証明書は不要の場合）です。</p>

表 B-28 <ssl-config> の属性 (続き)

名前	説明
provider	<p>値: 文字列 デフォルト: <code>com.sun.net.ssl.internal.ssl.Provider</code></p> <p>Java Secure Socket Extension (JSSE) を使用する場合、この属性を使用してプロバイダを指定できます。</p> <p>デフォルトでは、OC4J は通常 Sun 社の SSL 実装を採用します。しかし、SOAP や <code>http_client</code> などの場合、OC4J は Oracle の SSL 実装を採用する場合があります。</p>
factory	<p>値: 文字列</p> <p>JSSE を使用しない場合、<code>factory</code> 属性を使用して <code>SSLServerSocketFactory</code> の実装を指定します。</p> <p>サード・パーティの <code>SSLServerSocketFactory</code> 実装を使用する場合、<ssl-config> 要素の <property> サブ要素を使用してパラメータをファクトリに送信できます。</p> <p><code>factory</code> 属性およびそのパラメータは、OC4J 10g (10.1.3.4.0) では推奨されません。</p>

セッション状態表の概要

この付録では、OC4J のデータベース永続性メカニズムで使用されるデータベース表のスキーマについて説明します。このメカニズムの詳細は、9-10 ページの「[データベース・レプリケーションの構成](#)」を参照してください。

OC4J_HTTP_SESSION

この表には、アプリケーションの識別子やセッションのユーザー設定プロパティなど、単一の HTTP セッションに対するメタデータが格納されます。ID が主キーです。

OC4J_HTTP_SESSION 表と OC4J_HTTP_SESSION_VALUE 表の間には、1 対多のリレーションシップがあります。OC4J_HTTP_SESSION 表の各エントリーには、OC4J_HTTP_SESSION_VALUE 表のエントリーが 0 (ゼロ) 以上あります。

表 C-1 OC4J_HTTP_SESSION 表の説明

名前	NULL 指定	データ型	説明
ID	NOT_NULL	VARCHAR2 (100)	一意のセッション ID。
APPLICATION_ID	NULL	VARCHAR2 (100)	セッションが属するアプリケーションに割り当てられる OC4J の内部 ID。
IP	NULL	NUMBER (38)	アプリケーションのホストとなるマシンの IP アドレス。
LAST_ACCESSED	NULL	NUMBER (38)	現在のレコードが最後に更新された時刻。
USER_NAME	NULL	VARCHAR2 (50)	セッションのアプリケーション・ユーザー設定値に対するユーザー名。
MAX_INACTIVE_TIME	NULL	NUMBER (38)	期限切れになるまで、セッションのアイドル状態を維持できる最大時間。この最大時間を超えると、セッション・データは保存されません。
CREATION_TIME	NULL	NUMBER (38)	表が作成された時刻。

OC4J_HTTP_SESSION_VALUE

この表には、各 HTTP セッションのプロパティとアプリケーション・ユーザーによって設定された値が格納されます。値はバイナリ・ラージ・オブジェクト (BLOB) として格納されます。ID と KEY_FIELD 値を組み合わせて主キーを構成します。

表 C-2 OC4J_HTTP_SESSION_VALUE 表の説明

名前	NULL 指定	データ型	説明
ID	NOT_NULL	VARCHAR2 (100)	一意のセッション ID。
KEY_FIELD	NOT_NULL	VARCHAR2 (100)	アプリケーション・ユーザーによってセッションに設定されたプロパティの名前。
VALUE_FIELD	NULL	BLOB	セッションに設定されたプロパティの値。

OC4J_EJB_SESSION

この表には、ステートフル・セッション Bean の現在の状態が格納されます。状態データはバイナリ・ラージ・オブジェクト (BLOB) として格納されます。ID が主キーです。

表 C-3 OC4J_EJB_SESSION 表の説明

名前	NULL 指定	データ型	説明
ID	NOT_NULL	VARCHAR2 (100)	一意のセッション ID。
VALUE_FIELD	NULL	BLOB	セッション Bean の現在の状態データ。
LOCATION	NULL	NUMBER (38)	セッション Bean のバインド先の JNDI 名。
CHECKSUM	NULL	NUMBER (38)	バイトが正しい形式かどうかを検証するために内部的に使用されます。
PASSIVATE	NULL	NUMBER (38)	Bean が受動化されているかどうかを示すブール値。true の場合、受動化された Bean はディスクから取得されます。
LAST_ACCESSED	NULL	NUMBER (38)	現在のレコードが最後に更新された時刻。
USER_NAME	NULL	VARCHAR2 (50)	セッションのアプリケーション・ユーザー設定値に対するユーザー名。
MAX_INACTIVE_TIME	NULL	NUMBER (38)	期限切れになるまで、セッションのアイドル状態を維持できる最大時間。この最大時間を超えると、セッション・データは保存されません。
CREATION_TIME	NULL	NUMBER (38)	表が作成された時刻。

サード・パーティ・ライセンス

この付録では、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・ライセンスを示します。

ANTLR

このプログラムには、ANTLR のサード・パーティ・コードが組み込まれています。ANTLR 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（ANTLR 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。

The ANTLR License

Software License

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

Apache

このプログラムには、Apache Software Foundation（Apache）のサード・パーティ・コードが組み込まれています。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。

Apache のライセンス契約は、次の付属の Apache コンポーネントに適用されます。

- Apache HTTP Server
- Apache JServ
- mod_jserv
- 正規表現パッケージのバージョン 1.3
- Apache Expression Language（commons-el.jar に同梱）
- mod_mm 1.1.3
- Apache XML Signature および Apache XML Encryption のバージョン 1.4（Java 版）およびバージョン 1.0（C++ 版）
- log4j 1.1.1
- BCEL バージョン 5
- XML-RPC バージョン 1.1
- Batik バージョン 1.5.1
- ANT 1.6.2 および 1.6.5
- Crimson バージョン 1.1.3
- ant.jar
- wsif.jar
- bcel.jar

- soap.jar
- Jakarta CLI 1.0
- jakarta-regexp-1.3.jar
- JSP 標準タグ・ライブラリ 1.0.6 および 1.1
- Struts 1.1
- Velocity 1.3
- svnClientAdapter
- commons-logging.jar
- commons-el.jar
- standard.jar
- jstl.jar

The Apache Software License

License for Apache Web Server 1.3.29

```

/* =====
* The Apache Software License, Version 1.1
*
* Copyright (c) 2000-2002 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.
 * =====
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation. For more
 * information on the Apache Software Foundation, please see
 * <<http://www.apache.org/>>.
 *
 * Portions of this software are based upon public domain software
 * originally written at the National Center for Supercomputing
 Applications,
 * University of Illinois, Urbana-Champaign.

License for Apache Web Server 2.0

Copyright (c) 1999-2004, The Apache Software Foundation
 Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
 file except in compliance with the License. You may obtain a copy of the License at
<http://www.apache.org/licenses/LICENSE-2.0>
 Unless required by applicable law or agreed to in writing, software distributed under
 the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 KIND, either express or implied. See the License for the specific language governing
 permissions and limitations under the License.
 Copyright (c) 1999-2004, The Apache Software Foundation
 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.

Apache SOAP

このプログラムには、Apache Software Foundation (Apache) のサード・パーティ・コードが組み込まれています。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 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

Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

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 モジュール

このプログラムには、Tim Bunce 氏のサード・パーティ・コードが組み込まれています。Tim Bunce 氏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を求められています。ただし、Oracle プログラム (Tim Bunce 氏の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、Tim Bunce 氏の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 Tim Bunce 氏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

FastCGI

このプログラムには、Open Market, Inc のサード・パーティ・コードが組み込まれています。Open Market のライセンス条件に基づき、Oracle は、次の条件で Open Market ソフトウェアのライセンスをユーザーに付与することが求められています。この製品に付随する Oracle プログラム・ライセンスに含まれる条件は、Open Market ソフトウェアには適用されず、Open Market ソフトウェアを使用する権利は、次に規定されている内容に限られます。Oracle は、Open Market ソフトウェアのパフォーマンスについては責任を負いません。また、Open Market ソフトウェアの技術サポートは提供しておらず、Open Market ソフトウェアの使用により生じた損害については、一切責任を負いません。

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.

Info-ZIP Unzip パッケージ

このプログラムには、Info-ZIP のサード・パーティ・コードが組み込まれています。Info-ZIP 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 (Info-ZIP 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、Info-ZIP 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 Info-ZIP から提供されません。

The Info-ZIP Unzip Package License

Copyright (c) 1990-1999 Info-ZIP. All rights reserved. For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoince Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "AS IS," without warranty of any kind, express or implied. In no event shall InfoZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software."

JSR 110

このプログラムには、IBM 社 (IBM) のサード・パーティ・コードが組み込まれています。IBM 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 (IBM 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、IBM 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 IBM から提供されません。

Copyright IBM Corporation 2003 - All rights reserved

Java APIs for the WSDL specification are available at:
<http://www-124.ibm.com/developerworks/projects/wsdl4j/>

Jaxen

このプログラムには、Apache Software Foundation (Apache) および Jaxen Project (Jaxen) のサード・パーティ・コードが組み込まれています。Apache および Jaxen 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 および Jaxen の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。

The Jaxen License

Copyright (C) 2000-2002 bob mcwhirter & James Strachan. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

The name "Jaxen" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact licensejaxen.org.

Products derived from this software may not be called "Jaxen", nor may "Jaxen" appear in their name, without prior written permission from the Jaxen Project Management (pmjaxen.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgment equivalent to the following: "This product includes software developed by the Jaxen Project (<http://www.jaxen.org/>).". Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jaxen.org/>.

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 Jaxen AUTHORS OR THE PROJECT 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.

This software consists of voluntary contributions made by many individuals on behalf of the Jaxen Project and was originally created by bob mcwhirter and James Strachan . For more information on the Jaxen Project, please see <http://www.jaxen.org/>.

JGroups

このプログラムには、GNU のサード・パーティ・コードが組み込まれています。GNU 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（GNU 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、GNU 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 GNU から提供されません。

The GNU License

GNU Lesser General Public License
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its

derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and an idea of what it does.> Copyright (C) <year>  
<name of author>
```

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library 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 the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

mod_mm および mod_ssl

このプログラムには、Ralf S. Engelschall (Engelschall) のサード・パーティ・コードが組み込まれています。Engelschall 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 (Engelschall 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、mod_mm 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 Engelschall から提供されません。

mod_mm

Copyright (c) 1999 - 2000 Ralf S. Engelschall. All rights reserved.
This product includes software developed by Ralf S. Engelschall <rse@engelschall.com>
for use in the mod_ssl project (<http://www.modssl.org/>).

mod_ssl

Copyright (c) 1998-2001 Ralf S. Engelschall. All rights reserved.
This product includes software developed by Ralf S. Engelschall <rse@engelschall.com>
for use in the mod_ssl project (<http://www.modssl.org/>).

OpenSSL

このプログラムには、OpenSSL Project のサード・パーティ・コードが組み込まれています。OpenSSL Project 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 (OpenSSL 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。

OpenSSL License

```
/* =====
 * Copyright (c) 1998-2005 The OpenSSL Project. 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. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *
```

```

* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "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 OpenSSL PROJECT 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.
* =====
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to.  The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code.  The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* 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 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. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*    Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).

```

```

* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND
* ANY EXPRESS 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 AUTHOR OR 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.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

```

Perl

このプログラムには、Comprehensive Perl Archive Network (CPAN) のサード・パーティ・コードが組み込まれています。CPAN 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 (CPAN 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。

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 1.29 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.
 * =====
 */

```

mod_perl 1.99_16 License

Copyright (c) 1999-2004, The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 1999-2004, The Apache Software Foundation

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.

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

SAXPath

このプログラムには、SAXPath のサード・パーティ・コードが組み込まれています。SAXPath 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 (SAXPath 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、SAXPath 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 SAXPath から提供されません。

The SAXPath License

Copyright (C) 2000-2002 werken digital. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

The name "SAXPath" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@saxpath.org.

Products derived from this software may not be called "SAXPath", nor may "SAXPath" appear in their name, without prior written permission from the SAXPath Project Management (pm@saxpath.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgment equivalent to the following: "This product includes software developed by the SAXPath Project (<http://www.saxpath.org/>).". Alternatively, the acknowledgment may be graphical using the logos available at <http://www.saxpath.org/>.

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 SAXPath AUTHORS OR THE PROJECT 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. This software consists of voluntary contributions made by many individuals on behalf of the SAXPath Project and was originally created by bob mcwhirter and James Strachan . For more information on the SAXPath Project, please see <http://www.saxpath.org/>.

W3C DOM

このプログラムには、World Wide Web Consortium (W3C) のサード・パーティ・コードが組み込まれています。W3C 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 (W3C 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、W3C 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 W3C から提供されません。

The W3C License

W3C® SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.) THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

索引

記号

<cluster> 要素, 9-12
<file> 要素, 11-4
<log> 要素, 11-4, 11-5
<odl> 要素, 11-5
<session-tracking> 要素, 13-11
<web-app> 要素, 13-10

数字

2 フェーズ・コミット・トランザクション・コーディネータ, 1-5

A

-addDataSourceConnectionPool コマンド, 6-23
-addDestination コマンド, 6-30
-addJMSConnectionFactory コマンド, 6-29
-addManagedDataSource コマンド, 6-25
-addNativeDataSource コマンド, 6-26
admin_client.jar ツール
 使用, 3-4, 6-1
 停止, 6-22
admin.jar ツール
 アンデプロイ, 6-17
 管理, 3-4
 使用, 7-1
 停止, 5-3, 5-4
AJP, 概要, 1-10
Ant タスク
 Ant タスクを使用したスクリプト・デプロイ, 1-9
 OC4J, 1-9
Apache JServ Protocol, 「AJP」を参照
Application Server Control
 Web サイトの作成, 13-9
 概要, 3-2
 起動および停止, 6-22
associateUsingThirdTable プロパティ, 4-10

B

-bindWebApp コマンド, 6-15

C

cookie-domain 属性, 13-11
Cookie ドメイン, 13-11

createinstance コーティリティ, 8-7, 8-8

D

dedicated.connection 設定, 4-9
dedicated.rmicontext プロパティ, 4-9, 4-10
default-web-site.xml ファイル, 13-2
default アプリケーション, 1-11
DefineColumnType プロパティ, 4-11
-describeSharedLibrary コマンド, 6-21
DTD, OC4J への登録, 14-1

E

EJB 3.0 のサポート, 1-4
EJB モジュール, 更新, 6-18, 7-7

G

GenerateIIOP プロパティ, 4-9
-getDataSourcesDescriptor コマンド, 6-28
-getDestinations コマンド, 6-31
-getJMSConnectionFactory コマンド, 6-30

H

http.method.trace.allow プロパティ, 4-13
http.request.debug プロパティ, 4-8, 4-13
HTTPS
 クライアント認証, 13-13
 セキュアな Web サイト, 13-11
HTTP メソッド, トレース, 4-13

I

InitialContext, 4-9

J

J2EE
 サポートされる API, 1-3
 定義, 1-2
J2EE Management のサポート, 12-2
J2EE_HOME 環境変数, 2-2, 4-9
j2ee-logging.xml, 11-10
Java Management Extensions のサポート, 12-2
JAVA_HOME 環境変数, 2-2
java.awt.headless プロパティ, 4-8

java.ext.dirs プロパティ, 4-8
java.io.tmpdir プロパティ, 4-8
java.lang.OutOfMemory エラー, A-3
JConsole, 6-32
JDK, サポートされるバージョン, 1-2
JMX クライアントからの OC4J の管理, 6-33
JMX 通知, サブスクリプト, 12-6
JMX のサポート, 12-2
JSR-77 のサポート, 12-2
JVM, 1-2

K

KeepIOPCode プロパティ, 4-9

L

-listSharedLibraries コマンド, 6-21
LoadBalanceOnLookup プロパティ, 4-9
loadbalancer.jar, 9-2

M

MBean
 アクセス, 12-5
 アプリケーション固有, 12-6
 永続ポリシー, 12-5
 概要, 12-2
 使用, 12-5
 通知, 12-6
MBeanServer, 6-32
mod_oc4j
 ロード・バランシング・アルゴリズムの選択, 8-28
 ロード・バランシング・オプション, 8-26
mod_oc4j.conf
 Oc4jRoutingWeight ディレクティブ, 8-27
 Oc4jSelectMethod ディレクティブ, 8-27
 ロード・バランシングの構成, 8-27
mod_oc4j モジュール, 1-10, 8-23
-modifySharedLibrary コマンド, 6-20

N

needs-client-auth 属性, 13-13

O

OC4J
 Ant タスク, 1-9
 mod_oc4j, 8-26
 インスタンスの新規作成, 8-7
 管理, 3-4, 6-1, 7-1
 起動, 5-2
 コマンドライン・インタフェース, 3-4, 6-1, 7-1
 コマンドライン・オプション, 4-5
 再起動, 5-5, 6-22
 システム・プロパティ, 4-5
 停止, 5-3, 5-4, 6-22
 トラブルシューティング, A-1
 バージョン, 7-3
 ロード・バランシング, 8-26
OC4J_EJB_SESSION, C-3
oc4j_extended, 2-1

OC4J_HTTP_SESSION, C-1
OC4J_HTTP_SESSION_VALUE, C-2
oc4j.cmd バッチ・ファイル, 3-5
oc4j-connectors.xml, B-3
oc4j.jar ツール
 起動, 5-2
oc4j-ra.xml, B-5
Oc4jRoutingWeight ディレクティブ, 8-27
Oc4jSelectMethod ディレクティブ, 8-27
OC4J インスタンスのグループの停止, 6-22
OC4J インスタンスの追加作成, 8-7
oc4j シェル・スクリプト, 3-5
oc4j 実行可能スクリプト, 3-5
OC4J の停止, 6-22
OC4J のトラブルシューティング, A-1
OC4J を管理するためのリモート・クライアント, 6-31
OC4J をリモートから管理するための JMX クライアント,
 6-33
ODL アーカイブ, 11-6
ODL ログ, 11-6
ONS, 8-2
opmnassociate ツール, 8-16
opmnctl
 config port コマンド, 13-5
 OC4J の起動に使用, 5-2
 Web サイトのポートの構成, 13-5
 クラスタの構成, 8-15
OPMN 管理のレプリケーション, 9-8
Oracle Diagnostic Logging, 11-5
Oracle Diagnostic Logging (ODL)
 ファイルの名前付け, 11-7
 「ロギング」も参照
 ログ循環, 11-6
Oracle HTTP Server, 1-10
 mod_oc4j モジュール, 8-23
 概要, 1-10
 動的 OC4J 検出, 8-29
 ルーティング ID, 8-23
 ロード・バランシング, 8-23
Oracle JAAS Provider ユーザー・コンテキスト, B-4
Oracle Notification Server, 8-2
ORACLE_HOME 環境変数, 2-2
oracle.dms.gate 設定, 4-10
oracle.dms.sensors 設定, 4-10
oracle.j2ee.rmi.loadBalance プロパティ, 4-10
oracle.mdb.fastUndeploy プロパティ, 4-10
oracle-webservices.xml, B-5
oracle ログ出力, 構成, 11-10
Out of Memory エラー, 4-8

P

-publishSharedLibrary コマンド, 6-19

R

-removeDataSourceConnectionPool コマンド, 6-24
-removeDestination コマンド, 6-31
removeinstance コーティリティ, 8-10
-removeJMSConnectionFactory コマンド, 6-29
-removeManagedDataSource コマンド, 6-26
-removeNativeDataSource コマンド, 6-27
-removeSharedLibrary コマンド, 6-21

-restart コマンド, 6-22

S

Secure Sockets Layer, 「SSL」を参照

server.xml ファイル, B-6

-shutdown コマンド, 6-22

SSL

クライアント認証, 13-13

セキュアな Web サイト, 13-11

-start コマンド, 6-22

-stop コマンド, 6-22

system-application.xml, 1-10, B-3

system アプリケーション, 1-10, B-3

概要, 1-10

構成, 1-10

T

taskmanager-granularity 属性, 10-2

-testDatabaseConnection コマンド, 6-27

-testDataSourceConnectionPool コマンド, 6-24

-testDataSource コマンド, 6-28

U

-updateEJBModule コマンド, 6-18

V

-validateURI コマンド, 6-5

W

Web サーバー

Oracle HTTP Server, 1-10

ネイティブな OC4J リスナー, 1-8

Web サービスのサポート, 1-4

Web サイト

opmn.xml での参照, 13-10

OPMN での構成, 13-3

server.xml での参照, 13-9

起動および停止, 13-14

クライアント認証, 13-13

構成ファイル, 13-8

サイトの新規作成, 13-8

スタンドアロン OC4J での構成, 13-3

セキュア, 13-11

複数のサイトへのアプリケーションのバインド,
13-10

Web 通信

Oracle Application Server インストール, 1-10

スタンドアロン OC4J インストール, 1-8

ws.debug プロパティ, 4-13

X

XML スキーマ, 参照, B-2

XML ファイル, 変更済ファイルのリロード, 7-4

XSD

OC4J への登録, 14-1

参照, B-2

あ

アクセス・ロギング, 13-14

ODL 形式, 11-6, 13-16

Web モジュールに対する無効化, 13-17

構成, 13-14

テキストベース, 13-15

アプリケーション

system アプリケーション, B-3

アンデプロイ, 6-8, 7-4

起動、再起動または停止, 6-22

再起動, 7-7

デプロイ, 6-8, 7-4

複数の Web サイトへのバインド, 13-10

アプリケーションの起動, 6-22

アプリケーションの停止, 6-22

アプリケーション・マウント・ポイント, 8-29

い

インスタンス, 追加の OC4J の作成, 8-7

え

エンティティ・リゾルバ, 14-1

か

ガベージ・コレクション, サーバーのパフォーマンスへの影響, A-3

環境変数

J2EE_HOME, 2-2, 4-9

JAVA_HOME, 2-2

ORACLE_HOME, 2-2, 4-8

設定, 2-2

管理環境での OC4J インスタンスの停止, 6-22

管理クライアント・ユーティリティ, 6-33

き

共有ライブラリ

インストール, 6-19

管理, 6-18, 6-19

作成, 6-18, 6-19

く

クラスタリング

10g リリース 3 (10.1.3) における変更点, 8-3

JGroups, 9-7

peer-to-peer レプリケーション, 9-8

アイランド, 9-2

概要, 9-2

データベース・スキーマ, C-1

データベース・レプリケーション, 9-10

動的 peer-to-peer レプリケーション, 9-8

動的ノード検出, 8-13

トポロジ, 8-2

ノード構成, 8-2

マルチキャスト・レプリケーション, 9-7

無効化, 9-12

レプリケーションのオプション, 9-4

グローバル Web アプリケーション, 1-11

リ

構成ファイル

- OC4J 固有のリスト, B-4
 - server.xml, B-6
 - system-application.xml, 1-10
 - 概要, B-2
 - 現行リリースにおける変更点, 1-5
- コマンドライン・オプション, 4-5

さ

再起動

- OC4J, 5-5, 6-22
- アプリケーション, 6-22

し

- システム MBean ブラウザ, 12-5
- システム・プロパティ, 4-5

す

- スキーマ, 参照, B-2
- スクリプト・デプロイ
 - Ant タスクの使用, 1-9
- スタンドアロン環境での OC4J の起動, 5-2
- スタンドアロン構成での OC4J の停止, 5-3, 5-4, 6-22
- スレッド・プール, 構成, 10-5

せ

- セキュアな Web サイト, 13-11
- セキュリティ, OC4J および Oracle HTTP Server の構成, 13-12

た

- タスク・マネージャ
 - 実行間隔, 10-2
 - 精度の設定, 10-2

て

データソース

- 新しい構成への変換, 7-10
- アプリケーション用に作成, 7-8
- 削除, 7-10
- テスト, 7-10

ね

- ネットワーク・インタフェース・カードのバインド, 9-7

は

- バージョン, OC4J リリース, 7-3
- パスワード
 - OC4J の変更, 3-7
 - クラスタ内のすべてのインスタンスに対する変更, 3-7
- パフォーマンス, oracle.dms.sensors 設定, 4-10

パフォーマンス設定

- dedicated.connection, 4-9
- dedicated.rmicontext, 4-9, 4-10
- DefineColumnType, 4-11
- LoadBalanceOnLookup, 4-9
- oracle.dms.gate, 4-10
- oracle.j2ee.rmi.loadBalance, 4-10
- タスク・マネージャの精度, 10-2

ま

- マウント・ポイント, 8-29

ら

ライブラリ

- 共有ライブラリのインストール, 6-19
- 共有ライブラリの管理, 6-18, 6-19
- 共有ライブラリの作成, 6-18, 6-19

り

リソース・アダプタ

- アンデプロイ, 7-12
- デプロイ, 7-12

- リモート・クライアントを使用した OC4J の管理, 6-31
- リモートの Oracle Application Server への接続, 6-32

る

ルーティング ID

- Web ルーティングの管理に使用, 8-23
- 作成, 8-23

ろ

ロード・バランシング

- Oracle HTTP Server, 8-23
- アルゴリズム, mod_oc4j, 8-26

ロギング

- Oracle Diagnostic Logging, 11-5
- プレーン・テキスト, 11-3
- ログ・ファイルのサマリー, 11-2
- ログ・ファイルの循環, 11-5