

Oracle® Communication and Mobility Server

インストレーション・ガイド

10g リリース 3 (10.1.3)

部品番号 : B50836-01

2008 年 9 月

Oracle Communication and Mobility Server インストール・ガイド, 10g リリース 3 (10.1.3)

部品番号 : B50836-01

原本名 : Oracle Communication and Mobility Server Installation Guide, 10g Release 3 (10.1.3)

原本部品番号 : E12657-02

Copyright © 2006, 2008, Oracle. All rights reserved.

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

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

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle 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	v
対象読者	vi
ドキュメントのアクセシビリティについて	vi
関連ドキュメント	vi
表記規則	vii
サポートおよびサービス	vii
1 製品およびインストールの概要	
製品の概要	1-2
今回のリリースの新機能	1-2
インストールの前提条件	1-2
システム要件	1-2
ポート要件	1-3
ポートが使用中かどうかの確認	1-3
インストール・モード	1-3
スタンドアロン開発者モードのインストール	1-4
既存の Oracle Application Server へのインストール	1-4
インストールされるコンポーネント	1-4
2 Oracle Communication and Mobility Server のインストール	
高可用性環境での OCMS インスタンス・クラスタのインストール	2-2
クラスタ環境での Edge Proxy を使用した OCMS の構成	2-2
インストールのサイズ設定	2-2
複数インスタンス・インストールでのポートの割当て	2-2
Edge Proxy のインストール	2-3
Oracle Communication and Mobility Server のインストール	2-3
インストール・タイプの選択	2-4
プライマリ・サーバー・アドレスの指定	2-5
OC4J Admin パスワードの指定	2-6
Oracle DB の詳細の指定	2-7
SDP データファイル・ディレクトリ	2-8
DB スキーマ	2-9
STUN サーバー構成の指定	2-9
テスト・ユーザーの構成	2-10
テスト・ユーザーの詳細	2-11
SIP コンテナの構成	2-12

サマリー情報	2-13
Oracle Remote Method Invocation (RMI) ポートのリスト	2-14
OCMS インストールの確認	2-14
Oracle Communication and Mobility Server の起動と停止	2-15
Edge Proxy の起動と停止	2-16
削除	2-16
インストールの問題のトラブルシューティング	2-16
ポートの競合	2-16
インストール中のネットワーク接続の切断	2-16

3 OCMS のインストールおよび機能の確認

Oracle Communicator のインストールと OCMS インストールの確認	3-2
サンプル・ユーザーのプロビジョニング	3-2
ログ・レベルの設定	3-3
Oracle Communicator のインストールと構成	3-3
Oracle Communicator FileTransferServlet のインストール	3-6
サブレットの登録の確認	3-6
Presence サーバーのテスト	3-8
ユーザーのプレゼンスのサブスクライブ	3-8
ユーザーのプレゼンス公開のテスト	3-9
イベント通知の受信のテスト	3-10
SIP のテスト・コール	3-10
SIP から PSTN へのテスト・コール	3-10
Ethereal による OCMS ネットワーク・トラフィックの監視	3-11

4 バックエンド Oracle RAC データベースに対する Oracle Communication and Mobility Server のインストール

RAC 情報の収集	4-2
RAC に対する Oracle Communication and Mobility Server のインストール	4-2
RAC データベースでのサービスの作成	4-2
Oracle Communication and Mobility Server のインストール	4-3
RAC データベースを使用するための Oracle Communication and Mobility Server のインストール後の構成	4-5

5 Presence の大規模デプロイ・インストール

概要	5-2
定義	5-2
Presence クラスタ	5-3
XDM クラスタ	5-3
Presence マルチノード・トポロジ	5-4
コンポーネントの概要	5-5
ロード・バランス	5-5
User Dispatcher	5-5
Presence サーバー	5-5
XDM サーバー	5-5
Aggregation Proxy	5-5
データベース	5-5
Presence ノード	5-5

XDM ノード	5-6
インストール	5-7
サンプル・ネットワーク	5-7
Oracle Application Server 10.1.3.4 のインストール	5-9
管理ノードのインストール	5-9
Presence ノードのインストール	5-10
Oracle Application Server のインストール	5-10
User Dispatcher のインストール	5-10
追加インスタンスの作成	5-11
OC4J インスタンスの構成	5-12
共有ライブラリを使用するためのインスタンスの構成	5-12
ロギングの構成	5-12
JVM の起動および停止パラメータの構成	5-13
SIP ポートの構成	5-14
適切な IP アドレスでリスニングするための Sip サブレット・コンテナの構成	5-14
xcap 構成ディレクトリの構成	5-15
springbeans.xml の構成	5-15
構成設定の確認	5-15
Presence のデプロイおよび構成	5-16
PresRulesXCAPUri および PIDFManipulationXCAPUri の変更	5-16
UserAgentFactoryService ポートの更新	5-17
JGroups の有効化	5-17
User Dispatcher の構成	5-17
インストールのチューニング	5-18
WebCenter インスタンスの無効化	5-18
ホーム・インスタンスの無効化	5-18
XDM ノードのインストール	5-18
Oracle Application Server 10.1.3.2 のインストール	5-19
Oracle Application Server への 10.1.3.4 パッチの適用	5-19
User Dispatcher のインストール	5-19
追加 OC4J インスタンスの作成	5-19
OC4J インスタンスの構成	5-20
インスタンスの停止	5-21
共有ライブラリを使用するためのインスタンスの構成	5-21
ロギングの構成	5-21
JVM の起動および停止パラメータの構成	5-21
SIP ポートの構成	5-22
適切な IP アドレスでリスニングするための Sip サブレット・コンテナの構成	5-23
xcap 構成ディレクトリの構成	5-23
Aggregation Proxy の構成	5-24
springbeans.xml の構成	5-24
構成設定の確認	5-24
接続プールの作成	5-25
XDMS のデプロイおよび構成	5-26
PublicXCAPRootUrl および PublicContentServerRootUrl の変更	5-26
UserAgentFactoryService ポートの更新	5-27
JGroups の有効化	5-27
データベースの構成	5-28
sash の構成	5-29
確認	5-29

User Dispatcher の構成	5-29
インストールのチューニング	5-30
オーバーロード・ポリシーの更新	5-30
WebCenter インスタンスの無効化	5-30
ホーム・インスタンスの無効化	5-30
ASG インスタンスの無効化	5-30
ロード・バランサの構成	5-31
新しいプールの作成	5-31
新しい仮想サーバーの作成	5-32

6 インストール後の作業

インストール後の管理タスクの実行	6-2
データベースのチューニング	6-2

索引

はじめに

「はじめに」には、次の項が含まれています。

- [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

このガイドは、Oracle Communication and Mobility Server をインストールして使用する、システム管理者と開発者を対象としています。

ドキュメントのアクセシビリティについて

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

ドキュメント内のサンプル・コードのアクセシビリティについて

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

外部 Web サイトのドキュメントのアクセシビリティについて

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

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。

関連ドキュメント

詳細は、Oracle Communication and Mobility Server、Oracle Containers for J2EE、Oracle Application Server および Oracle TimesTen In-Memory Database 製品セットの次のマニュアルを参照してください。

- 『Oracle Communication and Mobility Server 管理者ガイド』
- 『Oracle Containers for J2EE 構成および管理ガイド』
- 『Oracle Containers for J2EE デプロイメント・ガイド』
- Oracle Application Server のインストール・ガイド
- 『Oracle Application Server 管理者ガイド』
- 『Oracle Communication and Mobility Server リリース・ノート』

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。

製品およびインストールの概要

この章では、Oracle Communication and Mobility Server (OCMS) および推奨されるトポロジについて説明します。次の項目が含まれています。

- 製品の概要
- インストールの前提条件
- インストール・モード
- インストールされるコンポーネント

製品の概要

OCMS は、SIP 対応ブロードバンド・ネットワークまたは携帯電話ネットワーク経由でメッセージを送受信する必要があるアプリケーション用のキャリア・グレード SIP (Session Initiation Protocol) および J2EE 実行プラットフォームです。このプラットフォームを SIP アプリケーション・サーバーと呼びます。

このようなアプリケーションの例を次に示します。

- IP 電話
- テレビ電話
- インスタント・メッセージング
- 短縮ダイヤル・サービス
- 転送サービス
- サード・パーティ・コール・コントロール
- 緊急通報サービス

OCMS は、ブロードバンドまたは付加価値サービスを提供する 3GPP IP Multimedia System (IMS) ネットワークのどちらにもデプロイできます。

今回のリリースの新機能

今回のリリースの Oracle Communication and Mobility Server には、拡張機能および新機能が含まれます。これらの拡張機能および新機能の詳細は、『Oracle Communication and Mobility Server 管理者ガイド』および http://www.oracle.com/technology/products/ocms/otn_front.htm の Oracle Technology Network を参照してください。

インストールの前提条件

ここでは、OCMS のインストールおよび特定の OCMS 機能を使用する際の前提条件について説明します。

システム要件

OCMS をインストールするシステムは、次の要件を満たしている必要があります。

- Red Hat Enterprise Linux AS 4 (アップデート 5 以上)、AS 5 (アップデート 1 以上)。
- 開発目的の場合のみ推奨 : Service Pack 3 以上を適用した Microsoft Windows 2000、Service Pack 1 以上を適用した Microsoft Windows Server 2003 (32 ビット)、または Service Pack 2 を適用した Microsoft Windows XP Professional。
- Java 2 Platform, Enterprise Edition Software Development Kit (JDK) 5.0, (JDK バージョン 1.5)。Oracle Application Server インストール・モードでは、Oracle Application Server から JDK 1.5.0 を使用します。スタンドアロン開発者モードでは、OCMS のインストールに JDK 1.5.0 がバンドルされます。
- Subscriber Data Services および Presence のための Oracle 10g Database (10.2.0.3 以上) または Oracle Database 11g (11.1.0.6 以上)。
- Subscriber Data Services の詳細は、「インストールされるコンポーネント」を参照してください。
- LAN 接続および IP アドレスとホスト名。

注意： リストのオペレーティング・システムは、Oracle Communication and Mobility Server で保証およびサポートされています。他のオペレーティング・システムは保証されていません。

ポート要件

OCMS では、SIP コンテナ、Presence サーバーおよび Edge Proxy（インストールされている場合）の TCP および UDP 通信にポートを使用する必要があります。SIP コンテナおよび Edge Proxy のポートにはデフォルトのポート番号が指定されます。または、ユーザーが指定するポート番号を使用することもできます。Presence ポートは、インストール中は設定できませんが、MBean ブラウザで設定できます。インストール後にポート番号を再設定する方法は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

インストール時に、選択したポートが確認されます。ポートが別のアプリケーションで使用されている場合は、インストールに失敗し、エラー・メッセージが表示されます。

表 1-1 に、Edge Proxy アプリケーションをインストールする場合に OCMS が使用するデフォルト・ポートを示します。

表 1-1 デフォルト・ポート : Edge Proxy をインストールした場合

SIP ポート	Edge Proxy ポート	Presence ポート
5070	5060	5081

表 1-2 に、Edge Proxy アプリケーションをインストールしない場合に OCMS が使用するデフォルト・ポートを示します。

表 1-2 デフォルト・ポート : Edge Proxy をインストールしない場合

SIP ポート	Edge Proxy ポート	Presence ポート
5060	該当なし	5070

ポートが使用中かどうかの確認

Windows でポートが使用されているかどうかは、次の netstat コマンドを実行して確認できます。

```
C:¥> netstat -an | find "portnum"
```

ポート番号の前後には二重引用符が必要です。

Linux でポートが使用されているかどうかは、次の netstat コマンドを実行して確認できます。

```
netstat -a | grep <portnum>
```

インストール・モード

Oracle Communication and Mobility Server は、次のインストール・モードでインストールできます。

- [スタンドアロン開発者モードのインストール](#)
- [既存の Oracle Application Server へのインストール](#)

表 1-3 は、各インストール・モードの要件の一覧です。

表 1-3 インストール・モードの要件

要件	Oracle Application Server モード	スタンドアロン開発者モード
Oracle Containers for J2EE (OC4J)	インストール前に必要。	不要。インストールに含まれる。
Java Development Kit (JDK) 1.5	Oracle Application Server により提供。	含まれる。

スタンドアロン開発者モードのインストール

スタンドアロン開発者モードでは、開発者がアプリケーション・サーバーなしの最小インストール環境で、SIP アプリケーションの開発とテストを行えます。このインストール・モードでは、Oracle Containers for J2EE (OC4J) がインストールされます。

スタンドアロン開発者モードは、管理機能が制限されるため、通常は本番デプロイにはお薦めしません。

既存の Oracle Application Server へのインストール

推奨インストール・モードでは、OCMS を Oracle Application Server リリース 10.1.3.4 以上の環境にインストールします。このインストール・モードでは、OCMS は、Oracle 高可用性 (HA)、クラスタリング、レプリケーションなどの OC4J 機能を使用できます。

Oracle Communication and Mobility Server は、OC4J コンテナとともにデプロイされます。このコンテナは、Oracle 10g Enterprise Manager Application Server Control コンソールを使用して管理します。Application Server Control の機能を使用して、アプリケーションの起動、停止、再起動、デプロイ、アンデプロイおよび再デプロイを行えます。

さらに、Application Server Control MBean ブラウザでは、OCMS コンポーネントの構成と管理も行えます。OCMS MBean (マネージド Bean) の属性を構成すると、DNS (ドメイン・ネーム・システム) の構成、Presence の構成と管理、OCMS SIP サーバー本体の基本構成 (ポート、IP、ホスト・アドレス) などの管理タスクを実行できます。

Oracle Application Server 環境での OCMS の管理および構成の詳細は、『Oracle Communication and Mobility Server 管理者ガイド』の「SIP サーバーの管理」を参照してください。

Oracle Application Server が提供する機能の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

インストールされるコンポーネント

ユーザーがインストール時に選択解除しないかぎり、次のアプリケーションをインストールするようにデフォルトで選択されます。

- **SIP コンテナ:** SIP サブレット・コンテナは J2EE アプリケーション・サーバーを拡張し、SIP アプリケーションにセキュリティ、同時実行性、ライフサイクル管理、トランザクション、デプロイおよびその他のサービスを含む実行時環境を提供します。JSR116 準拠の SIP サブレット・コンテナは、着信 SIP トラフィックをリスニングするためのトランスポート・プロトコル、IP アドレスおよびポート番号を組み合わせた、SIP リクエストおよびレスポンスの送受信のネットワーク・サービスを提供します。OCMS SIP サブレット・コンテナは、OC4J で実行される Oracle Application Server の既存のインスタンス上にインストールできます。専用の OC4J のスタンドアロン・インスタンス上で実行することもできます。標準的な OCMS SIP サブレット・コンテナは、OC4J を J2EE コンテナとして使用する Oracle Application Server インスタンスと、サーバーを監視する Oracle Process Manager and Notification Server (OPMN) とで構成されます。OCMS は、現在この構成でのみ高可用性デプロイをサポートします。
- **Subscriber Data Services:** Subscriber Data Services は、アプリケーションで OCMS が提供する認証とセキュリティが必要な場合にインストールされます。Subscriber Data Services では Oracle Database が必要です。
- **Proxy Registrar:** Proxy Registrar は、SIP プロキシ・サーバーとレジストラを組み合わせた機能を提供します。サブスクリイバの登録、サブスクリイバの場所の検索、登録先へのリクエストのプロキシなどのタスクを行います。

Proxy Registrar は、Location Service と、Proxy Registrar の登録サブコンポーネントを機能させるために Subscriber Data Services に依存します。

- **Presence:** Presence アプリケーションを使用すると、サービス・プロバイダはエンド・ユーザーにプレゼンス・サービスを提供したり、プレゼンス情報に基づいて他のサービスを提供できます（プレゼンス情報に基づくインテリジェント・コール・ルーティングなど）。

Presence アプリケーションは、SIP コンテナに依存します。

- **Aggregation Proxy:** Aggregation Proxy は、任意の XCAP リクエストと Web Services コールを認証します。その後、XCAP リクエストと Web Services コールは、それぞれのサーバーにプロキシされます。詳細は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

Aggregation Proxy は、SIP コンテナと Subscriber Data Services に依存します。

- **Application Router:** Application Router は、着信 SIP リクエストを正しいアプリケーションにルーティングする SIP アプリケーションです。アプリケーションの作成や独自の新しいアプリケーションのデプロイに必須のコンポーネントです。

Application Router は、処理する各 SIP リクエストにルート・ヘッダーを付けて、リクエストをルーティングします。1つのリクエストに、それぞれが異なる宛先 URI を表す多数のルート・ヘッダーを使用できます。SIP リクエストは、一連の宛先 URI 経由で送信されるか、最初の宛先に達したときに新しい URI にプロキシされます。

Application Router アプリケーションは、SIP コンテナと Subscriber Data Services に依存します。

次のアプリケーションはデフォルトではインストールされません。インストールするには対象アプリケーションのリストからユーザーが選択する必要があります。

- **Parlay X Presence Web Services:** Presence Web Services は、Parlay X Presence Web Services をサポートします。これは、Parlay X 2.1 Web Services インタフェースの実装です。

Presence Web Service アプリケーションは、SIP コンテナに依存します。

- **STUN サーバー:** STUN サーバーは、STUN (Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators) サーバー・プロトコルを実装します。これは、NAT トラバーサル・メカニズムとして機能し、クライアントがクライアントを表す外部 IP アドレスとポートを検出できるようにします。

STUN サーバーは、SIP コンテナに依存します。

- **Edge Proxy:** Edge Proxy は、特定の OCMS SIP Server インスタンスへの SIP リクエストをプロキシすることで、リクエストのルーティングの SIP 分散を提供します。Edge Proxy は、個々のクライアントと SIP サーバー間に論理的な経路を形成し、特定のクライアント・セッションから送信された SIP トラフィックが常に同じサーバーで処理されるようにします。SIP クライアント数が増加した場合は、Edge Proxy サーバーを追加し、スケーラビリティの高い SIP クライアント処理ができます。

Edge Proxy は、Oracle Application Server モードによるインストールでのみ使用できます。

- **User Dispatcher:** User Dispatcher を使用すると、Presence アプリケーションのようなステートフル・アプリケーションを拡張できるようになります。User Dispatcher は、SIP および XCAP (HTTP 上) リクエストを一貫した基準でそれぞれの適切な宛先にディスパッチするプロキシです。

これらの機能の詳しい説明と構成情報は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

Oracle Communication and Mobility Server のインストール

この章では、Oracle Communication and Mobility Server (OCMS) のインストールを実行する方法について説明します。次の項目が含まれています。

- [高可用性環境での OCMS インスタンス・クラスタのインストール](#)
- [Oracle Communication and Mobility Server のインストール](#)
- [Oracle Communication and Mobility Server の起動と停止](#)
- [Edge Proxy の起動と停止](#)
- [削除](#)
- [インストールの問題のトラブルシューティング](#)

高可用性環境での OCMS インスタンス・クラスタのインストール

OCMS は、評価または開発用に単一ノードとしてインストールすることも、高可用性の本番トポロジ用に複数ノードにインストールすることもできます。高可用性の本番トポロジを実現するには、他の OCMS ノードとともに 1 つ以上の Edge Proxy ノードが必要です。Edge Proxy は、スケーラビリティと高可用性を提供します。Edge Proxy は、高度な本番レベルのトポロジに必要で、通常は SIP 分散を提供するために使用されます。Edge Proxy は、SIP を認識しないロード・バランサと OCMS クラスタ間で使用された場合、OCMS SIP アプリケーション・サーバー間の着信 SIP トラフィックを分散させます。

推奨されるデプロイ・トポロジと高可用性の構成の詳細は、『Oracle Communication and Mobility Server 管理者ガイド』の「デプロイ・トポロジ」と「高可用性の構成」の章を参照してください。

クラスタ環境での Edge Proxy を使用した OCMS の構成

管理者は、通常 Edge Proxy を別々のノードにインストールします。Edge Proxy を OCMS インストールの一環としてインストールするには、Oracle Application Server インストール・モードで実行する必要があります。Edge Proxy は、OPMN クラスタリングを通して複数の OCMS インスタンスを認識します。これには、各 OCMS インスタンスが一意的 Oracle ホームを参照するような、クラスタ化された Oracle Application Server 10.1.3.4 環境が必要です。

インストールのサイズ設定

OCMS インストールに推奨される Edge Proxy の数は、インストールのスケーラビリティおよび高可用性の要件と、SIP クライアントおよび OCMS インスタンスの数によって決まります。クラスタ環境で高可用性を保証するには、少なくとも 2 つの Edge Proxy を使用することをお勧めします。SIP クライアントまたは OCMS インスタンス数の増加に応じて、Edge Proxy サーバーを追加できます。詳細は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

3 つ以上の OCMS インスタンスを含む OCMS の場合、Edge Proxy は通常個別にインストールされた OCMS インスタンスで、Edge Proxy アプリケーションのみで構成されます（つまり、1 つの OCMS インスタンスに対して Edge Proxy のみがインストールするように選択されます）。インスタンス数が 2 つ以下の OCMS の場合、Edge Proxy を OCMS インスタンスと同時にインストールできます。

複数インスタンス・インストールでのポートの割当て

OCMS の OCMS インスタンスと Edge Proxy のみのインスタンスの数を決定したら、各インスタンスに割り当てるポートを決定する必要があります。マシンが 1 台のインストールの場合、OCMS の各インスタンスには一意に定義されたポートが必要です。OCMS インストーラは、インストール時にポートが使用できることを確認します。例 2-1 は、1 台の Edge Proxy サーバーを使用した複数インスタンスの OCMS でポートを割り当てる方法を示しています。

例 2-1 1 台の Edge Proxy を使用した複数インスタンスの OCMS でのポートの割当て

OCMS instance 1: Custom installation consisting of Edge Proxy only
Edge Proxy Port: 5060 (default)

OCMS instance 2: Typical installation
SIP Port: 5080 (user-configured during installation)

OCMS instance 3: Typical installation
SIP Port: 5090 (user-configured during installation)

OCMS instance 4: Typical installation
SIP Port: 5100 (user-configured during installation)

Edge Proxy のインストール

OCMS に Edge Proxy をインストールするには、次の手順を実行します。

1. Oracle Universal Installer の「**使用可能な製品コンポーネント**」画面で、**Edge Proxy** コンポーネントを選択します。

注意：他のコンポーネントは選択せず Edge Proxy のみを選択する必要があります。

2. 「次へ」をクリックします。
3. OCMS のインストールを完了します。
4. **opmnctl status** コマンドを実行して、Edge Proxy のステータスを確認します。
インストール後、Edge Proxy は自動的に実行され、「Alive」のステータスを報告します。
5. Oracle Universal Installer を実行し、上の手順を繰り返して、追加の Edge Proxy を構成します。
6. 『Oracle Communication and Mobility Server 管理者ガイド』の次の手順を実行します。
 - 高可用性のための OCMS SIP コンテナの構成
 - 高可用性のための Edge Proxy ノードの構成
 - 高可用性 SIP サーブレット・アプリケーションの構成
7. このガイドの手順を使用して、Edge Proxy 以外の OCMS インスタンスをインストールします。

Edge Proxy と OCMS インスタンスすべてをインストールすると、Oracle Application Server のインストール中に「AS Control を起動」を選択した場合は、Oracle Application Server インスタンスから Enterprise Manager 経由で OCMS インストールのトポロジを表示できます。

Oracle Communication and Mobility Server のインストール

OCMS を Microsoft Windows または Linux オペレーティング・システムにインストールできません。

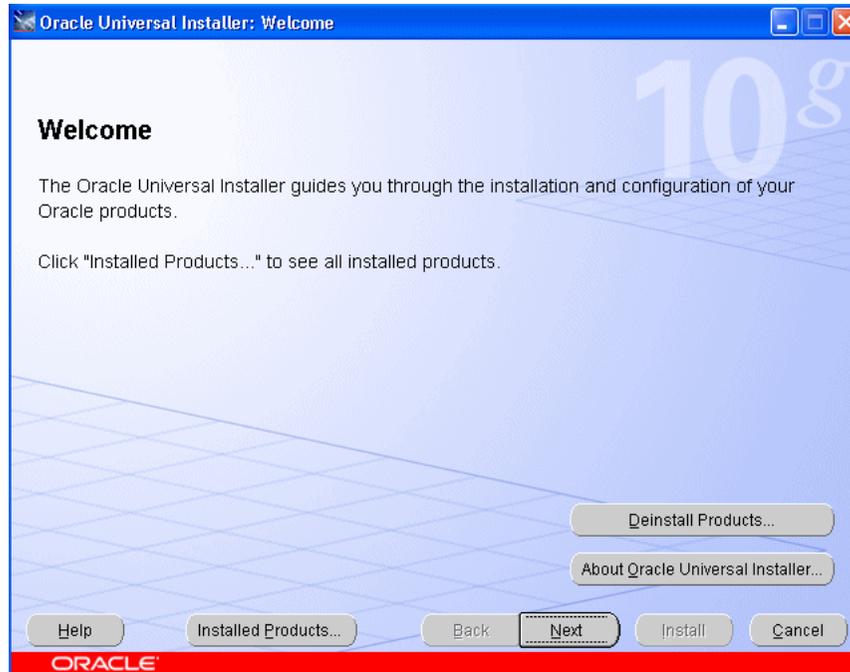
OCMS をインストールするには、次の手順を実行します。

1. 実行中の SIP クライアント・アプリケーションを終了します。
2. 使用しているオペレーティング・システムでインストール実行可能ファイルを実行します。
 - Microsoft Windows オペレーティング・システムの場合、**setup.exe** を実行します。
 - Linux オペレーティング・システムの場合、**runInstaller** を実行します。

Oracle Universal Installer が起動し、Java 2 Platform, Standard Edition Development Kit (JDK) 1.5.0.0.6 を自動的にインストールします。

「ようこそ」画面が表示されます。

図 2-1 「ようこそ」画面



注意： Oracle Universal Installer の各画面には「ヘルプ」ボタンがあります。画面の詳細を確認するには、「ヘルプ」を使用します。

3. 「次へ」をクリックして、続行します。「ファイルの場所の指定」画面が表示されます。
4. できればステージの場所を変更しないことをお勧めします。既存の Oracle Application Server にインストールするときは、インストール場所として既存の ORACLE_HOME を選択してください。どのフィールドも空白のままにしないでください。「次へ」をクリックして、続行します。

インストール・タイプの選択

この画面では、インストール・タイプを選択できます。

1. 実行するインストール・タイプを選択します。
 - スタンドアロン開発者モードでインストール：管理機能が制限されているため、スタンドアロン開発者モードは開発ユーザーにのみお勧めします。このオプションでは、Oracle Containers for J2EE (OC4J) ソフトウェアのリリース 10.1.3.4 がインストールされ、アプリケーション・サーバーは必要ありません。
 - 既存の Oracle Application Server にインストール（リリース 10.1.3.4 はサポートされています）：本番環境のデプロイにはこのインストール・モードの選択をお勧めします。Oracle Universal Installer により、指定されたアプリケーション・サーバー・ホームに Oracle Application Server が含まれているかどうかを確認されます。

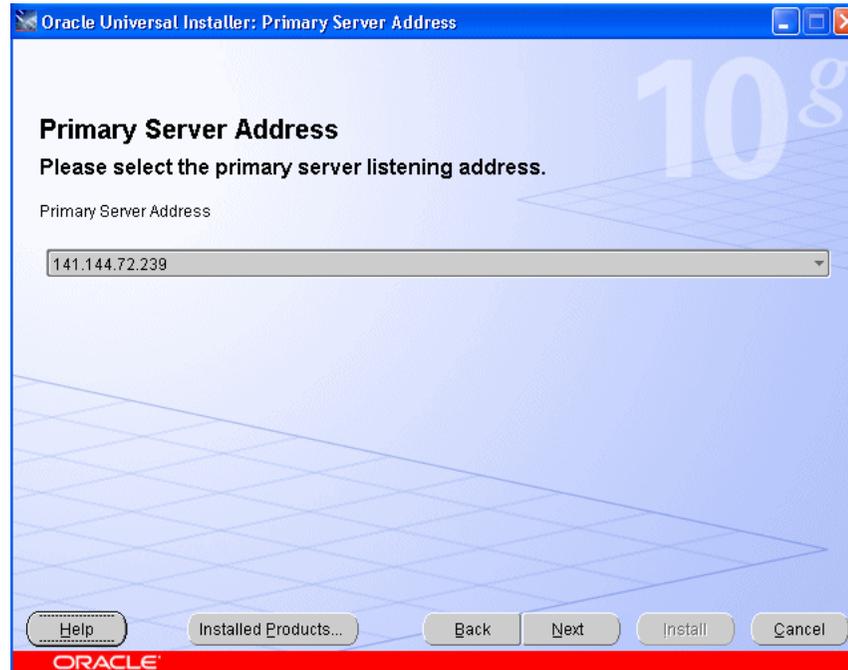
注意： 既存の Oracle Application Server にインストールする場合、Oracle Universal Installer は OC4J インスタンスに自動的に ocms という名前を付けます。OCMS をインストールする前に、ORACLE_HOME に ocms という名前の OC4J インスタンスが他にないことを確認します。

2. 「次へ」をクリックして、続行します。「使用可能な製品コンポーネント」画面が表示されます。

プライマリ・サーバー・アドレスの指定

プライマリ・サーバー・アドレスは、クライアントが OCMS と通信するために使用するアドレスです。

図 2-2 プライマリ・サーバー・アドレス画面



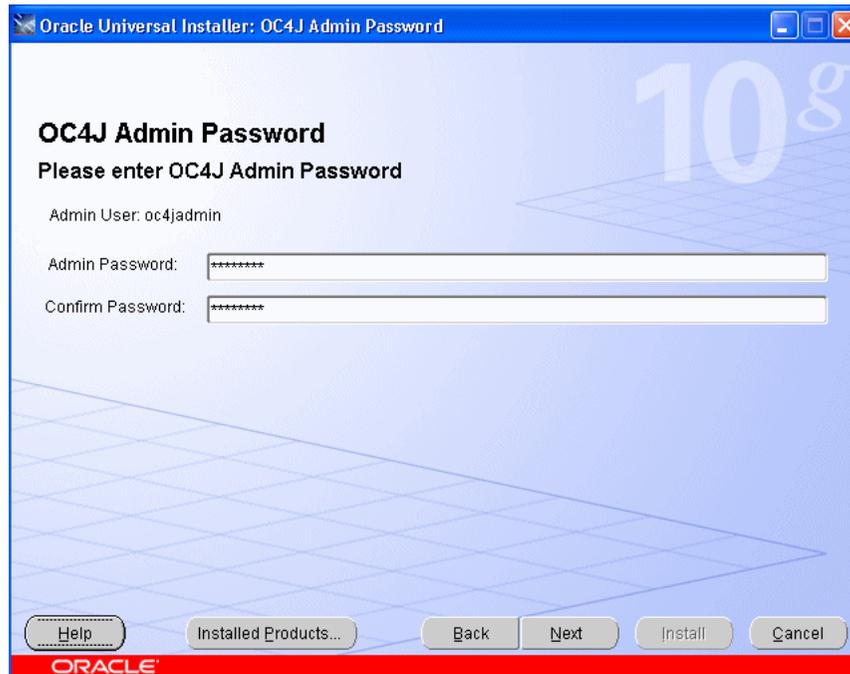
1. ドロップダウン・メニューからプライマリ・サーバーのリスニング・アドレスを選択します。
2. 「次へ」をクリックして、進みます。OC4J Admin パスワード画面が表示されます。

OC4J Admin パスワードの指定

スタンドアロン・モードでインストールしている場合は、OCMS インスタンスを停止するときにパスワードによる検証が行われます。

Oracle Application Server モードでインストールしている場合は、管理者パスワードを入力すると、Oracle Application Server によってこれらの資格証明の検証がすぐに行われます。

図 2-3 OC4J Admin パスワード画面



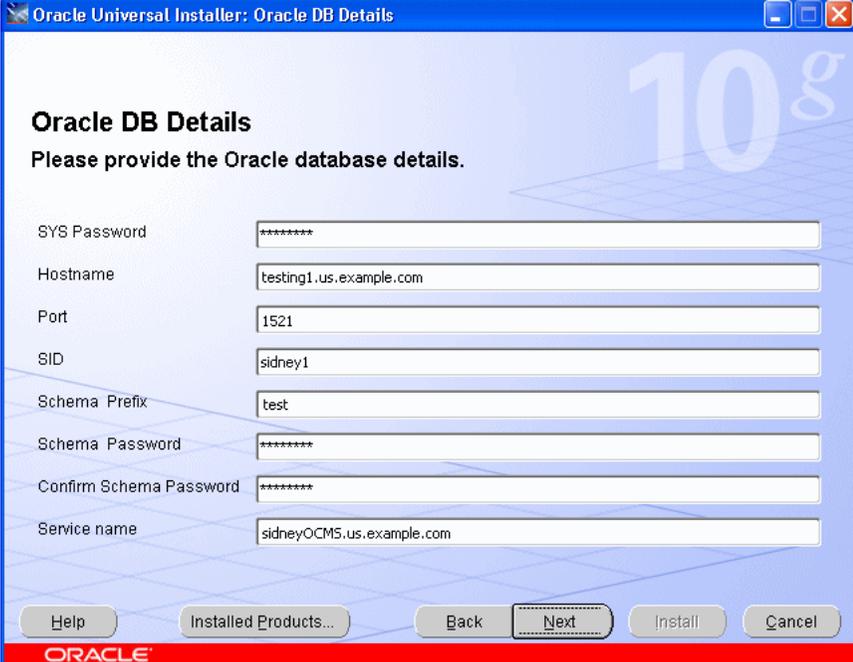
1. 管理者のユーザー名を入力します。
2. 管理者のパスワードを入力します。
3. 「次へ」をクリックして、進みます。Oracle DB の詳細画面が表示されます。

注意： これまでの初期画面の後では、インストールのための画面が表示される順序は、選択した製品およびコンポーネントに応じて異なります。次に説明する画面の順序は、インストールが同じ順序で行われるという意味ではありません。

Oracle DB の詳細の指定

この画面を使用して、Oracle データベースの詳細を入力します。

図 2-4



Oracle Universal Installer: Oracle DB Details

Oracle DB Details
Please provide the Oracle database details.

SYS Password: *****

Hostname: testing1.us.example.com

Port: 1521

SID: sidney1

Schema Prefix: test

Schema Password: *****

Confirm Schema Password: *****

Service name: sidneyOCM5.us.example.com

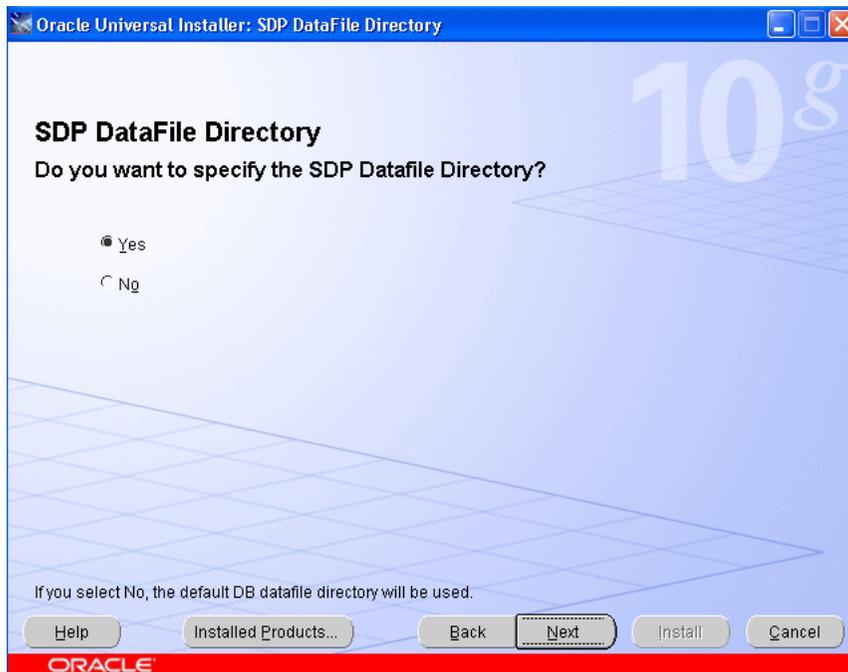
Buttons: Help, Installed Products..., Back, Next, Install, Cancel

1. 次のフィールドに値を入力します（データベースの資格証明はデータベースに接続するときに検証されます）。
 - SYS パスワード
 - ホスト名
 - ポート
 - SID
 - スキーマ接頭辞
 - スキーマ・パスワード（および確認用の同じパスワード）
 - サービス名
2. 「次へ」をクリックして、進みます。

SDP データファイル・ディレクトリ

この画面を使用して、SDP データファイル・ディレクトリを指定するかどうかを指定します。

図 2-5 SDP データファイル・ディレクトリ画面



- ディレクトリの指定を選択すると、次の画面（SDP データファイルの場所）でディレクトリの情報を入力できます。

注意： データベースのデータファイル・ディレクトリが存在しており書き込み可能であることを確認する必要があります。そうでない場合は構成ツールが失敗します。

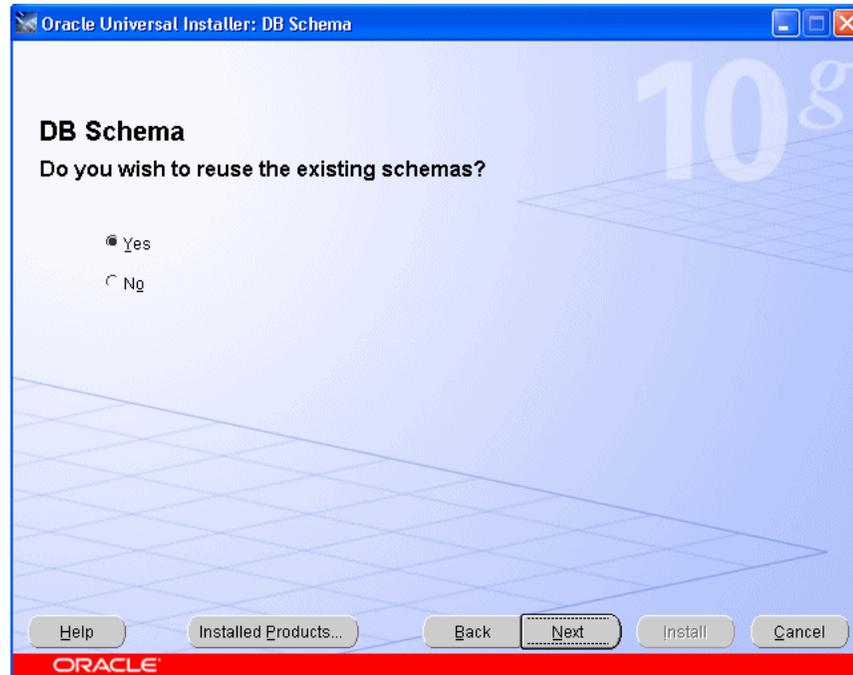
- 場所を指定しない場合は、デフォルトの DB データファイル・ディレクトリが使用されます。

「次へ」をクリックして、進みます。

DB スキーマ

既存スキーマの再利用に対して「はい」または「いいえ」を選択し、「次へ」をクリックして進みます。前に STUN サーバーのインストールを選択した場合は、STUN サーバー構成画面が表示されます。

図 2-6 DB スキーマ画面



STUN サーバー構成の指定

STUN サーバーのインストールを選択した場合、プライマリおよびセカンダリの STUN サーバーのホストとポートを入力するよう求められます。プライマリとセカンダリ両方の STUN サーバーを構成する必要があります。

STUN サーバーのホストとポートを指定するには、次のようにします。

1. プライマリ・ホスト・アドレス・フィールドにプライマリ STUN サーバーのホスト名または IP アドレスを入力します。
2. プライマリ・ポートを入力します。
3. セカンダリ・ホスト・アドレス・フィールドにセカンダリ STUN サーバーのホスト名または IP アドレスを入力します。
4. セカンダリ・ポートを入力します。
5. 「次へ」をクリックして、続行します。

テスト・ユーザーの構成

OCMS のインストール時に、複数の事前定義済テスト・ユーザーを作成できます。テスト・ユーザーを作成するには、「はい」をクリックしてから、「OK」をクリックして進みます。テスト・ユーザーの詳細画面が表示されます。

図 2-7 テスト・ユーザー画面



テスト・ユーザーの詳細

この画面を使用して、テスト・ユーザーの詳細を入力します。

図 2-8 テスト・ユーザーの詳細画面

1. テスト・ユーザーの数には、1～12 から選択します。
2. ユーザー名の接頭辞に、テスト・ユーザーの接頭辞を入力します。

デフォルトの接頭辞は `test.user` です。各テスト・ユーザーのユーザー名には、この文字列の接頭辞が付きます。

たとえば、テスト・ユーザーの数で3を選択し、ユーザー名の接頭辞に `test.user` と入力した場合、次のテスト・ユーザーが作成されます。

```
test.user1@sip-domain
test.user2@sip-domain
test.user3@sip-domain
```

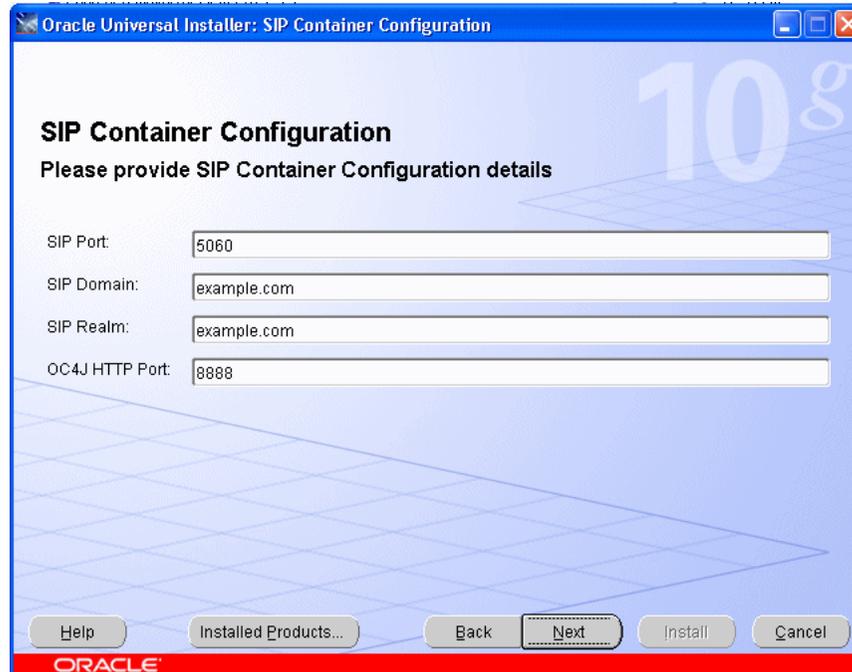
警告： このリリースの OCMS では小文字のユーザー名がサポートされていません。ユーザー名の接頭辞には、小文字のユーザー名のみを使用してください。

3. 「パスワード」に、テスト・ユーザーのパスワードを入力します。
テスト・ユーザーごとにこのパスワードが必要です。
4. 確認のため、「パスワード」に、テスト・ユーザーのパスワードをもう一度入力します。
5. 「次へ」をクリックします。

SIP コンテナの構成

次の SIP コンテナ情報を指定します。

図 2-9 SIP コンテナの構成画面



- **SIP ポート** : デフォルトの SIP ポートが表示されます。Edge Proxy をインストールする場合は 5070、Edge Proxy をインストールしない場合は 5060 です。
SIP ポートが使用中であることを伝えるメッセージが表示された場合は、すべてのクライアント・アプリケーションを終了し、「戻る」を選択して、このウィンドウに戻ります。
- **SIP ドメイン** : OCMS をインストールするマシンのドメインまたはホスト名を指定します。デフォルトは example.com です。
- **SIP レルム** : 認証に使用する SIP レルムを指定します。これは、OCMS をインストールするマシンのドメインまたはホスト名でもあります。デフォルトは example.com です。
- **OC4J HTTP ポート (スタンドアロン開発者モード専用)** : スタンドアロン開発者モードで OC4J を通じて OCMS を管理するための HTTP ポートを指定します。デフォルト・ポートは 8888 です。別のアプリケーションでこのポートを使用する場合は、異なるポートを選択できます。

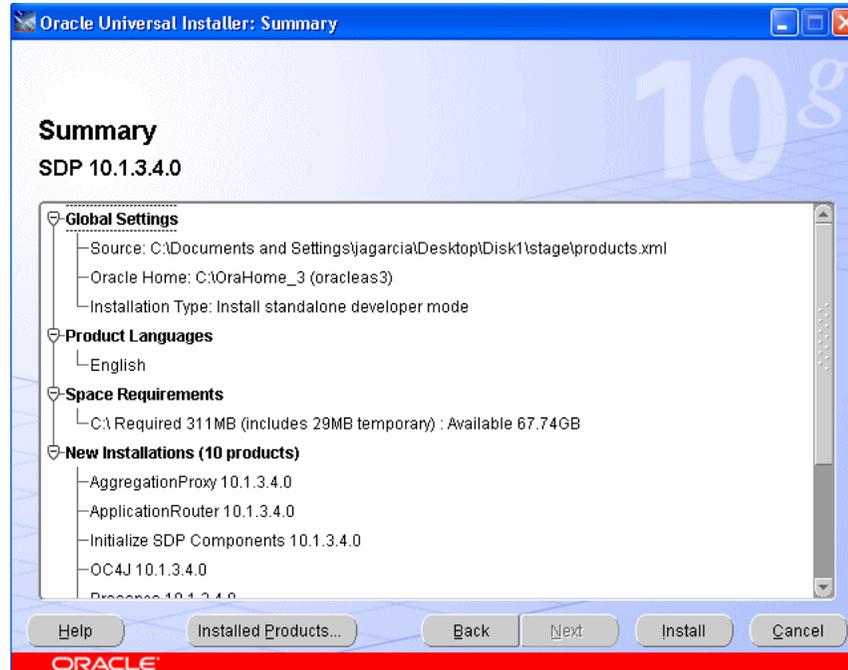
Oracle Application Server モードには、OC4J HTTP ポートを設定するオプションはありません。このモードでは、Oracle Application Server インストールのインストーラによって OC4J HTTP ポートが自動検出されるためです。HTTP_Server コンポーネントがインストールされていない Oracle Application Server へのインストールの場合は、7785 ~ 7799 の最初の空きポートがインストーラによって自動的に選択されます。

注意 : Oracle Universal Installer では、1 つの SIP ドメインと 1 つの SIP レルムのみが構成が可能です。追加の SIP ドメインと SIP レルムは、インストール後に SIP サブレット・コンテナの MBean の Domains および Realms 属性を使用して構成できます。1 つの SIP レルムで複数の SIP ドメインを使用できます。詳細は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

サマリー情報

OCMS インストールのサマリー画面には、インストール環境、インストールするコンポーネント、および構成した設定とポート番号が表示されます。

図 2-10 「サマリー」画面



1. 後で参照できるように、インストール構成セクションに表示されているポート番号などのインストールのサマリー情報を記録しておいてください。
この情報は、SIP クライアントの構成時に必要です。
2. すべてのインストールのサマリー情報が正しいことと、適切な OCMS アプリケーションがインストールされることを確認します。
3. 「インストール」をクリックして、OCMS コンポーネントのインストールを開始します。選択した製品およびコンポーネントがすべてインストールされると、「インストールの終了」画面が表示されます。

注意： インストール中にエラーが発生した場合は、
\$HOME/orainventory/logs のインストール・ログ・ファイルを調べてください。

4. 「終了」をクリックして、インストールを終了します。

Oracle Remote Method Invocation (RMI) ポートのリスト

Oracle Application Server Containers for J2EE (OC4J) では、独自の Remote Method Invocation (RMI) /Oracle RMI (ORMI) プロトコルを使用して、OC4J コンテナ間で EJB の相互起動ができます。Oracle RMI の詳細は、『Oracle Application Server Containers for J2EE サービス・ガイド』を参照してください。

特定の JMX アプリケーションが Oracle RMI ポートに接続する必要がある場合、RMI に割り当てられているポートを知る必要があります。RMI ポートは、OPMN (Oracle Process Management and Notification) によって動的に割り当てられます。

次のコマンドを使用すると、最新のポート割当てが一覧表示されます。

```
opmnctl status -l
```

次に例を示します (わかりやすくするため、一部の列は省略されています)。

図 2-11 opmnctl コマンドからの出力

```
Processes in Instance: <instancename>_<myhost>.com
-----
ias-component | process-type | pid | ... | ports
-----
OC4JGroup:ocms | OC4J:ocms | 7156 | ... | jms:12603,ajp:12503,rmis:12703,sip:5060,rmi:12403
OC4JGroup:ocms | OC4J:OC4J_WebCent | N/A | ... | N/A
OC4JGroup:ocms | OC4J:home | 7154 | ... | jms:12601,ajp:12501,rmis:12701,rmi:12401
HTTP_Server | HTTP_Server | 7153 | ... | https1:4443,http2:7200,http1:7777
ASG | ASG | N/A | ... | N/A
```

表の ports 列には、opmn によって選択されたポートが表示されています。次の例は、OC4J の OCMS インスタンス (OC4J:ocms) に対応しています。

```
jms:12603,ajp:12503,rmis:12703,sip:5060,rmi:12403
```

OCMS インストールの確認

次の手順を実行して、OCMS が正しくインストールされ、動作していることを確認します。

1. OCMS サーバーを起動し実行した状態で、次のコマンドを実行して、SIP コンテナがポート 5060 (デフォルトの SIP ポート) で SIP トラフィックをリスニングしていることを確認します。

```
netstat -a | grep 5060
```

5060 は、インストール中に選択したポート番号に置き換えてください。次の出力が表示されます。

```
tcp      0      0 <hostname>:5060  *:*          LISTEN
udp      0      0 <hostname>:5060  *:*
```

2. 次の Oracle Enterprise Manager Web ページに移動します。
`http://<ip_address>:<port_number>/em`
ポート番号には、次のいずれかの値を使用します。
 - スタンドアロン開発者モードのインストールの場合は、8888 を使用します。
 - HTTP サーバーを使用した Oracle AS インストールの場合は、HTTP サーバー・ポートを使用します。
 - HTTP サーバーを使用しない Oracle AS インストールの場合は、7785 を使用します。
 - HTTPS の場合は、Oracle AS インストールの HTTPS ポートを使用します。
3. OC4J 管理者のユーザー名とパスワードを入力します。
4. Oracle ホームで OCMS のインストール・ディレクトリが指定されていることを確認します。
5. サーバーのステータスが「稼働中」であることを確認します。
6. 「アプリケーション」タブを選択します。

7. インストールしたアプリケーションがデプロイされ、すべてが起動し、実行されていることを確認します。1 つ以上のアプリケーションが、`subscriberdataservices` の子アプリケーションとしてデプロイされている場合があります。
標準インストールの場合、`ocmsrouteladerear`、`proxyregistrar`、`subscriberdataservices` および `presence` が表示されます。
 8. ホーム・ページに戻り、「アプリケーション」タブを選択します。
 9. **JMX/ システム MBean ブラウザ**を選択します。左側で `SipContainer` を展開し、`SipServletContainer` をクリックして、MBean プロパティの値を確認します。
- OCMS の確認はこれで完了です。

Oracle Communication and Mobility Server の起動と停止

インストールが終了したら、Oracle Communication and Mobility Server を起動することができます。次の手順で説明するように、OCMS を手動で起動または停止できます。

Linux オペレーティング・システムで OCMS を起動するには、次のコマンドを入力します。

- `cd $ORACLE_HOME/sdp/bin`
`./startocms.sh`

Linux オペレーティング・システムで OCMS を停止するには、次のコマンドを入力します。

- `cd $ORACLE_HOME/sdp/bin`
`./stopocms.sh`

Windows オペレーティング・システムで OCMS を起動するには、次のコマンドを入力します。

- `<ocms_directory>%sdp%bin` ディレクトリから `startocms.bat` ファイルを実行します。

Windows オペレーティング・システムで OCMS を停止するには、次のコマンドを入力します。

- `<ocms_directory>%sdp%bin` ディレクトリから `stopocms.bat` ファイルを実行します。

既存の Application Server へのインストールの場合は、次のコマンドを実行して OCMS を起動することもできます。

```
$ORACLE_HOME/bin/opmnctl startproc process-type=ocms
```

既存の Application Server で OCMS を停止するには次のようにします。

```
$ORACLE_HOME/bin/opmnctl stopproc process-type=ocms
```

Edge Proxy の起動と停止

既存の Application Server では次のコマンドを使用して Edge Proxy を起動および停止できます。

```
$ORACLE_HOME/bin/opmnctl startproc process-type=EdgeProxy
```

既存の Application Server で Edge Proxy を停止するには次のようにします。

```
$ORACLE_HOME/bin/opmnctl stopproc process-type=ocms
```

削除

個々の製品またはコンポーネントを削除することはできません。製品を削除するためには、OCMS を削除する必要があります (Oracle Universal Installer を使用)。

インストールの問題のトラブルシューティング

ここでは、インストールに関連する問題を解決するための情報を示します (これらの問題は一部のユーザーで発生しています)。

ポートの競合

このガイドの「製品およびインストールの概要」の「[ポート要件](#)」で説明したとおり、OCMS では SIP、Presence および Edge Proxy の通信に特定のポートを使用する必要があります。

「[ポートが使用中かどうかの確認](#)」の指示に従って、ポートが使用中かどうかを確認します。

競合がある場合は、ポート番号を変更するか、そのポート番号を使用している他のサービスを停止することができます。

インストール中のネットワーク接続の切断

OCMS のインストール中は、アクティブなネットワーク接続が必要です。

インストール中にネットワーク接続が切断するといった不測の事態が発生した場合、インストールは失敗します。installActions<data_time>.log ファイル (Oracle インベントリ・ディレクトリ) を調べてこのイベントのログを確認します。

注意: 製品の問題の詳細は、このドキュメントに付属するリリース・ノートを参照してください。

OCMS のインストールおよび機能の確認

この章では、Oracle Communication and Mobility Server (OCMS) のインストールを確認する方法について説明します。次の項目が含まれています。

- [Oracle Communicator のインストールと OCMS インストールの確認](#)

Oracle Communicator のインストールと OCMS インストールの確認

アプリケーションをデプロイする前に、2人のユーザーをプロビジョニングし、SIP クライアントの Oracle Communicator をインストールして、接続テストを行います。Oracle Communicator クライアントは、デプロイメント・ディスクリプタで構成された接続先 IP アドレスとポートをリスニングします。

Oracle Communicator を使用して、次のことをテストできます。

- **Presence** サーバーとの相互作用: ユーザーのプレゼンスをサブスクライブして公開し、イベント通知を受け取ります。たとえば、ユーザーがオンラインになったことを示すポップアップを監視します。
- **音声通信**: 一方のユーザーからもう一方のユーザーに電話をかけて、VoIP 機能を確認します。
- **インスタント・メッセージング**: 一方のユーザーからもう一方のユーザーにメッセージを送信して、インスタント・メッセージングの機能を確認します。

次のタスクを実行して、Oracle Communicator クライアントをセットアップし、OCMS をテストします。

- **サンプル・ユーザーのプロビジョニング**: 各サンプル・ユーザーは、Oracle Communicator クライアントのインスタンスへのログインに使用されます。
- **ログ・レベルの設定**: ログ・レベルを `info` に設定し、すべての SIP トラフィックをリアルタイムで表示できるようにします。
- **Oracle Communicator のインストールと構成**: Oracle Communicator クライアントのインスタンスを2つインストールし、主要な OCMS SIP サブレットの機能をテストします。各インスタンスは、別々のコンピュータにインストールし、実行する必要があります。
- **サブレットの登録の確認**: サブレットが正しく登録されていることを確認します。
- **Presence サーバーのテスト**: ユーザーのプレゼンスの公開とサブスクライブ、およびイベント通知の受信をテストします。
- **SIP のテスト・コール**: SIP ネットワーク経由で電話をかけます。
- **SIP から PSTN へのテスト・コール**: 公衆交換電話網 (PSTN) でテスト・ユーザーから電話器に電話をかけます。

サンプル・ユーザーのプロビジョニング

インストール中にテスト・ユーザーを作成した場合は、Oracle Communicator を使用して OCMS に接続する準備はできています。Oracle Communicator をインストールおよび構成して、インストール中に作成したテスト・ユーザーの1人としてサインインします。

テスト・ユーザーを作成していない場合、SIP サブレットをテストするには、ディレクトリ `cd $ORACLE_HOME/sdp/sash/sbin` に移動し、SASH を起動して、2人のユーザーをプロビジョニングする必要があります。

ユーザーのプロビジョニングの詳細は、『Oracle Communication and Mobility Server 管理者ガイド』の「ユーザーとアプリケーションのプロビジョニング」を参照してください。

警告: このリリースの OCMS では小文字のユーザー名がサポートされています。小文字のユーザー名のみを使用してください。

ログ・レベルの設定

ログ・レベルを info に設定して、すべてのトラフィックを記録します。サブレットを登録し、クライアントと通信する間、ログを監視します。詳細は、『Oracle Communication and Mobility Server 管理者ガイド』のコア・コンポーネントのログ・レベルの設定に関する項を参照してください。

この章で行うテストを監視できるように、ログ・ウィンドウを開いたままにしておいてください。

Oracle Communicator のインストールと構成

2 台のコンピュータを使用できる場合は、それぞれのコンピュータに Oracle Communicator クライアントのインスタンスをインストールし、構成できます。2 台のコンピュータを使用できない場合は、同じコンピュータで Oracle Communicator の 2 つのインスタンスを起動できません。その場合、まず OCMS を実行していることを確認し、2 つのアカウントをプロビジョニングしてから、同じコンピュータで Oracle Communicator の 2 つのインスタンスを起動します。

Oracle Communicator クライアントをインストールし、構成するには、次のようにします。

1. Oracle Communication and Mobility Server が動作していることを確認します。
2. インストール・ファイルを起動し、画面の指示に従います。
 - 初期画面で、「Next」をクリックします。
 - Oracle Communicator のインストール・ディレクトリを選択し、「Next」をクリックします。
 - インストールのプリファレンスを選択し、「Next」をクリックします。
 - 「Next」をクリックして、インストールを開始します。
 - インストールが完了したら、「Finish」をクリックします。
3. 「スタート」→「すべてのプログラム」→「Oracle」→「Oracle Communicator」を選択して Oracle Communicator クライアントを実行し、Audio Setup ウィザードを完了します。

AudioSetup ウィザードでは、コンピュータのオーディオ・ハードウェアを構成およびテストし、サウンド・デバイスを構成できます。次の構成を実行します。

- コンピュータにヘッドセットを接続し、「Next」をクリックします。
- テスト・パラグラフを声に出して読みます。ボリュームを調整するには、「Volume」ボタンをクリックし、必要に応じて変更します。

声が検出されたかどうかを示されます。

- 「Finish」をクリックして、オーディオの構成を完了します。

Oracle Communicator Create Account Wizard が表示され、新しいアカウントを作成するよう求められます。

4. 作成するアカウント名を入力し、「Next」をクリックします。このアカウント名には、Sash コマンドラインを使用してプロビジョニングしたものを使用します。
5. username@example.com という形式で、最初のユーザーの SIP アドレスを入力します。
6. ユーザーのフルネームを入力します。

これは、各メッセージとともに送信される名前です。

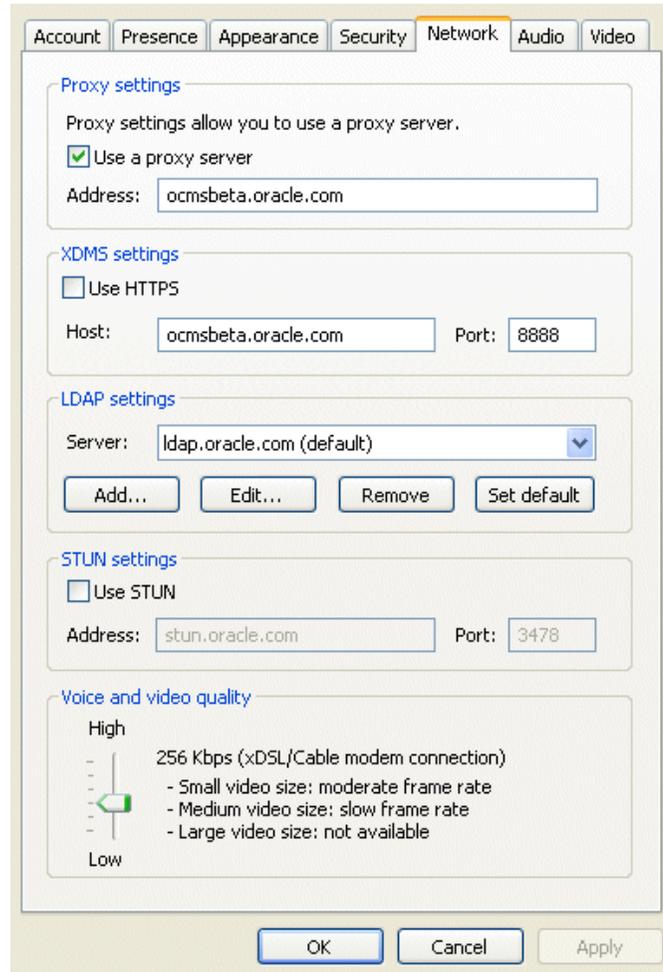
7. 「Finish」をクリックして、ウィザードを終了します。
Select Account ウィンドウが表示されます (図 3-1)。

図 3-1 Oracle Communicator アカウントの選択



8. ドロップダウン・リストから定義したアカウントを選択し、「OK」をクリックします。
9. ユーザー名とパスワードを入力します。
10. 設定のアイコンをクリックし、「Preferences」→「Network」の順に選択します。
11. 「Use a proxy server」を選択し、横のボックス (図 3-2) に SIP プロキシ・サーバーのアドレスを入力して、「OK」をクリックします。
12. Oracle Application Server が HTTPS サーバーを使用して構成されている場合は、「Use HTTPS」をチェックします。
13. 「XDMS Settings」セクションの下のデフォルト値 `xcap.<machine>.<domain>` から有効な値に変更します。
ポート番号には、次のいずれかの値を使用します。
 - スタンドアロン開発者モードのインストールの場合は、8888 を使用します。
 - HTTP サーバーを使用した Oracle AS インストールの場合は、HTTP サーバー・ポートを使用します。
 - HTTP サーバーを使用しない Oracle AS インストールの場合は、7785 を使用します。
 - HTTPS の場合は、Oracle AS インストールの HTTPS ポートを使用します。
 この構成で、Oracle Communicator クライアントが、互いにプレゼンス情報を表示し、電話をかけられるようになります。

図 3-2 プロキシ・サーバーの構成



14. タスク・バーのクライアント・アイコンを右クリックし、「Sign Out」を選択して、クライアントからサインアウトします。
15. 両方のクライアントで Oracle Communicator にログインし、クライアント間でメッセージを送受信できることを確認します。

クライアントで、緑のステータス LED とともに、「Connected to example.com」というメッセージが表示されると、正常に接続しています。

Oracle Communicator FileTransferServlet のインストール

FileTransferServlet を使用すると、Oracle Communicator ユーザー間でファイルを転送できます。このサーブレットの EAR (Enterprise Archive) ファイルは、OCMS と同じディスクに存在しますが、OCMS のインストール・プロセスの一部には含まれません。FileTransferServlet は、通常、OCMS と同じ OC4J コンテナには存在しないためです。そのため、FileTransferServlet は、個別にインストールする必要があります。つまり、Application Server Control コンソールで提供されるウィザードまたは `admin_client.jar` コマンドライン・ツールを使用して、OC4J コンテナにその EAR ファイルをデプロイします。

ヒント: 大量のデプロイが発生する場合は、パフォーマンスを確保するために FileTransferServlet を別のノードにデプロイすることをお勧めします。

FileTransferServlet は、Windows または Linux オペレーティング・システムが稼働する任意の OC4J ノードにデプロイできます。

注意: FileTransferServlet をホストする OC4J ノードでは、HTTP の GET および POST 操作をサポートしている必要があります。

ファイル転送をサポートするよう Oracle Communicator を構成する方法の詳細は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。OC4J にアプリケーションをデプロイする方法の詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

サーブレットの登録の確認

SIP クライアントと SIP サーブレット・コンテナ間で発生する最初の相互作用は登録です。SIP サーブレット・コンテナに初めて接続するとき、SIP クライアントはサーバーに REGISTER メッセージを送信します。SIP サーブレット・コンテナが正しく構成されている場合、すべてが正常であることを示す標準のメッセージとして 200 OK が返されます。

SIP サーブレット・コンテナが 401 Unauthorized というレスポンスを返した場合、Oracle Communicator にはユーザー名とパスワードの入力を求めるポップアップ・ボックスが表示されます。資格証明を入力すると、サーバーにもう一度 REGISTER メッセージが認証情報とともに送信されます。

このやりとりは、ログ・レベルを `info` に設定し（「[ログ・レベルの設定](#)」を参照）、SIP クライアントの実行時にログを見て監視できます。

サーブレットの登録を確認するには、次のようにします。

1. ログ・ウィンドウを開きます（ログ・レベルが設定されていることを確認してください）。
2. 「スタート」 → 「すべてのプログラム」 → 「Oracle」 → 「Oracle Communicator」を選択します。
3. プロンプトで、ユーザー名とパスワードを入力します（ユーザーのプロビジョニングについては、『Oracle Communication and Mobility Server 管理者ガイド』の「ユーザーとアプリケーションのプロビジョニング」を参照）。

4. SIP クライアントの別のインスタンスを実行し（可能な場合は別のコンピュータで）、異なるユーザー名とパスワードでログインします。

REGISTER メソッドと SIP サーブレット・コンテナのレスポンスを示すサンプル・ログを下に示します。

```
DEBUG [traffic] (NetworkEventWorker-11@) < REGISTER sip:example.com SIP/2.0
Call-ID: 77c49573-572e-46cc-893c-dc3b693eec90 CSeq: 1 REGISTER To:
<sip:alice@example.com> From: "alice@example.com"
<sip:alice@example.com>;tag=16feb568-cb96-4349-a825-a401e8189753 Max-Forwards: 70
User-Agent: OCMS-CallTron/4.5.7.1445 Contact:
<sip:alice@10.0.0.10:5062;transport=TCP>;q=1.00;agentid="4960c58b-286c-4426-bc37-b5
c6d91dcd7d";methods="INVITE, NOTIFY, MESSAGE, ACK, BYE, CANCEL";expires=600
Content-Length: 0 Via: SIP/2.0/TCP
10.0.0.10:5062;branch=z9hG4bK-1b9f16b3-10f9-44eb-8d50-dedd2cda5ed3.1;rport
```

```
DEBUG [traffic] (NetworkEventWorker-13@) > SIP/2.0 200 OK Via: SIP/2.0/TCP
10.0.0.10:1267;received=10.0.0.10;branch=z9hG4bK-eb1c7aba-7aa6-44fc-98e8-e3656713e0
9c.1;rport=1267 To: <sip:alice@example.com>;tag=6ed1a858-1083331acac--7ffa From:
"alice@example.com"
<sip:alice@example.com>;tag=89f9161e-762e-4750-9cea-970825f67848 Call-ID:
77c49573-572e-46cc-893c-dc3b693eec90 CSeq: 2 REGISTER Server:
OCMS-transactron/3.2.0-48 Content-Length: 0
```

Presence サーバーのテスト

Oracle Communicator では、Presence サーバーの主要機能をテストできます。SIP クライアントの 2 つのインスタンスにそれぞれ異なるユーザーとしてログインして実行することで、次のことを行えます。

- ユーザーのオンライン・ステータスの表示権限をリクエストして、ユーザーのプレゼンスをサブスクライブする（「[ユーザーのプレゼンスのサブスクライブ](#)」）。
- ユーザーのプレゼンスを公開する（「[ユーザーのプレゼンス公開のテスト](#)」）。
- ユーザーのログオン時間など、イベントの通知を受け取る（「[イベント通知の受信のテスト](#)」）。

ユーザーのプレゼンスのサブスクライブ

プレゼンスのサブスクライブとは、ユーザーのステータス（Available、Away など）や「Gone fishing」というメッセージなど、特定ユーザーのプレゼンス・データに対するユーザーのアクセス権を定義するドキュメントです。ユーザーは、特定ユーザーのプレゼンス・データのサブスクライブ権限をリクエストする必要があります。権限が付与されると、サブスクリプションが保存され、ドキュメントとして格納されます。

この項では、Oracle Communicator クライアントの 2 つのインスタンスを使用して、ユーザーのプレゼンスのサブスクリプションのリクエストと付与を行います。

ユーザーのプレゼンスをサブスクライブするには、次のようにします。

1. 各テスト・ユーザーが、Oracle Communicator クライアントの別々のインスタンス（可能な場合は異なるコンピュータ）にログインしていることを確認します。
2. 最初のテスト・ユーザーがログインした Oracle Communicator クライアントで、「**Contacts**」ボタンをクリックし、「**Add Contact**」を選択します（[図 3-3](#)）。

ヒント： Oracle Communicator への連絡先の追加方法の詳細は、Oracle Communicator のオンライン・ヘルプを参照してください。

図 3-3 Oracle Communicator の連絡先の追加

- 表示される「Contact Properties」タブの対応するボックスに、前に定義した 2 番目のテスト・ユーザーの SIP アドレスを入力します。
- 「Add Contact」をクリックします。
- 2 番目の Oracle Communicator クライアントのインスタンスを観察します。テスト・ユーザー 1 がテスト・ユーザー 2 のプレゼンスのサブスクライブ権限をリクエストしたことを伝えるメッセージが表示されます。
- テスト・ユーザー 2 の Oracle Communicator クライアントで、「OK」をクリックして、テスト・ユーザー 1 に対してプレゼンスのサブスクリプション権限を付与し、ユーザーをテスト・ユーザー 2 の連絡先リストに追加します。

ユーザーのプレゼンス公開のテスト

Oracle Communicator では、ユーザーは自分のプレゼンスを「Available」から「Away」などに変更できます。選択したプレゼンスのステータスがサーバーに保存されると、実際に公開され、公開者のプレゼンスをサブスクライブするユーザーがそのユーザーのプレゼンス・データを表示できるようになります。

ユーザーのプレゼンスを公開するには、次のようにします。

- 最初のテスト・ユーザーの Oracle Communicator クライアントで、クライアントの下部にある「Presence Status」ドロップダウン・リストをクリックします。
- ドロップダウン・リストから「Away」を選択します。
- 2 番目のテスト・ユーザーの Oracle Communicator クライアントを観察します。テスト・ユーザー 1 のステータスが「Away」として表示されます。

イベント通知の受信のテスト

テスト・ユーザーの1人がログオフしてからもう一度ログインしたときに、OCMS からイベント通知が送信されるかどうかをテストします。2番目のテスト・ユーザーがオンラインに戻ると、最初のテスト・ユーザーのコンピュータには、テスト・ユーザー2がオンラインであることを知らせるメッセージが表示されます。

イベントの通知を受け取るには、次のようにします。

1. テスト・ユーザー2をテスト・ユーザー1の連絡先リストに追加します（連絡先リストへのユーザーの追加方法は、「[ユーザーのプレゼンスのサブスクライブ](#)」を参照）。
2. 設定のアイコンをクリックし、「**Sign out**」を選択して、テスト・ユーザー2をログアウトさせます。
3. しばらく待ってから、ドロップダウン・リストからテスト・ユーザー2のアカウントを選択し、「**OK**」をクリックしてユーザー2をログインさせます。
4. テスト・ユーザー1がログインしている Oracle Communicator クライアントを実行中のコンピュータを観察します。

テスト・ユーザー2がオンラインになったことを知らせるメッセージが表示されます。

SIP のテスト・コール

一方のテスト・ユーザーからもう一方のテスト・ユーザーに電話して、OCMS の機能をテストします。スピーカーとマイク、またはマルチメディア・ヘッドセットを備えた別々のコンピュータで Oracle Communicator のインスタンスを実行します。

電話をかけるには、次のようにします。

1. 最初の Oracle Communicator クライアントで、テスト・ユーザー2を右クリックし、「**Call**」をクリックします。
2. 2番目の Oracle Communicator クライアントで「**Answer Call**」をクリックして、電話に出ます。
3. 話をして、相手の声が聞こえるかどうかを確認します。

SIP から PSTN へのテスト・コール

SIP から PSTN へのゲートウェイが、SIP ネットワークからの通話を公衆交換電話網（PSTN）にルーティングするようにインストールされている場合、Oracle Communicator から PSTN の電話番号にテスト・コールを行えます。

1. Oracle Communicator クライアントを起動します。
2. 「Quickcall」フィールドで、通話情報を次のように入力します。

```
sip:<phone number>@<IP address of SIP to PSTN gateway>
```

注意： DNS サーバーが構成されている場合は、IP アドレスのかわりにホスト名を使用できます。

Ethereal による OCMS ネットワーク・トラフィックの監視

Ethereal® は、OCMS ネットワーク・トラフィックの監視に使用できるネットワーク・プロトコル・アナライザです。具体的には、SIP、Presence および Edge Proxy トラフィックに使用されるポートでの TCP および UDP トラフィック（SIP クライアントは TCP または UDP を使用して SIP メッセージを送信）を監視できます。

次の手順を実行して、Ethereal で OCMS ネットワーク・トラフィックを監視します。

1. **Ethereal** の Web サイト <http://www.ethereal.com/> の指示に従って、**Ethereal** をダウンロードし、インストールします。

Ethereal のインストール時に、ライブ・ネットワーク・データを取り込むためのソフトウェア・パッケージ WinPcap もインストールする必要があります。

Ethereal をインストールするには、ハード・ディスクに約 55MB の空き容量が必要です。

2. 「**スタート**」、「**すべてのプログラム**」を選択し、**Ethereal** を実行します。
3. OCMS が使用するポートで TCP および UDP トラフィックを監視するように **Ethereal** を構成します。通常は、ポート 5060 ~ 5080 までを監視します。**Ethereal** ヘルプのキャプチャ・フィルタの説明を参照してください。
4. キャプチャ出力ファイルを構成します。
5. 「**Capture**」メニューから「**Start**」を選択して、ネットワーク・トラフィックのキャプチャを開始します。
6. キャプチャしたデータを表示します。

OCMS ネットワーク・トラフィックの問題のトラブルシューティングでは、主に **Ethereal** でキャプチャしたデータを分析します。アプリケーション・サーバー・ログは、データの二次ソースとして使用します。

バックエンド Oracle RAC データベースに対する Oracle Communication and Mobility Server のインストール

この章では、バックエンド Oracle RAC データベースで機能するように Oracle Communication and Mobility Server 10.1.3.4 をインストールして構成する方法を説明します。このドキュメントでは、Oracle Communication and Mobility Server で使用できるように、 n ノードの RAC データベースがインストールされて構成されていることを前提としています。このドキュメントでは、Oracle RAC データベースのインストールおよび構成方法については説明しません。内容は次のとおりです。

- [RAC 情報の収集](#)
- [RAC に対する Oracle Communication and Mobility Server のインストール](#)
- [RAC データベースを使用するための Oracle Communication and Mobility Server のインストール後の構成](#)

このドキュメントの説明は、Oracle Application Server の既存インスタンスへの Oracle Communication and Mobility Server 10.1.3.4 のインストールに適用されます。Oracle RAC データベースの構成は、Oracle Communication and Mobility Server のスタンドアロン（開発者）インストールではサポートされません。また、接続キャッシング、高速接続フェイルオーバー、および Oracle RAC データベースの高度な構成機能については、このドキュメントでは説明しません。

RAC 情報の収集

Oracle RAC データベースについて次の情報を取得する必要があります。

- データベース名
- RAC データベースのノード数
- 各ノードに関する次の情報
 - ホスト名
 - ポート番号
 - Oracle SID
 - ORACLE_HOME および ORACLE_CRS_HOME
 - sys ユーザーのパスワード

通常、本番環境のデプロイには、ホスト名の他に各 DB ノードの仮想 IP アドレスも含まれます。インストールと構成の際には、通常のホスト名ではなく仮想 IP アドレスに対応するホスト名を使用する必要があります。

RAC に対する Oracle Communication and Mobility Server のインストール

次の手順を実行して、Oracle Communication and Mobility Server をインストールし、バックエンド RAC を使用するように構成します。

- [RAC データベースでのサービスの作成](#)
- [Oracle Communication and Mobility Server のインストール](#)

RAC データベースでのサービスの作成

Oracle Communication and Mobility Server をインストールする前に、Oracle Communication and Mobility Server が RAC データベースに接続するために使用するデータベース・サービスをデータベースで作成する必要があります。このデータベース・サービスは、Oracle RAC 環境のすべてのノードで使用できるように作成します。

いずれかの RAC ノードで次のコマンドを実行します。

```
cd $ORA_CRS_HOME/bin
./srvctl add service -d <database name>
                        -s <service name>
                        -r <comma separated list of
                           ORACLE_SIDs for the RAC nodes>
```

作成したサービスのステータスを調べるために、次のコマンドを実行します。

```
./srvctl status database -d <database name> -s <service name>
```

次のようなレスポンスを受け取ります。

```
Service <service name> is not running.
```

新しいサービスが実行されていないことがわかります。次のコマンドを発行して新しいサービスを明示的に起動する必要があります。

```
./srvctl start service -d <database name> -s <service name>
```

次のレスポンスが返されます。

```
Service <service name> is running on instance(s) (inst. 1, inst. 2, etc.)
```

これで、Oracle Communication and Mobility Server のインストールを開始することができます。

Oracle Communication and Mobility Server のインストール

Oracle Universal Installer では、インストール時に RAC データベースに対する Oracle Communication and Mobility Server を構成することはできません。ただし、次の方法を使用すると、Oracle Communication and Mobility Server を RAC データベースに対して構成することができます。

- インストール時に RAC データベースのいずれかのノードを指定して、Oracle Communication and Mobility Server をインストールします。
- データソースをインストール後に変更して、Oracle Communication and Mobility Server を n ノードの RAC インスタンスに対して構成します。

既存の Oracle Application Server に対する Oracle Communication and Mobility Server のインストールを開始し（詳細は「[Oracle Communication and Mobility Server のインストール](#)」を参照）、データベースの構成画面まで進みます。この画面で、作成した RAC データベース・サービスに含まれるいずれかのデータベース・ノードに対応する情報を入力します。

たとえば、次の DB 構成画面（[図 4-1](#)）には最初の DB ノードに対応する情報が表示されます。

図 4-1 最初の DB ノードのデータベース情報

または、図 4-2 に示すようにデータベース・サービスの第 2（あるいはその他）のノードに対応する情報を指定することもできます。

図 4-2 後続の DB ノードのデータベース情報

「次へ」をクリックすると、SDP データファイルの場所を指定するための画面が表示されます。この画面で「いいえ」を選択して、バックエンド RAC データベースのデフォルトのデータファイルの場所（共有ネットワーク・ドライブ）に SDP データファイルを作成します。

「次へ」をクリックし、Oracle Communication and Mobility Server のインストールを完了します。

この時点で、Oracle Communication and Mobility Server スキーマはデータベース・サービスのどのノードからもアクセスできます。これは、次のように SQLPLUS を使用してデータベース・サービスの任意のノードに接続することで確認できます（この場合、マシンの名前は testing1.us.example.com）。

```
cd $ORACLE_HOME/bin
./sqlplus /nolog
SQL> conn userx_orasdpsds/myPassword1@testing1.us.example.com:1521/
sidneyOCMS.us.example.com
SQL> select * from tab;
```

表が表示されたことを確認して、SQLPLUS を終了します。ここで、このマシンから他のノード、たとえば testing2.us.example.com に接続します。

```
./sqlplus /nolog
SQL> conn userx_orasdpsds/myPassword1@testing2.us.example.com:1521/
sidneyOCMS.us.example.com
SQL> select * from tab;
```

この接続でも表が表示されることを確認します。

Oracle Communication and Mobility Server インストールが正常に終了した場合は、Oracle Communication and Mobility Server 環境が機能します。ただし、Oracle Communication and Mobility Server からのデータベース接続は、RAC データベースの構成済データベース・ノード 1 つのみです。RAC データベースへのデータベース接続を変更するにはインストール後の構成手順が必要です。

RAC データベースを使用するための Oracle Communication and Mobility Server のインストール後の構成

Oracle Communication and Mobility Server サーバー・マシンで次のコマンドを実行します。

```
cd $ORACLE_HOME/j2ee/ocms/config
```

この場所にある data-sources.xml を編集します。

Subscriber Data Services、Location Service および XDMS に対して次のような JDBC URL があります。

```
url="jdbc:oracle:thin:@//testing1.us.example.com:1521/sidneyOCMS.us.example.com"
```

これを次のように変更します。

```
url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=on) (ADDRESS=(PROTOCOL=tcp) (HOST=testing1.us.example.com) (PORT=1521)) (ADDRESS=(PROTOCOL=tcp) (HOST=testing2.us.example.com) (PORT=1521)) (ADDRESS=(PROTOCOL=tcp) (HOST=testing3.us.example.com) (PORT=1521)) (ADDRESS=(PROTOCOL=tcp) (HOST=testing4.us.example.com) (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=sidneyOCMS.us.example.com)))"
```

ホスト名、ポート番号、データベース・サービス名およびインスタンス数を、使用している RAC データベースの対応する値で置き換えます。

data-sources.xml を保存し、サーバーを再起動します。

Oracle Communication and Mobility Server のデータベース接続が RAC データベースのすべてのノードに対して有効であることを確認するには、次のコマンドを発行します。

```
netstat -a | grep 1521
```

Oracle Communication and Mobility Server サーバーからすべての RAC ノード（この例ではポート 1521 上）に対して、TCP 接続が確立されていることを確認します。

data-sources.xml ファイルを編集するかわりに、Oracle Enterprise Manager を使用して Oracle Communication and Mobility Server JDBC リソースを変更することもできます。

Presence の大規模デプロイ・インストール

この章では、Presence の大規模デプロイを行う方法について説明します。内容は次のとおりです。

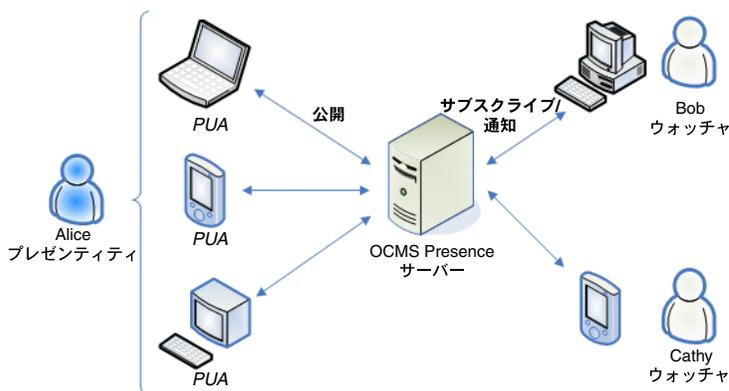
- [概要](#)
- [Presence マルチノード・トポロジ](#)
- [インストール](#)

概要

OCMS には独自の Presence サーバーがあり、クライアントがビジー、可能、会議中などのプレゼンス情報を公開するために使用できます。また、プレゼンス情報の変更時に通知を希望するとして登録したユーザーに、その情報を表示することができます。エンドクライアントの観点では、マルチ・ノードの Presence サーバーはシングル・ノードと同じです。エンド・ユーザーに対してサーバーはブラックボックスであり、違いに気付くことはありません。

ここでは、基本の Presence の機能や、OCMS 10.1.3.4 の新しい要素である User Dispatcher を使用して Presence サーバーを拡張する方法について説明します。Presence サーバーの一般的な機能の詳細は、『Oracle Communication and Mobility Server 管理者ガイド』を参照してください。

図 5-1 Presence サーバーの一般的な使用方法



定義

大規模なプレゼンス・サービスを適切にインストールして管理するには、きわめて多くの概念を理解する必要があります。ここでは、主要な概念について定義して説明します。

表 5-1 Presence の主要用語

用語	定義
Presence クラスタ	Y 個のロード・バランサが対応する X 個の Presence ノードで構成されるクラスタ。詳細は、「 Presence クラスタ 」を参照してください。
Presence ノード	X 個の Presence サーバー・インスタンス (PS) に対応する 1 つの User Dispatcher で構成される物理マシン (ノード)。
Presence サーバー (PS)	Presence イベントパッケージに対するサブスクリプションと公開のリクエストおよび <code>presence.wininfo</code> に対するサブスクリプションを処理する実際の Presence サーバー (インスタンス)。Presence サーバーとプレゼンス・サービスを混同しないでください。
プレゼンス・サービス	プレゼンス・サービスという語は RFC 2778 で定義されており、プレゼンスに関連するすべてのトラフィックを処理するサービスを非常に広義に意味します。これには、ウォッチャ情報のトラフィック、プレゼンティティをサブスクリプションするウォッチャ許可を決定するためのトラフィック、実際のプレゼンス更新などが含まれます。
XDM	XML ドキュメント管理
XDMC	XDM クライアント。XDM ネットワークにアクセスする任意のクライアント。Presence サーバー・インスタンスは XDM クラスタにアクセスするため、XDM クライアントとしても機能します。
XDM クラスタ	Y 個のロード・バランサが対応する X 個の XDM ノードで構成されるクラスタ。詳細は、「 XDM クラスタ 」を参照してください。

表 5-1 Presence の主要用語（続き）

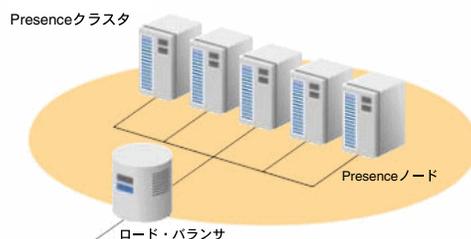
用語	定義
XDM ノード	1つの Aggregation Proxy、1つの User Dispatcher、X 個の XDM サーバー (XDMS) で構成される物理マシン (ノード)。
XDM サーバー (XDMS)	XML ドキュメントを格納する実際の XDMS インスタンス。XCAP を介してそれらのドキュメントの間合せと処理を行うことができます。XDMS を使用すると、SIP SUBSCRIBE/NOTIFY を使用した、ドキュメントにおける変更のサブスクライブも可能になります。XDMS インスタンスは、Presence サーバー・インスタンスが Presence ノードで行うのと同じ役割を XDM ノードで果します。

Presence クラスタ

Presence クラスタは、図 5-2 に示すように 1 つ以上のロード・バランサが対応する Presence ノードのセットです。Presence クラスタは、該当する場合に NOTIFY:s を送信するために、Presence イベントパッケージに対するサブスクライブと公開のリクエストを受信して処理します。また、Presence クラスタは、presence.wininfo イベントパッケージに対するサブスクライブ・リクエストも受け取って処理します。

Presence クラスタは、XDM クラスタとやりとりして、処理を完了するために必要な情報を取得します。XDM クラスタに問い合わせる情報には、ユーザーのプレゼンス・ルールや pidf-manipulation ドキュメント（つまりユーザーのハードステート）が含まれます。

図 5-2 Presence クラスタ

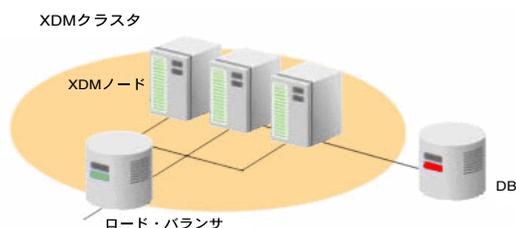


XDM クラスタ

XDM クラスタは、図 5-3 に示すように 1 つ以上のロード・バランサが対応する XDM ノードのセットです。XDM クラスタは、すべての XDM 関連トラフィック（つまり、XCAP トラフィックと ua-profile イベントパッケージへの SIP サブスクライブ・トラフィック）を処理します。つまり、XML ドキュメントの操作に必要なすべての処理を行います。XDM クラスタは、XML ドキュメントを格納するためにデータベースを使用しますが、データベース（およびそのクラスタ）は XDM クラスタには含まれません。

XDM クラスタはすべての XML ドキュメントを処理するため、各ノードは共有 XDMS と PS XDMS の両方になります。

図 5-3 XDM クラスタ



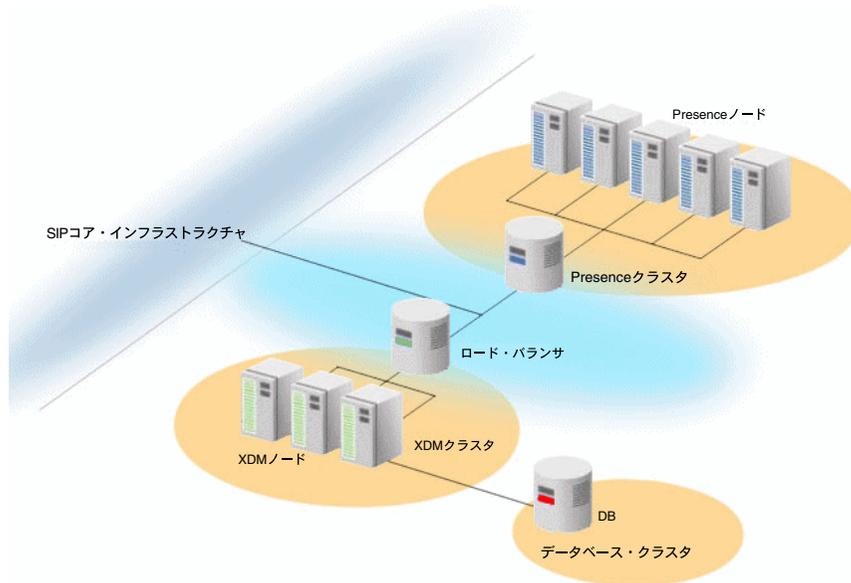
Presence マルチノード・トポロジ

図 5-1 は、クライアントが 1 つの Presence サーバーとやりとりする様子を示しています。より大きな規模のユーザー・ベースを扱うためには、このような単一サーバーを拡張する必要があります。このためには、Presence ノードと XDM ノードをシステムにさらに追加します。このようにして拡張したプレゼンス・サービスは 2 つのクラスタに分かれます。「定義」で定義した Presence クラスタと XDM クラスタです。

注意：最終的なプレゼンス・サービスが複数の別個のクラスタ、ノードおよびコンポーネントで構成されていることを管理者が理解することが重要です。ただし、この事実をエンド・ユーザーが認識することはありません。

図 5-4 に、必要なすべてのコンポーネントを備えた完全な Presence クラスタと XDM クラスタを示します。この図では、2 つのクラスタ Presence と XDM が別個のクラスタとして扱われ、最初のトラフィックにおいてこれら 2 つのネットワークへの経路は、常にそれぞれのロード・バランサを介することも示されます。Presence サーバーは、たとえばプレゼンス・ルール・ドキュメントに対するサブスクリプションを設定するとき、実際には XDM クラスタのロード・バランサを経由します。ただし、サブスクリプションが確立されると、その後のリクエストはロード・バランサを通らずに、サブスクリプションをホストする XDMS インスタンスに直接送られます。XDM クラスタのすべてのノードは、Presence クラスタから直接アクセスできます。実際、PS はプレゼンス・ルール・ドキュメントをフェッチするときに、XDMS インスタンスに直接アクセスします。

図 5-4 大規模デプロイの 2 つのクラスタ



この図では、2 つのロード・バランサ (Presence クラスタの前に 1 つと XDM クラスタの前に 1 つ) が示されていますが、それらは通常は同じ物理マシンに格納されることに注意してください。

コンポーネントの概要

2つの異なるノード（Presence ノードと XDM ノード）は、それぞれ小規模なコンポーネントのセットで構成されます。これらのコンポーネントについてこの項で定義して説明します。コンポーネントの違いと目的を理解することが重要です。実際のインストールを実行するときは、これらの様々なコンポーネントが物理ノードにデプロイされる要素になります。

ロード・バランサ

ロード・バランサの目的は、他のコンポーネントにトラフィックを分散することです。ロード・バランサは下位レベルのコンポーネントを認識し、SIP トラフィックを常に User Dispatcher に分散しますが、XCAP トラフィックは Aggregation Proxy に分散します。

User Dispatcher

User Dispatcher が行う処理は、受信リクエストからのユーザー・アイデンティティの抽出です。ユーザーに基づき、トラフィック（SIP と XCAP の両方）をサブアプリケーションに応じて PS または XDMS にディスパッチします。

Presence サーバー

Presence サーバーは、Presence イベントパッケージに対する SUBSCRIBE および PUBLISH リクエストを受信して処理し、該当する場合に NOTIFY を送信するコンポーネントです。presence.wininfo イベントパッケージに対する受信 SUBSCRIBE リクエストも処理します。PS は、プレゼンス・ルールと pidf-manipulation（プレゼンスのハードステート）ドキュメントを得るために XDMS とやりとりします。

XDM サーバー

XDMS の主な目的は、XML ドキュメントのリモート・ファイル記憶域として動作することです。これらのドキュメントは XCAP を使用して操作できます。XCAP には、クライアントがドキュメント内の変更に対するサブスクリプションを設定できる SIP インタフェースもあります。イベント・パッケージは ua-profile です。XDMS は、サブスクライバに NOTIFY:s を送信することでドキュメントの状態を伝えます。

Aggregation Proxy

Aggregation Proxy の役割は、すべての着信 XCAP トラフィックを認証してから、それらのリクエストを User Dispatcher にプロキシ処理することです。したがって、Aggregation Proxy が XDMS に直接アクセスすることはありません。XDMS へのアクセスは User Dispatcher が実行します。

データベース

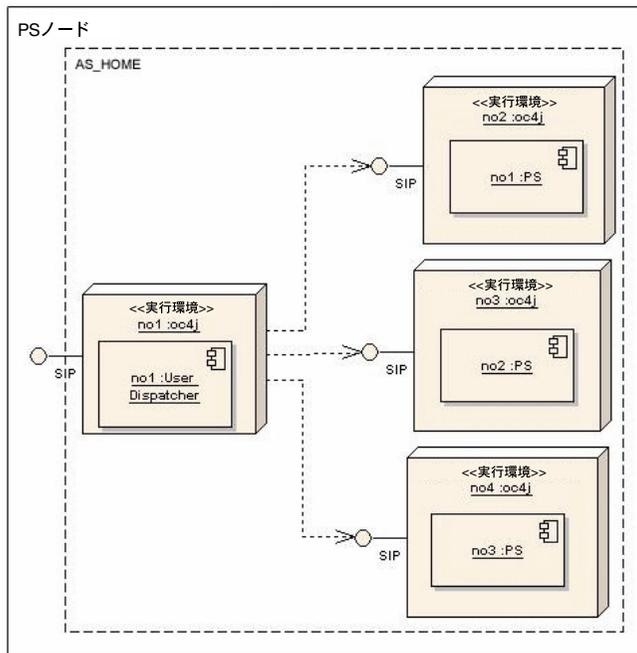
データベースは、XDMS で管理される XML ドキュメントが物理的に格納される場所です。

Presence ノード

Presence ノードは Presence クラスタのメイン・コンポーネントであり、受信トラフィックを正しい Presence サーバー・インスタンスにディスパッチします。ブラックボックスの観点ではユーザーにプレゼンス情報を提供します。User Dispatcher はシングル・ノード・デプロイでもマルチノード・デプロイでも同じ目的を果すことを理解してください（目的は受信トラフィックを特定の PS インスタンスにディスパッチすることです。このインスタンスが同じ物理ノードで実行しているかいないかは、User Dispatcher には関係ありません）。User Dispatcher は特定のノードを完全アドレス（IP アドレスおよびポート）で識別します。ローカル・インスタンスという概念はありません。

図 5-5 に、典型的な Presence ノードのレイアウトを示します。このノードには必ず User Dispatcher がデプロイされ、ノードそのものへの主経路として機能します。通常、User Dispatcher はポート 5060 (SIP のデフォルト・ポート) をリスニングし、そのノード上の他の Presence サーバーは他のポートでリスニングします。このように、シングル・ノードはクライアントに対して 1 つの Presence サーバーとして示されますが、実際には User Dispatcher の後ろで複数のインスタンスが実行されています。Presence ノードにデプロイされる各コンポーネントはそれぞれの別個の JVM で実行されます (つまり、User Dispatcher と PS インスタンスはすべて独自の OC4J インスタンスで実行されます)。

図 5-5 Presence ノードにデプロイされるコンポーネント



これらの OC4J インスタンス (この例では 4 つ) はすべて同じ Oracle Application Server (同一 AS_HOME) 内で実行していることに注意してください。

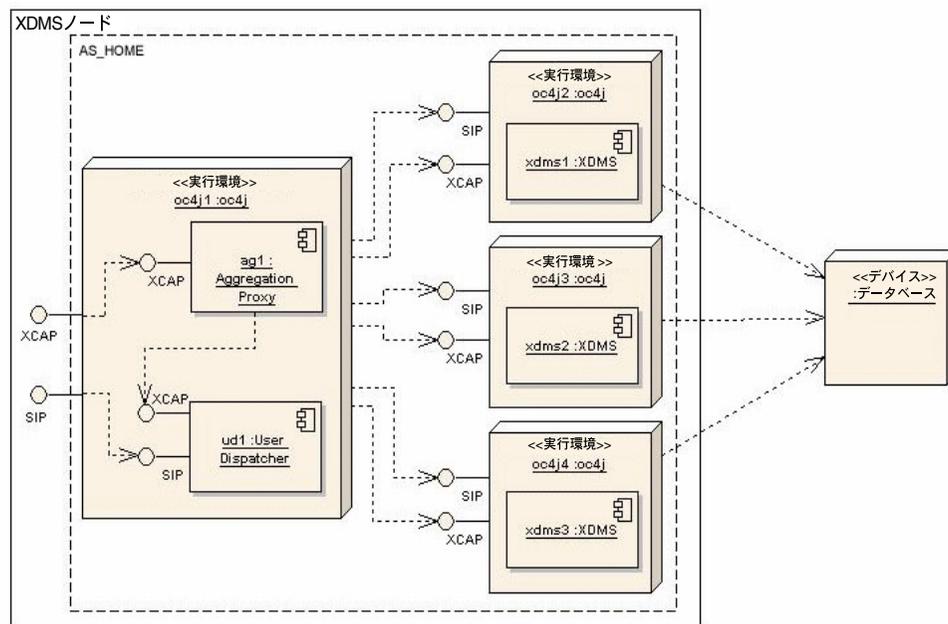
XDM ノード

XDM ノードには、図 5-6 のように必ず Aggregation Proxy がデプロイされます。これは、通常ポート 80 で XCAP トラフィックをリスニングします (XCAP は HTTP を介して送信される)。Aggregation Proxy は受信トラフィックを認証し、認証が正常に終了すると、リクエストを User Dispatcher に転送します。Presence ノードと同様に、XDM ノードにも User Dispatcher がデプロイされます (通常はポート 5060)。SIP トラフィックに関しては XDM ノードと Presence ノードの違いはまったくありません。2 つのタイプのノードの違いは、User Dispatcher が XCAP トラフィックのディスパッチも行うという点です。User Dispatcher は、SIP の場合と同様に、ユーザー ID をリクエストから抽出し、それに基づいてリクエストを特定の XDMS インスタンスにマップして、その XDMS インスタンスにリクエストを転送します。

また、X 個の XDMS インスタンスがデプロイされ、User Dispatcher は SIP トラフィックと XCAP トラフィックの両方をそれらにディスパッチします。Presence ノード上の PS インスタンスの場合と同様に、各 XDMS インスタンスは互いに認識せず、独立して実行します。

また、Aggregation Proxy と User Dispatcher が同じ OC4J コンテナにデプロイされ、同じ JVM を使用することにも注意してください。また、すべての OC4J インスタンスは (Presence ノードの場合と同様に) 同一の Oracle Application Server 内で実行します。

図 5-6 XDMS ノード



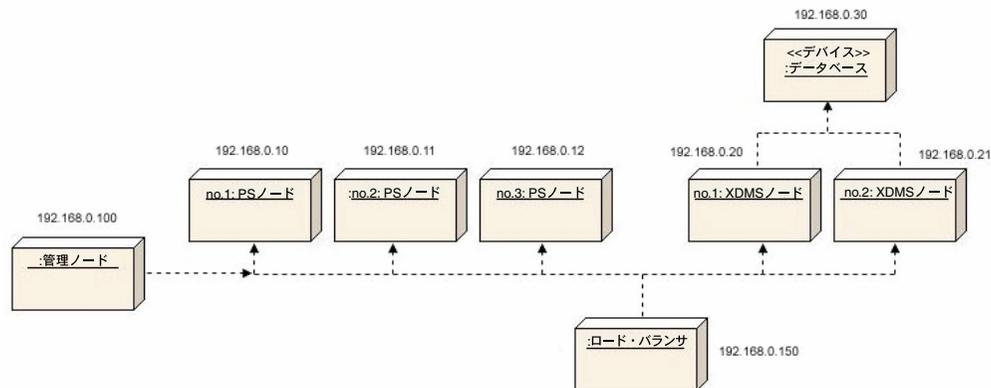
インストール

ここまで、Presence の大規模デプロイの一般的なレイアウトとコンポーネントについて説明しました。この項では、システムのインストール方法について説明します。

サンプル・ネットワーク

必要な手順を簡単に説明するため、図 5-7 に示すネットワークを例として使用します。

図 5-7 サンプル・ネットワーク



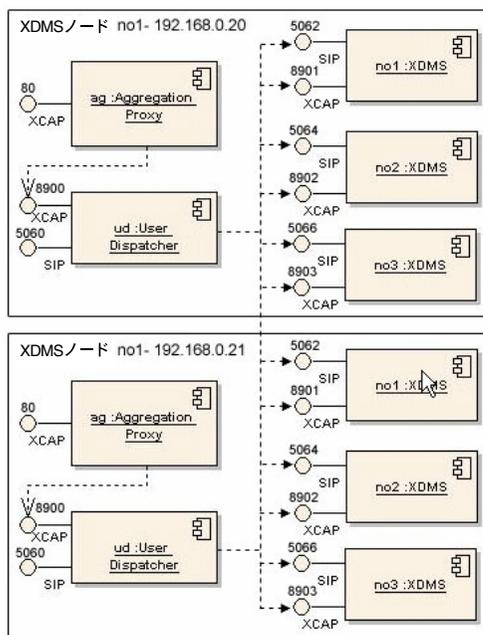
このネットワークは3つのPSノードで構成され、それらが Presence クラスタを形成しています。XDM クラスタは2つのXDMSノードで構成され、アドレスが192.168.0.30のデータベースにアクセスしています。両方のクラスタは同じ物理ロード・バランサを共有しています。ロード・バランサによりネットワークが外部ネットワークと内部ネットワークに分けられていないことに注意してください。このため、ロード・バランサのIPアドレスは192.168.0.150のみです。

管理しやすくするために、1つの管理ノードがネットワークに追加されています。他のすべてのノードとインスタンスの構成はこのノードを介して実行されます。

2つのクラスタについて詳しく説明します。まず、すでに説明した3つの Presence ノードで構成される Presence クラスタがあります。図 5-8 に、Presence クラスタの詳細な図を示します。各 Presence ノードには、1つの User Dispatcher がデプロイされ、3つの PS インスタンスがあります。これら4つのコンポーネントはそれぞれが独自の OC4J インスタンスで実行されていますが、細部は図を簡略化するために省いています。

図 5-8 に、XDM クラスタの詳細な図を示します。この例では、3つの XDMS インスタンスが実行されており、1つの User Dispatcher と 1つの Aggregation Proxy が対応しています。「XDM ノード」で説明したように、Aggregation Proxy と User Dispatcher は同一の OC4J インスタンスで実行されますが、3つの XDMS インスタンスはそれぞれ独自の OC4J で実行されます。これらすべての OC4J インスタンス (4つのインスタンス) は、同じ Oracle Application Server 内で実行されます。

図 5-8 クラスタの例



Oracle Application Server 10.1.3.4 のインストール

最終システムにインストールされるすべてのノード（Presence ノード、XDM ノードまたは管理ノード）では、同じ基本インストールが行われます（つまり、すべてが Oracle Application Server 10.1.3.4 を実行します）。したがって、次の項では Oracle Application Server のインストールおよび設定方法について詳しく説明します。最初に Oracle Application Server 10.1.3.2 をインストールしてから、10.1.3.4 パッチ・セットを適用します。主な手順は次のとおりです。

1. Oracle Universal Installer を起動して Oracle Application Server 10.1.3.2 をインストールします。
2. 「**拡張インストール**」を選択します。
3. インストール・タイプとして「**Oracle WebCenter Framework**」（下から 2 番目）を選択します。

インストールするノードが管理ノードでないかぎり、「このインスタンスで Oracle Enterprise Manager 10g ASControl を起動する」チェック・ボックスを選択しないでください。ASControl はシステム内の 1 つのノードのみで実行します。管理ノードに配置されます。システムのその他のすべてのノード（PS ノードおよび XDMS ノード）は管理ノードを介して制御されます。

4. 検出アドレスを入力し、そのマルチキャスト・アドレスのために使用する値がクラスタのすべてのノードで同じになるようにします。つまり、PS ノードおよび XDMS ノードの検出アドレスは管理ノードと同じであることが必要です。違う場合、それらのノードは管理ノードによって検出されません。詳細は、『Oracle Application Server インストレーション・ガイド』を参照してください。

AS 10.1.3.2 のインストールが完了したら、続けて次の手順を実行し、AS 10.1.3.4 パッチ・セットをインストールします。

1. 10.1.3.4 パッチ用の Oracle Universal Installer を実行して、10.1.3.4 WebCenter パッチを適用します。
2. Java 5 アップデート 14 をインストールします。推奨する手順は次のとおりです。
 - JDK 1.5 アップデート 14 用の Sun Java インストーラを実行し、選択したディレクトリ (<jdk-directory> と表記) に JDK をインストールします。
 - \$ORACLE_HOME に移動し、ファイル jdk の名前を jdk.install.backup に変更して、インストールされている JDK のバックアップを作成します。
 - <jdk-directory> を指す jdk という名前のシンボリック・リンクを \$ORACLE_HOME に作成します。

管理ノードのインストール

Oracle Application Server Control を使用して、クラスタ内のすべてのノードを構成および管理することができます。クラスタに 1 つの管理ノードを配置すると、システムの操作を容易に行うことができ、変更した構成をすべてのノードですぐに有効にすることができます。

この管理ノードをインストールするには、Oracle Application Server を 1 つのノードにインストールし（サンプル・ネットワークでは、このノードは 192.168.0.100 で実行）、Application Server Control の実行を有効にします。「このインスタンスで Oracle Enterprise Manager 10g ASControl を起動する」チェック・ボックスを選択してください。管理ノードのインストールに必要な手順はこれですべてです。

システムのすべてのノードは、同じ検出アドレスを使用するように構成する必要があります。このサンプルでは、マルチキャスト・アドレス 235.0.0.1:6789 を使用しています。このアドレスをシステムのすべてのノードで使用する必要があります。

また、管理ノードは setupLinux.tar.gz ファイル（解凍済）を含む必要があります。これにより、後でデプロイする .ear ファイルをあらかじめ管理ノードに配置でき、デプロイの準備が整います。

Presence ノードのインストール

次の手順は、Presence ノードをインストールするために実行する必要があります。1つの Presence クラスタには多数の Presence ノードが含まれるため、次の各手順はノードごとに繰り返す必要があります。

注意： デプロイをより円滑に行うためにはすべてのノードは同時にインストールおよび構成する必要があります。

実行する手順は次のとおりです。

1. Oracle Application Server 10.1.3.4 をインストールします。
2. Oracle Universal Installer を使用して、User Dispatcher を含む最初のインスタンスをインストールします。Sip コンテナはデフォルトでインストールされます。
3. 他に必要な数の OC4J インスタンスを作成します。
4. これらの OC4J インスタンスを構成します。
5. これらの新たに作成した OC4J インスタンスに Presence をデプロイして構成します。
6. User Dispatcher を構成します。
7. インストール後の作業あるいはインストールのチューニングを行います。

Oracle Application Server のインストール

各 Presence ノードは Oracle Application Server 上で実行するため、最初に Oracle Application Server をインストールする必要があります。必要な手順は「[Oracle Application Server 10.1.3.4 のインストール](#)」で説明していますが、これは管理ノードではないため、「このインスタンスで Oracle Enterprise Manager 10g ASControl を起動する」チェック・ボックスを選択しないようにしてください。

検出アドレスについては、管理ノードで選択したアドレスと同じものを選択してください。サンプル・ネットワークでは、管理ノードに選択されている検出アドレスは 235.0.0.1:6789 です。ここでもこのアドレスを使用します。

User Dispatcher のインストール

次の手順に従って User Dispatcher をインストールします。

1. OCMS 用の Oracle Universal Installer を使用して、「User Dispatcher」および「SipContainer」のみを選択します。
2. このインスタンスに対してデフォルトの SIP ポート値 5060 を使用します。
3. Oracle Universal Installer のインストール手順を完了します。

インストーラによって、Oracle Application Server で実行する ocms という名前の OC4J インスタンスが作成されます。このインスタンスにデプロイされるアプリケーションは User Dispatcher のみです。

追加インスタンスの作成

作成するインスタンスの数は使用可能なメモリーによって変わります。新しいインスタンスを設定するたびに 2.5GB のメモリーが消費されます。使用可能なメモリーに収まる最大限のインスタンスを作成することをお勧めします。オペレーティング・システムのためにメモリーを残しておく必要があることに注意してください。

最終的なチューニングを後から実行してシステム・パフォーマンスを最適化することができます。

続行する前に、`$ORACLE_HOME/opmn/bin/opmnctl stopall` と入力して、実行中のインスタンスを停止します。

`$ORACLE_HOME/bin` の `createinstance` コマンドを使用して OC4J インスタンスを追加作成します。次に例を示します。

```
cd $ORACLE_HOME/bin
./createinstance -instanceName ps1 -groupName presence -httpPort 8901 -defaultAdminPass
```

ここで、`ps1` はインスタンスの名前、グループの名前は `presence` です。グループが存在しない場合は作成されます。

`-httpPort 8901` によってポート番号が指定されます。管理しやすくするために、連続した使用可能ポート番号を使用することをお勧めします。サンプル・ネットワークでは、8901、8902 および 8903 を、それぞれ OC4J インスタンス `ps1`、`ps2` および `ps3` に対して使用します。

ここでは、`-defaultAdminPass` によってパスワードの設定が省略されています。これは、インスタンスのパスワード設定を省略するように `createinstance` に指定します。後から別のコマンドを発行して、このインスタンスのパスワードを設定します。このオプションを含めないと、`createinstance` によって、パスワードを入力するように求められます。作成するインスタンスが少数の場合はこの方法を利用できますが、インスタンス作成のスクリプトを実行する場合は `-defaultAdminPass` を使用した方が効率がよくなります。

作成したインスタンスにパスワードを設定するには、次のコマンドを実行します。

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../<instance-name> -jar jazn.jar -activateadmin <password>.
```

たとえば、`ps1` インスタンスのパスワードを `myPassword1` に設定するには、次のように実行します。

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps1 -jar jazn.jar -activateadmin myPassword1
```

これら 2 つのコマンドを、十分なインスタンスを作成するために必要な回数繰り返します。サンプル・ネットワークでは、3 つの `Presence` インスタンスにそれぞれ対応する 3 つの OC4J インスタンス `ps1`、`ps2` および `ps3` を作成するために、これらのコマンドを次のように 3 回実行する必要があります。

```
cd $ORACLE_HOME/bin
./createinstance -instanceName ps1 -groupName presence -httpPort 8901 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps1 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName ps2 -groupName presence -httpPort 8902 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps2 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName ps3 -groupName presence -httpPort 8903 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps3 -jar jazn.jar -activateadmin myPassword1
```

OC4J インスタンスの構成

前述の手順では、実際の Presence サーバーがデプロイされる OC4J インスタンスを新たに作成する方法について説明しました。デプロイする前に、Sip サブレット・コンテナを取得するようにそれらの新しいインスタンスを構成することも必要です。必要な jar は、User Dispatcher をインストールしたときにすでに Oracle Application Server にインストールされています。このため、実行する必要があるのは、各 ps X インスタンスが共有ライブラリから Sip サブレット・コンテナ jar を取得するように構成することだけです。また、適切な起動とロギングのために新しい OC4J インスタンスを構成する必要もあります。すべての新しい OC4J インスタンスに対して実行する必要がある全手順は次のとおりです。

1. Sip サブレット・コンテナに必要なすべての jar が含まれる共有ライブラリを取得するようにインスタンスを構成します。このためには boot.xml ファイルを編集します。
2. ロギングを構成します。XDMS インスタンスで処理されたロギングを新しいインスタンスが使用するためには、j2ee-logging.xml ファイルを編集する必要があります。
3. インスタンスの起動と停止のパラメータを編集して、Sip サブレット・コンテナがロードされるようにします。このためには \$ORACLE_HOME/opmn.xml ファイルを編集します。
4. 新しいインスタンスがリスニングする SIP ポートを指定します。
5. Sip サブレット・コンテナが適切な IP アドレスをリスニングするように構成します。
6. xcap config ディレクトリを追加します。
7. 構成を確認します。

共有ライブラリを使用するためのインスタンスの構成 Sip サブレット・コンテナに必要な jar が含まれる共有ライブラリを取得するようにインスタンスを構成します。このためには、OCMS インストーラによって作成された ocms インスタンスから boot.xml を次のようにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/boot.xml $ORACLE_HOME/j2ee/<instance name>/config/
```

これで、ocms インスタンスの適切に構成された boot.xml が別のインスタンスにコピーされます。この手順を新たに作成したすべてのインスタンスについて繰り返します。サンプル・ネットワークでは、<instance name> を ps1、ps2 および ps3 で置き換えてこのコマンドを 3 回発行し、boot.xml を 3 つの OC4J インスタンスにコピーします。

ロギングの構成 共有ライブラリをロードするために boot.xml ファイルをコピーしたのと同様に、ロギングのための構成ファイルをコピーすることができます。ocms の config ディレクトリにある j2ee-logging.xml ファイルをすべてのインスタンスにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/j2ee-logging.xml $ORACLE_HOME/j2ee/<instance name>/config/
```

サンプル・ネットワークでは、<instance name> を ps1、ps2 および ps3 で置き換えてこのコマンドを 3 回発行し、j2ee-logging.xml を 3 つの OC4J インスタンスにコピーします。

JVMの起動および停止パラメータの構成 JVMパラメータを構成して、Sip サブレット・コンテナがロードされるようにインスタンスの起動と停止を設定します。これには、`opmn.xml` ファイルを編集して、すべての Presence の OC4J インスタンス（サンプル・ネットワークでは `ps1`、`ps2` および `ps3`）および `ocms` インスタンスについて起動および停止のパラメータを設定します。

起動パラメータ：

```
<data id="java-options" value="-server -Xmx2500M -Xms2500M -Xloggc:/<instance name>/
sdp/logs/gc.ps1.log -XX:+PrintGCDetails -XX:NewRatio=3 -XX:+UseConcMarkSweepGC
-XX:+UseParNewGC -XX:PermSize=128m -XX:MaxPermSize=128m -Xss128k
-Dhttp.maxFileCacheEntries=-1 -Djava.security.policy=${ORACLE_HOME}/j2ee/
<instance name>/config/java2.policy -Djava.awt.headless=true -Dhttp.webdir.enable=false
-DopmnPingInterval=1
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.
sipservletcontainer.deployer.Oc4jApplicationHook "/>
```

停止パラメータ：

```
<data id="java-options" value="-Djava.security.policy=${ORACLE_HOME}/j2ee/
<instance name>/config/java2.policy -Djava.awt.headless=true -Dhttp.webdir.enable=false
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.
sipservletcontainer.deployer.Oc4jApplicationHook"/>
```

これらのパラメータの意味は次のとおりです。

-Xmx2500M: 最大 JVM メモリーを 2.5GB に設定します。

-Xms2500M: 最小 JVM メモリーを 2.5GB に設定します。

-XX:+PrintGCDetails: コレクション・アクティビティのログギングを有効にします。

-XX:NewRatio=3: ヒープの新世代と旧世代の比率を 1:3 に設定します（つまり、エデン領域とサブイバ領域の合計サイズが合計ヒープ・サイズの 4 分の 1 になります）。

-XX:+UseConcMarkSweepGC: 同時マークアンドスイープ・ガベージ・コレクタ（同時ロー・ポーズ・コレクタとも呼ばれる）を有効にします。

-XX:+UseParNewGC: パラレル・スレッドを使用します。

-XX:PermSize=128m: 永続世代の初期サイズを 128MB に設定します。

-XX:MaxPermSize=128m: 永続世代の最大サイズを 128MB に設定します。

-Xss128k: 各スレッドの `stat` サイズを 128KB に設定します。

-Dhttp.maxFileCacheEntries=-1: Oracle Application Server にバンドルされている HTTP サーバーでのキャッシングを無効にします。このリリースでは、Presence サーバーの良好なパフォーマンスを得るためにキャッシングを無効にする必要があります。

-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipservletcontainer.deployer.Oc4jApplicationHook: OC4J で Sip コンテナをロードできるようにします。

注意： インストールした `ocms` インスタンスに、複数の JVM インストールに必要なパラメータがすべて含まれない場合、前述のパラメータを `ocms` インスタンスにも設定するようにしてください。

SIP ポートの構成 起動および停止パラメータの設定が終了したので、次に Sip サブレット・コンテナが正しいポートでリスニングするように構成する必要があります。デフォルトでは、Sip サブレット・コンテナは、SIP をポート 5060、SIPS を 5061 でリスニングします。ファイル \$ORACLE_HOME/opmn/conf/opmn.xml では、OCMS インストーラによって作成された ocms インスタンスが次のようにこれらのデフォルト値で構成されています。

```
<port id="sip" range="5060"/>
<port id="sips" range="5061"/>
```

ocms インスタンスのこれらの値はそのままにしておく必要があります。User Dispatcher はこれらのポートで受信トラフィックをリスニングするためです。Presence ノードのために作成した他の OC4J インスタンスについては、ポート競合を避けるためにそれぞれに別のポートを構成する必要があります。管理しやすくするために、連続した使用可能ポートを使用することをお勧めします。サンプル・ネットワークでは、Presence インスタンスに 5062 から始まるポートを割り当てています。構成は次のようになります。

ps1

```
<port id="sip" range="5062"/>
<port id="sips" range="5063"/>
```

ps2

```
<port id="sip" range="5064"/>
<port id="sips" range="5065"/>
```

ps3

```
<port id="sip" range="5066"/>
<port id="sips" range="5067"/>
```

適切な IP アドレスでリスニングするための Sip サブレット・コンテナの構成 デフォルトでは、Sip サブレット・コンテナは IP アドレス 127.0.0.1 でリスニングします。このアドレスを、このマシンの外部からアドレス指定できる IP アドレスに変更する必要があります。前述の手順で、コンテナをロードするように Oracle Application Server を構成しましたが、実際の Sip サブレット・コンテナの構成ファイルは取得していません。デフォルトの構成ファイルを生成するには、Oracle Application Server を起動してから停止する必要があります。デフォルトの構成ファイルが生成されたら、デプロイに合わせてファイルを編集することができます。

Oracle Application Server を起動するには次のようにします。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

起動して実行したら、次のように停止します。

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

sdp という名前の新しい構成ディレクトリが、\$ORACLE_HOME/j2ee/<instance name>/config/sdp/ に作成されています。各インスタンスの OCMS の構成に関連するすべての情報がここにあります。各インスタンスには SipServletContainer.xml という名前のファイルがあります。各インスタンスについてこのファイルを編集し、IPAddress 属性を 127.0.0.1 から Sip サブレット・コンテナがリスニングする IP アドレスに変更する必要があります。ただし、このファイルはこのマシンのすべての Sip サブレット・コンテナで同じにする必要があるため、それぞれを手動で編集する必要はありません。ocms ディレクトリにあるファイルをコピーして再利用します。コマンドは次のとおりです。

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/SipServletContainer.xml $ORACLE_HOME/j2ee/
<instance name>/config/sdp/
```

<instance-name> を Presence の OC4J インスタンスの名前で置き換えます。サンプル・ネットワークでは、このコマンドを 3 回 (ps1、ps2 および ps3 に対して 1 回ずつ) 実行します。これで Sip サブレット・コンテナがすべてのインスタンスに対して構成されたため、サーバーを起動することができます。Oracle Application Server ですべての OC4J インスタンスを起動するには、次のようにします。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

xcap 構成ディレクトリの構成 ua-profile イベント・パッケージは Presence ノードで明示的には使用されませんが、存在するように構成する必要があります。このため、xcap 構成ファイルをすべての Presence OC4J インスタンスにコピーしてください。これを行うには、xcap ディレクトリを各インスタンスの sdp 構成ディレクトリ \$ORACLE_HOME/j2ee/<instance-name>/config/sdp/ にコピーします。サンプル・ネットワークでは、xcap ディレクトリを次のディレクトリにコピーする必要があります。

```
$ORACLE_HOME/j2ee/ps1/config/sdp/
$ORACLE_HOME/j2ee/ps2/config/sdp/
$ORACLE_HOME/j2ee/ps3/config/sdp/
```

xcap ディレクトリは、OCMS インストーラを抽出したディレクトリにあります。たとえば、インストーラをディレクトリ OCMS_INSTALLER に抽出した場合、xcap ディレクトリは次の場所にあります。

```
OCMS_
INSTALLER/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/DataFiles/Expanded/Shiphome/
shiphome-archive/shiphome-archive/xcapconf/conf
```

springbeans.xml の構成 次のように、springbeans.xml を ocms インスタンスの sdp 構成ディレクトリからすべての Presence インスタンスの sdp 構成ディレクトリにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/springbeans.xml $ORACLE_HOME/j2ee/
<instance name>/config/sdp/
```

サンプル・ネットワークでは、<instance-name> を ps1、ps2 および ps3 でそれぞれ置き換えて、このコマンドを 3 回実行します。

構成設定の確認 インストーラを続行する前に、ログを調べてこれまでの操作が正常に処理されたことを確認します。また、netstat コマンドを発行して、サーバーが正しいポートでリスニングしていることも確認できます。

ログ・ファイル: ログ・ファイルを定期的に監視することも必要です。構成に問題がある場合にはログ・ファイルによって明らかになります。確認する必要がある主なログ・ファイルは 2 つです。1 つ目は、\$ORACLE_HOME/opmn/logs にある各インスタンスのコンソール出力です。通常、Presence ノードの場合は次のようなファイル名になります。

```
presence-~<instance-name>-default_group-1.log
```

ocms インスタンスに対するファイルと、opmn そのもののログ・ファイルもあります。常にこれらのファイルをよく注意してください。異常が発生する大半のケースでは、これらのログ・ファイルによって潜在的な問題に関する十分な情報が得られるためです。

もう 1 つの重要なログ・ファイルは、Sip サーブレット・コンテナそのものによって生成されるログです。各インスタンス（たとえば ps1）には sdp ログ・ディレクトリがあり、trace.log ファイルが含まれます。このファイルにもよく注意してください。このファイルは ocms インスタンスに対しても存在します。すべてのインスタンスのすべての trace.log ファイルを確認するようにしてください。

opmnctl: opmnctl ツールは、サーバーの起動と停止に使用されます。このツールでは、実行中のインスタンスのステータスも表示されます。基本コマンドは次のとおりです。

```
$ORACLE_HOME/opmn/bin/opmnctl status
```

すべてのプロセスと、それらが実行しているかどうかが表示されます。一般的な出力を次に示します。

```
Processes in Instance: priv5.priv5
-----
ias-component | process-type | pid | status
-----
OC4JGroup:presence | OC4J:ps3 | 4868 | Alive
OC4JGroup:presence | OC4J:ps2 | 4869 | Alive
OC4JGroup:presence | OC4J:ps1 | 4867 | Alive
OC4JGroup:default_group | OC4J:ocms | 4866 | Alive
OC4JGroup:default_group | OC4J:home | 4865 | Alive
ASG | ASG | N/A | Down
```

この出力は、ps1 ~ ps3 がこのマシンで実行していることを示します。ocms インスタンスも起動して実行しています。User Dispatcher がこのインスタンスで実行されていることに注意してください。opmnctl を使用して、各インスタンスがリスニングしているすべてのポートを表示することもできます。このためには、次の例のようにスイッチ -l を opmnctl に指定します。

```
$ORACLE_HOME/opmn/bin/opmnctl status -l
```

これにより、表 5-1 のような出力が表示されます。この表の出力は一部省略されています。通常は、プロセス ID、起動時刻、その他の情報が含まれます。

すべての構成が適切な場合は、ocms が SIP ポート 5060、ps1 が 5062 などのようにリスニングしていることが表示されます。

表 5-2 プロセス・タイプ別の使用ポート

プロセス・タイプ	ポート
OC4J:ps3	jms:12605, sip:5066, http:8903, rmis:12705, sip:5066, sips:5067, rmi:12405
OC4J:ps2	jms:12604, sip:5064, http:8902, rmis:12704, sip:5064, sips:5065, rmi:12404
OC4J:ps1	jms:12603, sip:5062, http:8901, rmis:12703, sip:5062, sips:5063, rmi:12403
OC4J:ocms	jms:12602, sip:5060, http:7785, rmis:12702, sip:5060, sips:5061, rmi:12402
OC4J:home	jms:12601, http:8888, rmis:12701, rmi:12401

Presence のデプロイおよび構成

Oracle Enterprise Manager でグループ・ビューを使用し、アプリケーションをそのグループのすべてのインスタンスにデプロイします。デプロイ時にはアプリケーションの名前として presence を使用することをお勧めします。デプロイが終了したら、すべてのインスタンスで次の項目を構成する必要があります。

- PresRulesXCAPUri および PIDFManipulationXCAPUri の変更
- UserAgentFactoryServiceImpl.xml の更新
- JGroups の有効化

PresRulesXCAPUri および PIDFManipulationXCAPUri の変更 各 OC4J Presence ノード・インスタンスについて、Presence アプリケーション Mbean に移動し、次の属性の値を設定します。

```
- PIDFManipulationXCAPUri - sip:<xdmsHostIP>;transport=TCP;lr
- PresRulesXCAPUri - sip:<xdmsHostIP>;transport=TCP;lr
```

サンプル・ネットワークでは、XDMS プールが 192.168.0.150:5062 のロード・バランサ上にあるため、すべての Presence インスタンス (ps1、ps2 および ps3) に対して設定は次のようになります。

```
PIDFManipulationXCAPUri - sip:192.168.0.150:5062;transport=TCP;lr
PresRulesXCAPUri - sip:192.168.0.150:5062;transport=TCP;lr
```

UserAgentFactoryService ポートの更新 各 OC4J Presence ノード・インスタンスについて、Presence アプリケーションの UserAgentFactoryService MBean に移動し、ポートの競合を避けるために、マシン上の Presence インスタンスそれぞれについて Port 属性の値を一意に設定します。管理しやすくするために、連続した使用可能ポートを使用することをお勧めします。サンプル・ネットワークでは、インスタンス ps1、ps2 および ps3 に対して 5070、5071 および 5072 をそれぞれ使用しています。

JGroups の有効化 各 OC4J Presence ノード・インスタンスについて、PackageManager MBean に移動し、JGroupsBroadcastEnabled - true のように属性の値を設定します。

JGroups 構成でデフォルト値を使用するように JgroupXMLConfigPath の値は空にしておきます。デフォルトの JGroups 構成では、次の値が使用されます。

- マルチキャスト・アドレス : 230.0.0.1
- マルチキャスト・ポート : 7426
- TTL: 1

User Dispatcher の構成

デプロイ内のすべての Presence インスタンスに SIP トラフィックをルーティングできるように User Dispatcher を構成します。User Dispatcher が存在するマシン上のすべての Presence インスタンスと、デプロイ内の他のマシンのすべての Presence インスタンスに SIP トラフィックを送るように、すべての User Dispatcher を構成する必要があります。つまり、各 Presence ノード (Presence マシン) の User Dispatcher は、他のノードのすべての Presence インスタンスを認識する必要があります。User Dispatcher が SIP トラフィックをすべての Presence サーバーにルーティングできるように構成するには、次の手順を実行します。

1. 管理ノードの Enterprise Manager にログインします。
2. クラスタ・ビューで、構成する User Dispatcher が存在する Presence ノードを選択します。
3. 「アプリケーションの選択」 → 「userdispatcher アプリケーション定義の MBeans」を選択します。
4. 「presence-pool」をクリックし、「サーバー」を選択します。
5. デプロイ内のすべての Presence サーバーを指す SIP URI を追加します。URI の形式は次のとおりです。

```
sip:<ip-address>:<port>;transport=tcp;lr
```

サンプル・ネットワークには 3 つの Presence ノード (マシン) があり、それぞれに 3 つの Presence サーバーがあります。Presence サーバーは合計 9 つです。各 User Dispatcher が SIP トラフィックを 9 つの Presence サーバーにルーティングするように構成する必要があります。各 User Dispatcher の Presence プールに次の内容を追加します。

```
sip:192.168.0.10:5062;transport=tcp;lr
sip:192.168.0.10:5064;transport=tcp;lr
sip:192.168.0.10:5066;transport=tcp;lr
sip:192.168.0.11:5062;transport=tcp;lr
sip:192.168.0.11:5064;transport=tcp;lr
sip:192.168.0.11:5066;transport=tcp;lr
sip:192.168.0.12:5062;transport=tcp;lr
sip:192.168.0.12:5064;transport=tcp;lr
sip:192.168.0.12:5066;transport=tcp;lr
```

インストールのチューニング

ここでは、インストールしたコンポーネント同士が共存して適切に稼働できるように、チューニングする方法について説明します。

WebCenter インスタンスの無効化 インストールが終了したら、不要にリソースを消費しないように WebCenter インスタンスを無効にします。これには、`$ORACLE_HOME/opmn/conf/`にある `opmn.xml` ファイルを編集します。サンプルの `<process-type id="OC4J_WebCenter" module-id="OC4J" status="disabled">` のように、ステータスを `enabled` から `disabled` に変更します。変更を有効にするために、次のようにサーバーを再起動します。

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

停止した後で再び起動します。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

ホーム・インスタンスの無効化 すべてのインストールが終了したら、管理ノードを除くすべてのノードでホーム OC4J インスタンスを停止します。管理ノードは、Enterprise Manager コンソールにログインして、デプロイ全体の構成を表示または変更するために使用します。ホーム・インスタンスを停止するには、すべての Presence ノードの `opmn.xml` ファイルを編集して、WebCenter インスタンスの場合と同じ方法でホーム・インスタンスを無効にします。ホームが必要になるのは、`createinstance` コマンドを使用して新しいインスタンスを作成するときのみです。

XDM ノードのインストール

XDM ノードのインストールは Presence ノードのインストールと似ています。どちらのタイプのノードにも User Dispatcher がデプロイされ、その他に Presence アプリケーションの `ear` ファイルがデプロイされた X 個のインスタンスが実行しています。違いは、XDM ノードには AggregationProxy と Subscriber Data Services もデプロイされている点です。また、User Dispatcher と XDMS アプリケーションの構成がわずかに異なります。

注意： デプロイをより円滑に行うためにはすべてのノードは同時にインストールおよび構成する必要があります。

XDM ノードをインストールする手順は次のとおりです。

1. マシンに Oracle Application Server 10.1.3.2 をインストールします。
2. Oracle Application Server に 10.1.3.4 パッチを適用します。
3. OCMS インストーラを使用して、User Dispatcher、Aggregation Proxy および Subscriber Data Services を含める最初のインスタンスをインストールします。Sip サブレット・コンテナもデフォルトでインストールされます。
4. 他に必要な数の OC4J インスタンスを作成します。
5. これらの新たに作成した OC4J インスタンスを構成します。
6. これらの新たに作成した OC4J インスタンスに Presence アプリケーション `ear` をデプロイして構成します。
7. User Dispatcher を構成します。
8. インストールをチューニングします。

注意： 最初の XDM ノードをインストールしたら、後続の XDM ノードも最初のノードと同じデータベースを指すようにします。

後続の XDM ノードをインストールするときも、「Do you wish to reuse the existing schemas?」というオプションを選択する必要があります。

Oracle Application Server 10.1.3.2 のインストール

最初に Oracle Application Server を XDMS ノードにインストールする必要があります。

1. Oracle Universal Installer を起動して Oracle Application Server 10.1.3.2 をインストールします。
2. 「**拡張インストール**」を選択します。
3. インストール・タイプとして「**Oracle WebCenter Framework**」を選択します。
4. 検出アドレスを入力し、そのマルチキャスト・アドレスのために使用する値がクラスタのすべてのノードで同じになるようにします。つまり、PS ノードおよび XDMS ノードの検出アドレスは管理ノードと同じであることが必要です。違う場合、それらのノードは管理ノードによって検出されません。詳細は、『Oracle Application Server インストール・ガイド』を参照してください。

Oracle Application Server への 10.1.3.4 パッチの適用

次に Oracle Application Server パッチを適用する必要があります。

1. 10.1.3.4 パッチ用の Oracle Universal Installer を実行して、10.1.3.4 WebCenter パッチを適用します。
2. Java 5 アップデート 14 をインストールします。推奨する手順は次のとおりです。
 - JDK 1.5 アップデート 14 用の Sun Java インストーラを実行し、選択したディレクトリ (<jdk-directory> と表記) に JDK をインストールします。
 - \$ORACLE_HOME に移動し、ファイル jdk の名前を jdk.install.backup に変更して、インストールされている JDK のバックアップを作成します。
 - <jdk-directory> を指す jdk という名前のシンボリック・リンクを \$ORACLE_HOME に作成します。

User Dispatcher のインストール

次の手順に従って User Dispatcher をインストールします。

1. OCMS 用の Oracle Universal Installer を使用して、「Sip サーブレット・コンテナ」、「User Dispatcher」、「Subscriber Data Services」および「Aggregation Proxy」を選択します。
2. このインスタンスに対してデフォルトの SIP ポート値 5060 を使用します。
3. Oracle Universal Installer のインストール手順を完了します。

インストーラによって、Oracle Application Server で実行する ocms という名前の OC4J インスタンスがもう 1 つ作成されます。このインスタンスにデプロイされるアプリケーションは、User Dispatcher、Subscriber Data Services および Aggregation Proxy です。

追加 OC4J インスタンスの作成

\$ORACLE_HOME/bin の createinstance コマンドを使用して OC4J インスタンスを追加作成します。次に例を示します。

```
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms1 -groupName xdms -httpPort 8901 -defaultAdminPass
```

ここで、xdms1 はインスタンスの名前、グループの名前は xdms です。グループが存在しない場合は作成されます。

-httpPort 8901 によってポート番号が指定されます。管理しやすくするために、連続した使用可能ポート番号を使用することをお勧めします。サンプル・ネットワークでは、8901、8902 および 8903 をそれぞれ OC4J インスタンス xdms1、xdms2 および xdms3 に対して使用します。

ここでは、`-defaultAdminPass` によってパスワードの設定が省略されています。これは、インスタンスのパスワード設定を省略するように `createinstance` に指定します。後から別のコマンドを発行して、このインスタンスのパスワードを設定します。このオプションを含めないと、`createinstance` によって、パスワードを入力するように求められます。作成するインスタンスが少数の場合はこの方法を利用できますが、インスタンス作成のスクリプトを実行する場合は `-defaultAdminPass` を使用した方が効率がよくなります。

作成したインスタンスにパスワードを設定するには、次のコマンドを実行します。

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../<instance-name> -jar jazn.jar -activateadmin <password>.
```

たとえば、`xdms` インスタンスのパスワードを `myPassword1` に設定するには、次のように実行します。

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms1 -jar jazn.jar -activateadmin myPassword1
```

これら 2 つのコマンドを、十分なインスタンスを作成するために必要な回数繰り返します。サンプル・ネットワークでは、3 つの XDMS インスタンスにそれぞれ対応する 3 つの OC4J インスタンス `xdms1`、`xdms2` および `xdms3` を作成するために、これらのコマンドを次のように 3 回実行する必要があります。

```
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms1 -groupName xdms -httpPort 8901 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms1 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms2 -groupName xdms -httpPort 8902 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms2 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms3 -groupName xdms -httpPort 8903 -defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms3 -jar jazn.jar -activateadmin myPassword1
```

OC4J インスタンスの構成

前述の手順では、実際の XDMS サーバーがデプロイされる OC4J インスタンスを新たに作成する方法について説明しました。デプロイする前に、`Sip` サブレット・コンテナを取得するようにこれらの新しいインスタンスを構成することも必要です。必要な `jar` は、`User Dispatcher` をインストールしたときにすでに `Oracle Application Server` にインストールされています。このため、実行する必要があるのは、`xdmsX` インスタンスそれぞれが共有ライブラリから `Sip` サブレット・コンテナ `jar` を取得するように構成することだけです。また、適切な起動とロギングのために新しい OC4J インスタンスを構成する必要もあります。すべての新しい OC4J インスタンスに対して実行する必要がある全手順は次のとおりです。

1. `Sip` サブレット・コンテナに必要なすべての `jar` が含まれる共有ライブラリを取得するようにインスタンスを構成します。このためには `boot.xml` ファイルを編集します。
2. ロギングを構成します。XDMS インスタンスで処理されたロギングを新しいインスタンスが使用するためには、`j2ee-logging.xml` ファイルを編集する必要があります。
3. インスタンスの起動と停止のパラメータを編集して、`Sip` サブレット・コンテナがロードされるようにします。このためには `$ORACLE_HOME/opmn.xml` ファイルを編集します。
4. 新しいインスタンスがリスニングする SIP ポートを指定します。
5. `Sip` サブレット・コンテナが適切な IP アドレスをリスニングするように構成します。
6. `xcap config` ディレクトリを追加します。
7. 構成を確認します。

インスタンスの停止 `$ORACLE_HOME/opmn/bin/opmnctl stopall`

共有ライブラリを使用するためのインスタンスの構成 Sip サブレット・コンテナに必要な jar が含まれる共有ライブラリを取得するようにインスタンスを構成します。このためには、OCMS インストーラによって作成された ocms インスタンスから `boot.xml` を次のようにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/boot.xml $ORACLE_HOME/j2ee/<instance name>/config/
```

これで、ocms インスタンスの適切に構成された `boot.xml` が別のインスタンスにコピーされます。この手順を新たに作成したすべてのインスタンスについて繰り返します。サンプル・ネットワークでは、`<instance name>` を `xdms1`、`xdms2` および `xdms3` で置き換えてこのコマンドを 3 回発行し、`boot.xml` を 3 つの OC4J インスタンスにコピーします。

ロギングの構成 共有ライブラリをロードするために `boot.xml` ファイルをコピーしたのと同様に、ロギングのための構成ファイルをコピーすることができます。ocms の `config` ディレクトリにある `j2ee-logging.xml` ファイルをすべてのインスタンスにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/j2ee-logging.xml $ORACLE_HOME/j2ee/
<instance name>/config/
```

サンプル・ネットワークでは、`<instance name>` を `xdms1`、`xdms2` および `xdms3` で置き換えてこのコマンドを 3 回発行し、`j2ee-logging.xml` を 3 つの OC4J インスタンスにコピーします。

JVM の起動および停止パラメータの構成 JVM パラメータを構成して、Sip サブレット・コンテナがロードされるようにインスタンスの起動と停止を設定します。これには、`opmn.xml` ファイルを編集して、すべての Presence の OC4J インスタンス（サンプル・ネットワークでは `xdms1`、`xdms2` および `xdms3`）および ocms インスタンスについて起動および停止のパラメータを設定します。

起動パラメータ：

```
<data id="java-options" value="-server -Xmx2500M -Xms2500M
-Xloggc:/home/sdp/logs/gc.xdms1.log -XX:+PrintGCDetails -XX:NewRatio=3
-XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -Dhttp.maxFileInfoCacheEntries=-1 -Djava.security.policy=$ORACLE_HOME/
j2ee/home/config/java2.policy -Djava.awt.headless=true -Dhttp.webdir.enable=false
-DopmnPingInterval=1
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipser
vletcontainer.deployer.Oc4jApplicationHook "/>
```

停止パラメータ：

```
<data id="java-options" value="-Djava.security.policy=$ORACLE_HOME/
j2ee/home/config/java2.policy -Djava.awt.headless=true -Dhttp.webdir.enable=false
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipser
vletcontainer.deployer.Oc4jApplicationHook"/>
```

これらのパラメータの意味は次のとおりです。

-Xmx2500M: 最大 JVM メモリーを 2.5GB に設定します。

-Xms2500M: 最小 JVM メモリーを 2.5GB に設定します。

-XX:+PrintGCDetails: コレクション・アクティビティのロギングを有効にします。

-XX:NewRatio=3: ヒープの新世界と旧世界の比率を 1:3 に設定します（つまり、エデン領域とサブイバ領域の合計サイズが合計ヒープ・サイズの 4 分の 1 になります）。

-XX:+UseConcMarkSweepGC: 同時マークアンドスイープ・ガベージ・コレクタ（同時ロー・ポーズ・コレクタとも呼ばれる）を有効にします。

-XX:+UseParNewGC: パラレル・スレッドを使用します。

-XX:PermSize=128m: 永続世代の初期サイズを 128MB に設定します。

-XX:MaxPermSize=128m: 永続世代の最大サイズを 128MB に設定します。

-Xss128k: 各スレッドの stat サイズを 128KB に設定します。

-Dhttp.maxFileCacheEntries=-1: Oracle Application Server にバンドルされている HTTP サーバーでのキャッシングを無効にします。このリリースでは、Presence サーバーの良好なパフォーマンスを得るためにキャッシングを無効にする必要があります。

-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipservletcontainer.deployer.Oc4jApplicationHook: OC4J で Sip コンテナをロードできるようにします。

注意: インストールした ocms インスタンスに、複数の JVM インストールに必要なパラメータがすべて含まれない場合、前述のパラメータを ocms インスタンスにも設定するようにしてください。

SIP ポートの構成 起動および停止パラメータの設定が終了したので、次に Sip サブレット・コンテナが正しいポートでリスニングするように構成する必要があります。デフォルトでは、Sip サブレット・コンテナは、SIP をポート 5060、SIPS を 5061 でリスニングします。ファイル \$ORACLE_HOME/opmn/conf/opmn.xml では、OCMS インストーラによって作成された ocms インスタンスが次のようにこれらのデフォルト値で構成されています。

```
<port id="sip" range="5060"/>
<port id="sips" range="5061"/>
```

ocms インスタンスのこれらの値はそのままにしておく必要があります。User Dispatcher はこれらのポートで受信トラフィックをリスニングするためです。Presence ノードのために作成した他の OC4J インスタンスについては、ポート競合を避けるためにそれぞれに別のポートを構成する必要があります。管理しやすくするために、連続した使用可能ポートを使用することをお勧めします。サンプル・ネットワークでは、Presence インスタンスに 5062 から始まるポートを割り当てています。構成は次のようになります。

ps1

```
<port id="sip" range="5062"/>
<port id="sips" range="5063"/>
```

ps2

```
<port id="sip" range="5064"/>
<port id="sips" range="5065"/>
```

ps3

```
<port id="sip" range="5066"/>
<port id="sips" range="5067"/>
```

適切な IP アドレスでリスニングするための Sip サーブレット・コンテナの構成 デフォルトでは、Sip サーブレット・コンテナは IP アドレス 127.0.0.1 でリスニングします。このアドレスを、このマシンの外部からアドレス指定できる IP アドレスに変更する必要があります。前述の手順で、コンテナをロードするように Oracle Application Server を構成しましたが、実際の Sip サーブレット・コンテナの構成ファイルは取得していません。デフォルトの構成ファイルを生成するには、Oracle Application Server を起動してから停止する必要があります。デフォルトの構成ファイルが生成されたら、デプロイに合わせてファイルを編集することができます。

Oracle Application Server を起動するには次のようにします。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

起動して実行したら、次のように停止します。

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

sdp という名前の新しい構成ディレクトリが、\$ORACLE_HOME/j2ee/<instance name>/config/sdp/ に作成されています。各インスタンスの OCMS の構成に関連するすべての情報がここにあります。各インスタンスには SipServletContainer.xml という名前のファイルがあります。各インスタンスについてこのファイルを編集し、IPAddress 属性を 127.0.0.1 から Sip サーブレット・コンテナがリスニングする IP アドレスに変更する必要があります。ただし、このファイルはこのマシンのすべての Sip サーブレット・コンテナで同じにする必要があるため、それぞれを手動で編集する必要はありません。ocms ディレクトリにあるファイルをコピーして再利用します。コマンドは次のとおりです。

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/SipServletContainer.xml $ORACLE_HOME/j2ee/
<instance name>/config/sdp/
```

<instance-name> を Presence の OC4J インスタンスの名前で置き換えます。サンプル・ネットワークでは、このコマンドを 3 回 (xdms1、xdms2 および xdms3 に対して 1 回ずつ) 実行します。これで Sip サーブレット・コンテナがすべてのインスタンスに対して構成されたため、サーバーを起動することができます。Oracle Application Server ですべての OC4J インスタンスを起動するには、次のようにします。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

xcap 構成ディレクトリの構成 ua-profile イベント・パッケージは Presence ノードで明示的には使用されませんが、存在するように構成する必要があります。このため、xcap 構成ファイルをすべての Presence OC4J インスタンスにコピーしてください。これを行うには、xcap ディレクトリを各インスタンスの sdp 構成ディレクトリ \$ORACLE_HOME/j2ee/<instance-name>/config/sdp/ にコピーします。サンプル・ネットワークでは、xcap ディレクトリを次のディレクトリにコピーする必要があります。

```
$ORACLE_HOME/j2ee/xdms1/config/sdp/
$ORACLE_HOME/j2ee/xdms2/config/sdp/
$ORACLE_HOME/j2ee/xdms3/config/sdp/
```

xcap ディレクトリは、OCMS インストーラを抽出したディレクトリにあります。たとえば、インストーラをディレクトリ OCMS_INSTALLER に抽出した場合、xcap ディレクトリは次の場所にあります。

```
OCMS_
INSTALLER/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/DataFiles/Expanded/Shiphome/
shiphome-archive/xcapconf/conf
```

Aggregation Proxy の構成 各 XDMS ノードの Aggregation Proxy が、同じノード上の User Dispatcher を指すように構成する必要があります。次の手順に従って Aggregation Proxy を構成します。

1. 管理ノードの Oracle Enterprise Manager コンソールにログオンし、Aggregation Proxy Mbean に移動します。
2. XCAPRoot が同じノードの User Dispatcher のコンテキスト・ルートになるように変更します。User Dispatcher コンテキスト・ノードのデフォルト値は userdispatcher です。これを変更していないかぎり、XCAPRoot 属性の値を /userdispatcher に設定します。XCAPHost や XCAPPport など他の属性を変更する必要はありません。これらはインストーラによって適切に設定されます。

springbeans.xml の構成 次のように、springbeans.xml を ocms インスタンスの sdp 構成ディレクトリからすべての Presence インスタンスの sdp 構成ディレクトリにコピーします。

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/springbeans.xml $ORACLE_HOME/j2ee/<instance name>/config/sdp/
```

サンプル・ネットワークでは、<instance-name> を xdms1、xdms2 および xdms3 でそれぞれ置き換えて、このコマンドを 3 回実行します。

構成設定の確認 インストールを続行する前に、ログを調べてこれまでの操作が正常に処理されたことを確認します。また、netstat コマンドを発行して、サーバーが正しいポートでリスニングしていることも確認できます。

ログ・ファイル: ログ・ファイルを定期的に監視することも必要です。構成に問題がある場合にはログ・ファイルによって明らかになります。確認する必要がある主なログ・ファイルは 2 つです。1 つ目は、\$ORACLE_HOME/opmn/logs にある各インスタンスのコンソール出力です。通常、xdms ノードの場合は次のようなファイル名になります。

```
xdms~<instance-name>-default_group~1.log
```

ocms インスタンスに対するファイルと、opmn そのもののログ・ファイルもあります。常にこれらのファイルをよく注意してください。異常が発生する大半のケースでは、これらのログ・ファイルによって潜在的な問題に関する十分な情報が得られるためです。

もう 1 つの重要なログ・ファイルは、Sip サブレット・コンテナそのものによって生成されるログです。各インスタンス（たとえば xdms1）には sdp ログ・ディレクトリがあり、trace.log ファイルが含まれます。このファイルにもよく注意してください。このファイルは ocms インスタンスに対しても存在します。すべてのインスタンスのすべての trace.log ファイルを確認するようにしてください。

opmnctl: opmnctl ツールは、サーバーの起動と停止に使用されます。このツールでは、実行中のインスタンスのステータスも表示されます。基本コマンドは次のとおりです。

```
$ORACLE_HOME/opmn/bin/opmnctl status
```

すべてのプロセスと、それらが実行しているかどうかが表示されます。一般的な出力を次に示します。

```
Processes in Instance: priv5.priv5
-----
ias-component | process-type | pid | status
-----
OC4JGroup:xdms | OC4J:xdms3 | 4868 | Alive
OC4JGroup:xdms | OC4J:xdms2 | 4869 | Alive
OC4JGroup:xdms | OC4J:xdms1 | 4867 | Alive
OC4JGroup:default_group | OC4J:ocms | 4866 | Alive
OC4JGroup:default_group | OC4J:home | 4865 | Alive
ASG | ASG | N/A | Down
```

この出力は、xdms1 ~ xdms3 がこのマシンで実行していることを示します。ocms インスタンスも起動して実行しています。User Dispatcher がこのインスタンスで実行されていることに注意してください。opmnctl を使用して、各インスタンスがリスニングしているすべてのポートを表示することもできます。このためには、次の例のようにスイッチ -l を opmnctl に指定します。

```
$ORACLE_HOME/opmn/bin/opmnctl status -l
```

これにより、表 5-1 のような出力が表示されます。この表の出力は一部省略されています。通常は、プロセス ID、起動時刻、その他の情報が含まれます。

すべての構成が適切な場合は、ocms が SIP ポート 5060、xdms1 が 5062 などのようにリスニングしていることが表示されます。

表 5-3 プロセス・タイプ別の使用ポート

プロセス・タイプ	ポート
OC4J:xdms3	jms:12605, sip:5066, http:8903, rmis:12705, sip:5066, sips:5067, rmi:12405
OC4J:xdms2	jms:12604, sip:5064, http:8902, rmis:12704, sip:5064, sips:5065, rmi:12404
OC4J:xdms1	jms:12603, sip:5062, http:8901, rmis:12703, sip:5062, sips:5063, rmi:12403
OC4J:ocms	jms:12602, sip:5060, http:7785, rmis:12702, sip:5060, sips:5061, rmi:12402
OC4J:home	jms:12601, http:8888, rmis:12701, rmi:12401

接続プールの作成 XDMS インスタンスの接続プールとデータソースを作成します。ユーザー・プレゼンス・ルールと PIDF ドキュメントは Oracle Database (サンプル・ネットワークでは 192.168.0.30) に格納されていることに注意してください。これらのドキュメントにアクセスできるように XDMS ノードを構成する必要があります。次のプロパティを使用して接続プールとデータソースを設定します。

- 接続プール名: SDP XDMS Oracle Connection Pool
- コネクション・ファクトリ・クラス: oracle.jdbc.pool.OracleDataSource
- URL: jdbc:oracle:thin: @//<db-hostname>:<db-port>/<db-name>
- OC4J ユーザー名: oc4jadmin
- OC4J パスワード: myPassword1
- データベース・ユーザー名: SDP_ORASDPXDMS
- データベース・パスワード: myDBPassword1
- データソース名: OcmsXdmsDs

ユーザー名とパスワードは、それらの資格証明で Oracle Database にアクセスできるものであればデプロイに合わせて変更できます。接続プールとデータソースを作成するには、管理ノードで次のコマンドを実行します。

```
java -jar ORACLE_HOME/j2ee/home/admin_client.jar
deployer:clusterj:opmn://<management-host-ip-address> <oc4j-username> <admin_pwd>
-addDataSourceConnectionPool -applicationName <name> -name <connection pool name> -name
<connection pool name> -factoryClass <factory class> -dbPassword <db password> -dbUser
<db username> -url <url>
```

```
java -jar ORACLE_HOME/j2ee/home/admin_client.jar
deployer:cluster:opmn://<management-host-ip-address>/xdms <oc4j-username> <password>
-addManagedDataSource -applicationName default -name "<data-source-name>" -jndiLocation
java:jdbc/<data-source-name> -connectionPoolName "<connection-pool-name>"
```

サンプル・ネットワークでは、Oracle Database はホスト 192.168.0.30 のポート 1521 にあります。データベース名は orcl11g、管理ノードは 192.168.0.100 にあります。サンプル・ネットワークを前述のように構成する場合は、次のコマンドを管理ノードで実行して、XDMS の JDBC データソースを作成します。

```
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -addDataSourceConnectionPool -applicationName default -name "SDP XDMS
Oracle Connection Pool" -factoryClass oracle.jdbc.pool.OracleDataSource -dbUser
SDP_ORASDPXDMS -dbPassword myDBPassword1 -url
jdbc:oracle:thin:@//192.168.0.30:1521/orcl11g
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -addManagedDataSource -applicationName default -name "OcmsXdmsDs"
-jndiLocation java:jdbc/OcmsXdmsDs -connectionPoolName "SDP XDMS Oracle Connection
Pool"
```

データソースまたは接続プール（あるいは両方）を後から削除する必要がある場合は、次のコマンドを発行します。

```
java -jar admin_client.jar deployer:cluster:opmn://<management-host-ip-address>/
xdms <oc4j-username> <oc4j-password> -removeManagedDataSource -name
"<data-source-name>"
java -jar admin_client.jar deployer:cluster:opmn://<management-host-ip-address>/
xdms <oc4j-username> <oc4j-password> -removeDataSourceConnectionPool -name
"<connection-pool-name>"
```

サンプル・ネットワークでは、データソースと接続プールを削除するためのコマンドはそれぞれ次のようになります。

```
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -removeManagedDataSource -name "OcmsXdmsDs"
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -removeDataSourceConnectionPool -name "SDP XDMS OracleConnection Pool"
```

XDMS のデプロイおよび構成

Oracle Enterprise Manager でグループ・ビューを使用し、Presence アプリケーションの .ear ファイルをグループのすべてのインスタンスにデプロイします（Presence と XDMS の両方で同じ .ear ファイルを使用します）。デプロイ時にはアプリケーションの名前として xdms を使用することをお勧めします。デプロイが終了したら、すべてのインスタンスで次の項目を構成する必要があります。

- PublicXCAPRootUrl および PublicContentServerRootUrl の変更
- UserAgentFactoryServiceImpl.xml の更新
- JGroups の有効化

PublicXCAPRootUrl および PublicContentServerRootUrl の変更 各 OC4J XDMS ノード・インスタンスについて、「XDMS」→「XCAPConfig」アプリケーション MBean に移動し、次の属性の値を設定します。

- PublicXCAPRootUrl - http://<node-ip>:<node-http-port>/services
- PublicContentServerRootUrl - http://<node-ip>:<node-http-port>/contentserver

ここで、node-ip は XDMS インスタンスが存在するマシンの IP アドレス、node-http-port は XDMS サーバーが実行している OC4J インスタンスの HTTP ポートです。これは、OC4J インスタンスを createinstance コマンドを使用して作成するときに、-httpPort XXXX オプションを使用して指定する値です。

サンプル・ネットワークでは、1つの XDMS ノードに3つの XDMS インスタンスがあります。このため、ホスト 192.168.0.20 の XDMS インスタンスについては次のように値を設定します。

xdms1:

```
PublicXCAPRootUrl - http://192.168.0.20:8091/sevices
PublicContentServerRootUrl - http://192.168.0.20:8091/contentserver
```

xdms2:

```
PublicXCAPRootUrl - http://192.168.0.20:8092/sevices
PublicContentServerRootUrl - http://192.168.0.20:8092/contentserver
```

xdms3:

```
PublicXCAPRootUrl - http://192.168.0.20:8093/sevices
PublicContentServerRootUrl - http://192.168.0.20:8093/contentserver
```

サンプル・ネットワークにおいて 192.168.0.21 にある 2 番目の XDMS ノード (マシン) の設定は、前の例と似ていますが、属性の IP アドレスが 192.168.0.20 ではなく 192.168.0.21 になります。

UserAgentFactoryService ポートの更新 各 OC4J XDMS ノード・インスタンスについて、XDMS アプリケーションの `UserAgentFactoryService MBean` に移動し、ポートの競合を避けるために、マシン上の XDMS インスタンスそれぞれについて `Port` 属性の値を一意に設定します。管理しやすくするために、連続した使用可能ポートを使用することをお勧めします。サンプル・ネットワークでは、インスタンス `xdms1`、`xdms2` および `xdms3` に対して 5070、5071 および 5072 をそれぞれ使用しています。

JGroups の有効化 各 OC4J XDMS ノード・インスタンスについて、`PackageManager MBean` に移動し、`JGroupsBroadcastEnabled - true` のように属性の値を設定します。

XDMS サーバーと Presence サーバーが同じポートおよびアドレスで JGroups の通知をリスニングしないようにします。このようにしないと、エラーが発生します。Presence サーバーではデフォルトの JGroups 構成を使用したため、XDMS サーバーでは異なる構成を使用する必要があります。このためには、XDMS サーバー・パッケージ・マネージャが認識するように、JGroups 構成ファイルへのパスを指定します。最小の構成ファイルの例を次に示します。

```
<config>
  <UDP bind_addr="[host-ip-address]" mcast_addr="[multicast-address]"
mcast_port="[multicast-port]" ip_ttl="1"/>
</config>
```

replacing the variables as follows:

[host-ip-address] - the IP address of the host running the XDMS server

[multicast-address] - the multicast address on which all participants in the group will listen for messages.

[multicast-port] - the multicast port on which all participants in the group will listen for messages.

すべてのノードのすべての XDMS サーバー・インスタンスで、`multicast-address` および `multicast-port` の値が同じになるようにしてください。また、通常、同一ホスト上の XDMS サーバーの `host-ip-address` は同じになります。ただし、XDMS サーバーの `multicast-address:multicast-port` の組合せは、Presence サーバーとは異なる必要があります。Presence サーバーでは、デフォルト構成値 (230.0.0.1:7426) を `multicast-host:multicast-port` で使用したことに注意してください。場所を選択してこのファイルを保存し、`EventPackageManager Mbean` の `JGroupXMLConfigPath` 属性がこのファイルを指すように編集します。

`JGroupXMLConfigPath` - <absolute path to the jgroups XML configuration file>

サンプル・ネットワークでは、192.168.0.20 のホストで次の JGroups 構成ファイルを使用します。

```
<config>
  <UDP bind_addr="192.168.0.20" mcast_addr="234.0.0.1" mcast_port="1234" ip_ttl="1"/>
</config>
```

このノード上のすべての XDMS インスタンスの構成は同じです。したがって、このファイルを \$ORACLE_HOME/j2ee/ocms/config/sdp/jgroups.xml に保存し、各 XDMS インスタンス xdms1、xdms2 および xdms3 について EventPackageManager Mbean を次の設定になるように編集します。

```
JGroupBroadcastEnabled - true
JGroupXMLConfigPath - <path-to-ORACLE_HOME>/j2ee/ocms/config/sdp/jgroups.xml
```

192.168.0.21 の XDMS ホストも、ホスト IP アドレスを適切に置き換えて同じ方法で構成します。

データベースの構成 次の構成手順を実行します。

1. orasdpxdms.create.oracle.sql および xcapservice.create.oracle.sql ファイルをデータベースが実行しているマシンにコピーします。これらのファイルは、<installer files extraction location>/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/DataFiles/Expanded/DBFiles/ にあります。
2. データベースが実行しているマシンにログインし、sql ファイルをコピーしたディレクトリに移動してください。
3. sqlplus を使用して sysdba としてデータベースに接続します。これは、bash\$ sqlplus / as sysdba のように実行します。
4. 前述の sql ファイルを次のように sqlplus プロンプトで実行します。

```
sqlplus> @orasdpxdms.create.oracle.sql PREFIX DATADIR PASSWORD
```

ここで、PREFIX はスキーマとユーザーに使用する接頭辞、DATADIR は作成したデータベース・ファイルが存在するパス、PASSWORD は作成されるユーザーのパスワードです。

次に例を示します。

```
sqlplus> @orasdpxdms.create.oracle.sql TEST "C:¥oraexe¥oradata¥XE" myPassword1
```

これにより、データベース・ファイルが C:¥oraexe¥oradata¥XE に作成されます。スキーマ名とユーザー名は TEST で始まるようになり、ユーザーのパスワードは myPassword1 になります。

5. もう 1 つの sql ファイルを次のように実行します。

```
sqlplus> @xcapservice.create.oracle.sql PREFIX
```

この場合の PREFIX は、orasdpxdms.create.oracle.sql ファイルの実行時の指定と同じにする必要があります。

sash の構成 次の構成手順を実行します。

1. `$ORACLE_HOME/sdp/sash/sbin` に移動します。
2. ファイル `xdms-create-default-appusage.txt` を作成します。
3. このファイルに次の行を追加します。

```
xcap appusage create applicationUsage=pres-rules configurationFilename=
presrules_au.xml
```

```
xcap appusage create applicationUsage=resource-lists
configurationFilename=resource-lists_au.xml
```

```
xcap appusage create applicationUsage=pidf-manipulation
configurationFilename=pidfmanipulation_au.xml
```

4. デフォルトのアプリケーション使用でデータベースをシードするには、次のコマンドを実行します。

```
- bash$ sash -a presenceapplication --username oc4jadmin --password PASSWORD --file
xdms-create-default-appusage.txt
```

ここで、PASSWORD は oc4jadmin のパスワードです。

確認 次のように sash にログオンします。

```
bash$ sash -a presenceapplication --username oc4jadmin --password PASSWORD
```

sash プロンプトで次のように入力します。

```
sash# xcap appusage list
```

構成が正しい場合は、これによって `resource-lists`、`pidf-manipulation` および `pres-rules` という 3 つの値が返されます。

User Dispatcher の構成

デプロイ内のすべての XDMS インスタンスに SIP トラフィックをルーティングできるように User Dispatcher を構成します。User Dispatcher が存在するマシン上のすべての Presence インスタンスと、デプロイ内の他のマシンのすべての XDMS インスタンスに SIP トラフィックを送るように、すべての User Dispatcher を構成する必要があります。つまり、各 XDMS ノード (XDMS マシン) の User Dispatcher は、他のノードのすべての XDMS インスタンスを認識する必要があります。User Dispatcher が SIP トラフィックをすべての XDMS サーバーにルーティングできるように構成するには、次の手順を実行します。

1. 管理ノードの Enterprise Manager にログインします。
2. クラスタ・ビューで、構成する User Dispatcher が存在する XDMS ノードを選択します。
3. 「アプリケーションの選択」 → 「userdispatcher アプリケーション定義の MBeans」を選択します。
4. 「xdms-sip-pool」をクリックし、「サーバー」を選択します。
5. デプロイ内のすべての XDMS サーバーを指す SIP URI を追加します。URI の形式は次のとおりです。

```
sip:<ip-address>:<port>;transport=tcp;lr
```

サンプル・ネットワークには2つのXDMSノード(マシン)があり、それぞれに3つのXDMSサーバーがあります。XDMSサーバーは合計6つです。各User DispatcherがSIPトラフィックを6つのXDMSサーバーにルーティングするように構成する必要があります。各User Dispatcherのxdms sipプールに次の内容を追加します。

```

sip:192.168.0.20:5062;transport=tcp;lr
sip:192.168.0.20:5064;transport=tcp;lr
sip:192.168.0.20:5066;transport=tcp;lr
sip:192.168.0.21:5062;transport=tcp;lr
sip:192.168.0.21:5064;transport=tcp;lr
sip:192.168.0.21:5066;transport=tcp;lr

```

6. 「xdms-http-pool」をクリックし、「サーバー」を選択します。
7. デプロイ内のすべてのXDMSインスタンスを指すHTTP URIを追加します。URIの形式は次のとおりです。

```
http://<ip-address>:<port>/services
```

サンプル・ネットワークには2つのXDMSノード(マシン)があり、それぞれに3つのXDMSサーバーがあります。XDMSサーバーは合計6つです。各User DispatcherがHTTPトラフィックを6つのXDMSサーバーにルーティングするように構成する必要があります。各User Dispatcherのxdms sipプールに次の内容を追加します。

```

http://192.168.0.20:8901/services/
http://192.168.0.20:8902/services/
http://192.168.0.20:8903/services/
http://192.168.0.21:8901/services/
http://192.168.0.21:8902/services/
http://192.168.0.21:8903/services/

```

インストールのチューニング

ここでは、インストールしたコンポーネント同士が共存して適切に稼働できるように、チューニングする方法について説明します。

オーバーロード・ポリシーの更新 User Dispatcherを除くすべてのPresenceインスタンスとXDMSインスタンスについて、OverloadPolicy.xmlのSipSessionTableMaxSize属性を400000に変更します。OverloadPolicy.xmlは\$ORACLE_HOME/j2ee/<instance-name>/config/sdpにあります。

WebCenter インスタンスの無効化 インストールが終了したら、不要にリソースを消費しないようにWebCenterインスタンスを無効にします。これには、\$ORACLE_HOME/opmn/conf/にあるopmn.xmlファイルを編集します。サンプルの<process-type id="OC4J_WebCenter" module-id="OC4J" status="disabled">のように、ステータスをenabledからdisabledに変更します。変更を有効にするために、次のようにサーバーを再起動します。

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

停止した後で再び起動します。

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

ホーム・インスタンスの無効化 すべてのインストールが終了したら、管理ノードを除くすべてのノードでホームOC4Jインスタンスを停止します。管理ノードは、Enterprise Managerコンソールにログインして、デプロイ全体の構成を表示または変更するために使用します。ホーム・インスタンスを停止するには、すべてのPresenceノードのopmn.xmlファイルを編集して、WebCenterインスタンスの場合と同じ方法でホーム・インスタンスを無効にします。ホームが必要になるのは、createinstanceコマンドを使用して新しいインスタンスを作成するときのみです。

ASG インスタンスの無効化 パフォーマンスを向上させるためにASGインスタンスを停止します。

ロード・バランサの構成

外部からの観点では、ネットワーク全体は1つのノードのように見えます。これは、1つ以上のロード・バランサを Presence クラスタと XDM クラスタの前面に配置することで実現します。ここでは、サンプル・ネットワークを使用して、F5 の BigIP ロード・バランサを設定する方法について説明します。

全体的な視点で考えると、SIP トラフィック用に2つのプール、XCAP トラフィック用に1つのプールを作成する必要があります。1つの SIP プールにはシステムのすべての Presence ノードが含まれ、もう1つのプールにはすべての XDM ノードが含まれます。XCAP トラフィックのプールには XDM ノードのリストのみが含まれます。XDM ノードのみが XCAP トラフィックを処理するためです。

外部クライアントはこれらのプールには直接接続せず、仮想サーバーに接続します。この仮想サーバーには特定のプールが含まれます。外部クライアントが認識するのはこの仮想サーバーであり、これによって、システム全体が1つのマシンのように見えます。

BigIP のすべての構成は、Web ベースの管理インターフェースで行います。機能や構成オプションの詳細は、BigIP のドキュメントを参照してください。

新しいプールの作成

新しいプールを作成するには、「Local Traffic」→「Virtual Servers」→「Pools」を選択して「Pool」ページにナビゲートします。新しいプールを作成する一般的な手順は次のとおりです。

1. 構成は「Basic」のまま、プールの適切な名前を選択して、作成を開始します。
2. 状態モニターについては gateway_icmp を選択します。
3. 「Round Robin」がこの場合のニーズに合うロード・バランシング・メソッドです。これを選択します。
4. 「Priority Group Activation」は無効のままにします。
5. 「New Members」では、この特定のプールに含めるすべてのノードを追加します。ノードごとに、アドレスとポートを入力して「Add」をクリックします。この特定のプールに含めるすべてのノードについてこの操作を繰り返します。

サンプル・ネットワークを使用する場合は、最初のプール、すなわち Presence 用の SIP プールを作成するために次の情報を入力します。

1. 任意の名前を指定できます。このサンプルでは、このプールの名前として ps_sip を使用します。
2. 状態モニターについては gateway_icmp を選択します。
3. 「Round Robin」がこの場合のニーズに合うロード・バランシング・メソッドです。これを選択します。
4. 「Priority Group Activation」は無効のままにします。
5. このプールに追加するメンバーは Presence ノードすべてです。サンプル・ネットワークには3つの Presence ノードがあり、このプールはこれらの3つのインスタンスを指す必要があります。実際の Presence インスタンスに対応するのは User Dispatcher であるため、実際にはロード・バランサが User Dispatcher を指すようにします。したがって、次の3つのアドレスをこのプールのメンバーとして追加します。
 - IP アドレス: 192.168.0.10、ポート: 5060
 - IP アドレス: 192.168.0.11、ポート: 5060
 - IP アドレス: 192.168.0.12、ポート: 5060

これで、Presence のプールが作成されました。同じ方法で、XDM ネットワークに送信される SIP トラフィックのためにもう 1 つのプールを次のように作成できます。

1. 名前 : `xdms_sip`
2. 前述の手順と同様にします。
3. 前述の手順と同様にします。
4. 前述の手順と同様にします。
5. メンバーは、サンプル・ネットワークの 2 つの XDMS ノードです。Presence のプールと同様に、このプールがそれらの XDMS ノード上で実行している User Dispatcher を指すようにします。したがって、次の 2 つのメンバーがこのノードに追加されます。

IP アドレス : 192.168.0.20、ポート : 5060

IP アドレス : 192.168.0.21、ポート : 5060

XCAP トラフィックのプールは他の 2 つのプールと同じです。このプールは、SIP トラフィックではなく HTTP トラフィック (XCAP は HTTP 上で送信される) をディスパッチしますが、概念は同じです。

1. 名前 : `xdms_http`
2. 前述の手順と同様にします。
3. 前述の手順と同様にします。
4. 前述の手順と同様にします。
5. メンバーはここでも 2 つの XDMS ノードですが、XCAP トラフィックは認証の必要があることに注意してください。このため、このトラフィックは Aggregation Proxy を介して送信する必要があります。したがって、これらのメンバーは User Dispatcher を指すのではなく、Aggregation Proxy がリスニングする HTTP ポートを指すようにします。サンプル・ネットワークの場合は次のアドレスを追加します。

IP アドレス : 192.168.0.20、ポート : 80

IP アドレス : 192.168.0.21、ポート : 80

これでサンプル・ネットワークに必要な 3 つのプールが作成されました。次の手順は、各プールに対応する仮想サーバーの構成です。

新しい仮想サーバーの作成

仮想サーバー (VS) は外部クライアントが操作するためのものです。クライアントが特定の VS に接続すると、その VS がリクエストをいずれかのプールにプロキシ処理し、プールがメンバーの 1 つにリクエストをディスパッチします。サンプル・ネットワークでは、3 つのプールそれぞれに対応する VS を 1 つずつ作成します。新しい VS を作成する一般的な手順を次に示します。仮想サーバーは「Local Traffic」に表示されます。

1. 「Create」をクリックして、仮想サーバーの作成を開始します。
2. この VS は TCP を使用するため、適当な名前として `ps_sip_tcp` と付けます。
3. サンプル・ネットワークでは、ロード・バランサのインタフェースは 1 つのみであり (少なくとも 1 つだけは有効にし、その他は不要)、その 1 つのインタフェースが 192.168.0.150 でリスニングします。宛先アドレスにはこのアドレスを入力します。
4. ポートは使用可能な任意のポートを使用できますが、このサンプルでは 5060 を使用します。
5. この VS は TCP を実行するため、そのように選択します。
6. 「Auto Map for the SNAT Pool」を有効にします。
7. 「Resource」には、Presence ノードを含むプールを指定します。サンプル・ネットワークでは、このプールの名前は `ps_sip` で、デフォルトのプールです。この VS が使用するのはこのプールのみです。

8. 「**Finish**」をクリックし、新しい仮想サーバーを作成します。

次に作成する VS は、XDM クラスタに送られる SIP トラフィックに対する VS です。

1. 「**Create**」をクリックして、仮想サーバーの作成を開始します。
2. 名前は `xdms_sip_tcp` とします。
3. 宛先アドレスは `192.168.0.150` です。
4. `5060` は使用中であるため、`5062` を選択します。
5. この VS は TCP を実行するため、そのように選択します。
6. 「**Auto Map for the SNAT Pool**」を有効にします。
7. 「**Resource**」には、`xdms_sip` という名前のプールを指定します。
8. 「**Finish**」をクリックします。

仮想サーバー (VS) は外部クライアントが操作するためのものです。クライアントが特定の VS に接続すると、その VS がリクエストをいずれかのプールにプロキシ処理し、プールがメンバーの 1 つにリクエストをディスパッチします。サンプル・ネットワークでは、3 つのプールそれぞれに対応する VS を 1 つずつ作成します。新しい VS を作成する一般的な手順を次に示します。仮想サーバーは「**Local Traffic**」に表示されます。

1. 「**Create**」をクリックして、仮想サーバーの作成を開始します。
2. 名前は `xdms_http_tcp` とします。
3. 宛先アドレスは `192.168.0.150` です。
4. ポートは `80` です。
5. プロトコルは TCP です。
6. 「**Auto Map for the SNAT Pool**」を有効にします。
7. 「**Resource**」には、`xdms_http` という名前のプールを指定します。
8. 「**Finish**」をクリックします。

これで、F5 BipIP ロード・バランサの構成が完了します。様々なオプションの詳細は、BigIP のドキュメントを参照してください。

6

インストール後の作業

この章では、インストールと構成に関する一般的な問題とその解決方法を説明します。次の項目が含まれています。

- [インストール後の管理タスクの実行](#)
- [データベースのチューニング](#)

インストール後の管理タスクの実行

OCMS をインストールし、Oracle Communicator を使用して OCMS SIP サーバー・コンテナに接続して、インストールを確認しました。OCMS のインストールはこれで完了です。ただし、アプリケーションを OCMS にデプロイする前に、OCMS 管理者が次の管理タスクを実行します。

- SIP サーブレット・コンテナと Application Router を構成します。『Oracle Communication and Mobility Server 管理者ガイド』の「SIP サーブレット・コンテナの構成」を参照してください。
- Presence、Proxy-Registrar、Edge Proxy および Aggregation Proxy アプリケーションをインストールした場合は、必要な構成を実行します。『Oracle Communication and Mobility Server 管理者ガイド』の「SIP アプリケーションの管理」の SIP アプリケーションの構成に関する項を参照してください。
- Sapphire Shell (Sash) コマンドライン・ユーティリティを使用して、Oracle データベースに OCMS ユーザーをプロビジョニングします。『Oracle Communication and Mobility Server 管理者ガイド』の「ユーザーとアプリケーションのプロビジョニング」の章を参照してください。
- SIP アプリケーションでの認可および認証の実行方法と SIP サーブレットのセキュリティの設定方法を構成します。『Oracle Communication and Mobility Server 管理者ガイド』の「OCMS のセキュリティ」の章を参照してください。
- OCMS のロギングを構成します。『Oracle Communication and Mobility Server 管理者ガイド』のロギングに関する章を参照してください。
- 本番環境に OCMS をインストールするときは、SIP ロード・トラフィックと同期データを分離させて、互いに干渉しないようにするために、少なくとも 2 枚のネットワーク・カードの使用を検討してください。
- データベースのチューニングの詳細は、データベースのドキュメントを参照してください。

データベースのチューニング

Oracle Communication and Mobility Server をそのまま実行する場合は、データベースの特別なチューニングは必要ありません。ただし、OCMS を本番環境にデプロイする前には次の操作を行うことをお勧めします。

- OCMS からデータベースへの接続数を増やします。一般的なインストールでは、Subscriber Data Services (<prefix>_oraspsds)、Location Service (<prefix>_oraspls) および XDMS (<prefix>_oraspxdms) に 1 つずつ、3 つのスキーマ・ユーザーが OCMS によって作成されます。そのまま使用する場合、データベース接続の最小数と最大数は、これらの各スキーマで 4 に構成されます。
- データベース・サーバーのプロセス数を増やします。これは、前述の指示に従って OCMS のデータベース接続数を増やす場合や、データベースに他のアプリケーションも接続している場合に特に重要です。
- SGA および PGA キャッシュ・サイズを増やします。PGA および SGA サイズを増やすと、キャッシュに格納できるデータの容量が増加します。
- 表領域が不足する場合は各スキーマにデータベース・ファイルを追加します。そのまま実行する場合、構成される各スキーマに対して最大サイズ 250MB のデータベース・ファイルが 1 つずつしか作成されません。
- データベースのチューニング方法の詳細は、データベースのドキュメントを参照してください。

数字

3GPP IP Multimedia System, 1-2

A

Aggregation Proxy, 1-5, 5-5
Application Router, 1-5
Application Server Control, 5-9

E

Edge Proxy, 1-3, 1-5, 2-2
 インストール, 2-3
Ethereal, 3-11

H

HTTPS, 3-4

I

IMS, 1-2

J

JDK 1.5, 1-3
JVM, 5-13

M

MBean ブラウザ, 1-3

N

netstat コマンド, 1-3

O

OC4J

 Admin, 2-6
 HTTP ポート, 2-12
 構成, 5-12

OCMS, 1-1

 Aggregation Proxy, 1-5
 Application Router, 1-5
 Edge Proxy, 1-5
 OCMS の起動, 2-15

 OCMS の停止, 2-15

 Oracle Application Server モード, 1-4
 Presence, 1-5

 Presence Web Services, 1-5

 Proxy Registrar, 1-4

 SIP から PSTN への通話, 3-10

 SIP コンテナの構成, 2-12

 SIP サーブレット・コンテナ, 1-4

 STUN サーバー, 1-5

 Subscriber Data Services, 1-4

 インストール, 2-1, 2-3

 インストールの確認, 2-14, 3-1

 インストール・モード, 1-3

 コンポーネント, 1-4

 サーブレットの登録の確認, 3-6

 システム要件, 1-2

 スタンドアロン開発者モード, 1-4

 テスト・ユーザーの構成, 2-10

 ネットワーク・トラフィックの監視, 3-11

 ポート要件, 1-3

OCMS インストールの確認, 2-14

OCMS の確認, 3-1

OCMS の起動, 2-15

OCMS の停止, 2-15

OPMN, 1-4

opmnctl status コマンド, 2-14

Oracle Application Server Control, 5-9

Oracle Application Server モード, 1-4

Oracle Communicator, 3-2

 XDMS 設定, 3-4

 イベント通知, 3-10

 インストール, 3-3

 プレゼンス公開のテスト, 3-9

 プロキシ・サーバーの構成, 3-4

 連絡先の追加, 3-8

Oracle Enterprise Manager

 ポート番号, 2-14

Oracle Process Manager and Notification Server, 1-4

Oracle RAC データベース, 4-1

Oracle Remote Method Invocation ポート, 2-14

Oracle RMI (ORMI) プロトコル, 2-14

P

Presence, 1-5

 Presence クラスタ, 5-2

 Presence サーバー (PS), 5-2

 Presence ノード, 5-2

テスト, 3-8
マルチノード・トポロジ, 5-4
Presence の大規模デプロイ
インストール, 5-1
Presence ポート, 1-3
Proxy Registrar, 1-4
PSTN
SIP から PSTN への通話, 3-10

R

RAC
インストール, 4-2
Remote Method Invocation ポート, 2-14
RMI ポート, 2-14

S

SDP データファイル・ディレクトリ, 2-8
Session Initiation Protocol, 1-2
SIP, 1-2
SIP コンテナ, 2-12
SIP ドメイン, 2-12
SIP ポート, 2-12
SIP レルム, 2-12
STUN サーバー, 1-5
プライマリおよびセカンダリ・アドレスの指定, 2-9
Subscriber Data Services, 1-4
SYS パスワード, 2-7

U

User Dispatcher, 1-5, 5-5

X

XDM, 5-2
XDMC, 5-2
XDMS 設定, 3-4
XDM クラスタ, 5-2
XDM ノード, 5-3

あ

アクセシビリティ, vi

い

インストール
確認, 3-1
クラスタ, 2-2
サマリー, 2-13
前提条件, 1-2
タイプ, 2-4
インストール・モード, 1-3

か

仮想サーバー (VS), 5-32
関連ドキュメント, vi

く

クラスタリング, 2-1

こ

高可用性, 2-1, 2-2

さ

サブレットの登録, 3-6
サイジング, 2-2
削除, 2-16
サンプル・ネットワーク (Presence 大規模デプロイ用), 5-7

し

システム要件, 1-2

す

スキーマ
データベース, 2-9
スタンドアロン開発者モード, 1-4

せ

製品の概要, 1-2
前提条件, インストール, 1-2

ち

チューニング (Presence 大規模デプロイ用), 5-18

て

テスト・ユーザー, 2-10

と

トポロジ, 1-1
Presence マルチノード, 5-4

ふ

プール
作成, 5-31
プライマリ・サーバー・アドレス, 2-5
プレゼンス, 3-9
サブスクリプション, 3-8
プレゼンス・サービス, 5-2
プロキシ・サーバー, 3-4

ほ

ポートの競合, 1-3, 2-16
ポート番号
Oracle Enterprise Manager, 2-14

ゆ

ユーザー, 3-2

ユーザーのプロビジョニング, 3-2

ユーザー名

小文字の要件, 2-11

ろ

ロード・バランサ, 5-5

ログ・レベル, 3-3

