

## **Oracle® Identity Manager**

Tools リファレンス

リリース 9.1.0.1

部品番号 : B53780-01

2009 年 4 月

Oracle Identity Manager Tools リファレンス , リリース 9.1.0.1

部品番号 : B53780-01

Oracle Identity Manager Tools Reference, Release 9.1.0.1

原本部品番号 : E14067-01

原著者 : Debapriya Datta

原本協力者 : Lyju Vadassery

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

#### 制限付権利の説明

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントが、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供される場合は、次の Notice 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアを危険が伴うアプリケーションで使用する場合、このソフトウェア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。このソフトウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、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。

Oracle は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

このソフトウェアおよびドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても、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
ドキュメントの更新 .....	vi
表記規則 .....	vii
オンライン・ヘルプ .....	vii
サポートおよびサービス .....	vii
<b>1 統合の概要</b>	
1.1 統合の問題 .....	1-2
1.2 Oracle Identity Manager のソリューション .....	1-2
1.3 アダプタについて .....	1-2
1.4 「Adapter Factory」 フォームのタブ .....	1-4
1.4.1 Adapter Tasks .....	1-4
1.4.2 Execution Schedule .....	1-4
1.4.3 Resources .....	1-4
1.4.4 Variable List .....	1-5
1.4.5 Usage Lookup .....	1-5
1.4.6 Responses .....	1-5
<b>2 スタート・ガイド</b>	
2.1 Oracle Identity Manager の構成の概要 .....	2-2
2.2 JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成 .....	2-2
2.3 Remote Manager のインストール .....	2-2
2.4 信頼関係の追加 .....	2-3
<b>3 アダプタの作成</b>	
3.1 アダプタ作成の概要 .....	3-2
3.2 アダプタの定義 .....	3-2
3.3 アダプタの無効化と再有効化 .....	3-3
3.4 アダプタ変数について .....	3-3
3.4.1 アダプタ変数の作成 .....	3-3
3.4.2 アダプタ変数の変更 .....	3-5
3.4.3 アダプタ変数の削除 .....	3-5
3.5 アダプタ・タスクの作成 .....	3-6

3.5.1	Java タスクの作成 .....	3-7
3.5.2	リモート・タスクの作成 .....	3-11
3.5.3	ストアド・プロシージャ・タスクの作成 .....	3-12
3.5.4	ユーティリティ・タスクの作成 .....	3-15
3.5.5	Oracle Identity Manager API タスクの作成 .....	3-16
3.5.6	アダプタ変数値の再割当て .....	3-18
3.5.7	エラー・ハンドラ・タスクの追加 .....	3-19
3.5.8	ロジック・タスクの作成 .....	3-20
3.6	アダプタ・タスクの変更 .....	3-22
3.7	タスクの順序とネスティングの変更 .....	3-23
3.8	アダプタ・タスクの削除 .....	3-24
3.9	レスポンスの使用 .....	3-25
3.9.1	レスポンスの作成 .....	3-25
3.9.2	レスポンスの変更 .....	3-26
3.9.3	レスポンスの削除 .....	3-26
3.10	ルール・ジェネレータとエンティティ・アダプタのスケジュール設定 .....	3-26
3.10.1	ルール・ジェネレータとエンティティ・アダプタのスケジュール設定 .....	3-27

## 4 プロセス・タスク・アダプタについて

4.1	プロセスおよびプロセス・タスクの概要 .....	4-2
4.2	プロセス・タスク・アダプタの動作 .....	4-3
4.3	プロセス・タスク・アダプタのプロセス・タスクへのアタッチ .....	4-4
4.4	プロセス・タスク・アダプタのプロセス・タスクからの削除 .....	4-7
4.4.1	プロセス・タスク・アダプタのプロセス・タスクからの削除 .....	4-7

## 5 タスク割当てアダプタの適用

5.1	概要 .....	5-2
5.2	タスク割当てアダプタのプロセス・タスクへのアタッチ .....	5-3
5.2.1	タスク割当てアダプタのプロセス・タスクへのアタッチ .....	5-3
5.3	タスク割当てアダプタのプロセス・タスクからの削除 .....	5-6
5.3.1	タスク割当てアダプタのプロセス・タスクからの削除 .....	5-6

## 6 ルール・ジェネレータについて

6.1	概要 .....	6-2
6.2	ルール・ジェネレータのアダプタ変数のマッピング .....	6-3
6.3	ルール・ジェネレータとプロセスの関連付け .....	6-6
6.4	ルール・ジェネレータのフォーム・フィールドからの削除 .....	6-6

## 7 事前移入アダプタの使用

7.1	概要 .....	7-2
7.2	事前移入アダプタのフォーム・フィールドへのアタッチ .....	7-3
7.3	事前移入アダプタのフォーム・フィールドからの削除 .....	7-6

## 8 エンティティの管理

## 9 アダプタのコンパイル

9.1	アダプタの自動コンパイル .....	9-2
9.2	アダプタの手動コンパイル .....	9-2

## 10 アダプタのエクスポートおよびインポート

## 11 Remote Manager IT リソースの作成とテスト

11.1	インストール後の構成 .....	11-2
11.2	Remote Manager IT リソースの作成とテスト .....	11-4

## 12 SPML Web サービス

12.1	SPML Web サービスの概要 .....	12-2
12.1.1	SPML Web サービスの機能のアーキテクチャ .....	12-2
12.2	SPML Web サービスでサポートされているプロビジョニング処理 .....	12-4
12.3	SPML Web サービスのデプロイ .....	12-7
12.3.1	Oracle WebLogic Server への SPML Web サービスのデプロイ .....	12-7
12.3.2	IBM WebSphere Application Server への SPML Web サービスのデプロイ .....	12-8
12.3.3	JBoss Application Server への SPML Web サービスのデプロイ .....	12-8
12.3.4	Oracle Application Server への SPML Web サービスのデプロイ .....	12-9
12.4	Oracle Web Services Manager を使用し、SPML Web サービスをデプロイするセキュリティの有効化 .....	12-10
12.4.1	Oracle WSM Server Agent の構成 .....	12-11
12.4.1.1	サーバー・エージェントの追加 .....	12-11
12.4.1.2	サーバー・エージェントのポリシーの定義 .....	12-11
12.4.1.3	サーバー・エージェントのインジェクト .....	12-12
12.4.1.4	SPML Web サービスのデプロイ .....	12-13
12.4.2	Oracle WSM Gateway の構成 .....	12-14
12.4.2.1	Oracle WSM Gateway の登録 .....	12-14
12.4.2.2	SPML Web サービスのゲートウェイへの登録 .....	12-14
12.4.2.3	カスタム・ポリシーのゲートウェイへの追加 .....	12-15
12.4.2.4	SPML Web サービスのデプロイ .....	12-16
12.4.2.5	WSDL ファイルの表示 .....	12-16
12.5	デプロイ後のタスク .....	12-16
12.6	SSL 通信の有効化 .....	12-16
12.6.1	JBoss Application Server .....	12-17
12.6.1.1	前提条件 .....	12-17
12.6.1.2	SSL 証明書の設定 .....	12-17
12.6.2	Oracle WebLogic Server .....	12-18
12.6.2.1	前提条件 .....	12-18
12.6.2.2	SSL 証明書の設定 .....	12-18
12.6.3	IBM WebSphere Application Server .....	12-20
12.6.3.1	前提条件 .....	12-20
12.6.3.2	SSL 証明書の設定 .....	12-21
12.6.4	Oracle Application Server .....	12-21
12.6.4.1	前提条件 .....	12-21
12.6.4.2	SSL 証明書の設定 .....	12-22

12.6.5	Oracle HTTP Server への HTTP 通信に対する SSL の有効化 .....	12-23
12.6.5.1	証明書のエクスポート .....	12-23
12.7	SPML Web サービスのクライアントの開発 .....	12-24
12.7.1	サポートされている SPML の処理 .....	12-25
12.7.2	認証 .....	12-25
12.7.3	SPML リクエストに含まれるフィールド .....	12-26
12.7.4	SOAP ヘッダーの構造 .....	12-28
12.7.5	SOAP SPML メッセージの例 .....	12-30

## **A Design Console ユーザー用リファレンス**

## **B SPML メッセージの例**

## **索引**

---

---

# はじめに

このガイドでは、アダプタの作成、プロセス・タスクの割当ておよび削除、アダプタのコンパイル、インポートおよびエクスポートの際に利用できる情報が提供されます。

## 対象読者

このガイドは、Oracle Identity Manager Design Console のユーザーを対象としています。このガイドの情報は、アダプタを作成する開発者や、プロセス・タスク、Oracle Identity Manager で事前定義されたフォームおよびユーザーが作成したフォームにアダプタをアタッチする Oracle Identity Manager 管理者に有用です。

## 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、このリリースの Oracle Identity Manager ドキュメント・セットの他のドキュメントを参照してください。

## ドキュメントの更新

オラクル社は、最新かつ最適な情報の提供に努めています。Oracle Identity Manager のドキュメント・セットの更新情報は、次に示す Oracle Technology Network のサイトを参照してください。

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



## 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）、およびファイル、ディレクトリ、属性、パラメータの名前を示します。

## オンライン・ヘルプ

Oracle Identity Manager には、必要な時に関連する情報を検索できるように設計されているオンライン・ヘルプ・システムがあります。ヘルプ・システム・ライブラリ内の様々なドキュメントは、メイン画面の「ヘルプ」メニューから利用できます。特定のブックを開く場合は、「ヘルプ」メニューから選択してください。

また、Oracle Identity Manager には、様々なフォームに対する状況依存ヘルプがあります。状況依存ヘルプ・トピックの内容は、Oracle Identity Manager メイン画面のアクティブなフォームを反映します。状況依存ヘルプ・トピックを起動する場合は、目的のフォームがアクティブであることを確認してください。次に、[F1] を押します。

---

**注意：** [F1] キーを押すと、状況依存ヘルプだけではなく、Oracle Identity Manager オンライン・ヘルプ・システムも呼び出せます。

---

## 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](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。

---

---

---

## 統合の概要

この章では、アクセス権の管理アプリケーションをその他のソフトウェア・ソリューションと統合する際の課題と、Oracle Identity Manager が提供するソリューションについて説明します。次の項目について説明します。

- 統合の問題
- Oracle Identity Manager のソリューション
- アダプタについて
- 「Adapter Factory」 フォームのタブ

## 1.1 統合の問題

現在、アクセス権限の管理アプリケーションの正常な動作には、他のソフトウェア・ソリューションとの統合が不可欠です。しかし、たくさんのリソースが存在する上に、これらのリソースを結び付けるための単一の統合規格は存在しません。

この問題への取り組みの1つに、すべての統合でサポートされる共通機能の使用があります。この場合、コードを記述する開発者が必要です。そして、既存のソフトウェア・リソースが変更されるたびに、または新しく追加されるたびに、コードを記述する必要があります。

## 1.2 Oracle Identity Manager のソリューション

アダプタ・ファクトリは Oracle Identity Manager が提供するコード生成ツールです。これにより、アダプタと呼ばれる Java クラスの作成が容易になります。

アダプタには、次の利点があります。

- Oracle Identity Manager の内部ロジックおよび機能性を拡張します。
- リソースの API を使用してリソースに接続することで、あらゆるソフトウェア・リソースを結び付けます。
- Oracle Identity Manager と外部システムの統合を可能にします。
- コードを手動で記述することなく生成できます。
- 軽量で、ニーズに特化しています。
- アダプタの定義がすべてリポジトリに格納されているため、管理が容易です。このリポジトリは GUI を介して編集できます。
- ある Oracle Identity Manager ユーザーが統合に関するドメインの知識を保持する一方で、別のユーザーがアダプタを維持することができます。
- 効率的に修正およびアップグレードできます。

## 1.3 アダプタについて

前述のとおり、アダプタとは、Oracle Identity Manager ユーザーによってアダプタ・ファクトリで作成された Java クラスです。また、これは論理フロー内のアトミック・ファンクション・コールの集約です。これらのアトミック・ステップを、アダプタ・タスクと呼びます。

---

**注意：** Oracle Identity Manager は、JDBC および LDAP 用の Java API を使用して、データベースやディレクトリ・サーバーなどの外部システムに接続できます。また、それ以外の API (C、C++、VB、COM/DCOM など) に対しても、Oracle Identity Manager が API と直接に通信できるように、Java ラッパーを作成できます。

---

アダプタには5つのタイプがあります。

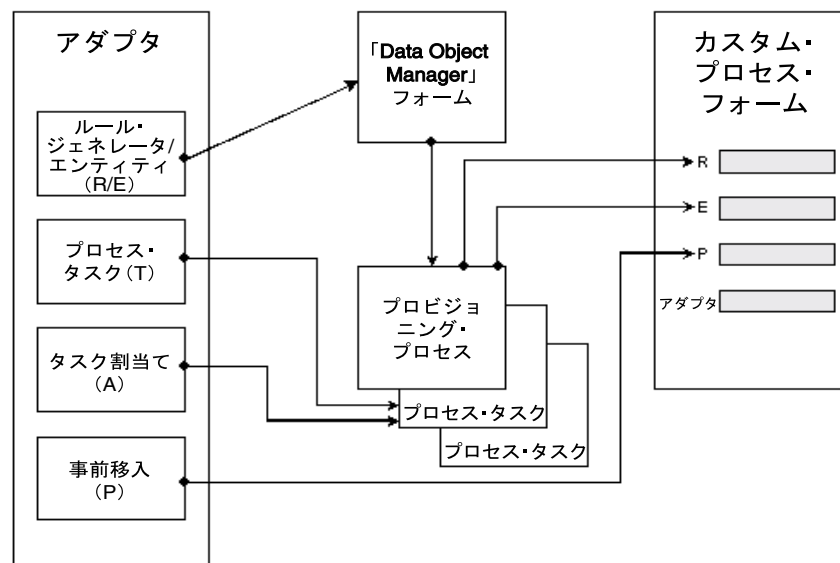
- プロセス・タスク・アダプタは、Oracle Identity Manager によるプロセス・タスクの完了を自動化します。
- タスク割当てアダプタは、Oracle Identity Manager によるユーザーまたはグループへのプロセス・タスクの割当てを自動化します。
- ルール・ジェネレータは、ビジネス・ルールを Oracle Identity Manager フォームまたはユーザー定義フォーム (「Form Designer」フォームを使用して作成) のフィールドに組み込みます。これにより、フィールドは自動的に移入され、Oracle Identity Manager データベースに保存されます。

**注意:** 「Form Designer」 フォームの詳細は、『Oracle Identity Manager デザイン・コンソール・ガイド』を参照してください。

- 事前移入アダプタは、ルール・ジェネレータ・アダプタの一種で、ユーザー作成フォーム・フィールドにアタッチできます。このタイプのアダプタで生成されたデータは、自動または手動で表示できます。また、Oracle Identity Manager では、基準を使用し、指定したフォーム・フィールドにどの事前移入アダプタを適用するかを決定します。次に、Oracle Identity Manager データベースに情報を保存することなく、この情報を指定したフォーム・フィールドに移入します。
- エンティティ・アダプタは、Oracle Identity Manager またはユーザーが作成したフォーム・フィールドにアタッチされます。エンティティ・アダプタは、Oracle Identity Manager で挿入、更新および削除の前後にトリガーされます。これが発生すると、アダプタがアタッチされたフィールドは自動的に移入され、Oracle Identity Manager データベースに保存されます。

図 1-1 は、5つのタイプのアダプタの機能を示します。

図 1-1 アダプタの機能



5つのタイプのアダプタの詳細は、次の章から順を追って説明します。

## 1.4 「Adapter Factory」フォームのタブ

Design Console の「Adapter Factory」フォームには次のタブがあります。

- Adapter Tasks
- Execution Schedule
- Resources
- Variable List
- Usage Lookup
- Responses

それぞれのタブの詳細を、次の項で説明します。

### 1.4.1 Adapter Tasks

「Adapter Tasks」タブでは、アダプタのアトミック・ファンクション・コールを作成および管理できます。このファンクション・コールを、アダプタ・タスクと呼びます。

### 1.4.2 Execution Schedule

「Execution Schedule」タブでは、Oracle Identity Manager でルール・ジェネレータまたはエンティティ・アダプタをトリガーするタイミングを指定できます。Oracle Identity Manager にスケジュールを指定して、挿入前または更新前（あるいはその両方）に、ルール・ジェネレータ（アダプタ・タイプ R）を実行できます。エンティティ・アダプタ（アダプタ・タイプ E）を挿入、更新および削除の前後に実行するよう Oracle Identity Manager を構成することもできます。

---

---

**注意：**プロセス・タスクにアタッチされたプロセス・タスク・アダプタとタスク割当てアダプタは、プロセス・タスクのステータスが「Pending」になるとトリガーされます。そのため、Oracle Identity Manager でこのタイプのアダプタをトリガーするタイミングを指定する必要がなく、Oracle Identity Manager ではこのアダプタの「Execution Schedule」タブは無効化されています。

また、Oracle Identity Manager では挿入前に常に事前移入アダプタがトリガーされるため、事前移入アダプタに対するこのタブのチェック・ボックスは無効化されています。

---

---

### 1.4.3 Resources

「Resources」タブでは、次のことができます。

- 「Java APIs」サブタブをクリックすると、アダプタで使用中の Java API を確認できます。
- 「Other」サブタブをクリックすると、必要に応じて、Java API ではないファイルをアダプタに記録できます。

## 1.4.4 Variable List

「Variable List」タブでは、次のことができます。

- アダプタ変数を作成、変更および削除します。
- 各変数のデータ型を設定し、説明を付けます。
- アダプタ変数を、リテラル参照またはアダプタ参照にマップします。プロセス・タスクまたはフォーム・フィールドにアタッチされるまで、マッピングを延期できます。

アダプタ変数をプロセス・タスクにアタッチしてプロセス・タスクを実行する時に、アダプタ変数の値を解決することもできます。これにより、この変数にマップするプロセス固有のデータが使用可能になります。

## 1.4.5 Usage Lookup

プロセス・タスクまたはタスク割当てアダプタの場合、「Usage Lookup」タブには、アダプタがアタッチされたプロセス・タスクおよびそのプロセス・タスクをメンバーに持つプロセスが表示されます。

ルール・ジェネレータ、またはエンティティ・アダプタの場合、このタブには、Oracle Identity Manager フォームおよびアダプタがアタッチされた関連データ・オブジェクトが表示されます。また、アダプタの実行スケジュールが、Oracle Identity Manager がアダプタをトリガーする順序を表す順序番号とともに表示されます。

事前移入アダプタの場合、このタブには、ユーザー定義フォームとアダプタがアタッチされたフォーム・フィールドが表示されます。また、アダプタに関連付けられた事前移入ルールも表示されます。

---

---

**注意：** 様々なタイプのアダプタの詳細は、[第3章「アダプタの作成」](#)を参照してください。

---

---

## 1.4.6 Responses

「Responses」タブは、プロセス・タスクに対する有効なレスポンスを定義するために使用します。レスポンスは、アダプタの実行結果に基づきます。外部システムによって返される様々なエラー・メッセージは、プロセス・タスクのコンテキストにおいて意味を持つような方法でこれらのレスポンスにマップできます。プロセス・タスクにアダプタをアタッチする時に、アダプタ・レスポンス・コードに基づいて、プロセス・タスク（およびそれに続くオブジェクト・ステータス）のステータス・バケット（「Pending」、「Completed」、「Rejected」）を設定できます。

**ヒント：** Oracle Identity Manager では、プロセス・タスク・アダプタに対してのみ「Responses」タブが有効化されます。タスク割当てアダプタ、ルール・ジェネレータ・アダプタ、事前移入アダプタまたはエンティティ・アダプタの場合、このタブは無効化されます。





---

## スタート・ガイド

この章では、Oracle Identity Manager の構成と Remote Manager のインストールについて説明します。内容は次のとおりです。

- Oracle Identity Manager の構成の概要
- JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成
- Remote Manager のインストール
- 信頼関係の追加

## 2.1 Oracle Identity Manager の構成の概要

アダプタ・タスクを構成するには、Oracle Identity Manager がターゲット API の JAR ファイルおよび接続するサード・パーティ・アプリケーションにアクセスできることを確認してください。

アダプタで Java タスクを使用する場合、適切な Java API を検索するように Oracle Identity Manager を構成する必要があります。このためには、これらの API を含む .jar ファイルを、`OIM_HOME/xellerate` フォルダ・パスの `JavaTasks` サブディレクトリ（たとえば、`C:\¥oracle¥Xellerate¥JavaTasks`）に配置する必要があります。その結果、これらの Java API に関連付けられた Java クラスにアクセスし、作成した Java タスクで使用することができます。

サード・パーティ・システムと直接通信せずに、プロキシのように動作する Oracle Identity Manager コンポーネントを使用する必要がある場合があります。このコンポーネントは、Remote Manager と呼ばれます。

Remote Manager は、次の目的で使用します。

- Oracle Identity Manager を通じた非リモート API の呼出し
- セキュアな接続での Secure Sockets Layer (SSL) をサポートしない API の呼出し

次の項の手順では、次の実行方法を説明します。

- Oracle Identity Manager を JAR およびクラス・ファイルを参照するように構成します (Java タスク用)。
- Remote Manager を構成します。

### 関連項目：

Java タスクの詳細は、3-7 ページの「[Java タスクの作成](#)」

Remote Manager の詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』

## 2.2 JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成

Oracle Identity Manager を JAR およびクラス・ファイルを参照するように構成する手順は、次のとおりです。

1. `OIM_HOME/xellerate` フォルダ・パスにある `JavaTasks` サブディレクトリを開きます。たとえば、`C:\¥oracle¥Xellerate¥JavaTasks` などです。
2. JAR ファイルをサブディレクトリに置きます。このファイルを使用して、サーバーを再起動せずに、アダプタ内に Java タスクを作成できます。

## 2.3 Remote Manager のインストール

Remote Manager を構成するには、使用しているアプリケーション・サーバー用の Oracle Identity Manager のインストール・ガイドに記載されている手順に従います。

## 2.4 信頼関係の追加

信頼できる証明書をインポートするには、コマンド・プロンプトで次の場所に移動します。

```
JAVA_HOME\jre\lib\security
```

次のコマンドを入力します。

```
JAVA_HOME/jre/bin/keytool -import -file  
XLREMOTE_HOME\xlremote\config\xlserver.cert -keystore cacerts -trustcacerts -  
alias rmtrust1 -storepass changeit
```

```
Owner: CN=Customer, OU=Customer, O=Customer, L=City, ST=NY, C=US  
Issuer: CN=Customer, OU=Customer, O=Customer, L=City, ST=NY, C=US  
Serial number: 41dee35a  
Valid from: Fri Jan 07 11:30:34 PST 2005 until: Mon Jan 05 11:30:34 PST 2015  
Certificate fingerprints:  
    MD5: B0:F2:33:C8:69:E4:25:A3:CB:59:E8:51:27:EE:5C:52  
    SHA1: 3D:6A:6D:14:33:B5:5C:19:85:CC:EE:77:7F:7F:22:1D:56:48:47:4D  
Trust this certificate? [no]: yes
```

証明書がキーストアに追加されます。



---

## アダプタの作成

この章では、アダプタの作成および定義の方法を説明します。内容は次のとおりです。

- アダプタ作成の概要
- アダプタの定義
- アダプタの無効化と再有効化
- アダプタ変数について
- アダプタ・タスクの作成
- アダプタ・タスクの変更
- タスクの順序とネスティングの変更
- アダプタ・タスクの削除
- レスポンスの使用
- ルール・ジェネレータとエンティティ・アダプタのスケジュール設定

## 3.1 アダプタ作成の概要

Oracle Identity Manager の設定が終了すると、次の5つのタイプの Oracle Identity Manager アダプタを作成できます。次のようなものがあります。

- プロセス・タスク・アダプタ
- タスク割当てアダプタ
- ルール・ジェネレータ
- 事前移入アダプタ
- エンティティ・アダプタ

**関連項目：** アダプタの詳細は、1-2 ページの「[アダプタについて](#)」

## 3.2 アダプタの定義

アダプタを定義する手順は、次のとおりです。

1. Oracle Identity Manager にログインします。
2. 「Adapter Factory」フォームを開きます。このフォームは、Design Console の「Development Tools」フォルダにあります。
3. 「Adapter Name」フィールドにアダプタの名前（Create Solaris User など）を入力します。

---

**注意：** アダプタ名には特殊文字を使用できますが、ランタイム・エラーが発生する可能性があるため、それらの文字を使用しないことをお勧めします。

---

4. 「Adapter Type」検索フィールドをダブルクリックします。  
5つのタイプの Oracle Identity Manager アダプタが表示された「Lookup」ウィンドウが開きます。次のようなものがあります。
  - プロセス・タスク
  - ルール・ジェネレータ
  - 事前移入ルール・ジェネレータ
  - エンティティ
  - タスク割当て
5. プロセス・タスクを自動化するアダプタを有効化するには、「**Process Task (T)**」を選択します。  
ビジネス・ルールを Oracle Identity Manager またはユーザー定義フォーム・フィールドに組み込むには、「**Rule Generator (R)**」を選択します。たとえば、フォームの「User ID」フィールド用に、ユーザーの姓と名の頭文字を結合するように Oracle Identity Manager を構成できます。  
ユーザーが作成したフォーム・フィールドにある種類のルール・ジェネレータ・アダプタをアタッチし、次のことを実行できます。
  - アダプタによって生成されたデータの自動または手動表示
  - 指定されたフォーム・フィールドにどのアダプタを適用するかを Oracle Identity Manager が判断するための基準の使用

Oracle Identity Manager またはユーザー定義フォーム・フィールドにアダプタをアタッチし、挿入、更新および削除の前後に Oracle Identity Manager でアダプタをトリガーするには、「Entity (E)」を選択します。

アダプタでユーザーまたはグループに対するプロセス・タスクの割当てを自動化するには、「Task Assignment (A)」を選択します。

6. 使用するアダプタ・タイプ（「Process Task (T)」など）を選択します。次に、「OK」をクリックします。

**関連項目：**「Form Designer」フォームの詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』

7. 「Description」フィールドにアダプタの説明（「このアダプタは、Solaris 環境で新規ユーザーを作成するために使用します」など）を入力します。
8. ツールバーで「Save」をクリックします。

アダプタが Oracle Identity Manager データベースに格納されます。

### 3.3 アダプタの無効化と再有効化

アダプタを無効化してプロセス・タスクまたはフォーム・フィールドで使用できないようにするには、「Disable Adapter」オプションを選択してアダプタを保存します。

再度有効化するには、「Disable Adapter」オプションの選択を解除し、アダプタを保存します。

### 3.4 アダプタ変数について

新規に作成したアダプタを動作させるため、アダプタ・タスクのパラメータにデータをマップできます。そのために、実行時にデータをマップするプレースホルダを作成します。これは、アダプタ変数ともいいます。

---

**注意：** アダプタ変数はすべてのアダプタ・タスクで再使用できます。

---

アダプタ変数がアダプタの実行に不要になった場合は、アダプタから削除できます。アダプタ変数の削除後、アダプタを再コンパイルしてください。

#### 3.4.1 アダプタ変数の作成

アダプタ変数を作成する手順は、次のとおりです。

1. アダプタ変数を追加するアダプタ（Create Solaris User アダプタなど）を選択します。
2. 「Variable List」タブを選択します。
3. 「Add」をクリックします。  
「Add a Variable」ウィンドウが表示されます。
4. アダプタ変数をアクティブ化した後に Oracle Identity Manager で変更できないようにするには、「Final」を選択します。
5. 「Variable Name」フィールドにアダプタ変数の名前（SolarisUserID など）を入力します。

---

**注意：** アダプタ変数名に空白は使用できません。

---

6. 「Type」メニューからアダプタ変数の分類（「String」など）を選択します。使用可能な項目は次のとおりです。
  - Object
  - IT Resource
  - String
  - Boolean
  - Character
  - Byte
  - Date
  - Integer
  - Float
  - Long
  - Short
  - Double
7. 「Description」テキスト領域には、アダプタ変数の説明を入力できます。
8. 「Map To」メニューから、アダプタ変数を表 3-1 の項目のいずれかにマップできます。

表 3-1 「Map To」メニューの項目

名前	説明
Literal	このアダプタ変数は定数（またはリテラル）にマップされます。
Resolve at Runtime	このアダプタ変数は後で実行時にマッピングされます。このオプションを選択すると、アダプタの再利用性が向上します。
Adapter References	このアダプタ変数によって、Oracle Identity Manager データベース参照または Oracle Identity Manager データ・オブジェクト参照にアクセスできます。
System Date	このアダプタ変数が Oracle Identity Manager でトリガーされると、サーバーの現在の日付と時間がマッピングされます。 <b>注意：</b> このオプションは「Date」タイプを選択した場合にのみ表示されます。

**注意：**「Object」タイプを選択すると、「Qualifier」メニューが「Add a Variable」ウィンドウに表示されます。このメニューから次のいずれかを選択できます。

- Database Reference この項目を選択すると、アダプタ変数が、Oracle Identity Manager が現在実行されているデータベースの参照にマップされます。
- Data Object Reference この項目を選択すると、アダプタ変数が、Oracle Identity Manager データ・オブジェクトにマップされます。

**注意：**「IT Resource」タイプを選択すると、「Resource Type」メニューが「Add a Variable」ウィンドウに表示されます。このメニューから、「IT Resource Type Definition」フォームを使用して作成した IT リソース・タイプのいずれかを選択できます。これにより、この IT リソース・タイプのパラメータにアダプタ変数をマップできます。



9. 「Add a Variable」ウィンドウのツールバーで、「Save」をクリックします。アダプタ変数の情報が、Oracle Identity Manager データベースに格納されます。

「Add a Variable」ウィンドウを閉じ、メイン画面をアクティブにします。作成したアダプタ変数の名前、分類タイプ、マッピング選択および説明が、「Variable List」タブの子表に表示されます。

これで、このアダプタ変数が「Adapter Factory」フォームのアダプタに加わりました。Oracle Identity Manager データベースに保存され、アダプタ変数が使用可能になります。

### 3.4.2 アダプタ変数の変更

アダプタ変数を変更する手順は、次のとおりです。

1. 編集するアダプタ変数を含むアダプタ (Create Solaris User アダプタなど) を選択します。
2. 「Variable List」タブをクリックし、変更するアダプタ変数の行ヘッダーをダブルクリックします。アダプタ変数の情報を示す「Edit a Variable」ウィンドウが表示されます。
3. 必要な編集を行います。たとえば、アダプタ変数のデータ型を、「String」から「Character」に変更します。
4. 「Edit a Variable」ツールバーで、「Save」をクリックします。アダプタ変数の変更情報が、Oracle Identity Manager データベースに格納されます。
5. 「Edit a Variable」ウィンドウを閉じ、メイン画面をアクティブにします。変更したアダプタ変数が、「Adapter Factory」フォームの子表に表示されます。

---

---

**注意:** アダプタ変数のデータ型を変更した場合は特に、データ・マッピングを確認し、アダプタを再コンパイルしてください。

---

---

### 3.4.3 アダプタ変数の削除

アダプタ変数がアダプタの実行に不要となった場合は、アダプタから削除できます。これを実行する手順は次のとおりです。

1. 削除するアダプタ変数を含むアダプタ (Create Solaris User アダプタなど) を選択します。
2. 「Variable List」タブを選択します。
3. このタブのリストから、削除するアダプタ変数を選択します。
4. 「Delete」をクリックします。
5. 変数を削除した後は、アダプタを再コンパイルしてください。

アダプタ変数が子表に表示されなくなります。アダプタ変数は削除されました。

## 3.5 アダプタ・タスクの作成

アダプタの構成と変数の作成が終了すると、アダプタのアトミック・ファンクション・コールを作成できます。このファンクション・コールを、アダプタ・タスクと呼びます。

Oracle Identity Manager では、次のアダプタ・タスクを作成できます。

- **Java タスク** : Java API を呼び出してアダプタと外部ソースとの通信を可能にします。
- **リモート・タスク** : アダプタによる API のメソッドの呼出しを可能にします。この API は Oracle Identity Manager の外部にあるコンピュータに存在します。  
このタイプのタスクは、主にネットワークが有効ではないサード・パーティ API との統合に使用されます。**Remote Manager** は、リモート・コンピュータに配置されているリモート API メソッドを実行します。また、サード・パーティ API が **Secure Sockets Layer (SSL)** を使用しない場合、**Remote Manager** を使用して、SSL で保護された通信でサード・パーティ API を呼び出すことができます。リモート・タスクは、ネットワークが有効であり、スケーラビリティのために **Oracle Identity Manager** に配置されていないサード・パーティ API との統合にも使用されます。この場合も、リモート API は **Remote Manager** によって実行されます。ただし、サード・パーティ API でネットワークが有効であるため、**Remote Manager** がターゲット・システム上に存在している必要はありません。
- **ストアド・プロシージャ・タスク** : **Oracle Identity Manager** で特定のデータベース・スキーマ内に存在する SQL プログラムをマップして実行できます。このプログラムを、ストアド・プロシージャと呼びます。これには、効率を高めるために事前にコンパイルされた SQL 文などの情報が含まれます。  
ストアド・プロシージャ・タスクをアダプタに組み込み、このアダプタをプロセス・タスクにアタッチすると、**Oracle Identity Manager** で、ストアド・プロシージャをネットワーク上でアクセス可能なあらゆる **Oracle Database** または **Microsoft SQL Server** データベースに組み込むことができます。これには、ストアド・プロシージャからのプリミティブ値の取得も含まれます。
- **ユーティリティ・タスク** : アダプタに、**Oracle Identity Manager** にパッケージされたメソッドおよび API を移入できます。また、このタイプのタスクを使用すると、Java 標準ライブラリ API にアクセスできます。
- **Oracle Identity Manager API タスク** : アダプタ・タスクから **Oracle Identity Manager** で公開された API にアクセスできます。これにより、アダプタ・コードの移植性が高まります。
- **変数設定タスク** : アダプタ内に変数を設定できます。
- **エラー・ハンドラ・タスク** : 実行時に発生するアダプタ関連のエラーを表示できます。また、エラーの原因と適切な解決方法を確認できます。
- **ロジック・タスク** : アダプタ内に条件文を構築できます。

次のタイプのロジック・タスクを作成できます。

- FOR ループ
- WHILE ループ
- IF 文
- ELSE 文
- ELSE IF 文
- BREAK 文
- RETURN 文
- CONTINUE 文
- SET VARIABLE 文
- Handle Error 文

**関連項目：**構築できるロジック・タスクのタイプの詳細は、3-20 ページの「ロジック・タスクの作成」

区別しやすくするために、Oracle Identity Manager では、アダプタ・タスクのタイプがそれぞれアイコンで表示されます。タスク名の先頭にあるアイコンは、タスク・タイプを視覚的に示しています。たとえば、「J」は Java タスクを表し、「LT」はロジック・タスクを表します。

アイコンのリストを表示するには、「Adapter Tasks」タブを選択し、「Legend」をクリックします。次のアイコンのリストを示した「Legend」ウィンドウが表示されます。

- 関数タスク
  - Java
  - リモート
  - ストアド・プロシージャ
- ユーティリティ・タスク
  - ユーティリティ
  - Oracle Identity Manager API
- 論理タスク

### 3.5.1 Java タスクの作成

Oracle Identity Manager は Java API を通じて外部ソースとハンドシェイクします。このためには、Oracle Identity Manager によってトリガーされた場合に外部ソースとの通信を開始するタスクを、アダプタに追加する必要があります。このタイプのタスクを Java タスクと言います。

Java タスクを作成する手順は次のとおりです。

1. Java タスクを追加するアダプタ（Update Solaris Password アダプタなど）を選択します。
2. 「Adapter Tasks」タブを選択します。
3. 「Add」をクリックします。

「Adapter Task Selection」ウィンドウが表示されたら、「Functional Task」オプションを選択します。

4. オプションの右側の表示領域から Java 項目を選択し、「Continue」をクリックします。

「Object Instance Selection」ウィンドウが表示されます。

表 3-2 で、「Object Instance Selection」ウィンドウのオプションを説明します。

**表 3-2 「Object Instance Selection」ウィンドウのオプション**

オプション	説明
New Object Instance	このオプションをクリックして、新規 Java オブジェクト・インスタンスを作成します。
Persistent Instance	このオプションをクリックし、隣接するコンボ・ボックスをクリックし、ドロップダウン・メニューからオブジェクト・インスタンスを選択すると、永続オブジェクトにメソッドを呼び出すことができます。
Task Return Value Instance	このオプションをクリックし、コンボ・ボックスをクリックし、ドロップダウン・リストからアダプタ・タスクを選択すると、先に定義したアダプタ・タスクによって戻されたオブジェクトにこのメソッドを呼び出すことができます。

---

**注意：**「Persistent Instance」オプションが無効化されている場合は、アダプタに永続オブジェクトが定義されていません。同様に、「Task Return Value Instance」オプションが無効化されている場合は、タスクに関連付けられた Java オブジェクトの戻り値がありません。

---

5. オプション（「New Object Instance」など）をクリックして、「Continue」をクリックします。「Add an Adapter Factory Task」ウィンドウが表示されます。

表 3-3 で、「Add an Adapter Factory Task」ウィンドウの様々な領域を説明します。

**表 3-3 「Add an Adapter Factory Task」ウィンドウの領域**

名前	説明
Task Name	このフィールドには Java タスクの名前が表示されます。
Persistent Instance	この Java オブジェクトが再利用される場合には、チェック・ボックスが選択され、タスク・インスタンスの名前が隣のフィールドに入力されています。
API Source	このコンボ・ボックスには、アクセス可能なすべての JAR およびクラス・ファイルのリストが表示されます。
Application API	このコンボ・ボックスには、アクセス権を持ち、「API Source」リストで選択した JAR ファイルに属するすべてのクラス・ファイルのリストが表示されます。
Constructors	このテキスト領域には、Java オブジェクトで使用可能なすべてのコンストラクタが表示されます。
Methods	このテキスト領域には、Java オブジェクトで使用可能なすべてのメソッドのリストが表示されます。
Application Method Parameters	この領域には、選択したコンストラクタとメソッドのパラメータが表示されます。このパラメータは、アダプタ変数と Oracle Identity Manager コンポーネントにマップされます。

6. 「Task Name」フィールドに、作成するタスクの名前（Update Password など）を入力します。
7. (オプション。) Java オブジェクトを再利用可能にするには、「Persistent Instance」を選択し、チェック・ボックスの右側にあるテキスト・フィールドにこのタスクのインスタンスの名前を入力します。

---

**注意：**インスタンス名に空白が含まれていないことを確認してください。

---



---

**注意：**ターゲット・リソースとのセッションを、1 回のみではなく、アダプタの使用中に何度も参照するには、「Persistent Instance」を選択します。

---

**ヒント：**Java オブジェクトを永続に設定すると、Java オブジェクトを次に作成する時に、「Object Instance Selection」ウィンドウの「Persistent Instance」リストに表示されます。また、同一 Java オブジェクトのすべてのアダプタ・タスクにコンストラクタをマップする必要はありません。

- 「API Source」を選択します。Oracle Identity Manager が `OIM_HOME/xellerate` フォルダ・パスの `JavaTasks` サブディレクトリ（たとえば `C:\¥oracle¥Xellerate¥JavaTasks`）から参照する JAR ファイルが表示されます。

**関連項目：** Oracle Identity Manager でサード・パーティ JAR ファイルを Java タスクとともに使用方法については、2-2 ページの「[JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成](#)」

- 「Application API」を選択します。「API Source」で選択した JAR ファイルに属するクラス・ファイルが表示されます。
- 「Constructors」領域で、選択した Java クラスの初期化に使用するメソッドを選択します。
- 「Methods」領域で、Java タスクで使用するメソッドを選択します。
- ツールバーで「Save」をクリックします。

Java タスクに関する情報が、Oracle Identity Manager データベースに格納されます。Java タスクのコンストラクタとメソッドのパラメータにアクセス可能になります。これらのパラメータは、「Add an Adapter Factory Task」ウィンドウの「Application Method Parameters」領域に表示されます。
- マッピングを設定する必要がある Java クラスのコンストラクタおよびメソッドを表示するには、「Constructors」および「Methods」アイコンの左側にあるプラス・アイコンをクリックします。
- マッピングを設定する必要があるコンストラクタまたはメソッドのパラメータを選択します。
- 「Description」テキスト領域には、このマッピングの説明を入力できます。
- 「Map To」コンボ・ボックスをクリックし、コンストラクタまたはメソッドのパラメータにマップ可能な項目を選択します（「Adapter Variable」など）。
- マッピングを正しく設定します。

**関連項目：** 設定するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「[アダプタ・マッピング情報](#)」

- 「Set」をクリックします。

選択されたコンストラクタまたはメソッドのパラメータが、青で表示されます。これはマップ済であることを表します。

**ヒント：** パラメータのマッピングを削除するには、そのパラメータを右クリックし、表示されるポップアップ・メニューから「Un-Map Parameter」を選択します。

- 「Application Method Parameters」領域に表示されるコンストラクタおよびメソッドのすべてのパラメータに対して、手順 15～18 を繰り返します。
- 「Add an Adapter Factory Task」ウィンドウのツールバーで「Save」をクリックします。Java タスクに関する情報が、Oracle Identity Manager データベースに格納されます。
- ツールバーで「Close」をクリックします。「Add an Adapter Factory Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。作成した Java タスク（Update Password など）が「Adapter Factory」フォームに表示されます。

22. (オプション。) アダプタに追加の Java タスクを作成するには、手順 3 ~ 21 を繰り返します。

**ヒント:** アダプタには、様々なタイプのアダプタ・タスクを作成および追加できます。

アダプタが論理的に完全で、アダプタ・タスクのすべての変数がマップ済の場合、アダプタをコンパイルして、プロセス・タスクまたはフォーム・フィールドで使用できます。

23. アダプタをコンパイルするには、「**Build**」をクリックします。

「**Compile Status**」フィールドのテキストが、「**Recompile**」から「**OK**」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにアタッチできます。

24. (オプション。) Oracle Identity Manager で生成されたコードを表示するには、ツール・バーで「**Notes**」をクリックします。

Oracle Identity Manager で生成されたコードを含む「**Notes**」ウィンドウが表示されます。

---

**注意:** 「**Build**」をクリックした後に、「**Compile Status**」フィールドに **CODE GEN ERROR** が表示される場合は、アダプタの検証およびコンパイル中に、Oracle Identity Manager で次のいずれかのタイプのエラーが発生しています。

■ 検証エラー

Oracle Identity Manager がアダプタをチェックし、有効性を検証している時に、エラーが発見されました。アダプタ・タスクのパラメータがマッピングされていない場合、パラメータのマッピングが正しくない場合または配置されたアダプタ・タスクが使用できない場合に、このエラーが発生します。

Oracle Identity Manager では、アダプタの検証が終了してからアダプタのコードが生成されるため、Oracle Identity Manager で検証エラーが発生すると、コードの生成は行われません。

■ Java コンパイル・エラー

アダプタの有効性が Oracle Identity Manager で検証されました。しかし、Oracle Identity Manager でアダプタのコンパイル中に、エラーが発見されました。アダプタ・タスク・パラメータに割り当てられたデータ型が正しくない場合に、このエラーが発生します。

Oracle Identity Manager でアダプタが検証されているため、コードは生成されます。ただし、コンパイル中に Oracle Identity Manager でエラーが発生すると、コードの生成が停止します。

---

**ヒント:** Java タスクを作成してアダプタに追加すると、「**Adapter Factory**」フォームの「**Resources**」タブにアクセスして、次の情報を確認できます。

- Java タスクの作成に使用した JAR ファイルおよびクラス・ファイル
- JAR ファイルおよびクラス・ファイルを含むディレクトリ・パスを表す名前

## 3.5.2 リモート・タスクの作成

リモート・タスクを使用すると、アダプタで **Remote Manager** を使用した API メソッドの呼出しができます。この API は **Oracle Identity Manager** の外部にあるコンピュータに存在します。この項では、リモート・タスクの作成方法を説明します。

---

---

**注意：** リモート・タスクを作成する前に、アダプタ変数を IT リソースの分類タイプに定義し、「IT Resource Type Definition」フォームを使用して作成した IT リソースのいずれか 1 つを選択してください。

---

---

1. リモート・タスクを追加するアダプタを選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 「Add」をクリックします。  
「Adapter Task Selection」ウィンドウが表示されます。
4. 「Functional Task」オプションを選択します。
5. ボタンの右側の表示領域から、リモート・タスクを作成するリモート項目を選択します。次に、「Continue」をクリックします。  
「Object Instance Selection」ウィンドウが表示されます。

---

---

**注意：** このウィンドウの選択肢の詳細は、3-7 ページの [「Java タスクの作成」](#) を参照してください。

---

---

6. 「Continue」をクリックします。  
「Add an Adapter Factory Task」ウィンドウが表示されます。
7. 「Task Name」フィールドに、作成するリモート・タスクの名前を入力します。
8. (オプション。) リモート・タスクを再利用可能にするには、「Persistent Instance」オプションを選択します。次に、チェック・ボックスの右側のテキスト・フィールドに、このタスクのインスタンス名を入力します。

---

---

**注意：** インスタンス名に空白が含まれていないことを確認してください。

---

---

### 関連項目：

「Add an Adapter Factory Task」ウィンドウの領域の詳細は、3-7 ページの [「Java タスクの作成」](#)

Oracle Identity Manager でサード・パーティ JAR ファイルを Java タスクとともに使用する方法については、2-2 ページの [「JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成」](#)

9. 「Add an Adapter Factory Task」ウィンドウで、JAR ファイル、クラス・ファイル、コンストラクタおよびメソッドを選択します。次に、コンストラクタおよびメソッドのパラメータのマッピングを設定します。

---

---

**注意：** 入力パラメータの 1 つに、IT リソースの分類タイプが含まれます。このパラメータを、IT リソースのアダプタ変数タイプと関連付ける必要があります。

---

---

**関連項目：** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」

10. 「Add an Adapter Factory Task」ウィンドウのツールバーで「**Save**」をクリックします。  
リモート・タスクに関する情報が、Oracle Identity Manager データベースに格納されます。
11. このウィンドウのツールバーで、「**Close**」をクリックします。  
「Add an Adapter Factory Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。作成したリモート・タスクが、「Adapter Factory」フォームに表示されます。
12. (オプション。) アダプタに追加のリモート・タスクを作成するには、手順 3～11 を繰り返します。  
これで、アダプタをコンパイルし、プロセス・タスクまたはフォーム・フィールドで使用できます。
13. アダプタをコンパイルするには、「**Build**」をクリックします。  
「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチし、それによって Oracle Identity Manager と外部 API が通信できます。

### 3.5.3 ストアド・プロシージャ・タスクの作成

Oracle Identity Manager では、特定のデータベース・スキーマ内に存在する SQL プログラムへのマップおよび実行ができます。この SQL プログラムを、ストアド・プロシージャと言います。ストアド・プロシージャには、効率を高めるために事前にコンパイルされた SQL 文などが含まれます。

そのためには、ストアド・プロシージャ・タスクをアダプタに追加する必要があります。Oracle Identity Manager でトリガーされると、このタスクはネットワーク上でアクセス可能なあらゆる Oracle Database または Microsoft SQL Server データベースにストアド・プロシージャを組み込みます。これには、ストアド・プロシージャからのプリミティブ値の取得も含まれます。

ストアド・プロシージャ・タスクを作成する手順は次のとおりです。

---

**注意：** データベース・スキーマのパラメータ値とサーバー・タイプは、「IT Resources」フォームで設定します。

スキーマのサーバー・タイプは、データベースに設定します。それ以外では、ストアド・プロシージャ・タスクの作成または実行（あるいはその両方）の間に、Oracle Identity Manager がデータベース・スキーマを参照できません。

また、Oracle Identity Manager は、パラメータで表される値（データベース名または URL など）を使用してスキーマに接続します。その結果、スキーマに含まれるストアド・プロシージャを実行できます。

---

1. Oracle Database を使用する Oracle Identity Manager インストールの場合、*OIM\_HOME/xellerate/ext/* ディレクトリから *OIM\_DC\_HOME/xlclient/ext* ディレクトリに *ojdbc14.jar* ファイルをコピーしてください。

Microsoft SQL Server を使用する Oracle Identity Manager インストールの場合、Microsoft から次のファイル入手し、それらを *OIM\_DC\_HOME/xlclient/ext* ディレクトリにコピーする必要があります。

- *msbase.jar*
- *mssqlserver.jar*
- *msutil.jar*



2. ストアド・プロシージャ・タスクを追加するアダプタ（Update User ID アダプタなど）を選択します。
3. 「Adapter Tasks」タブをクリックします。
4. 「Add」をクリックします。  
「Adapter Task Selection」ウィンドウが表示されます。
5. 「Functional Task」オプションを選択します。
6. オプションの右側の表示領域から、「Stored Procedure」を選択し、「Continue」をクリックします。「Add an Adapter Factory Task」ウィンドウが表示されます。  
次の表で、「Add an Adapter Factory Task」ウィンドウの領域を説明します。

表 3-4 「Add an Adapter Factory Task」ウィンドウの領域

名前	説明
Task Name	ストアド・プロシージャ・タスクの名前が表示されます。
Description	ストアド・プロシージャ・タスクの説明が表示されます。
Database	「IT Resources」フォームで定義されたデータベースをリストします。 <b>重要:</b> サーバー・タイプがデータベースである IT リソースのみが「Database」コンボ・ボックスに表示されます。
Schema	「Database」リストに表示されるデータベースに関連付けられたスキーマをリストします。
Procedure	「Schema」リストに表示されるデータベース・スキーマに存在するストアド・プロシージャをリストします。
Connection Status	Oracle Identity Manager とターゲット・ストアド・プロシージャを含むデータベースの接続状態が表示されます。 Oracle Identity Manager がデータベースに接続可能な場合は、「Connection Status」領域に「Connection Established」と表示されます。 <b>注意:</b> Oracle Identity Manager が接続できない場合は、表示領域に「Connection Failed」と表示されます。また、「Add an Adapter Factory Task」ウィンドウの「Notes」ボタンが有効になります。このボタンをクリックすると、接続を確立できなかった理由が表示されます。次に例を示します。 Exception Type: java.lang.ClassNotFoundExceptionMessage: java.lang.ClassNotFoundExceptionMessage: oracle.jdbc.driver.OracleDriver この例では、特定の Java クラスが見つからなかったために、Oracle Identity Manager が指定されたデータベースに接続できませんでした。
Parameters	ストアド・プロシージャにマップ可能なパラメータが表示されます。これらのパラメータは、データベース、スキーマおよびストアド・プロシージャを選択し、この情報を Oracle Identity Manager データベースに保存した後で表示されます。

7. 「Task Name」フィールドに、作成するストアド・プロシージャ・タスクの名前（Update ID など）を入力します。
8. 「Description」テキスト領域には、このストアド・プロシージャ・タスクの説明を入力できます。

9. 「Database」 リストをクリックします。「IT Resources」 フォームで定義されたデータベースが表示されます。

---

---

**注意：** Oracle Identity Manager が選択したデータベースに接続できない場合は、表示領域に「Connection Failed」と表示されます。また、「Add an Adapter Factory Task」 ウィンドウの「Notes」 ボタンが有効になります。このボタンをクリックすると、接続を確立できなかった理由が表示されます。

---

---

**ヒント：** Oracle Identity Manager で接続できるデータベースを選択した場合のみ、スキーマおよびストアド・プロシージャが表示されます。この選択に基づき、関連するスキーマとプロシージャが、対応するコンボ・ボックスに表示されます。

**関連項目：** 「IT Resources」 フォームを使用してストアド・プロシージャを含むデータベースの設定を定義する方法の詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』

10. 「Schema」 リストをクリックします。選択したデータベースに関連付けられたスキーマが表示されます。
11. 「Procedure」 リストをクリックします。「Schema」 コンボ・ボックスから選択したデータベース・スキーマに存在するストアド・プロシージャが表示されます。
12. 「Add an Adapter Factory Task」 ウィンドウのツールバーで「Save」 をクリックします。ストアド・プロシージャ・タスクに関する情報が、Oracle Identity Manager データベースに格納されます。

これで、ストアド・プロシージャのパラメータにマッピングを設定できます。これらのパラメータは、「Add an Adapter Factory Task」 ウィンドウの「Parameters」 領域に表示されます。

---

---

**注意：** Oracle Identity Manager は、選択したストアド・プロシージャのデータベースおよびスキーマを自動的にマップします。ただし、Oracle Identity Manager ではこのマッピングを上書きできません。

---

---

**関連項目：** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」

13. 「Add an Adapter Factory Task」 ウィンドウのツールバーで「Save」 をクリックします。ストアド・プロシージャ・タスクのパラメータに設定したマッピングが、Oracle Identity Manager データベースに格納されます。
14. このウィンドウのツールバーで、「Close」 をクリックします。  
「Add an Adapter Factory Task」 ウィンドウが閉じ、メイン画面が再びアクティブになります。作成したストアド・プロシージャ・タスク（Update ID など）が、「Adapter Factory」 フォームに表示されます。
15. (オプション。) アダプタに追加のストアド・プロシージャ・タスクを作成するには、手順 3～13 を繰り返します。
16. アダプタをコンパイルするには、「Build」 をクリックします。  
「Compile Status」 フィールドのテキストが、「Recompile」 から「OK」 に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチし、それによって Oracle Identity Manager で選択したストアド・プロシージャ・タスクにマップおよび実行できます。

### 3.5.4 ユーティリティ・タスクの作成

アダプタ・ファクトリは、アダプタ開発の効率を高めるユーティリティ・クラスおよびメソッドのライブラリとともに出荷されています。

ユーティリティ・クラスとメソッドは、**xlUtils.jar**、**xlIntegration.jar** および **rt.jar** ファイル内にあります。この JAR ファイルのいずれかのクラスまたはメソッドを使用して作成した Java タスクを、**ユーティリティ・タスク**と言います。

**関連項目：** xlUtils.jar、xlAPI.jar、xlIntegration.jar および rt.jar ファイルを含むクラス・ファイルの詳細は、Java API のドキュメント

1. ユーティリティ・タスクを追加するアダプタ（Update Solaris User Group アダプタなど）を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 「Add」をクリックします。  
「Adapter Task Selection」ウィンドウが表示されます。
4. 「Utility Task」オプションを選択します。
5. オプションの右側の表示領域から、「Utility」を選択し、「Continue」をクリックします。「Object Instance Selection」ウィンドウが表示されます。

**関連項目：** このウィンドウの選択肢の詳細は、3-7 ページの「[Java タスクの作成](#)」

6. 「Continue」をクリックします。「Add an Adapter Factory Task」ウィンドウが表示されます。
7. 「Task Name」フィールドに、作成するユーティリティ・タスクの名前（Update User Group など）を入力します。
8. (オプション。) ユーティリティ・タスクを再利用可能にするには、「Persistent Instance」を選択し、チェック・ボックスの右側にあるテキスト・フィールドにこのタスクのインスタンスの名前を入力します。

---

**注意：** インスタンス名に空白が含まれていないことを確認してください。

---

**関連項目：**

「Add an Adapter Factory Task」ウィンドウの領域の詳細は、3-7 ページの「[Java タスクの作成](#)」

2-2 ページの「[JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成](#)」

9. 「Application API」リストをクリックします。xlUtils.jar、xlIntegration.jar および rt.jar ファイルに属するクラス・ファイルが表示されます。

---

**注意：** xlUtils.jar、xlIntegration.jar および rt.jar には、ユーティリティ・タスクで使用できるすべてのクラス・ファイルが含まれています。したがって、「API Source」リストにアクセスする必要はありません。

---

10. 「Add an Adapter Factory Task」ウィンドウで、コンストラクタおよびメソッドを選択します。次に、コンストラクタおよびメソッドのパラメータのマッピングを設定します。

**関連項目：** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」

11. 「Add an Adapter Factory Task」ウィンドウのツールバーで「Save」をクリックします。ユーティリティ・タスクに関する情報が、Oracle Identity Manager データベースに格納されます。
12. このウィンドウのツールバーで、「Close」をクリックします。  
「Add an Adapter Factory Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。作成したユーティリティ・タスク（Update User Group など）が、「Adapter Factory」フォームに表示されます。
13. (オプション。) アダプタに追加のユーティリティ・タスクを作成するには、手順 3～12 を繰り返します。  
これで、アダプタをコンパイルし、プロセス・タスクまたはフォーム・フィールドで使用できます。
14. アダプタをコンパイルするには、「Build」をクリックします。  
「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチできます。

### 3.5.5 Oracle Identity Manager API タスクの作成

アダプタ・コードの移植性を高めるため、Oracle Identity Manager API タスクでは、Oracle Identity Manager で公開済の API をアダプタ・タスクで呼び出すことができます。これは、ハードコードされた SQL 文を介して Oracle Identity Manager データに直接アクセスするよりもよい方法です。

アダプタ・ファクトリは、Oracle Identity Manager API タスクを含むアダプタ開発の効率を高めるユーティリティ・クラスおよびメソッドのライブラリとともに出荷されています。このユーティリティ・クラスとメソッドは、xlAPI.jar ファイル内にあります。

**関連項目：** xlUtils.jar、xlAPI.jar、xlIntegration.jar および rt.jar ファイルを含むクラス・ファイルの詳細は、Java API のドキュメント

このタイプのアダプタ・タスクを作成する手順は、次のとおりです。

1. Oracle Identity Manager API タスクを追加するアダプタ（Get User's Password アダプタなど）を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 「Add」をクリックします。  
「Adapter Task Selection」ウィンドウが表示されます。
4. 「Utility Task」オプションを選択します。
5. オプションの右側の表示領域から、Xellerate API を選択し、「Continue」をクリックします。「Object Instance Selection」ウィンドウが表示されます。

**関連項目：** このウィンドウの詳細は、3-7 ページの「Java タスクの作成」

6. 「Continue」をクリックします。「Add an Adapter Factory Task」ウィンドウが表示されます。

7. 「Task Name」フィールドに、作成する Oracle Identity Manager API タスクの名前 (Retrieve Password など) を入力します。
8. (オプション。) Oracle Identity Manager API タスクを再利用可能にするには、「Persistent Instance」を選択します。次に、チェック・ボックスの右側のテキスト・フィールドに、このタスクのインスタンス名を入力します。

**ヒント:** インスタンス名に空白が含まれていないことを確認してください。

**関連項目:**

「Add an Adapter Factory Task」ウィンドウの領域の詳細は、3-7 ページの [「Java タスクの作成」](#)

Oracle Identity Manager でサード・パーティ JAR ファイルを Java タスクとともに使用する方法については、2-2 ページの [「JAR およびクラス・ファイル参照のための Oracle Identity Manager の構成」](#)

9. 「Application API」リストをクリックします。x1API.jar ファイルに属するクラス・ファイルが表示されます。

---

**注意:** x1API.jar ファイルには、Oracle Identity Manager API タスクで使用できるすべてのクラス・ファイルが含まれています。したがって、「API Source」リストにアクセスする必要はありません。

---

10. 「Add an Adapter Factory Task」ウィンドウで、クラス・ファイル、コンストラクタおよびメソッドを選択します。次に、コンストラクタおよびメソッドのパラメータのマッピングを設定します。

**関連項目:** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」

11. 「Add an Adapter Factory Task」ウィンドウのツールバーで **「Save」** をクリックします。Oracle Identity Manager API タスクに関する情報が、Oracle Identity Manager データベースに格納されます。
12. 「Add an Adapter Factory」ウィンドウを閉じ、メイン画面をアクティブにします。作成した Oracle Identity Manager API タスク (Retrieve Password など) が、「Adapter Factory」フォームに表示されます。
13. (オプション。) アダプタに追加の Oracle Identity Manager API タスクを作成するには、手順 3 ~ 12 を繰り返します。

これで、アダプタをコンパイルし、プロセス・タスクまたはフォーム・フィールドで使用できます。

14. アダプタをコンパイルするには、**「Build」** をクリックします。

**「Compile Status」** フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、Oracle Identity Manager がサード・パーティ・アプリケーションと通信できるように、アダプタをプロセス・タスクまたはフォーム・フィールドにアタッチできます。

### 3.5.6 アダプタ変数値の再割当て

アダプタが必要な目的を完了できるように、アダプタ変数の値を、別のアダプタ変数、異なるタイプのアダプタ・タスクまたは定数（またはリテラル）に再度割り当てる必要がある場合があります。アダプタ変数値の再割当てを可能にするタスクを、変数設定タスクと言います。

**関連項目：** 3-3 ページの「[アダプタ変数について](#)」

たとえば、ユーザー ID の長さが 11 文字未満の場合に、アダプタ変数の戻り値をアダプタ・タスク（UserName）の出力と同じ値に設定する変数設定タスクを作成できます。

変数設定タスクを作成する手順は次のとおりです。

1. 変数設定タスクを追加するアダプタ（Check the Solaris User ID アダプタなど）を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 「Add」をクリックします。「Adapter Task Selection」ウィンドウが表示されます。
4. 「Logic Task」オプションを選択します。
5. 表示領域から SET VARIABLE を選択し、「Continue」をクリックします。「Add Set Variable Task Parameters」ウィンドウが表示されます。
6. 「Variable Name」リストから、再割当てする値（アダプタ戻り値など）を持つアダプタ変数を選択します。
7. 「Operand Type」リストから、変数に値を提供するオペランドのタイプを選択します。

**ヒント：** アダプタ変数値を、別のアダプタ変数、異なるタイプのアダプタ・タスクまたはリテラルに再割当てできます。

表 3-5 を使用して、様々なオペランドのタイプを理解してください。

**表 3-5 オペランドのタイプ**

オペランド名	説明
Variable	このオペランド・タイプを選択すると、アダプタ変数が「Operand Qualifier」リストに表示されます。このリストから、再割当て値を指定するアダプタ変数を選択します。 <b>注意：</b> 「Operand Qualifier」コンボ・ボックスには、「Variable Name」コンボ・ボックスに表示されるアダプタ変数と同じデータ型のアダプタ変数のみが表示されます。
Adapter Tasks	このオペランド・タイプを選択すると、アダプタ・タスクが「Operand Qualifier」コンボ・ボックスに表示されます。このコンボ・ボックスから、再割当て値を指定するアダプタ・タスクを選択します。 <b>注意：</b> 「Operand Qualifier」コンボ・ボックスには、「Variable Name」コンボ・ボックスに表示されるアダプタ変数と同じデータ型のアダプタ・タスクのみが表示されます。
Literal	このオペランド・タイプを選択すると、リテラルのタイプが「Operand Qualifier」コンボ・ボックスに表示されます。このコンボ・ボックスから、再割当て値を指定するリテラルのタイプを選択します。次に、コンボ・ボックスの下に表示されるフィールドにリテラルを入力します。

次のタスクでは、アダプタ変数の戻り値を `UserName` アダプタ変数と等しくなるように設定します。

1. 「Add Set Variable Task Parameters」ウィンドウのツールバーで、「**Save**」をクリックします。作成した変数設定タスクが、Oracle Identity Manager データベースに格納されます。
2. 「Add Set Variable Task Parameters」ウィンドウのツールバーで、「**Close**」をクリックします。「Add Set Variable Task Parameters」ウィンドウが閉じ、メイン画面が再びアクティブになります。作成した変数設定タスク (`Set Adapter return value = UserName` など) が、「Adapter Factory」フォームに表示されます。
3. (オプション。) アダプタに追加の変数設定タスクを作成するには、手順 3 ~ 9 を繰り返します。  
これで、アダプタをコンパイルし、プロセス・タスクまたはフォーム・フィールドで使用できます。
4. アダプタをコンパイルするには、「**Build**」をクリックします。「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。Oracle Identity Manager がアダプタをコンパイルし、エラーは発見されませんでした。アダプタをプロセス・タスクまたはフォーム・フィールドにアタッチできます。

### 3.5.7 エラー・ハンドラ・タスクの追加

エラー・ハンドラ・タスクを作成する手順は次のとおりです。

1. アダプタ・タスクからエラーが返る場合があります。これが発生すると、アダプタがアタッチされたプロセス・タスクまたはフォーム・フィールドは拒否されます。  
カスタマイズ可能なエラー・メッセージをアタッチして、ユーザーに表示することができます。このメッセージをエラー・ハンドラ・タスクと言います。  
たとえば、ユーザー ID の長さが 10 文字を超える場合にエラー・メッセージを表示するエラー・ハンドラ・タスクをアダプタにアタッチできます。
2. エラー・ハンドラ・タスクを追加するアダプタ (Check the Solaris User ID アダプタなど) を選択します。
3. 「Adapter Tasks」タブをクリックします。
4. 「**Add**」をクリックします。  
「Adapter Task Selection」ウィンドウが表示されます。
5. 「**Logic Task**」オプションを選択します。
6. 表示領域からエラー処理を選択し、「**Continue**」をクリックします。「Add an Adapter Factory Task」ウィンドウが表示されます。
7. このウィンドウの「**Lookup**」フィールドをダブルクリックします。アダプタに追加可能なエラー・ハンドラ・タスクを示す「**Lookup**」ウィンドウが表示されます。

---

**注意:** この「Lookup」ウィンドウには、ADAPTER で始まるエラー・ハンドラ・タスク (ADAPTER.USERIDLENERR など) のみが表示されます。

アダプタに組み込むエラー・ハンドラ・タスクが表示されない場合、「Error Message Definition」フォームにアクセスして作成します。「Error Message Definition」フォームの詳細は、『Oracle Identity Manager デザイン・コンソール・ガイド』を参照してください。

---

8. 使用するエラー・ハンドラ・タスク (ADAPTER.USERIDLENERR など) を選択します。
9. 「**OK**」をクリックします。「Lookup」ウィンドウが閉じ、「Add an Adapter Factory Task」ウィンドウがアクティブになります。また、選択したエラー・ハンドラ・タスクが、このウィンドウのフィールドに表示されます。

10. 「Add an Adapter Factory Task」ウィンドウのツールバーで「Save」をクリックします。アダプタに組み込んだエラー・ハンドラ・タスクが、Oracle Identity Manager データベースに格納されます。
11. このウィンドウのツールバーで、「Close」をクリックします。  
「Add an Adapter Factory Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。追加したエラー・ハンドラ・タスク (Handle Error.ADAPTER.USERIDLENERR など) が、「Adapter Factory」フォームの子表に表示されます。
12. (オプション。) アダプタに追加のエラー・ハンドラ・タスクを作成するには、手順 3 ~ 10 を繰り返します。  
アダプタが論理的に完全で、アダプタ・タスクのすべての変数がマップ済の場合、アダプタをコンパイルして、プロセス・タスクまたはフォーム・フィールドで使用できます。
13. アダプタをコンパイルするには、「Build」をクリックします。  
「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチできます。

**関連項目：**

- 「Error Message Definition」フォームを使用したエラー・メッセージの作成の詳細は、『Oracle Identity Manager デザイン・コンソール・ガイド』
- カスタム・リソース・バンドルを使用したユーザー定義データのローカライズの詳細は、『Oracle Identity Manager グローバリゼーション・ガイド』

### 3.5.8 ロジック・タスクの作成

アダプタを定義するときに、アダプタに条件文を追加して論理フローを管理することができます。この条件文をロジック・タスクと言います。たとえば、ユーザー ID の長さが 10 文字を超える場合にアクションをトリガーするロジック・タスクを作成できます。

ロジック・タスクを作成する手順は次のとおりです。

1. ロジック・タスクを追加するアダプタ (Check the Solaris User ID アダプタなど) を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 「Add」をクリックします。「Adapter Task Selection」ウィンドウが表示されます。
4. 「Logic Task」オプションを選択します。
5. 表示領域から、作成するロジック・タスクのタイプを選択します。次に、「Continue」をクリックします。

---

**注意：** 条件式を選択して「Continue」をクリックすると、次のいずれかのアクションが発生します。

Oracle Identity Manager が条件文をアダプタに直接追加します。または、構成する条件式に関係するフィールドを含む別のウィンドウが表示されます。

---



特定の条件文を選択した場合に発生する内容については、表 3-6 を参照してください。

**表 3-6 特定の条件文から発生するアクション**

条件文	文をアダプタに直接追加	セカンダリ・ウィンドウの表示
FOR		X
WHILE		X
IF		X
ELSE	X	
ELSE IF		X
BREAK	X	
RETURN	X	
CONTINUE	X	

表 3-7 では、「Add Adapter Factory Logic Task Parameters」ウィンドウの様々な領域を説明します。

**表 3-7 「Add Adapter Factory Logic Task」ウィンドウの領域**

名前	説明
Operand Type	このコンボ・ボックスには、アダプタ・タスクおよびアダプタ変数などのオペランドのタイプが表示されます。
Comparator Combo Box	このコンボ・ボックスで、2つのオペランドの関係 (<, =, > など) を設定できます。
Operand Qualifier	このコンボ・ボックスには、オペランドの修飾子が表示されません。
Literal Text Box	「Literal」オペランド・タイプを選択した場合、このフィールドにリテラルを入力します。

**注意：** FOR 条件文を選択すると、「Add Adapter Factory Logic Task Parameters」ウィンドウが表示されます。ただし、異なるテキスト・ボックスとコンボ・ボックスが表示されます。

FOR 条件文の場合は、表 3-8 で「Add Adapter Factory Logic Task Parameters」ウィンドウの領域を確認してください。

**表 3-8 FOR 条件文の場合の「Add Adapter Factory Logic Task Parameters」**

名前	説明
Operand Type	このコンボ・ボックスには、アダプタ・タスクおよびアダプタ変数などのオペランドのタイプが表示されます。
Comparator Combo Box	このコンボ・ボックスで、2つのオペランドの関係 (<, =, > など) を設定できます。
Operand Qualifier	このコンボ・ボックスには、オペランドの修飾子が表示されません。
Increment Combo Box	この領域で、初期値の増減、およびその割合を設定できます。

**注意：** ELSE、BREAK、RETURN または CONTINUE 条件文を選択する場合は、手順 8 に進んでください。

6. 条件文にパラメータを設定します。  
このロジック・タスクでは、ユーザー ID の長さが 10 文字を超えているかどうかを確認します。
7. 「Add Adapter Factory Logic Task Parameters」ウィンドウのツールバーで、「Save」をクリックします。  
作成したロジック・タスクが、Oracle Identity Manager データベースに格納されます。
8. このウィンドウのツールバーで、「Close」をクリックします。「Add Adapter Factory Logic Task Parameters」ウィンドウが閉じ、メイン画面が再びアクティブになります。作成したロジック・タスク (If (Check ID Length > 10) など) が、「Adapter Factory」フォームに表示されます。
9. (オプション。) アダプタに追加のロジック・タスクを作成するには、手順 3～8 を繰り返します。

---

**注意：**ロジック・タスクの条件で実行できるすべてのアダプタ・タスクは、ロジック・タスクの下に正しくネストする必要があります。

---

**関連項目：**タスクのネストの詳細は、3-23 ページの「[タスクの順序とネ스팅の変更](#)」

これで、アダプタをコンパイルし、プロセス・タスクまたはフォーム・フィールドで使用できます。

10. アダプタをコンパイルするには、「Build」をクリックします。  
「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにアタッチできます。

## 3.6 アダプタ・タスクの変更

次の手順では、変更が必要になった場合のアダプタ・タスクの編集方法を説明します。アダプタ・タスクを変更する手順は次のとおりです。

1. 編集するアダプタ・タスクを含むアダプタ (Update Solaris User Group アダプタなど) を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 変更するアダプタ・タスクをダブルクリックします。  
選択したアダプタ・タスクに関する情報を示す「Edit Adapter Factory Task Parameters」ウィンドウが表示されます。このウィンドウで、必要な変更を行います。
4. 「Edit Adapter Factory Task Parameters」ウィンドウのツールバーで、「Save」をクリックします。  
変更した情報が、Oracle Identity Manager データベースに格納されます。
5. ツールバーで「Close」をクリックします。  
「Edit Adapter Factory Task Parameters」ウィンドウが閉じます。メイン画面が再びアクティブになります。変更したタスクが、「Adapter Factory」フォームの子表に表示されます。プロセス・タスクまたはフォーム・フィールドで使用できるように、アダプタを再コンパイルする必要があります。

6. アダプタをコンパイルするには、「Build」をクリックします。

「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチできます。

---

**注意：** Java、Xellerate API またはユーティリティ・タスク内の API コールは変更できません。アダプタ・タスクを削除し、再作成する必要があります。

また、「Compile Status」フィールドに CODE GEN ERROR が表示される場合は、アダプタのコンパイル中に、Oracle Identity Manager でエラーが発生しています。エラーを修正し、必要に応じてアダプタ・タスクを変更し、アダプタを再コンパイルしてください。

---

## 3.7 タスクの順序とネスティングの変更

アダプタに複数のタスクを追加する場合、アダプタを動作させるために、タスクを実行する順序を変更するか、あるタスクを別のタスク内に入れることができます。

次の手順では、タスクの順序とネスティングの変更方法を説明します。

---

**注意：** 必ずアダプタ・タスクのマッピングの依存状態を理解した上で、アダプタ・タスクの順序とネスティングを変更してください。

---

タスクの順序とネスティングを変更する手順は次のとおりです。

1. 順序を変更するかネストする（あるいはその両方）タスクを含むアダプタ（Check the Solaris User ID アダプタなど）を選択します。

2. 「Adapter Tasks」タブをクリックします。

現行のアダプタに属するタスクが表示されます。

この例では、次のように変更する必要があります。

- エラー・ハンドラ・タスクを、IF (Check ID Length > 10) ロジック・タスク内にネストする必要があります。
- 変数設定タスクを、ELSE ロジック・タスク内にネストする必要があります。
- IF ロジック・タスクを ELSE ロジック・タスクより前にします。

そのため、まず、ロジック・タスクを再編成する必要があります。次に、エラー・ハンドラ・タスクと変数設定タスクを、IF および ELSE ロジック・タスク内にそれぞれネストする必要があります。タスクを再編成する手順は次のとおりです。

3. 他のタスクの前に実行する必要があるタスクを選択し、「Up」矢印ボタンをクリックします。選択したタスクの位置が、その前のタスクと入れ替わります。

または

他のタスクの後に実行する必要があるタスクを選択し、「Down」矢印ボタンをクリックします。選択したタスクが、下にあったタスクの下に表示されます。

タスクをネストする / タスクのネスティングを削除する手順は、次のとおりです。

4. 他のタスク内に置く必要があるタスクを選択し、「Right」矢印ボタンをクリックします。選択したタスクが、その上に表示されるタスクの中にネストされます。

または

他のタスク内にネストしなくなったタスクを選択し、「Left」矢印ボタンをクリックします。選択したタスクで、その上に表示されるタスクの中でのネスティングが解除されます。

5. ツールバーで「Save」をクリックします。

アダプタ・タスクの順序とネスティングが、Oracle Identity Manager データベースに格納されます。アダプタが論理的に完全で、アダプタ・タスクのすべての変数がマップ済の場合、アダプタをコンパイルして、プロセス・タスクまたはフォーム・フィールドで使用できます。

6. アダプタをコンパイルするには、「Build」をクリックします。

「Compile Status」フィールドのテキストが、「Recompile」から「OK」に変わります。これは、Oracle Identity Manager がアダプタをコンパイルし、エラーが発見されなかったことを示します。これで、アダプタをプロセス・タスクまたはフォーム・フィールドにタッチできます。

---

**注意：**「Compile Status」フィールドに *CODE GEN ERROR* が表示される場合は、アダプタのコンパイル中に、Oracle Identity Manager でエラーが発生しています。エラーを修正し、必要に応じてアダプタ・タスクを変更し、アダプタを再コンパイルしてください。

---

## 3.8 アダプタ・タスクの削除

アダプタ・タスクがアダプタの実行に不要になった場合は、アダプタから削除する必要があります。アダプタ・タスクを削除する手順は次のとおりです。

1. 削除するタスクを含むアダプタ（Update Solaris User Group アダプタなど）を選択します。
2. 「Adapter Tasks」タブをクリックします。
3. 削除するタスク（CONTINUE ロジック・タスクなど）を選択します。
4. 「Delete」をクリックします。  
選択したタスクが削除され、子表に表示されなくなります。
5. ツールバーで「Save」をクリックします。
6. アダプタを再コンパイルします。

---

**注意：**アダプタ・タスクを削除する場合は、アダプタのロジックが矛盾なく維持されていることを確認してください。

---

## 3.9 レスポンスの使用

アダプタには**レスポンス**と呼ばれる多様な結果があります。このレスポンスに基づき、アダプタが他のプロセス・タスクをトリガーできます。

たとえば、アダプタから **True** レスポンスが返ると、プロセス・タスクのステータスは自動的に「Completed」に設定されます。しかし、アダプタから **False** レスポンスが返ると、プロセス・タスクのステータスは自動的に「Rejected」に設定され、別のプロセス・タスクがトリガーされます。

このレスポンスは、「Adapter Factory」フォームの「Responses」タブで追加、変更または削除できます。

次の手順では、レスポンスの作成、変更および削除の方法を説明します。

---

**注意：** これらのアダプタはプロセス・タスクにアタッチされているため、レスポンスはプロセス・タスク・アダプタのみで使用されます。ルール・ジェネレータ、事前移入アダプタおよびエンティティ・アダプタはプロセスには関連付けられません。タスク割当てアダプタも、レスポンスに関連付けられません。そのため、アクティブなアダプタが、タスク割当て、ルール・ジェネレータ、事前移入アダプタまたはエンティティ・アダプタである場合、Oracle Identity Manager の「Responses」タブは無効化されています。

---

### 3.9.1 レスポンスの作成

1. レスポンスを追加するアダプタ（Create Solaris User アダプタなど）を選択します。
2. 「Responses」タブをクリックします。
3. 「Add」をクリックします。

「Responses」タブに空白行が挿入されます。

4. 「Code Name」列内に表示されるフィールドをクリックします。
5. 生成可能なレスポンス・タイプを表すコード（True など）を入力します。
6. 「Description」列内に表示されるフィールドをクリックします。
7. レスポンスの説明（「ユーザーは正常に作成されました。」など）を入力します。
8. 「Status」列内に表示されるフィールドをダブルクリックします。

レスポンスに関連付け可能な様々なステータス・レベルを示す「Lookup」ポップアップ・ウィンドウが表示されます。

---

**注意：** Oracle Identity Manager のステータス・レベルの詳細は、[第4章「プロセス・タスク・アダプタについて」](#)を参照してください。

---

9. 目的のステータス・レベル（「Completed (C)」など）をクリックします。次に、「OK」をクリックします。

「Lookup」ウィンドウが閉じ、「Responses」タブが再びアクティブになります。

10. 「Add」ボタンをクリックして、「Code Name」フィールドには **False** と入力し、「Description」フィールドには「ユーザーは正常に作成されませんでした。」と入力して、もう1つのレスポンスを作成します。次に、「Lookup」ウィンドウにアクセスし、「Rejected (R)」のステータス・レベルをこのレスポンスに割り当てます。
11. ツールバーで「Save」をクリックします。

このアダプタに対して作成したレスポンスが、Oracle Identity Manager データベースに格納されます。このアダプタをプロセス・タスクにアタッチすると、レスポンスは「Process Definition」フォームの「Editing Task」ウィンドウにある「Responses」タブに表示されません。

### 3.9.2 レスポンスの変更

次の手順では、レスポンスを編集する方法を説明します。

1. 編集するレスポンスを含むアダプタ（Create Solaris User アダプタなど）を選択します。
2. 「Responses」タブをクリックします。
3. 変更する情報を含むレスポンスのフィールドをダブルクリックします。
  - a. フィールドがテキスト・フィールドの場合、Oracle Identity Manager ではフィールドが有効化されています。これでフィールドの内容を編集できます。
  - b. フィールドが「Lookup」フィールドの場合、レスポンスに関連付け可能な様々なステータス・レベルを示す「Lookup」ポップアップ・ウィンドウが表示されます。目的のステータス・レベルをクリックし、「OK」をクリックします。

たとえば、False レスポンスの「Status」列をダブルクリックし、「Suspended (S)」のステータス・レベルを選択して「OK」をクリックします。

4. ツールバーで「Save」をクリックします。

変更したレスポンスの情報が、Oracle Identity Manager データベースに格納されます。

### 3.9.3 レスポンスの削除

レスポンスが不要になった場合は、アダプタから削除できます。

1. 削除するレスポンスを含むアダプタを選択します。
2. 「Responses」タブをクリックします。
3. 削除するレスポンスを選択します。
4. 「Delete」をクリックします。

レスポンスが表示されなくなります。これは、Oracle Identity Manager がレスポンスを削除したことを意味します。

## 3.10 ルール・ジェネレータとエンティティ・アダプタのスケジュール設定

Oracle Identity Manager では、プロセス・タスク・アダプタまたはタスク割当てアダプタがプロセス・タスクにアタッチされ、プロセス・タスクのステータスが「Pending」である場合に、自動的にこれらをトリガーします。また、Oracle Identity Manager は、常に事前移入アダプタを挿入前にトリガーします。そのため、プロセス・タスク・アダプタ、タスク割当てアダプタまたは事前移入アダプタが実行される場合は、スケジュール設定はしません。

一方、ルール・ジェネレータおよびエンティティ・アダプタは、フォーム・フィールドにアタッチされます。ルール・ジェネレータまたはエンティティ・アダプタは、トリガーされないと、Oracle Identity Manager で実行できません。これは「Execution Schedule」タブで指定します。

---

---

**注意：**エンティティ・アダプタが、フィールド値の検証用にプロセス・フォームまたはオブジェクト・フォームにアタッチされている場合、ダイレクトまたはリクエスト・プロビジョニングの完了後、これらのフォームのデータを編集すると、アダプタがトリガーされます。

---

---

このタブを使用して、Oracle Identity Manager が挿入前または更新前にルール・ジェネレータまたはエンティティ・アダプタをトリガーするように決定できます。また、削除前、挿入後、更新後および削除後にエンティティ・アダプタを実行するようスケジュール設定できます。

この手順では、Oracle Identity Manager を構成して、ルール・ジェネレータまたはエンティティ・アダプタをトリガーする方法を説明します。

### 3.10.1 ルール・ジェネレータとエンティティ・アダプタのスケジュール設定

ルール・ジェネレータおよびエンティティ・アダプタにスケジュールを設定する手順は次のとおりです。

1. Oracle Identity Manager でトリガーするルール・ジェネレータまたはエンティティ・アダプタ (Solaris User ID Generator など) を選択します。

---

**注意：** プロセス・タスク・アダプタまたは事前移入アダプタの作業では、「Execution Schedule」タブは使用しません。その場合、このタブとコンテンツは無効化されています。

---

2. 「Execution Schedule」タブをクリックします。

「Execution Schedule」タブのコンテンツが表示されます。

次の表は、「Execution Schedule」タブの様々なチェック・ボックスの理解に役立ちます。

名前	説明
Pre-Insert	このチェック・ボックスをクリックすると、レコードをデータベースに挿入する前に、Oracle Identity Manager でルール・ジェネレータまたはエンティティ・アダプタをトリガーできます。
Pre-Update	このチェック・ボックスをクリックすると、レコードがデータベースで更新される前に、Oracle Identity Manager でルール・ジェネレータまたはエンティティ・アダプタをトリガーできます。
Pre-Delete	このチェック・ボックスをクリックすると、レコードがデータベースで削除される前に、Oracle Identity Manager でエンティティ・アダプタをトリガーできます。
Post-Insert	このチェック・ボックスをクリックすると、レコードをデータベースに挿入した後に、Oracle Identity Manager でエンティティ・アダプタをトリガーできます。
Post-Update	このチェック・ボックスをクリックすると、レコードがデータベースで更新された後に、Oracle Identity Manager でエンティティ・アダプタをトリガーできます。
Post-Delete	このチェック・ボックスをクリックすると、レコードがデータベースから削除された後に、Oracle Identity Manager でエンティティ・アダプタをトリガーできます。

---

**注意：** 「Execution Schedule」タブのチェック・ボックスをクリックして、Oracle Identity Manager がルール・ジェネレータまたはエンティティ・アダプタをいつトリガーできるかを定義します。「Data Object Manager」フォームでは、Oracle Identity Manager がルール・ジェネレータまたはエンティティ・アダプタをいつトリガーするかを指定できます。

「Data Object Manager」フォームの詳細は、6-3 ページの「[ルール・ジェネレータのアダプタ変数のマッピング](#)」を参照してください。

---

3. 目的のチェック・ボックスを有効化します。次に、ツールバーで「Save」をクリックします。

設定した Oracle Identity Manager でルール・ジェネレータまたはエンティティ・アダプタを実行するための基準は、Oracle Identity Manager データベースに格納されます。





---

## プロセス・タスク・アダプタについて

この章では、プロセス・タスク・アダプタについて説明します。内容は次のとおりです。

- プロセスおよびプロセス・タスクの概要
- プロセス・タスク・アダプタの動作
- プロセス・タスク・アダプタのプロセス・タスクへのアタッチ
- プロセス・タスク・アダプタのプロセス・タスクからの削除

## 4.1 プロセスおよびプロセス・タスクの概要

プロセス・タスク・アダプタは、アダプタ・ファクトリで作成した Java コードです。これにより、Oracle Identity Manager ではプロビジョニング・プロセスでプロセス・タスクを自動的に実行できます。

それぞれのプロセスおよびプロセス・タスクにはステータスがあり、完了の段階を示します。次の表に、プロセスまたはプロセス・タスクのステータスを、重要性の高いものから示します。

**注意：** UT、UCR および XLR のステータスについては、次の表では説明しません。これらのステータスの値は、推奨されておらず、Oracle Identity Manager の将来のリリースでは削除されます。

タスク・ステータス	説明
C	Completed: このプロセスまたはプロセス・タスクは正常に完了しました。
MC	Manually Completed: このプロセス・タスクは、Oracle Identity Manager ユーザーにより（手動で）正常に完了しました。
P	Pending: このプロセスまたはプロセス・タスクは、完了処理中です。先行するプロセスおよびプロセス・タスクはすべて完了しています。
PX	Pending Cancellation: このプロセス・タスクは取り消されますが、取り消す前に完了する必要があります。
R	Rejected: このプロセスまたはプロセス・タスクは、正常に完了しましたが、承認されていません。却下されたプロセス・タスクのステータスは、「Cancelled」または「Unsuccessfully Completed」のみに変更できます。
S	Suspended: このプロセスまたはプロセス・タスクは一時的に停止しています。
UC	Unsuccessfully Completed: このプロセス・タスクは、「Completed」に設定されています。しかし、タスクは以前に却下されたことがあります。
W	Waiting: このプロセスまたはプロセス・タスクは、先行するすべてのプロセス・タスクまたはプロセスが完了するまで、完了しません。
X	このプロセスまたはプロセス・タスクは停止しています。ステータスを変更できません。

プロセスのステータス・レベルは、そのプロセス・タスクの最も重要なステータス・レベルを表します。これが完了しないと、プロセスは完了しません。プロセスに異なるステータス・レベル（「Completed」、「Waiting」および「Rejected」）を持つ3つのプロセス・タスクがある場合、3つのすべてのプロセス・タスクを完了するまでは、プロセスが完了しません。「Rejected」が最高のステータス・レベルであるため、プロセスのステータス・レベルも「Rejected」となります。

プロセス・タスクは、次のいずれかの方法で管理できます。

- 「Organizations」または「Users」フォームの「Object Process Console」タブまたは Oracle Identity Manager Web Application で、手動で処理します。
- タスクのいずれか（または複数）が「Pending」ステータスに到達した際に自動的にトリガーされるように、Oracle Identity Manager プロセスを構成します。

Oracle Identity Manager でプロセス・タスクを自動的に実行する Java コードは、プロセス・タスク・アダプタと呼ばれます。開発者がプロセス・タスク・アダプタを作成および管理するための Oracle Identity Manager ツールは、アダプタ・ファクトリと呼ばれます。

---

---

**注意：**プロセス・タスクの手動処理の詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』を参照してください。

---

---

## 4.2 プロセス・タスク・アダプタの動作

プロセス・タスク・アダプタを作成し、「Process Definition」フォームの「Integration」タブを使用して、アダプタを適切なプロセス・タスクにアタッチします。このタブでは、「Resolve at Run time」またはアダプタ戻り変数のいずれかとして指定されたアダプタ変数を、適切な場所にマップすることもできます。

たとえば、adpSOLARISPASSWORDUPDATED という名前のアダプタが、Solaris プロセスの Password Updated タスクに関連付けられているとします。

アダプタを機能させるためにアダプタをプロセス・タスクにアタッチすると、別のフォームのフィールドからデータが必要になる場合があります。この例では、次の情報を取得するまで adpSOLARISPASSWORDUPDATED アダプタは動作しません。

- ユーザーの Oracle Identity Manager ID およびパスワード
- ユーザーの Solaris ID およびパスワード
- Solaris が位置する IP アドレス

そのため、UserID、Passwd、SolarisUserID、SolarisUserPasswd および ServerAddress アダプタ変数から、それぞれ情報を取得する必要があります。この 5 つの変数は、「Adapter Factory」フォームを使用して作成されます。それぞれのアダプタ変数の先頭の Y は、アダプタが正常にマップされたことを表します。

プロセスが必要な情報を取得するために使用するプロセス固有フィールドを作成するためのフォームは「Form Designer」と呼ばれます。作成したフィールドは、Oracle Identity Manager により表に格納されます。次に、この表をプロセスに関連付けると（「Process Definition」フォームの「Table Name」検索フィールドを使用）、このプロセスのタスクにアタッチされたアダプタは、表を使用して適切なデータを検索します。

この表を変更する場合は、「Form Designer」フォームで変更できます。

UD\_SOLARIS 表には、UD\_SOLARIS\_USERID と UD\_SOLARIS\_PASSWD という 2 つの表があります。「Form Designer」フォームでこのレコードにアクセスすると、表のフィールドを編集できます。

---

---

**注意：**「Form Designer」フォームでプロセス固有フィールドを作成および変更する方法の詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』を参照してください。

---

---

プロセス・タスク・アダプタを依存プロセス・タスクにアタッチすると、このプロセス・タスクのステータスは「Pending」になり（先行するプロセス・タスクのステータスは「Completed」）、Oracle Identity Manager は自動的にアダプタをトリガーします。プロセス・タスクが独立したタスクの場合、Oracle Identity Manager はプロセスがリクエストされた時点でアダプタを実行します。

実行されたアダプタの結果は、プロセス・タスクのステータスを表します。アダプタが正常に終了した場合、アダプタをアタッチしたプロセス・タスクのステータスは「Completed」です。

一方、アダプタで指定された機能が実行できない場合、アダプタをアタッチしたプロセス・タスクのステータスは「Rejected」です。エラーの原因を突き止めることで、正常に実行されるようにプロセス・タスクまたはアダプタ（あるいはその両方）を変更できます。

---

---

**注意:** プロセス・タスクが失敗した原因を確認する手順は次のとおりです。

プロセス・タスクを探します。プロセス・タスクをターゲット・ユーザーまたは組織にプロビジョニングしていない場合は、「To-Do List」または「Pending Approvals」の中にあります。タスクの検出手順は次のとおりです。

1. ユーザーとしてログインします。
  2. ウィンドウの左側の「To-Do List」リンクまたは「Pending Approvals」リンクを選択します。
- 
- 

## 4.3 プロセス・タスク・アダプタのプロセス・タスクへのアタッチ

前の章では、プロセス・タスク・アダプタの作成方法を説明しました。プロセス・タスクを自動的に実行するためには、プロセス・タスク・アダプタをプロセス・タスクにアタッチする必要があります。

アダプタをプロセス・タスクに関連付けるには、「Process Definition」フォームから「Integration」タブにアクセスします。このタブでは、アダプタ変数を適切な場所にマップすることもできます。

次の手順では、プロセス・タスク・アダプタをプロセス・タスクにアタッチする方法を説明します。

1. 「Process Management」フォルダにある「Process Definition」フォームを開きます。  
Oracle Identity Manager ワークスペースに「Process Definition」フォームが表示されます。
2. アダプタをアタッチするタスクを含むプロセスを選択します。選択したプロセスとそのタスクが、「Process Definition」フォームに表示されます。この例では、Solaris プロセスが選択されています。
3. アダプタをアタッチするタスクの行ヘッダーをダブルクリックします。タスクの情報（たとえば、Password Updated プロセス・タスクなど）を含む「Editing Task」ウィンドウが表示されます。
4. 「Integration」タブをクリックします。
5. 「Add」をクリックします。  
「Handler Selection」ウィンドウが表示されます。
6. 「Adapter」オプションをクリックして、Oracle Identity Manager のアダプタにアクセスします。  
プロセス・タスクをアタッチできるアダプタが表示されます。
7. この領域から、プロセス・タスクにアタッチするアダプタ (adpSOLARISPASSWORDUPDATED アダプタなど) を選択します。  
  
**ヒント:** 各アダプタの名前は、区別しやすいよう、最初の 3 文字が **adp** です。各アダプタの名前は、区別しやすいよう、最初の 3 文字が **adp** です。
8. 「Handler Selection」ウィンドウのツールバーで「Save」をクリックします。  
アダプタが正常にプロセス・タスクにアタッチされたことを示すダイアログ・ボックスが表示されます。

9. 「OK」をクリックします。

ダイアログ・ボックスが閉じ、「Integration」タブがアクティブになります。タブには次の項目が表示されます。

- プロセス・タスクにアタッチされたアダプタの名前
- アダプタのステータス
- アダプタ変数の名前、説明およびマッピング・ステータス

---

---

**注意：**アダプタのマッピング・ステータスは、次の3つのいずれかです。

**Ready.** アダプタは正常にコンパイルされ、すべての変数が正常にマップされています。

**Mapping Incomplete.** アダプタは正常にコンパイルされましたが、少なくとも1つの変数が正常にマップされていません。

**Adapter Unavailable.** アダプタは、正常にコンパイルされた後、変更および再コンパイルされました。

---

---

---

---

**注意：**アダプタにマップ可能な変数がない場合、「Adapter Variable」領域は空白です。また、「Status」フィールドには、アダプタを再コンパイルする必要があるかどうかによって、「Ready」または「Adapter Unavailable」のいずれかが表示されます。

---

---

---

---

**注意：**マップ可能なアダプタ変数は、「Resolve at Run time」に指定されているか、アダプタ戻り変数のいずれかです。

---

---

---

---

**注意：**アダプタをプロセス・タスクにアタッチすると、アダプタに定義したレスポンスが「Editing Task」ウィンドウの「Responses」タブに表示されます。

---

---

10. 「Integration」タブの「Adapter Variable」領域に表示されるそれぞれの変数にマッピングを設定します。それには、マップする変数の行ヘッダー（SolarisUserID など）をダブルクリックします。

「Data Mapping for Variable」ウィンドウが表示されます。

表 4-1 で、「Data Mapping for Variable」ウィンドウのフィールドについて説明します。

表 4-1 「Data Mapping for Variable」ウィンドウのフィールド

フィールド名	説明
Variable Name	このフィールドには、マッピングを設定するアダプタ変数の名前 (SolarisUserID など) が表示されます。
Data Type	このフィールドには、アダプタ変数のデータ型が表示されます (たとえば、SolarisUserID 変数のデータ型は文字列です)。
Map To	このフィールドには、アダプタ変数に設定可能なマッピング型 (「IT Resources」など) が含まれます。このコンボ・ボックスから、アダプタ変数をマップする場所または連絡先の特定のタイプを選択します。また、アダプタ変数をカスタム・プロセス・フォームにマップし、そのフォームに子表が含まれる場合、Oracle Identity Manager では隣接するコンボ・ボックスが有効化されます。このコンボ・ボックスから、アダプタ変数をマップする子表を選択します。場所、連絡先またはカスタム・プロセス・フォームの子表にアダプタ変数をマップしない場合、このコンボ・ボックスは無効化されます。
Qualifier	このフィールドには、「Map To」コンボ・ボックスで選択したマッピングの修飾子が含まれます (「IT Asset」など)。
IT Asset Type	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、特定の IT リソース (Solaris など) を選択できます。 アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。
IT Asset Property	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、マッピングの結果を受け取る特定のフィールド (「User Name」など) を選択できます。 アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。 <b>重要:</b> 「IT Asset Type」および「IT Asset Property」フィールドは、下位互換性のためにこのウィンドウに含まれています。アダプタ変数を「IT Resource」データ型で作成することをお勧めします。この場合、これらのフィールドは表示されません。
Literal Value	アダプタ変数をリテラルにマップする場合、このフィールドでリテラル値を指定します。 アダプタ変数をリテラルにマップしない場合、このフィールドは表示されません。
Old Value	このチェック・ボックスを選択して、選択した「Qualifier」フィールドの変更前の値に、アダプタ変数をマップします。 プロセス・タスクに関連付けられたプロセス・タスク・アダプタは、プロセス・フォームのフィールドの一部が変更されると、条件付きでトリガーされる場合があります。「Old Value」オプションをクリックし、プロセス・タスクを条件付きとしてマークした場合、フィールドを変更する前の値がアダプタに渡されます。これは、パスワードを使用するフィールドの場合に役立ちます。たとえば、パスワードに同じ値の設定を許可しない場合、比較に古い値を使用できます。 カスタム・プロセス・フォームの子表に属するフィールドにアダプタ変数をマップしていない場合は、このチェック・ボックスは無効化されています。

- 「Map To」、「Qualifier」、「IT Asset Type」、「IT Asset Property」、「Literal Value」および「Old Value」フィールドをすべて入力します。

---

**注意：** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』のアダプタ変数マッピングに関する項を参照してください。

---

- ツールバーで「Save」をクリックします。次に、「Close」をクリックします。  
「Data Mapping for Variable」ウィンドウが閉じます。「Integration」タブが再びアクティブになります。
- 「Editing Task」ウィンドウのツールバーで「Save」をクリックします。  
「Status」フィールドのコンテンツが、「Mapping Incomplete」から「Ready」に変化します。アダプタ変数のマッピング・ステータスも、「No (N)」から「Yes (Y)」へ変化します。
- ツールバーで「Close」をクリックします。  
「Editing Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。Password Updated タスクに追加したアダプタ (adpSOLARISPASSWORDUPDATED) が、「Process Definition」フォームに表示されます。  
これは、adpSOLARISPASSWORDUPDATED プロセス・タスク・アダプタが Password Updated プロセス・タスクにアタッチされたことを意味します。

**ヒント：** プロセス・タスク・アダプタをプロセス・タスクにアタッチすると、「Adapter Factory」フォームの「Usage Lookup」タブにアクセスすることで、関連付けられたプロセスおよびタスクを簡単に確認できます。

## 4.4 プロセス・タスク・アダプタのプロセス・タスクからの削除

Oracle Identity Manager で、プロセス・タスクを自動的に完了するためのプロセス・タスク・アダプタが不要になった場合、または別のアダプタをプロセス・タスクにアタッチする場合は、まずプロセス・タスクにアタッチしたアダプタを削除する必要があります。

次の手順では、プロセス・タスク・アダプタをプロセス・タスクから削除する方法を説明します。

### 4.4.1 プロセス・タスク・アダプタのプロセス・タスクからの削除

- 「Process Definition」フォームを開きます。  
Design Console ワークスペースに、「Process Definition」フォームが表示されます。
- アダプタを削除するタスクを含むプロセス (Solaris プロセスなど) を選択します。  
選択したプロセスとそのタスクが、「Process Definition」フォームに表示されます。
- アダプタを削除するプロセス・タスク (Password Updated タスクなど) の行ヘッダーをダブルクリックします。  
プロセス・タスクの情報を含む「Editing Task」ウィンドウが表示されます。
- 「Integration」タブをクリックします。  
「Integration」タブに、プロセス・タスクにアタッチされたアダプタの情報が表示されます。
- 「Remove」をクリックします。  
アダプタをプロセス・タスクから削除することを確認するダイアログ・ボックスが表示されます。

6. 「OK」をクリックします。  
アダプタがプロセス・タスクから削除されたことを示すダイアログ・ボックスが表示されます。
7. 「OK」をクリックします。  
アダプタのコンテンツが「Integration」タブに表示されなくなります。
8. ツールバーで「Close」をクリックします。  
「Editing Task」ウィンドウが閉じ、メイン画面が再びアクティブになります。Password Updated タスクに関連付けられていたアダプタ (adpSOLARISPASSWORDUPDATED) が、「Process Definition」フォームの子表に表示されなくなります。  
これは、アダプタがプロセス・タスクから削除されたことを意味します。



---

## タスク割当てアダプタの適用

この章では、タスク割当てアダプタの適用について説明します。内容は次のとおりです。

- 5-2 ページの「概要」
- 5-3 ページの「タスク割当てアダプタのプロセス・タスクへのアタッチ」
- 5-6 ページの「タスク割当てアダプタのプロセス・タスクからの削除」

## 5.1 概要

手動で完了する必要があるプロセス・タスクに対して、Oracle Identity Manager を構成し、特定のユーザーまたは特定のユーザー・グループに属するユーザーのいずれかへのタスク割当てを自動化できます。

Oracle Identity Manager でプロセス・タスクをユーザーまたはグループに自動的に割り当てる Java コードは、タスク割当てアダプタと呼ばれます。

効率を高めるために、タスク割当てアダプタは、異なるプロセス・タスクで繰り返し使用できます。また、複数のタスク割当てアダプタを特定のタスクに関連付けできます。そのため、Oracle Identity Manager がプロセス・タスクに対するタスク割当てアダプタを選択します。これは、タスク割当てルールを通じて行われます。

タスク割当てルールは基準です。これらの基準に従い、Oracle Identity Manager は、ターゲット・ユーザーまたはグループへのプロセス・タスクの割当てに使用されるタスク割当てアダプタを選択します。

---

---

**注意：**つまり、タスク割当てルールにより、Oracle Identity Manager は、プロセス・タスクをユーザーまたはグループに割り当てるかどうかを決定します。タスク割当てアダプタにより、Oracle Identity Manager は、プロセス・タスクの受信者にするユーザーまたはグループを決定します。

---

---

それぞれのタスク割当てアダプタには、タスク割当てルールが関連付けられます。また、ルールには優先順位があり、Oracle Identity Manager はその順序でルールをトリガーします。

この例では、Oracle Identity Manager は、Associate Adapter with User ルールを最初にトリガーします（最も優先順位が高いためです）。このルールの状態が TRUE の場合、ルールは成功です。その結果、Oracle Identity Manager は、関連するタスク割当てアダプタ（Assign Task to User アダプタ）を Get Solaris UID プロセス・タスクに関連付けます。

一方、ルールの状態が FALSE の場合、ルールは失敗です。Oracle Identity Manager は、次に優先順位が高いルールをトリガーします。このルールが成功すると、Oracle Identity Manager は、指定したアダプタをターゲット・プロセス・タスクに割り当てます。

この例では、Associate Adapter with User ルールが失敗した場合、Oracle Identity Manager が Associate Adapter with Group ルールをトリガーします。このルールが成功すると、Oracle Identity Manager は、関連するタスク割当てアダプタ（Assign Task to Group アダプタ）を Get Solaris UID プロセス・タスクにアタッチします。

このタイプのアダプタにアダプタ変数が含まれる場合、ルールをタスク割当てアダプタに割り当ててから、変数を適切な場所にマップする必要があります。そうしない場合、アダプタは機能しません。

Oracle Identity Manager で、プロセス・タスクをユーザーまたはグループに関連付けるためのタスク割当てアダプタが最終的に無効または不要になった場合は、アダプタをタスクから削除する必要があります。

次の項では、タスク割当てアダプタをプロセス・タスクにアタッチする方法およびプロセス・タスクから削除する方法を説明します。

## 5.2 タスク割当てアダプタのプロセス・タスクへのアタッチ

第3章「アダプタの作成」では、タスク割当てアダプタの作成方法を説明しました。ユーザーまたはグループのタスク割当てを Oracle Identity Manager で自動化するために、タスク割当てアダプタをプロセス・タスクにアタッチする必要があります。

タスク割当てアダプタをプロセス・タスクに関連付けるには、「Process Definition」フォームから「Assignment」タブにアクセスします。このタブでは、アダプタ変数を適切な場所にマップすることもできます。

次の手順では、タスク割当てアダプタをプロセス・タスクにアタッチする方法を説明します。

### 5.2.1 タスク割当てアダプタのプロセス・タスクへのアタッチ

1. 「Process Management」フォルダにある「Process Definition」フォームを開きます。  
Oracle Identity Manager ワークスペースに「Process Definition」フォームが表示されます。
2. アダプタをアタッチするタスクを含むプロセスを選択します。  
選択したプロセスとそのタスクが、「Process Definition」フォームに表示されます。
3. タスク割当てアダプタをアタッチするタスクの行ヘッダーをダブルクリックします。  
タスク（Get Solaris UUID プロセス・タスクなど）の情報を含む「Editing Task」ウィンドウが表示されます。
4. 「Assignment」タブをクリックします。「Assignment」ダイアログ・ボックスが表示されます。
5. このタブで、「Add」をクリックします。  
「Assignment」タブ内に空白行が表示されます。

次の表に、「Assignment」タブの関連フィールドを示します。

フィールド名	説明
Priority	このフィールドで、関連付けられたタスク割当てルールの優先度を設定します。
Rule	この検索フィールドで、ユーザーまたはグループへのプロセス・タスクの割当てを自動化するために、関連付けられたアダプタを使用するかどうかを決定するルールを選択します。
Target Type	この検索フィールドで、タスクを Oracle Identity Manager のユーザーまたはグループに割り当てるかどうかを指定します。
Adapter	この検索フィールドで、指定したタスク割当てルールに関連付けるアダプタを選択します。
Adapter Status	このフィールドには、アダプタ変数のマッピング・ステータスが表示されます。 アダプタのマッピング・ステータスの詳細は、4-4 ページの「 <a href="#">プロセス・タスク・アダプタのプロセス・タスクへのアタッチ</a> 」を参照してください。

6. 「Priority」フィールドをダブルクリックします。このフィールドで、関連付けられたタスク割当てルールの優先度を設定します。
7. 「Rule」検索フィールドをダブルクリックします。表示される「Lookup」ダイアログ・ボックスから、関連付けられたアダプタを使用してプロセス・タスクをユーザーまたはグループに自動的に割り当てるかどうかを決定するルールを選択します。
8. 「Target Type」検索フィールドをダブルクリックします。表示される「Lookup」ダイアログ・ボックスで、タスクを Oracle Identity Manager ユーザーまたはグループに割り当てるかどうかを指定します。

9. 「Adapter」 検索フィールドをダブルクリックします。表示される「Lookup」ダイアログ・ボックスで、手順 7 で選択したルールに関連付けるタスク割当てアダプタを指定します。
10. 「Assignment」 タブに表示されるツールバーで「Save」をクリックします。  
タスク割当てアダプタ変数のマッピング・ステータスが、「Adapter Status」フィールドに表示されます。次の表を使用し、アダプタのマッピング・ステータスに基づいて実行するアクションを決定してください。

マッピング・ステータス	アクション
Ready	アダプタにはマッピングできる変数はありません。つまり、いずれのアダプタ変数も戻り変数ではないか、「Resolve at Run time」として指定されていません。この場合は、手順 14 に進みます。
Mapping Incomplete	少なくとも 1 つのアダプタ変数にマッピングが必要です。この場合は、手順 11 に進みます。
Adapter Unavailable	アダプタは、正常にコンパイルされた後に変更されました。その結果、アダプタの再コンパイルが必要です。

**注意：** アダプタのマッピング・ステータスの詳細は、4-4 ページの「プロセス・タスク・アダプタのプロセス・タスクへのアタッチ」を参照してください。

11. 「Map」をクリックします。  
「Adapter Variable」ウィンドウが表示されます。次の情報が表示されます。
  - プロセス・タスクにアタッチされたタスク割当てアダプタの名前
  - アダプタのステータス
  - アダプタ変数のマッピングのステータス、名前および説明
12. このウィンドウの「Adapter Variable」領域に表示される各変数にマッピングを設定します。それには、マップする変数の行ヘッダー（UUID など）をダブルクリックします。  
「Edit Data Mapping for Variable」ダイアログ・ボックスが表示されます。

表 5-1 に、「Edit Data Mapping for Variable」ダイアログ・ボックスのフィールドを示します。

表 5-1 「Edit Data Mapping for Variable」ダイアログ・ボックスのフィールド

フィールド名	説明
Variable Name	このフィールドには、マッピングを設定するアダプタ変数の名前 (UUID など) が表示されます。
Data Type	このフィールドには、アダプタ変数のデータ型が表示されます (たとえば、UUID 変数のデータ型は String です)。
Map To	<p>このフィールドには、アダプタ変数に設定可能なマッピング型 (「IT Resources」など) が含まれます。</p> <p>アダプタ変数を場所または連絡先にマップする場合、Oracle Identity Manager では隣接するコンボ・ボックスが有効化されます。このコンボ・ボックスから、アダプタ変数をマップする場所または連絡先の特定のタイプを選択します。</p> <p>また、アダプタ変数をカスタム・プロセス・フォームにマップし、そのフォームに子表が含まれる場合、Oracle Identity Manager ではフォームに隣接するコンボ・ボックスが有効化されます。このコンボ・ボックスから、アダプタ変数をマップする子表を選択します。</p> <p>場所、連絡先またはカスタム・プロセス・フォームの子表にアダプタ変数をマップしない場合、このコンボ・ボックスは無効化されます。</p>
Qualifier	このフィールドには、「Map To」コンボ・ボックスで選択したマッピングの修飾子が含まれます (「IT Asset」など)。
IT Asset Type	<p>このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、特定の IT リソース (Solaris など) を選択できます。</p> <p>アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。</p>
IT Asset Property	<p>このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、マッピングの結果を受け取る特定のフィールド (「Unique ID」など) を選択できます。</p> <p>アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。</p> <p><b>重要:</b> 「IT Asset Type」および「IT Asset Property」フィールドは、下位互換性のためにこのウィンドウに含まれています。アダプタ変数を「IT Resource」データ型で作成することをお勧めします。この場合、これらのフィールドは表示されません。</p>
Literal Value	<p>アダプタ変数をリテラルにマップする場合、このフィールドでリテラル値を指定します。</p> <p>アダプタ変数をリテラルにマップしない場合、このフィールドは表示されません。</p>
Old Value	<p>このチェック・ボックスを選択して、選択した「Qualifier」フィールドの変更前の値に、アダプタ変数をマップします。</p> <p>プロセス・タスクに関連付けられたプロセス・タスク・アダプタは、プロセス・フォームのフィールドの一部が変更されると、条件付きでトリガーされる場合があります。「Old Value」オプションをクリックし、プロセス・タスクを条件付きとしてマークした場合、以前のフィールドの値がアダプタに渡されます。これは、パスワードを使用するフィールドの場合に役立ちます。</p> <p>たとえば、パスワードに同じ値の設定を許可しない場合、比較に古い値を使用できます。</p> <p>カスタム・プロセス・フォームの子表に属するフィールドにアダプタ変数をマップしていない場合は、このチェック・ボックスは無効化されています。</p>

13. 「Map To」、「Qualifier」、「IT Asset Type」、「IT Asset Property」、「Literal Value」 および 「Old Value」 フィールドをすべて入力します。

---

**注意：** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」を参照してください。

---

14. ツールバーで「Save」をクリックします。次に、「Close」をクリックします。  
「Edit Data Mapping for Variable」ウィンドウが閉じます。「Adapter Variable」ダイアログ・ボックスが再びアクティブになります。  
「Status」フィールドのコンテンツが、「Mapping Incomplete」から「Ready」に変化します。アダプタ変数のマッピング・ステータスも、「No (N)」から「Yes (Y)」へ変化します。
15. 「Save」をクリックします。次に、「Close」をクリックします。  
「Adapter Variable」ダイアログ・ボックスが閉じ、「Assignment」タブが再びアクティブになります。  
プロセス・タスクに割り当てたアダプタ (Assign Solaris Task など) のステータスが、「Ready」になります。
16. 「Assignment」タブに表示されるツールバーで、「Save」および「Close」をクリックします。  
「Assignment」タブが閉じ、メイン画面が再びアクティブになります。これは、タスク割当てアダプタが、プロセス・タスクにアタッチされたことを意味します。

---

**注意：** タスク割当てアダプタをプロセス・タスクにアタッチすると、「Adapter Factory」フォームの「Usage Lookup」タブにアクセスすることで、関連付けられたプロセスおよびタスクを簡単に確認できます。

---

## 5.3 タスク割当てアダプタのプロセス・タスクからの削除

Oracle Identity Manager でプロセス・タスクをユーザーまたはグループに関連付けるためのタスク割当てアダプタが無効または不要になった場合は、アダプタをタスクから削除する必要があります。

### 5.3.1 タスク割当てアダプタのプロセス・タスクからの削除

タスク割当てアダプタをプロセス・タスクから削除するには、次の手順を実行します。

1. 「Process Definition」フォームを開きます。  
Design Console ワークスペースに、「Process Definition」フォームが表示されます。
2. アダプタを削除するタスクを含むプロセス (Solaris 8 プロセスなど) を選択します。  
選択したプロセスとそのタスクが、「Process Definition」フォームに表示されます。
3. アダプタを削除するプロセス・タスク (Get Solaris UUID タスクなど) の行ヘッダーをダブルクリックします。  
プロセス・タスクの情報を含む「Editing Task」ダイアログ・ボックスが表示されます。
4. 「Assignment」タブをクリックします。  
プロセス・タスクにアタッチされたアダプタの情報を示す「Assignment」タブが表示されます。
5. プロセス・タスクから削除するアダプタを含む行を選択します。
6. 「Delete」をクリックします。アダプタが「Assignment」タブに表示されなくなります。

7. 「Save」をクリックします。次に、「Close」をクリックします。

「Assignment」タブが閉じ、メイン画面が再びアクティブになります。これは、タスク割当てアダプタがプロセス・タスクから削除されたことを示します。





---

---

## ルール・ジェネレータについて

この章では、ルール・ジェネレータについて説明します。内容は次のとおりです。

- 6-2 ページの「概要」
- 6-3 ページの「ルール・ジェネレータのアダプタ変数のマッピング」
- 6-6 ページの「ルール・ジェネレータとプロセスの関連付け」
- 6-6 ページの「ルール・ジェネレータのフォーム・フィールドからの削除」

## 6.1 概要

フィールド検証を実行し、Oracle Identity Manager にパッケージされたフォームまたは Oracle Identity Manager ユーザーが作成したフォームにデフォルト値を入力するためには、特定のビジネス・ルールを適用する必要があります。たとえば、「Users」フォームで、ユーザーの姓と名を連結して Oracle Identity Manager でユーザー ID を自動的に生成する場合があります。

このためには、フォームのフィールド値の変更を指定した特定のタイプのアダプタを作成する必要があります。これは「Adapter Factory」フォームを使用して実行できます。このアダプタをフォームにアタッチすると、フォームにあるすべてのレコードのフィールド値が Oracle Identity Manager により自動的に更新され、情報が Oracle Identity Manager データベースに保存されます。

自動的にフォームのフィールド値を生成、変更または検証するアダプタのタイプを、ルール・ジェネレータと呼びます。Oracle Identity Manager は、ルール・ジェネレータを挿入前および更新前にトリガーします。

アダプタ変数を含むルール・ジェネレータを作成する場合は、アダプタ変数を適切な場所にマップする必要があります。そうしない場合、アダプタは機能しません。

また、アダプタを動作させるために、このアダプタをプロビジョニング・プロセスにアタッチできます。プロセスをターゲット・ユーザーまたは組織へプロビジョニングすると、Oracle Identity Manager は関連付けたルール・ジェネレータをトリガーします。

プロビジョニング・プロセスに割り当てたルール・ジェネレータが、プロセスを完了するためには不要になる場合があります。このような場合は、ルール・ジェネレータをプロビジョニング・プロセスから削除できます。同様に、フォーム・フィールドにルール・ジェネレータをアタッチしてから、そのフォーム・フィールドに別のルール・ジェネレータを関連付けることができます。このような場合は、まず現在フォーム・フィールドにアタッチしているルール・ジェネレータを削除する必要があります。

次の項では、次の内容を説明します。

- ルール・ジェネレータのアダプタ変数をマップします。
- ルール・ジェネレータをプロビジョニング・プロセスに関連付けます。
- ルール・ジェネレータをフォーム・フィールドから削除します。

## 6.2 ルール・ジェネレータのアダプタ変数のマッピング

第3章「アダプタの作成」では、ルール・ジェネレータの作成方法を説明しました。ここで、ルール・ジェネレータのアダプタ変数を正しい場所にマップする必要があります。これにより、アダプタが確実に動作します。

アダプタ変数をマップするには、Design Console の「Development Tools/Business Rule Definition」フォルダから「Data Object Manager」フォームにアクセスします。

ルール・ジェネレータのアダプタ変数を正しい場所にマップする手順は次のとおりです。

1. 「Data Object Manager」フォームを開きます。Design Console ワークショップに「Data Object Manager」フォームが表示されます。

次の表で、「Data Object Manager」フォームの様々な領域を説明します。

名前	説明
「Form Description」フィールド	この検索フィールドで、ルール・ジェネレータをアタッチするフィールドを含むフォームを選択します。
「Data Object」フィールド	このフィールドには、選択したフォームで表されるデータ・オブジェクトの名前が表示されます。
「Attach Handlers」タブ	このタブには次の項目が表示されます。 <ul style="list-style-type: none"> <li>■ 選択したフォームにアタッチされたルール・ジェネレータ</li> <li>■ このフォームに関連付けたルール・ジェネレータの実行スケジュール</li> <li>■ ルール・ジェネレータが Oracle Identity Manager で実行される順序</li> <li>■ ユーザー・グループの挿入、更新および削除の権限</li> </ul>
「Map Adapters」タブ	このタブには次の項目が表示されます。 <ul style="list-style-type: none"> <li>■ フォームに関連付けられたルール・ジェネレータの名前</li> <li>■ アダプタのステータス</li> <li>■ ルール・ジェネレータのアダプタ変数の名前、説明およびマッピング・ステータス</li> </ul> <p><b>注意：</b>「Map Adapters」タブは、アダプタが現在のデータ・オブジェクトに関連付けられるまで無効化されています。アダプタのデータ・オブジェクトへの関連付けの詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』を参照してください。</p>

2. 「Form Description」フィールドをダブルクリックします。ルール・ジェネレータをアタッチできるフォームを示す「Lookup」ダイアログ・ボックスが表示されます。
3. 使用するフォーム（Solaris など）を選択します。次に、「OK」をクリックします。
4. ツールバーで「Save」をクリックします。

選択したフォーム、フォームのデータ・オブジェクトおよびフォームに関連付けられたルール・ジェネレータ・アダプタが表示されます。また、Oracle Identity Manager で「Map Adapters」タブが有効化されます。

この例では、Solaris フォームが選択されています。データ・オブジェクト (Thor.CarrierBase.tcUD\_SOLARIS) と、それに関連付けられた 4 つのルール・ジェネレータ・アダプタ (adpCONVERTTLOWERCASE、adpSOLARISHMDSTRINGGEN、adpSETSOLARISASSET および adpSETPASSWORDFROMMAIN) が表示されます。Oracle Identity Manager は、これらの 4 つのルール・ジェネレータを挿入前にトリガーします。

アダプタの順序番号に基づき、Oracle Identity Manager は、まず adpCONVERTTLOWERCASE アダプタをトリガーし、それから adpSOLARISHMDSTRINGGEN、adpSETSOLARISASSET および adpSETPASSWORDFROMMAIN アダプタの順にトリガーします。

**ヒント:** ルール・ジェネレータのトリガー順を変更する手順は次のとおりです。

1. 「Assign」をクリックします。「Event Handlers」ダイアログ・ボックスが表示されます。
2. ルール・ジェネレータ・フォームを選択します。
3. 「Up」および「Down」矢印ボタンをクリックし、ルール・ジェネレータの順序を変更します。

これらのルール・ジェネレータを正常に動作させるためには、アダプタ変数を正しい場所にマップする必要があります。

5. 「Map Adapters」タブをクリックします。
6. 「Name」コンボ・ボックスで、マッピング可能なアダプタ変数を含むルール・ジェネレータ (adpCONVERTTOLOWERCASE ルール・ジェネレータなど) を選択します。

「Map Adapters」タブには次の項目が表示されます。

- フォームにアタッチするルール・ジェネレータの名前
- ルール・ジェネレータのステータス
- ルール・ジェネレータのアダプタ変数の名前、説明およびマッピング・ステータス

---

**注意:** アダプタのマッピング・ステータスの詳細は、4-4 ページの「[プロセス・タスク・アダプタのプロセス・タスクへのアタッチ](#)」を参照してください。

---

7. 「Map Adapters」タブの「Adapter Variable」領域に表示される各変数にマッピングを設定します。それには、マップする変数の行ヘッダー (Data など) をダブルクリックします。「Data Mapping for Variable」ダイアログ・ボックスが表示されます。

表 6-1 で、「Data Mapping for Variable」ダイアログ・ボックスのフィールドについて説明します。

表 6-1 「Data Mapping for Variable」ダイアログ・ボックスのフィールド

フィールド名	説明
Variable Name	このフィールドには、マッピングを設定するアダプタ変数の名前 (Data など) が表示されます。
Data Type	このフィールドには、アダプタ変数のデータ型が表示されます (たとえば、Data アダプタ変数のデータ型は文字列です)。
Map To	このフィールドには、アダプタ変数 (User Definition など) に設定可能なマッピングのソースおよびターゲットの場所が含まれます。 アダプタ変数を場所または連絡先にマップする場合、Oracle Identity Manager では隣接するコンボ・ボックスが有効化されます。このコンボ・ボックスから、アダプタ変数をマップする場所または連絡先の特定のタイプを選択します。 アダプタ変数を場所または連絡先にマップしない場合、このコンボ・ボックスは無効です。
Qualifier	このフィールドには、「Map To」コンボ・ボックスで選択したマッピングの修飾子が含まれます («User Login» など)。
IT Asset Type	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、特定の IT リソース (Solaris など) を選択できます。 アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。
IT Asset Property	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、マッピングの結果を受け取る特定のフィールド («User Name» など) を選択できます。 アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。 <b>重要:</b> 「IT Asset Type」および「IT Asset Property」フィールドは、下位互換性のためにこのウィンドウに含まれています。アダプタ変数を「IT Resource」データ型で作成することをお勧めします。この場合、これらのフィールドは表示されません。
Literal Value	アダプタ変数をリテラルにマップする場合、このフィールドに特定のリテラル名 (IBM など) を入力します。 アダプタ変数をリテラルにマップしない場合、このフィールドは表示されません。

「Map To」、「Qualifier」、「IT Asset Type」、「IT Asset Property」および「Literal Value」のフィールドをすべて入力します。

**注意:** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」を参照してください。

8. 「Save」をクリックします。次に、「Close」をクリックします。

「Data Mapping for Variable」ウィンドウが閉じます。「Map Adapters」タブが再びアクティブになります。

9. メイン画面のツールバーで「Save」をクリックします。

マッピング可能なすべてのアダプタ変数について、手順 7 と 8 を繰り返します。

「Status」フィールドのコンテンツが、「Mapping Incomplete」から「Ready」に変化します。アダプタ変数のマッピング・ステータスも、「No (N)」から「Yes (Y)」へ変化します。

これは、ルール・ジェネレータのすべてのアダプタ変数が、正常にマップされたことを示します。これで、このルール・ジェネレータをプロビジョニング・プロセスにアタッチし、プロセスをターゲット・ユーザーまたは組織にプロビジョニングした際にルール・ジェネレータをトリガーできます。

**ヒント:** フォームに関連付けられたルール・ジェネレータにすべてのアダプタ変数をマップすると、「Adapter Factory」フォームの「Usage Lookup」タブにアクセスすることで、アタッチしたフォームおよびルール・ジェネレータの実行スケジュールを簡単に確認できます。

ルール・ジェネレータがプロセスに割り当てられ、プロセスがプロビジョニングされると、Oracle Identity Manager によってルール・ジェネレータが実行されます。

## 6.3 ルール・ジェネレータとプロセスの関連付け

ルール・ジェネレータのアダプタ変数を正しい場所にマップした後、ルール・ジェネレータをプロビジョニング・プロセスにアタッチする必要があります。プロセスをターゲット・ユーザーまたは組織へプロビジョニングすると、Oracle Identity Manager は関連付けたルール・ジェネレータをトリガーします。

同様に、プロビジョニング・プロセスに割り当てたルール・ジェネレータがプロセスを完了するために不要になった場合は、ルール・ジェネレータをプロビジョニング・プロセスから削除する必要があります。

ルール・ジェネレータをプロビジョニング・プロセスに割り当てる、またはルール・ジェネレータをプロビジョニング・プロセスから削除するには、「Process Definition」フォームの「Event Handlers/Adapters」タブにアクセスします。このフォームは「Process Management」フォルダ内にあります。

## 6.4 ルール・ジェネレータのフォーム・フィールドからの削除

フォーム・フィールドにルール・ジェネレータをアタッチしてから、そのフォーム・フィールドに別のルール・ジェネレータを関連付けることができる場合もあります。このような場合は、まず現在フォーム・フィールドにアタッチしているルール・ジェネレータを削除する必要があります。

---

**注意:** フォームのデータ・オブジェクトのクラス名がプロビジョニング・プロセスの表名と一致する場合に、ルール・ジェネレータをフォームから削除すると、そのプロビジョニング・プロセスにルール・ジェネレータを割り当てることはできません。

たとえば、adpCONVERTTOLOWERCASE ルール・ジェネレータを Solaris フォームから削除したとします。フォームに関連付けられたデータ・オブジェクトのクラス名が UD\_SOLARIS である場合、UD\_SOLARIS の表名を持つプロビジョニング・プロセスにルール・ジェネレータを割り当てることはできません。

---

ルール・ジェネレータをフォーム・フィールドから削除するには、次の手順を実行します。

1. 「Data Object Manager」フォームを開きます。
2. 削除するルール・ジェネレータを含むフォームを選択します。
3. 選択したフォームとそのルール・ジェネレータが、「Data Object Manager」フォームに表示されます。

4. フォーム・フィールドから削除するルール・ジェネレータをクリックします。
5. 「Delete」をクリックします。

選択したルール・ジェネレータが、「Data Object Manager」フォームに表示されなくなります。これは、ルール・ジェネレータがフォーム・フィールドから削除されたことを意味します。

---

**注意：**ルール・ジェネレータをフォーム・フィールドから削除する時に、エラー・ボックスが表示される場合、アダプタはすでにプロビジョニング・プロセスに関連付けられています。まず、ルール・ジェネレータをプロセスから削除してください。その後、フォーム・フィールドから削除できます。

---





---

## 事前移入アダプタの使用

この章では、事前移入アダプタについて説明します。内容は次のとおりです。

- 7-2 ページの「概要」
- 7-3 ページの「事前移入アダプタのフォーム・フィールドへのアタッチ」
- 7-6 ページの「事前移入アダプタのフォーム・フィールドからの削除」

## 7.1 概要

ユーザー作成フォームに、Oracle Identity Manager で移入可能なフィールドおよび Oracle Identity Manager ユーザーがデータを入力する必要のあるフィールドが含まれる場合があります。ユーザーがフィールドに入力する情報がシステム生成フィールドに表示されるデータによって変わる場合は、まず Oracle Identity Manager でこのフィールドを事前移入する必要があります。フォームが表示された際にシステム生成データを確認できるため、Oracle Identity Manager ユーザーは正しいフィールドに情報を入力できます。

このためには、ある種のルール・ジェネレータを作成する必要があります。それをシステム生成されるフィールドにアタッチすると、Oracle Identity Manager では、Oracle Identity Manager データベースに保存することなく、自動的に適切な情報をこのフィールドに移入します。

このタイプのルール・ジェネレータを、事前移入アダプタと呼びます。事前移入アダプタで生成したデータは、自動または手動で入力して表示できます。「Auto-prepopulate」チェック・ボックスをプロビジョニング・プロセスで選択した場合、Oracle Identity Manager で情報が自動的に表示されます。このチェック・ボックスが選択されていない場合、Oracle Identity Manager ユーザーは事前移入アダプタで生成されたデータを手動で生成して表示する必要があります。このためには、フォームをユーザーにプロビジョニングする時に、Web クライアントのダイレクト・プロビジョニング・ウィザードのフォーム・セクションにある事前移入ボタンをクリックする必要があります。

事前移入アダプタは、異なるフォーム・フィールドに繰り返し使用できます。また、複数の事前移入アダプタを特定のフィールドに関連付けるよう指定できます。そのため、どの事前移入アダプタをフォーム・フィールドに選択するか、Oracle Identity Manager であらかじめになっている必要があります。これは、事前移入ルールを使用して行われます。

事前移入ルールは基準です。これらのルールによって、Oracle Identity Manager で事前移入アダプタが選択されます。この事前移入アダプタがフィールドに割り当てられると、事前移入アダプタはフィールドに関連付けられます。

それぞれの事前移入アダプタには、事前移入ルールが関連付けられています。また、ルールには優先順位があり、Oracle Identity Manager はその順序でルールをトリガーします。

この例では、Oracle Identity Manager は、Rule for Uppercase User ID ルールを最初にトリガーします（最も優先順位が高いためです）。このルールの状態が TRUE の場合、ルールは成功です。その結果、Oracle Identity Manager は、関連する事前移入アダプタ（Display Uppercase Letters for User ID アダプタ）を「User ID」フィールドにアタッチします。

一方、ルールの状態が FALSE の場合、ルールは失敗です。Oracle Identity Manager は、次に優先順位が高いルールをトリガーします。このルールが成功すると、Oracle Identity Manager は、関連付けられたアダプタを指定するフィールドにアタッチします。

したがって、この例では、Rule for Uppercase User ID ルールが失敗した場合、Oracle Identity Manager は Rule for Lowercase User ID をトリガーします。このルールが成功すると、Oracle Identity Manager は、関連する事前移入アダプタ（Display Lowercase Letters for User ID アダプタ）を「User ID」フィールドにアタッチします。

このタイプのアダプタにアダプタ変数が含まれる場合、ルールを事前移入アダプタに割り当ててから、アダプタ変数を適切な場所にマップする必要があります。そうしない場合、アダプタは機能しません。

最終的にフィールドに関連付けられた事前移入アダプタが有効ではなくなった場合、アダプタをフィールドから削除する必要があります。

## 7.2 事前移入アダプタのフォーム・フィールドへのアタッチ

事前移入アダプタをフォーム・フィールドにアタッチするには、次の手順を実行します。

1. 事前移入アダプタをアタッチするフィールドを選択します。
2. 指定したフィールドに情報を移入するためにアダプタを使用するかどうかを決定するルールを選択します。
3. 指定したフィールドに関連付けるアダプタを選択します。
4. 選択したルールに優先順位を設定します。
5. 事前移入アダプタのアダプタ変数を正しい場所にマップします。

---

**注意：**事前移入アダプタをフォーム・フィールドにアタッチするには、次の点を確認する必要があります。

- フォームがアクティブな状態ではないこと。あてはまらない場合は、新しいフォーム・バージョンを作成します。
  - アタッチしたアダプタを使用する前に、フォームをアクティブ化する必要があります。
- 

6. 「Form Designer」 フォームを開きます。
7. 事前移入アダプタをアタッチするフォーム（Solaris など）を問い合わせます。
8. 「prepopulate」 タブをクリックします。

問い合わせたフォームにすでにアタッチされている事前移入アダプタが、タブ内に表示されます。

---

**注意：**アダプタがフォーム・フィールドにアタッチされていない場合、「prepopulate」タブには何も表示されません。

---

9. 「Add」 をクリックします。  
「prepopulate Adapters」 ダイアログ・ボックスが表示されます。

表 7-1 で、「prepopulate Adapters」ダイアログ・ボックスのフィールドを説明します。

**表 7-1 「Prepopulate Adapters」ダイアログ・ボックスのフィールド**

名前	説明
Field Name	このコンボ・ボックスには、事前移入アダプタをアタッチできるすべてのフォーム・フィールドのリストが表示されます。
Rule	この検索フィールドで、指定したフォーム・フィールドに情報を移入するために関連付けられたアダプタを使用するかどうかを決定するルールを選択します。
Adapter	この検索フィールドで、指定したフィールドに関連付けるアダプタを選択します。
Order	このフィールドで、選択したルールの優先順位を設定します。
Adapter Status	このフィールドには、アダプタ変数のマッピング・ステータスが表示されます。 アダプタのマッピング・ステータスの詳細は、4-4 ページの「プロセス・タスク・アダプタのプロセス・タスクへのアタッチ」を参照してください。
Adapter Variables	この領域には、次の項目が表示されます。 <ul style="list-style-type: none"> <li>■ <b>Mapped:</b> アダプタ変数のマッピング・ステータス。Y はアダプタ変数が正しくマップされたことを表し、N はこの変数が正しくマップされなかったことを表します。</li> <li>■ <b>Name:</b> アダプタ変数の名前。</li> <li>■ <b>Mapped to:</b> 変数がマッピングされるフォーム・フィールド。アダプタ変数がマッピングされていない場合、列内の対応するセルは空白です。</li> </ul>

10. 「Field Name」コンボ・ボックスから、事前移入アダプタをアタッチするフォーム・フィールド（「User ID」など）を選択します。
11. 「Rule」検索フィールドをダブルクリックします。表示される「Lookup」ダイアログ・ボックスで、指定したフォーム・フィールドに情報を移入するために関連付けられたアダプタを使用するかどうかを決定するルール（Rule for Lowercase User ID など）を選択します。
12. 「Adapter」検索フィールドをダブルクリックします。表示される「Lookup」ダイアログ・ボックスで、手順 10 で選択したフィールドに関連付けるアダプタ（Display Lowercase Letters for User ID など）を選択します。
13. 「Order」フィールドに、手順 11 で選択したルールの優先順位（2 など）を入力します。
14. 「prepopulate Adapters」ウィンドウのツールバーで「Save」をクリックします。
15. 「Mapping Incomplete」が「Adapter Status」フィールドに表示されます。これは、選択したアダプタに、正しくマップされていない変数が含まれていることを示しています。変数を正しい場所にマップする必要があります。そうしない場合、アダプタは動作しません。
16. 「prepopulate Adapters」ウィンドウの「Adapter Variable」領域に表示される各変数にマッピングを設定します。それには、マップする変数の行ヘッダー（UserID など）をダブルクリックします。  
「Map Adapter Variables」ウィンドウが表示されます。

表 7-2 で、「Map Adapter Variables」ウィンドウのフィールドについて説明します。

**表 7-2 「Map Adapter Variables」ウィンドウのフィールド**

フィールド名	説明
Variable Name	このフィールドには、マッピングを設定するアダプタ変数の名前 (UserID など) が表示されます。
Data Type	このフィールドには、アダプタ変数のデータ型が表示されます (たとえば、UserID アダプタ変数のデータ型は文字列です)。
Map To	このフィールドには、アダプタ変数に設定可能なマッピング型 (「Process Data」など) が含まれます。  アダプタ変数を場所または連絡先にマップする場合、Oracle Identity Manager では隣接するコンボ・ボックスが有効化されます。このコンボ・ボックスから、アダプタ変数をマップする場所または連絡先の特定のタイプを選択します。  アダプタ変数を場所または連絡先にマップしない場合、このコンボ・ボックスは無効です。
Qualifier	このフィールドには、「Map To」コンボ・ボックスで選択したマッピングの修飾子が含まれます (「User ID」など)。
IT Asset Type	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、特定の IT リソース (Solaris など) を選択できます。  アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。
IT Asset Property	このフィールドでは、アダプタ変数を IT リソースにマップし、この変数のデータ型が文字列である場合に、マッピングの結果を受け取る特定のフィールド (「User Name」など) を選択できます。  アダプタ変数を IT リソースにマップしない場合、またはアダプタ変数のデータ型が文字列ではない場合、このフィールドは表示されません。  <b>重要:</b> 「IT Asset Type」および「IT Asset Property」フィールドは、下位互換性のためにこのウィンドウに含まれています。アダプタ変数を「IT Resource」データ型で作成することをお勧めします。この場合、これらのフィールドは表示されません。
Literal Value	アダプタ変数をリテラルにマップする場合、このフィールドでリテラル値を指定します。  アダプタ変数をリテラルにマップしない場合、このフィールドは表示されません。

17. 「Map To」、「Qualifier」、「IT Asset Type」、「IT Asset Property」および「Literal Value」のフィールドをすべて入力します。

**注意:** 選択するマッピングの詳細は、『Oracle Identity Manager リファレンス』の「アダプタ・マッピング情報」を参照してください。

18. 「Map Adapter Variable」ウィンドウのツールバーで「Save」をクリックします。次に、「Close」をクリックします。

「Map Adapter Variables」ウィンドウが閉じます。「prepopulate Adapters」ウィンドウが再びアクティブになります。

「Adapter Status」フィールドのテキストが、「Mapping Incomplete」から「Ready」に変化します。アダプタ変数のマッピング・ステータスも、「No (N)」から「Yes (Y)」へ変化します。

19. 「prepopulate Adapters」ウィンドウのツールバーで「Close」をクリックします。

「prepopulate Adapters」ウィンドウが閉じ、「Form Designer」フォームが再びアクティブになります。「User ID」フォーム・フィールドにアタッチした事前移入アダプタ（Display Lowercase Letters for User ID）が、「Results of 1Q Sales 2003」フォームの「prepopulate」タブに表示されます。

このフォームを参照するプロセスを、ターゲット・ユーザーまたは組織にプロビジョニングすると、フォームが表示されます。Oracle Identity Manager は、優先順位が最も高い事前移入ルールが有効かどうかを確認します。有効な場合、Oracle Identity Manager は関連付けられた事前移入アダプタを指定するフィールド（「User ID」）に割り当てて実行します。この時、次のアクションのいずれか1つが発生します。

- 「Auto-prepopulate」チェック・ボックスをプロビジョニング・プロセスで選択した場合、事前移入アダプタで生成されたデータが自動的に Oracle Identity Manager で表示されません。
- 「Auto-prepopulate」チェック・ボックスが選択されていない場合、Oracle Identity Manager ユーザーは、事前移入アダプタで生成されたデータの表示を手動でトリガーする必要があります。このためには、管理者はフォームをユーザーにプロビジョニングする時に、Web クライアントのダイレクト・プロビジョニング・ウィザードのフォーム・セクションにある事前移入ボタンをクリックする必要があります。

**ヒント:** 事前移入アダプタをフォーム・フィールドに割り当て、事前移入ルールをアダプタに割り当てると、「Adapter Factory」フォームの「Usage Lookup」タブにアクセスすることで、アダプタ、フォーム・フィールドおよびルールの関連付けを簡単に確認できます。

## 7.3 事前移入アダプタのフォーム・フィールドからの削除

フォーム・フィールドに関連付けられた事前移入アダプタが有効ではなくなった場合、アダプタをフィールドから削除する必要があります。

---

**注意:** 事前移入アダプタをフォーム・フィールドから削除する前に、新しいバージョンのフォームを作成してください。

---

事前移入アダプタをフォーム・フィールドから削除する手順は次のとおりです。

1. 削除する事前移入アダプタを選択します。
2. 「Delete」をクリックします。事前移入アダプタがフォーム・フィールドから削除されません。フォームが起動している場合は、トリガーできません。
3. アダプタを削除した後に、フォームをアクティブ化する必要があります。

---

---

## エンティティの管理

ルール・ジェネレータ・アダプタと同様に、エンティティ・アダプタもフォーム・フィールドの値を自動的に生成、変更または検証し、その情報を Oracle Identity Manager データベースに保存します。しかし、ルール・ジェネレータとエンティティ・アダプタは、次の点で異なります。

- **スケジュールの実行。** Oracle Identity Manager では、挿入前、更新前、削除前、挿入後、更新後または削除後に、エンティティ・アダプタをトリガーできます。ルール・ジェネレータ・アダプタは、挿入前および更新前にのみ実行できます。
- **フィールド値の手動変更。** アダプタは、エンティティ・アダプタがアタッチされたフォーム・フィールドを移入します。エンティティ・アダプタによって上書きされるため、Oracle Identity Manager ユーザーはこの値を編集できません。結果として、変更内容はデータベースに保存されません。

同様に、アダプタは、ルール・ジェネレータがアタッチされたフォーム・フィールドを移入します。しかし、ルール・ジェネレータ・アダプタが生成する値よりも変更内容が優先されるため、Oracle Identity Manager ユーザーはこの値を編集できます。結果として、変更内容がデータベースに保存されます。

- **フォーム・フィールドの背景色。** ルール・ジェネレータをフォーム・フィールドにアタッチすると、フィールドは特定の色（たとえばピンクなど）で表示されます。これは、フィールドにルール・ジェネレータがアタッチされていることを視覚的に示すものです。一方、エンティティ・アダプタをフォーム・フィールドにアタッチした場合は、フィールドに背景色が付きません。

---

---

**注意：** Oracle Identity Manager フォームまたはプロビジョニング・プロセス（あるいはその両方）へのエンティティ・アダプタの値のマッピングの詳細は、第6章「ルール・ジェネレータについて」を参照してください。

---

---





---

## アダプタのコンパイル

複数のアダプタを作成し、それらをコンパイルする場合、2つの方法のいずれかで実行できます。1つ目は、それぞれのアダプタにアクセスし、個別にコンパイルする方法です。ただし、この方法は多くの時間と労力を必要とします。

より効率のよい方法は、コンパイルするアダプタを選択して一度にコンパイルするか、必要な場合にはボタンをクリックし、Oracle Identity Manager データベースに存在するすべてのアダプタをコンパイルする方法です。

## 9.1 アダプタの自動コンパイル

リリース 9.1.0 以降、デプロイメント・マネージャを使用してコネクタ・ファイルをインポートすると、アダプタが自動的にコンパイルされます。コンパイルされたアダプタのクラス・ファイルは、ファイルシステムではなく、Oracle Identity Manager データベースに格納され、実行時にロードされます。次の 2 つの API が、アダプタをプログラムでコンパイルする API として追加されました。

- `public void compileAdapter (String adapterName)`: この API では、1 つのアダプタをコンパイルし、コンパイルしたクラス・ファイルをデータベースに格納します。アダプタの名前をパラメータとして使用します。アダプタが見つからない場合、またはエラーが起きた場合は、API で適切な例外がスローされます。
- `public void compileAll`: この API では、システムのすべてのアダプタがコンパイルされます。コンパイル時にエラーが起きた場合は、タイプ `tcBulkException` の例外がスローされます。この例外は、コンパイル時に API で検出されたすべてのエラーで構成されます。

ただし、変更を加えた場合は、アダプタを手動で変更できます。

---

**注意:** `XL.CompilerPath` システム・プロパティの JDK ディレクトリのパスを設定する必要があります。これを行わないと、デプロイメント・マネージャを使用して XML ファイルをインポートしたときに行われるプロセスのアダプタ・コンパイル段階でエラーが起こります。

システム・プロパティの設定値の詳細は、『Oracle Identity Manager デザイン・コンソール・ガイド』の「System Configuration」フォームを参照してください。

---

## 9.2 アダプタの手動コンパイル

「Adapter Manager」フォームは、「Development Tools」フォルダにあります。これを使用して、複数のアダプタを同時にコンパイルします。

複数のアダプタを手動でコンパイルするには、次の手順を実行します。

1. 「Adapter Manager」フォームを開きます。
2. Oracle Identity Manager データベース内のすべてのアダプタをコンパイルするには、「**Compile All**」オプションを選択します。

複数のアダプタをコンパイルするには、コンパイルするアダプタを選択します。次に、「**Compile Selected**」オプションを選択します。

ステータスが「OK」ではないすべてのアダプタをコンパイルするには、「**Compile Previously Failed**」オプションを選択します。

3. 「**Start**」ボタンをクリックします。

Oracle Identity Manager により、手順 2 で指定した基準に合致するアダプタがコンパイルされます。

**ヒント:** Oracle Identity Manager では、「Adapter Manager」フォームに表示されるすべてのアダプタの記録を見ることができます。これにより、アダプタの詳細な情報を確認できます。

アダプタの記録を確認するには、該当するアダプタを選択します。次に、行ヘッダーをダブルクリックするか、アダプタを右クリックして、表示されるメニューから「Launch Adapter」コマンドを選択します。

---

## アダプタのエクスポートおよびインポート

---

この章では、アダプタのエクスポートおよびインポートについて説明します。

アダプタを1つのデータベースに作成してから、別のデータベースで使用する場合があります。手動でそのデータベースのためにアダプタを再構築するという方法があります。しかし、この方法は、時間、資金およびリソースの面で非生産的なだけでなく、アダプタの再作成中に開発者が不注意からエラーを発生させる可能性があります。その結果、アダプタが関連付けられたプロセスをターゲット・ユーザーまたは組織にプロビジョニングすると、アダプタで指定した機能が実行されず、Oracle Identity Manager は対応するリソースをユーザーまたは組織にプロビジョニングできません。

より効率的で、エラーの可能性が低いもう1つの方法は、次のとおりです。

1. 不具合のないアダプタを作成し、データベースに正しく適用します。デプロイメント・マネージャにより、このアダプタとそのすべての内部的な変数マッピングを含む \*.xml ファイルが内部的に作成されます。

また、デプロイメント・マネージャは、このファイルを指定した場所にエクスポートします。

2. ファイルのある場所からファイルを取り出し、ターゲット・データベースにインポートします。

デプロイメント・マネージャを使用して、アダプタをあるデータベースから別のデータベースに転送できます。デプロイメント・マネージャの使用方法は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』を参照してください。

---

**注意：** ターゲット・データベースにインポートしたアダプタは、そのデータベース内で再コンパイルする必要があります。また、アダプタを移行してから、サード・パーティ JAR ファイルをコピーしてください。そうしない場合、アダプタは動作しません。

---



---

## Remote Manager IT リソースの作成とテスト

この章では、Remote Manager IT リソースの作成とテストの手順について説明します。内容は次のとおりです。

- [インストール後の構成](#)
- [Remote Manager IT リソースの作成とテスト](#)

Remote Manager は Oracle Identity Manager のコンポーネントで、プロキシのように動作してサード・パーティ・システムと直接通信します。Remote Manager は、Oracle Identity Manager を通じた非リモート API の呼出しおよびセキュアな接続での Secure Sockets Layer (SSL) をサポートしていない API の呼出しに使用されます。

Remote Manager をインストールし、Oracle Identity Manager Server と Remote Manager の間に信頼関係を確立（証明書を信頼）すると、Remote Manager 用の IT リソースを作成し、それをテストできます。

## 11.1 インストール後の構成

Remote Manager のインストール後、アプリケーション・サーバーと Remote Manager の間で証明書が信頼されていることを確認できます。そのためには、まず Design Console の「Administration」フォルダで Remote Manager を開きます。「Remote Manager」フォームに、接続されているが信頼されていないすべての Remote Manager が表示されます。

次の手順を実行して、アプリケーション・サーバーと Remote Manager の間の信頼関係が証明書を通じて確立されていることを確認してください。この手順では、JBoss Application Server を例として使用します。keytool ユーティリティを証明書のインポート / エクスポートに使用します。

1. コマンド・プロンプトを使用してディレクトリ・パスを開き、keytool ユーティリティを使用して証明書のフィンガープリントの一覧を表示します。

```
XLREMOTE_HOME¥xlremote
```

2. 次のコマンドを入力します。

```
JAVA_HOME¥jre¥bin¥keytool -list -keystore config¥.xlkeystore
```

3. xellerate キーストアのデフォルト・パスワード (xellerate) を入力します。

```
Your keystore contains 1 entry
```

```
    xell, Jan 7, 2005, keyEntry,
```

```
    Certificate fingerprint (MD5):
```

```
    B0:F2:33:C8:69:E4:25:A3:CB:59:E8:51:27:EE:5C:52
```

証明書のフィンガープリントは、太字で表示されています。これを、キーストアの証明書の一覧と比較します。

4. アプリケーション・サーバーで使用する Java SDK フォルダを開きます。もう一度パスを入力し、keytool を使用してキーストア内の証明書の一覧を表示します。

```
JAVA_HOME¥jre¥lib¥security¥cacerts
```

5. 次のコマンドを入力し、信頼できる証明書の一覧を確認します。

```
JAVA_HOME¥bin¥keytool -keystore cacerts -storepass changeit  
-storetype jks -provider provider_name
```

キーストア・エントリは、次のように出力されます。

```
Your keystore contains 25 entries
equifaxsecureebusinessca1, Jul 23, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 64:9C:EF:2E:44:FC:C6:8F:52:07:D0:51:73:8F:CB:3D
verisignclass4ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 1B:D1:AD:17:8B:7F:22:13:24:F5:26:E2:5D:4E:B9:10
entrustglobalclientca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 9A:77:19:18:ED:96:CF:DF:1B:B7:0E:F5:8D:B9:88:2E
gtecybertrustglobalca, May 10, 2002, trustedCertEntry,
Certificate fingerprint (MD5): CA:3D:D3:68:F1:03:5C:D0:32:FA:B8:2B:59:E8:5A:DB
entrustgsslca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 9D:66:6A:CC:FF:D5:F5:43:B4:BF:8C:16:D1:2B:A8:99
verisignclass1ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 51:86:E8:1F:BC:B1:C3:71:B5:18:10:DB:5F:DC:F6:20
thawtepersonalbasicca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): E6:0B:D2:C9:CA:2D:88:DB:1A:71:0E:4B:78:EB:02:41
entrustsslca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): DF:F2:80:73:CC:F1:E6:61:73:FC:F5:42:E9:C5:7C:EE
thawtepersonalfreemailca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 1E:74:C3:86:3C:0C:35:C5:3E:C2:7F:EF:3C:AA:3C:D9
verisignclass3ca, Oct 24, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 10:FC:63:5D:F6:26:3E:0D:F3:25:BE:5F:79:CD:67:67
gtecybertrustca, May 10, 2002, trustedCertEntry,
Certificate fingerprint (MD5): C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
thawteserverca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): C5:70:C4:A2:ED:53:78:0C:C8:10:53:81:64:CB:D0:1D
equifaxsecureca, Jul 23, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 67:CB:9D:C0:13:24:8A:82:9B:B2:17:1E:D1:1B:EC:D4
thawtepersonalpremiumca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 3A:B2:DE:22:9A:20:93:49:F9:ED:C8:D2:8A:E7:68:0D
thawtepremiumserverca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): 06:9F:69:79:16:66:90:02:1B:8C:8C:A2:C3:07:6F:3A
entrust2048ca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): BA:21:EA:20:D6:DD:DB:8F:C1:57:8B:40:AD:A1:FC:FC
verisignserverca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5): 74:7B:82:03:43:F0:00:9E:6B:B3:EC:47:BF:85:A5:93
entrustclientca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 0C:41:2F:13:5B:A0:54:F5:96:66:2D:7E:CD:0E:03:F4
baltimorecybertrustca, May 10, 2002, trustedCertEntry,
Certificate fingerprint (MD5): AC:B6:94:A5:9C:17:E0:D7:91:52:9B:B1:97:06:A6:E4
geotrustglobalca, Jul 23, 2003, trustedCertEntry,
Certificate fingerprint (MD5): F7:75:AB:29:FB:51:4E:B7:77:5E:FF:05:3C:99:8E:F5
gtecybertrust5ca, May 10, 2002, trustedCertEntry,
Certificate fingerprint (MD5): 7D:6C:86:E4:FC:4D:D1:0B:00:BA:22:BB:4E:7C:6A:8E
equifaxsecureglobalebusinessca1, Jul 23, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 8F:5D:77:06:27:C4:98:3C:5B:93:78:E7:D7:7D:9B:CC
baltimorecodesigningca, May 10, 2002, trustedCertEntry,
Certificate fingerprint (MD5): 90:F5:28:49:56:D1:5D:2C:B0:53:D4:4B:EF:6F:90:22
equifaxsecureebusinessca2, Jul 23, 2003, trustedCertEntry,
Certificate fingerprint (MD5): AA:BF:BF:64:97:DA:98:1D:6F:C6:08:3A:95:70:33:CA
verisignclass2ca, Oct 24, 2003, trustedCertEntry,
Certificate fingerprint (MD5): B3:9C:25:B1:C3:2E:32:53:80:15:30:9D:4D:02:77:3E
```

見やすくするために、証明書フィンガープリントは、太字で強調されています。必要な証明書フィンガープリントの Certificate Fingerprint (MD5:B0:F2:33:C8:69:E4:25:A3:CB:59:E8:51:27:EE:5C:52) が信頼できる証明書に含まれていません。そのため証明書をインポートする場合があります。

6. 使用しているアプリケーション・サーバー用のインストレーション・ガイドの「Remote Manager 証明書の信頼」に記載されている手順を実行します。

## 11.2 Remote Manager IT リソースの作成とテスト

Remote Manager IT リソースを作成およびテストする手順は、次のとおりです。

---

**注意：** 非 SSL 接続で Remote Manager IT リソースをテストする場合は、次のファイルの <RMIOverSSL> プロパティを `false` に設定します。

```
OIM_HOME¥xellerate¥config¥xlconfig.xml
```

---

1. Design Console で、「Resource Object」フォームを開きます。
2. リソース・オブジェクトを作成します。この例では、次のパラメータが設定されています。
  - 名前: MyObj
  - 「Order for User」オプション: 有効化
  - Type: Application
  - 次のチェック・ボックスが使用可能
    - Allow Multiple
    - Auto Save
    - Self Request Allowed
    - Allow All
    - Auto Launch
3. リソース・オブジェクトに、IT リソースのタイプを作成します。「IT Resources Type Definition」フォームを開きます。この例では、次のパラメータが設定されています。
  - Server Type: MyObjServer

---

**注意：** Design Console で IT リソース・タイプ・パラメータを定義する際、暗号化するフィールドを指定できます。

---

4. Remote Manager に IT リソースを作成します。この例では、次のパラメータが設定されています。
  - IT リソース名: remote
  - タイプ名: Remote Manager

IT リソースに適切な URL とサーバー名があり、URL が示す場所に Remote Manager がインストールされていることを確認してください。

---

**注意：** URL に名前そのものが存在していないことを確認してください。たとえば、Remote Manager は次のようなサービス名と URL で構成されます。

```
サービス名: Rmanager、URL: rmi://w2kevandanwkstn:12346
```

---

5. すでに作成した MyObjServer IT リソース・タイプのインスタンスを作成します。「IT Resource Information」フォームを開きます。手順 4 で作成した Remote Manager (remote) が「Remote Manager」フィールドで選択されていることを確認してください。
6. 「IT Resource Information」フォームに情報を保存すると、該当する IT リソースに必要な詳細を追加できます。この例では、ユーザー名とパスワードを入力します。



7. 次のコードの JAR ファイルを作成します。

```
package testme;
import java.io.PrintStream;
public class test
{
    public test ()
    {
    }
    public static int addme(int i, int j)
    {
        /*6*/System.out.println(i + "+" + j + "=" + (i + j));
        /*7*/return i + j;
    }
    public static void main(String args[])
    {
        /* 11*/addme(5, 10);
    }
}
```

このコードが Remote Manager で実行されます。

8. JAR ファイルを `xlremote_home/JavaTasks` および `OIM_HOME/xellerate/JavaTasks` ディレクトリにコピーします。
9. Remote Manager で実行するアダプタを作成します。「Adapter Factory」フォームを開きます。この例では、次のパラメータが設定されています。
- Adapter Name: remotetest
  - Adapter Type: Process Task
- この例では、(.jar ファイル内のコード例に基づいて) このアダプタに 3 つの変数を作成できます。「Add」をクリックします。Java コードは 2 つの整数を引数とし、IT リソースを 3 番目の変数に取ります。
10. 最初の変数では、次のパラメータが設定されています。
- Variable Name: var1
  - Variable type: Integer
  - 「Map To」オプション: 「Resolve at Run time」と設定
11. 最初と同様の方法で、2 番目の変数を作成します。次のパラメータが設定されています。
- Variable Name: var2
  - Variable type: Integer
  - 「Map To」オプション: 「Resolve at Run time」と設定
12. 3 番目の変数を IT リソース用に作成します。次のようにパラメータが設定されています。
- Variable Name: ITRes
  - Variable type: ITResource
  - 「Map To」オプション: 「Resolve at Run time」と設定
  - Resource Type: MyObjServer

---

**注意:** 「Resource Type」フィールドは、手順 5 で作成した IT リソースと同じである必要があります。Remote Manager ではありません。

---

13. 新しいリモート Java タスクを追加します。「Adapter Factory」フォームで、「Add」をクリックします。「Functional Task」オプションがアクティブであることを確認してください。「Remote」オプションを選択します。「Continue」をクリックします。

14. 「Object Instance Selection」ダイアログ・ボックスが表示されます。新しいオブジェクト・インスタンスを作成します。「New Object Instance」オプションがアクティブであることを確認してください。「Continue」をクリックします。
15. 「Remote」ウィンドウが表示されます。この例では、次のパラメータが設定されています。
  - Task Name: remote
  - API Source: JavaTask フォルダ内の .jar ファイルを参照
  - Application API: Testme.test
  - Constructors: 0 public testme.test () に設定
  - Methods: testme.test.addme (int, int) に設定「Save」をクリックすると、IT リソースは自動的に引数として追加されます。アプリケーション・メソッド・パラメータをマッピングできます。
16. 「Parameter Data Mapping」リストの最初の項目を選択して、パラメータのマッピングを開始します。この出力パラメータは、整数です。次のマッピングが設定されています。
  - Map To: Adapter Variables
  - Name: Return variable
17. 「Set」をクリックします。
18. 2 番目のパラメータを選択してマップします。この入力パラメータは、整数です。次のマッピングが設定されています。
  - Map: Adapter Variables
  - Name: var1
19. 「Set」をクリックします。
20. 3 番目のパラメータを選択してマップします。この入力パラメータは、整数です。次のマッピングが設定されています。
  - Map To: Adapter Variables
  - Name: var2
21. 「Set」をクリックします。
22. 最後のパラメータを選択してマップします。アダプタに入力として渡した変数に、この IT リソースをマップします。次のマッピングが設定されています。
  - Map To: Adapter Variables
  - Name: ITRes
23. 「Set」をクリックします。
24. 「Set」をクリックします。次に、「Save」をクリックします。「Adapter Factory」フォームが表示されます。
25. 「Build」をクリックして、アダプタをコンパイルします。

アダプタを起動するには、アダプタを1つのタスクとして呼び出すプロビジョニング・プロセスを作成できます。これを実行する手順は次のとおりです。

1. 「Process Definition」 フォームを開きます。この例では、次のパラメータが設定されています。
  - 「Name」 フィールド: MyObjProv
  - 「Type」 フィールド: Provisioning
  - Object Name: MyObj
    - 次のチェック・ボックスが使用可能
    - Default Process
    - Auto Prepopulate
    - Auto Save Form
2. 「Save」 アイコンをクリックします。プロビジョニング・タスクが、自動的に「Tasks」 タブに表示されます。
3. 「Add」 をクリックして、新しいタスクを作成します。この例では、次のパラメータが設定されています。
  - 「Task Name」 フィールド: Call Remote Adapter
  - 「Task Description」 フィールド: タスクの機能の説明
4. 「Save」 アイコンをクリックします。次に、「Integration」 タブをクリックします。次に、「Add」 をクリックし、アダプタをこのタスクに追加します。「Handler Type」 ウィンドウが表示されます。
5. 「Adapter」 オプションを有効化し、実行するアダプタを選択します。
6. 「Save」 アイコンをクリックします。「Integration」 タブの「Name」 フィールドに、アダプタ名が表示されます。「Status」 フィールドには、マッピングが完了していないことが表示されます。「Adapter Variable」 ペインには、変数がマップされていないことが表示されます。
7. 最初の変数、アダプタ戻り値を選択し、「Map」 をクリックします。「Edit Data Mapping for Variable」 ウィンドウが表示されます。次のようにパラメータが設定されています。
  - Data Type: Object
  - Map To: Response Code
8. 2番目の変数、var1 を選択し、「Map」 をクリックします。「Edit Data Mapping for Variable」 ウィンドウが表示されます。次のようにパラメータが設定されています。
  - Data Type: Integer
  - Map To: Literal
  - Qualifier: Integer
  - Literal Value: 10
9. 3番目の変数、var2 を選択し、「Map」 をクリックします。「Edit Data Mapping for Variable」 ウィンドウが表示されます。次のようにパラメータが設定されています。
  - Data Type: set to Integer
  - Map To: Literal
  - Qualifier: Integer
  - Literal Value: 20

10. 4 番目の変数、ITRes を選択し、「Map」をクリックします。「Edit Data Mapping for Variable」ウィンドウが表示されます。次のようにパラメータが設定されています。
  - Data Type: IT Resource (MyObjServer)
  - Map To: IT Resource
  - Qualifier: MyObjServerInstance
11. 「Editing Task」ウィンドウの「Responses」タブをクリックします。「Add」をクリックし、アダプタから可能なレスポンスを追加します。この例では、可能なレスポンスは 30 のみです。「Description」を「Completed」に設定し、「Status」を C に設定します。
12. 「Task to Object Status Mapping」タブをクリックします。「Completed」カテゴリで、「Object Status」を「Provisioned」に設定します。
13. この時点で、ユーザーを新規に作成したリソースに直接プロビジョニングし、リモート・タスクの実行をテストできます。ただし、Remote Manager が実行中であることを、まず確認してください。「Remote Manager」フォームを開き、サービスが使用可能であることを検証します。
14. Oracle Identity Manager 管理およびユーザー・コンソールを起動し、管理者としてログインします。「ユーザー」、「管理」と移動し、このリソース (MyObj) をプロビジョニングするユーザーを選択します。選択したユーザーの「ユーザーの詳細」ページが表示されます。「このユーザーの詳細を表示」プルダウン・オプションで、「リソース・プロファイル」を選択します。
15. 「ユーザーの詳細」、「リソース・プロファイル」ページが表示されます。「新しいリソースのプロビジョニング」をクリックし、新規に作成したリソース (MyObj) を選択します。
16. ユーザーへのリソースのプロビジョニング・ウィザードが表示されます。「続行」をクリックし、プロビジョニング・プロセスを完了します。
17. 「リソースは正常にプロビジョニングされました」ページが表示されるまで、プロビジョニング・プロセスを続行します。
18. Remote Manager ログ・ファイルで、コードが実行されたことを確認します。Remote Manager ログ・ファイルは `OIM_HOME/xlremote/log` ディレクトリにあります。ログ・ファイルの最後の行は次のようになります。

```
DONE5+10=15
```

前述の行は、2つの入力された整数の合計が 15 であることを示しています。これは、コードが正しく実行され、リソース・オブジェクトがプロビジョニングされたことを示します。

---

---

## SPML Web サービス

この章では、Oracle Identity Manager の SPML Web サービス・インタフェースについて説明します。

この章の次の項では、SPML Web サービスの基本情報を提供します。

- [SPML Web サービスの概要](#)
- [SPML Web サービスでサポートされているプロビジョニング処理](#)

SPML Web サービスをデプロイするには、次の項のいずれかで説明されている方法を参照してください。

- [SPML Web サービスのデプロイ](#)
- [Oracle Web Services Manager を使用し、SPML Web サービスをデプロイするセキュリティの有効化](#)

次の項では、SPML Web サービスのデプロイ後に実行する手順について説明します。

- [デプロイ後のタスク](#)
- [SSL 通信の有効化](#)

次の項では、SPML Web サービスのクライアントを開発する際の手順の概要を説明します。

- [SPML Web サービスのクライアントの開発](#)

## 12.1 SPML Web サービスの概要

組織には、ユーザー・レコードの変更情報を交換するプロビジョニング・システムが複数ある場合があります。また、複数のプロビジョニング・システムとのやり取りをするアプリケーションがある場合もあります。コネクタによって、2つのプロビジョニング・システム間、またはアプリケーションとプロビジョニング・システムとの間のやり取りが可能になり、各アプリケーションと他のアプリケーションとの同期がとられます。しかし、これらのシステムの組合せごとにカスタム・コネクタを構成するのは、多くのコストを必要とします。

1つの共通言語のアプリケーションまたはこれらのすべてのシステムが理解するプロトコルが、この問題の解決策です。それは SPML です。

SPML Web サービスでは、SPML リクエストを解釈し、これを Oracle Identity Manager コールに変換する Oracle Identity Manager の上位層が提供されます。

**関連項目：** SPML v2.0 仕様の詳細は、次の Web ページを参照してください。

<http://www.oasis-open.org/specs/index.php#spmlv2.0>

SPML Web サービスでは、インバウンド・プロビジョニング・リクエストのみがサポートされます。リコンシリエーション・イベントが生成されないため、リコンシリエーションはサポートされません。たとえば、リソースを OIM ユーザーにプロビジョニングするリクエストをし、ここでリソース・プロセスと子表へのデータの移入が必要な場合、このリクエストはサポートされません。これは、SPML Web サービスでリソース・オブジェクトに関連付けられている情報が認識されないためです。

---

---

**注意：** アウトバウンド・プロビジョニング・リクエストは、SPML プロビジョニング・フォーマット・プロバイダを含む汎用テクノロジー・コネクタを使用して送信されます。汎用テクノロジー・コネクタの詳細は、『Oracle Identity Manager 管理およびユーザー・コンソール・ガイド』を参照してください。

---

---

### 12.1.1 SPML Web サービスの機能のアーキテクチャ

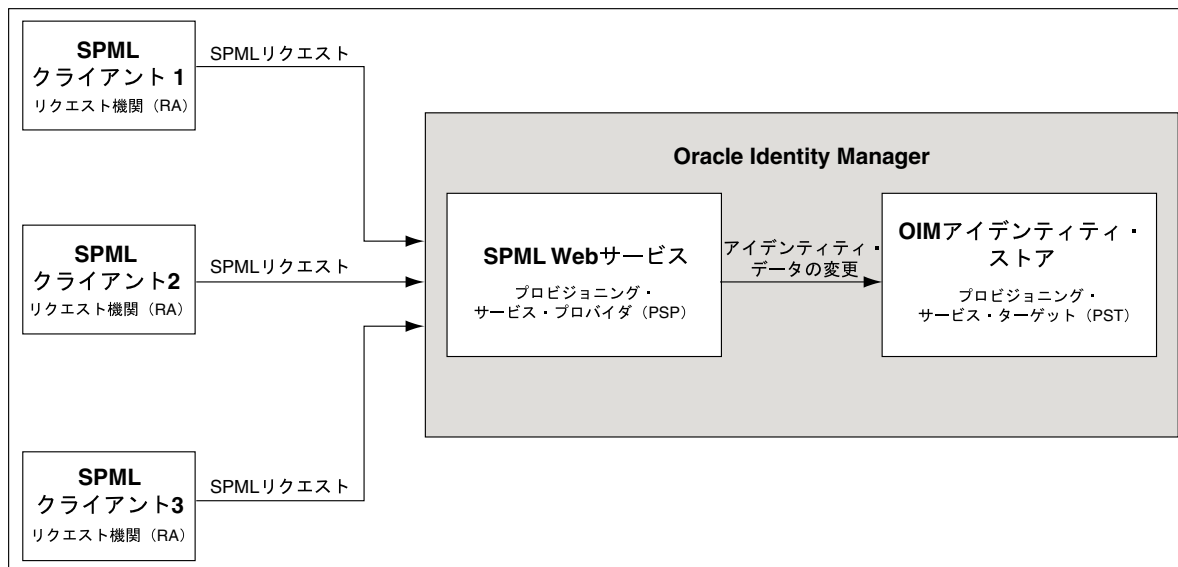
SPML Web サービスでは、SOAP メッセージの形で SPML リクエストが送受信されます。SPML モデルは、エンドツーエンド・プロビジョニング・シナリオに含まれる次のエンティティで構成されます。

- リクエスト局 (RA) : RA (リクエスタ) は、整形形式の SPML リクエストをプロビジョニング・サービス・プロバイダへ発行するコンポーネントです。
- プロビジョニング・サービス・プロバイダ (PSP) : PSP (プロバイダ) は、既知のリクエスタからの整形形式 SPML リクエストをリスニングおよび処理し、この結果を返すコンポーネントです。
- プロビジョニング・サービス・ターゲット (PST) : PST (ターゲット) は、プロビジョニング・アクションをプロバイダで使用可能にする先、またはエンドポイントを表します。

SPML プロトコルの実装により、プロビジョニング・リクエストの信頼できる交換、およびより複雑なアプリケーションレベルでのプロビジョニング機能を構築する基となるモデルが提供されます。SPML は、プロビジョニング・システムによって使用される管理リクエストを交換し、アイデンティティを管理および制御する言語です。

図 12-1 に、SPML Web サービスの機能のアーキテクチャを示します。

図 12-1 SPML アーキテクチャ



プロビジョニング・アプリケーションは、RA、PSP 両方の役割を担えます。次のようなシナリオを考えてみてください。

#### PSP としてのプロビジョニング・アプリケーション

このシナリオでは、クライアント・アプリケーションで、プロビジョニング・アプリケーションへ SPML リクエストが送信されます。プロビジョニング・アプリケーションでリクエストが実行され、クライアント・アプリケーションに SPML レスポンスが返されます。リクエストとレスポンスの交換は、同期、非同期のいずれでも行われます。これは、通常、インバウンド・シナリオと呼ばれます。Oracle Identity Manager では、これは SPML Web サービスを介して実装されます。

#### RA としてのプロビジョニング・アプリケーション

このシナリオでは、プロビジョニング・アプリケーションは SPML クライアントの役割を担い、リクエストを実行して SPML レスポンスを返す PST に、SPML リクエストを送信します。リクエストとレスポンスの交換は、同期でも非同期でも行われます。これは、通常、アウトバウンド・シナリオと呼ばれます。Oracle Identity Manager では、これは、SPML フォーマット・プロビジョニング・プロバイダを含む汎用テクノロジー・コネクタを介して実装されます。

#### RA および PSP としてのプロビジョニング・アプリケーション

---

**注意：** この機能は、Oracle Identity Manager リリース 9.1.0 ではサポートされていません。

---

このシナリオでは、クライアント・アプリケーションで、プロビジョニング・アプリケーションへ SPML リクエストが送信されます。ここで、プロビジョニング・アプリケーションは、リクエストを実行し、SPML レスポンスを返すプロビジョニング・ターゲットへリクエストを送ります。その後、プロビジョニング・アプリケーションで、クライアント・アプリケーションに SPML レスポンスが返されます。リクエスト / レスポンス / リクエストの交換は、同期でも非同期でも行われます。

## 12.2 SPML Web サービスでサポートされているプロビジョニング処理

SPML Web サービスでは、SPML v2.0 仕様で規定された最低準拠基準を満たす機能がサポートされます。この項では、SPML Web サービスでサポートされる各種処理について説明します。

---

**注意：** SPML Web サービスでは、UTF-8 エンコーディングのリクエストのみがサポートされます。

---

プロビジョニング・ターゲットでは、エンティティ（ユーザー、グループ、組織）を一意に識別するために `psoid` が使用されます。Oracle Identity Manager では、エンティティの指定に、オブジェクト・クラスとエンティティ・キーの組合せが必要です。エンティティ・キーは、Oracle Identity Manager でエンティティを作成すると返されるデータベース・キーです。Oracle Identity Manager での `psoid` の形式は次のとおりです。

```
objectclass:entitykey
```

たとえば、オブジェクト・クラスが `Users`、エンティティ・キーが `3` とします。この場合、指定する `psoid` は、`Users:3` です。

SPML Web サービスでは、次のプロビジョニング処理またはリクエストがサポートされます。

### ■ 追加処理

追加リクエスト処理では、リクエストされたターゲット・タイプのエントリが Oracle Identity Manager で作成されます。サポートされているターゲット・タイプは、`User`、`Group` および `Organization` です。この処理では、エントリの存在が確認され、エラーがあればこれがレポートされます。`memberOf` リレーションシップを使用してユーザーまたはグループを1つ以上のグループに関連付ける場合、この処理に、`User` および `Group` ターゲット・タイプの機能データを含めることもできます。この機能では、`administrator` 参照機能もサポートされ、1つ以上のグループを新規グループの管理者として割り当てる際に使用できます。

**関連項目：** 追加リクエストの例は B-1 ページの「[追加リクエスト](#)」

### ■ 変更処理

変更リクエスト処理では、リクエストされたターゲット・タイプの指定されたターゲット・エントリが Oracle Identity Manager で更新されます。サポートされているターゲット・タイプは、`User`、`Group` および `Organization` です。この処理では、エントリの存在が確認され、エラーがあればこれがレポートされます。エントリが存在する場合は、特定のデータでターゲットが更新されます。この処理では、機能データの変更も行われます。`User` ターゲット・タイプの場合、サポートされている参照機能は `memberOf` です。この処理は、ユーザーの参照の更新にも使用されます。ユーザーのグループへの追加やグループからの削除も行えます。

同様に、`Group` ターゲット・タイプの場合、サポートされている参照機能は、`memberOf` および `administrator` です。変更リクエストは、これらの参照の作成、置換または削除に使用できます。

**関連項目：** 変更リクエストの例は B-4 ページの「[変更リクエスト](#)」

### ■ 削除処理

削除リクエスト処理では、指定されたターゲット・エントリが Oracle Identity Manager から削除されます。サポートされているターゲット・タイプは、`User`、`Group` および `Organization` です。この処理では、エントリの存在が確認され、エラーがあればこれがレポートされます。エントリが存在する場合は、これが削除されます。再帰的削除処理はサポートされていません。



## ■ 参照の追加、置換または削除

参照は、他のエンティティから参照されるエンティティです。たとえば、ユーザー John Doe はグループのメンバーです。John Doe をそのグループから削除し、他のグループのメンバーにするとします。新しいグループでのこのユーザーの詳細の指定に参照を使用し、ユーザーを新しいグループのメンバーにできます。

次の参照が SPML でサポートされています。

- ユーザーのグループ・メンバーシップ参照の追加 / 置換 / 削除
- グループのグループ・メンバーシップ参照の追加 / 置換 / 削除
- グループの管理者参照の追加 / 置換 / 削除

## ■ ルックアップ処理

ルックアップ・リクエスト処理では、指定されたターゲット・エントリが Oracle Identity Manager でルックアップされます。サポートされているターゲット・タイプは、User、Group および Organization です。この処理ではエントリの存在が確認され、ルックアップ・データまたはエラー・レポートが返されます。ユーザーはユーザー ID を基に Oracle Identity Manager でルックアップされます。ルックアップ処理では、returnData タイプで指定した場合、オブジェクトに関連付けられている機能固有のデータを返すこともできます。

## ■ 検索処理

検索リクエスト処理では、指定されたターゲット・エントリが Oracle Identity Manager で検索されます。サポートされているターゲット・タイプは、User、Group および Organization です。この処理では、エントリの存在が確認され、エントリが使用可能かどうかに基づいて検索レスポンスが返されます。

検索リクエストでは、問合せの範囲はサポートされません。問合せに範囲がある場合でも、これは無視されます。検索問合せは、リクエストで指定された検索基準に AND 演算子を適用することで作成されます。SPML 検索リクエストでは、Directory Services Markup Language (DSML) フィルタを使用して異なる検索基準を指定できます。DSML フィルタでサポートされている唯一の処理は、equalityMatch です。

includeDataForCapability 要素はサポートされていません。returnData の値として指定した場合、検索処理ではすべての関連する参照が返されます。検索処理には、User、Group、Organization のどのコンテナに問い合わせるかに基づいてオブジェクト・クラス情報が必要とされます。つまり、オブジェクト・クラスが検索問合せの問合せ要素の 1 つとみなされます。

たとえば、Users オブジェクト・クラスの検索を行う場合、この情報をリクエストに組み込む必要があります。次のコード例でこれを示します。

```
<searchRequest returnData="identifier"xmlns="urn:oasis:names:tc:SPML:2:0:search"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
  <query scope="pso">
    <basePsoID ID=""/><and>
      <dsml:filter><dsml:equalityMatch name="Users.LastName">
        <dsml:values>Doe
        </dsml:values>
      </dsml:equalityMatch>
    </dsml:filter>
    <dsml:filter><dsml:equalityMatch name="Users.XellerateType">
      <dsml:values>End-User
      </dsml:values>
    </dsml:equalityMatch>
    </dsml:filter>
    <dsml:filter>
      <dsml:equalityMatch name="ObjectClass"><dsml:values>Users
      </dsml:values>
    </dsml:equalityMatch>
    </dsml:filter>
  </and>
```

```
</query>
</searchRequest>
```

検索が行われる basePsoID は、Organization または空の文字列である必要があります。(basePsoID は、psoid 形式 (Organization:key) と同じか、空の文字列のいずれかです。)

例: <basePsoID ID="Organization:7"/>, <basePsoID ID="" />

#### ■ パスワード処理

passwordRequest 処理には、setPasswordRequest と resetPasswordRequest の 2 つの処理があります。前者ではユーザーはパスワードを変更でき、後者では、Oracle Identity Manager でユーザーに定義されたパスワード・ポリシーに基づいて新しいパスワードがランダムに生成されます。サポートされるターゲット・タイプは、User です。resetPasswordRequest では、パスワードはデフォルトで default に設定されます。

#### ■ ユーザーの一時停止、再開またはアクティブ処理

SPML v2.0 仕様で規定された suspend 機能には次の 3 つの処理があります。

- suspend: Oracle Identity Manager でアクティブ状態のユーザーを無効にします。
- resume: Oracle Identity Manager で無効な状態のユーザーを有効にします。
- active: ユーザーの状態を返します (ユーザーが active 状態かどうかが表示されます)。

#### ■ ListTargets 処理

- listTargets は、プロビジョニングで使用可能なターゲットの決定に使用されます。この処理は、サポートされている各ターゲット・タイプについてプロビジョニング・システムでサポートされている属性の決定にも使用されます。この処理では、サポートされている 3 つのターゲット・タイプ (User、Group、Organization) すべてのスキーマ情報が返されます。

SPML Web サービスでは、次の処理はサポートされません。

- 検索 (iterate/closeterator/hasReference)
- 更新 (updates/iterate/closeterator)
- 非同期リクエスト (status/cancel)
- バッチ処理
- バルク更新 (bulkModify/bulkDelete)
- パスワードの期限切れまたは検証

---

**注意:** プロビジョニング・リクエストに日付属性を含める場合は、次の形式を使用する必要があります。

```
yyyy-MM-dd hh:mm:ss.ffffffffff
```

他の日付形式はサポートされていません。日付属性の割当てを含む追加リクエストの例は、B-8 ページの「日付書式付き追加リクエスト」を参照してください。

---

## 12.3 SPML Web サービスのデプロイ

---

**注意：** SPML Web サービスを Oracle Identity Manager とともに使用する場合は、Oracle Identity Manager のアップグレードの際に SPML Web サービスを再デプロイする必要があります。

EAR ファイルをカスタマイズした場合は、それらの変更を EAR ファイルに再適用してから再デプロイする必要があります。

---

SPML Web サービスは、OIMSpmlWS.ear という名前のエンタープライズ・アーカイブ (EAR) ファイルにパッケージされています。このファイルは Oracle Identity Manager のインストール時に生成されます。このファイルは次の場所に格納されます。

`OIM_HOME/SPMLWS`

アプリケーション・サーバーごとに異なる EAR ファイルがあり、各ファイルは SPMLWS ディレクトリの各アプリケーション・サーバーのフォルダに格納されています。この EAR ファイルは Oracle Identity Manager のインストール時に生成されます。

---

**注意：** Oracle Identity Manager と SPML Web サービスは、同じアプリケーション・サーバーにデプロイされる必要があります。これは、**コロケーション方式のデプロイ**として知られています。クラスタ環境では、Oracle Identity Manager がインストールされている各ノードに SPML Web サービスが必ずインストールされているようにしてください。

---

次のバッチ・ファイルを使用し、Oracle Identity Manager が稼働しているアプリケーション・サーバー上に SPML Web サービスをデプロイするスクリプトを実行します。

`OIM_HOME/xellerate/setup/spml_AppServerName`

このバッチ・ファイルを実行するには、操作環境に対応した手順を実行します。

---

**注意：** バッチ・ファイルを実行すると、次のログ・ファイルが作成されます。

`OIM_HOME/xellerate/logs/spml-AppServerName.log`

---

- [Oracle WebLogic Server への SPML Web サービスのデプロイ](#)
- [IBM WebSphere Application Server への SPML Web サービスのデプロイ](#)
- [JBoss Application Server への SPML Web サービスのデプロイ](#)
- [Oracle Application Server への SPML Web サービスのデプロイ](#)

### 12.3.1 Oracle WebLogic Server への SPML Web サービスのデプロイ

非クラスタ Oracle WebLogic Server インストールの場合：

次のコマンドを入力します。

UNIX の場合：

`OIM_HOME/setup/spml_weblogic.sh appserver_admin_password oim_db_user_password`

Microsoft Windows の場合：

`OIM_HOME%setup%spml_weblogic.cmd appserver_admin_password oim_db_user_password`

**クラスタ化された Oracle WebLogic Server インストールの場合：**

1. 管理者ノードで次のコマンドを入力します。

UNIX の場合：

```
OIM_HOME/xellerate/setup/spml_weblogic.sh appserver_admin_password  
oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME\xellerate\setup\spml_weblogic.cmd appserver_admin_password  
oim_db_user_password
```

2. 『Oracle Identity Manager Oracle WebLogic Server 用インストールおよび構成ガイド』の「Apache プロキシ・プラグインの構成」に示された手順を実行します。

## 12.3.2 IBM WebSphere Application Server への SPML Web サービスのデプロイ

**非クラスタ IBM WebSphere Application Server インストールの場合：**

次のコマンドを入力します。

UNIX の場合：

```
OIM_HOME/setup/spml_websphere.sh appserver_admin_password oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME\setup\spml_websphere.cmd appserver_admin_password oim_db_user_password
```

**クラスタ化された IBM WebSphere Application Server インストールの場合：**

1. 管理者ノードで次のコマンドを入力します。

UNIX の場合：

```
OIM_HOME/xellerate/setup/spml_websphere.sh appserver_admin_password  
oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME\xellerate\setup\spml_websphere.cmd appserver_admin_password  
oim_db_user_password
```

2. 『Oracle Identity Manager IBM WebSphere Application Server 用インストールおよび構成ガイド』の「IIS プラグインの構成」に示された手順を実行して plugin-cfg.xml ファイルを再生成します。

## 12.3.3 JBoss Application Server への SPML Web サービスのデプロイ

---

---

**注意：** クラスタ化された JBoss Application Server 環境での SPML Web サービスのデプロイはサポートされていません。

---

---

**非クラスタ JBoss Application Server インストールの場合：**

次のコマンドを入力します。

UNIX の場合：

```
OIM_HOME/xellerate/setup/spml_jboss.sh oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME\xellerate\setup\spml_jboss.cmd oim_db_user_password
```

## 12.3.4 Oracle Application Server への SPML Web サービスのデプロイ

非クラスタ Oracle Application Server インストールの場合：

1. 次のコマンドを入力します。

UNIX の場合：

```
OIM_HOME/setup/spml_oc4j.sh appserver_admin_password oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME%setup%spml_oc4j.cmd appserver_admin_password oim_db_user_password
```

2. 次のコマンド・スクリプトを入力します。

UNIX の場合：

```
OIM_HOME%xellerate%setup%spml_oc4j.sh appserver_admin_password oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME%xellerate%setup%spml_oc4j.cmd appserver_admin_password
oim_db_user_password
```

3. テキスト・エディタで次のファイルを開きます。

```
OC4J_HOME/j2ee/OC4J_instance/config/application.xml
```

4. application.xml ファイルの <imported-shared-libraries> セクションで、<import-shared-library name="apache.commons.logging"/> を <remove-inherited name="apache.commons.logging"/> に変更します。

つまり、<imported-shared-libraries> セクションは次のようにする必要があります。

```
<imported-shared-libraries>
  <import-shared-library name="adf.oracle.domain"/>
  <import-shared-library name="oracle.ifs.client"/>
  <remove-inherited name="apache.commons.logging"/>
</imported-shared-libraries>
```

クラスタ化された Oracle Application Server インストールの場合：

次の手順をクラスタの各ノードで実行します。

1. 次のコマンド・スクリプトを入力します。

UNIX の場合：

```
OIM_HOME%xellerate%setup%spml_oc4j.sh appserver_admin_password oim_db_user_password
```

Microsoft Windows の場合：

```
OIM_HOME%xellerate%setup%spml_oc4j.cmd appserver_admin_password
oim_db_user_password
```

2. テキスト・エディタで次のファイルを開きます。

```
OC4J_HOME/j2ee/OC4J_instance/config/application.xml
```

3. application.xml ファイルの <imported-shared-libraries> セクションで、<import-shared-library name="apache.commons.logging"/> を <remove-inherited name="apache.commons.logging"/> に変更します。

つまり、<imported-shared-libraries> セクションは次のようにする必要があります。

```
<imported-shared-libraries>
  <import-shared-library name="adf.oracle.domain"/>
  <import-shared-library name="oracle.ifs.client"/>
  <remove-inherited name="apache.commons.logging"/>
</imported-shared-libraries>
```

## 12.4 Oracle Web Services Manager を使用し、SPML Web サービスをデプロイするセキュリティの有効化

---

**注意：** Oracle Web Services Manager を使用して SPML Web サービスを保護する場合のみ、この項で説明する手順を実行します。

---

Oracle Web Services Manager (WSM) では、多様なデプロイ環境での Web サービスのインストール、構成および管理を容易にする機能が提供されます。

**関連項目：** Oracle WSM の詳細は、『Oracle Web Services Manager 管理者ガイド』

Oracle WSM を使用して、SPML Web サービスをセキュアにできます。SPML Web サービスに SOAP メッセージのリクエストが送信される際、Oracle WSM でこれが解釈されます。SOAP メッセージは、SOAP ヘッダーに Web Services Security (wsse) タグを含みます。この wsse:security タグに、認証される必要のあるユーザー資格証明が含まれます。SPML Web サービスをセキュアにするには、Oracle WSM Server Agent または Oracle WSM Gateway を使用します。

**関連項目：**

- Oracle WSM のインストールの詳細は、『Oracle Web Services Manager インストール・ガイド』

- Oracle WSM Server Agent および Oracle WSM Gateway の詳細は、『Oracle Web Services Manager デプロイメント・ガイド』

Oracle WSM Server Agent および Oracle WSM Gateway では、SPML Web サービスへのリクエストの転送前に SOAP メッセージから wsse:security タグが取り除かれます。Oracle WSM Server Agent または Oracle WSM Gateway をカスタム・ポリシー・ステップを使用するよう構成する必要があります。ここで、wsse:security タグからユーザー資格証明が抽出され、SOAP ヘッダーのカスタム・ヘッダー・タグに挿入されます。SPML Web サービスではここから資格証明が抽出されます。

次の項で、Oracle WSM Server Agent または Oracle WSM Gateway を使用して SPML Web サービスをセキュアにする構成手順を説明します。

- [Oracle WSM Server Agent の構成](#)
- [Oracle WSM Gateway の構成](#)

---

**注意：** Oracle WSM Server Agent では、IBM WebSphere Application Server および Oracle Application Server がサポートされます。Oracle WSM Gateway はすべてのアプリケーション・サーバーで使用できます。

---

## 12.4.1 Oracle WSM Server Agent の構成

SPML Web サービスをセキュアにするよう Oracle WSM Server Agent を構成するには、次の処理が必要です。

---

**注意：** Oracle WSM Server Agent は、SPML Web サービスのデプロイ前に構成します。

---

1. サーバー・エージェントの追加
2. サーバー・エージェントのポリシーの定義
3. サーバー・エージェントのインジェクト
4. SPML Web サービスのデプロイ

### 12.4.1.1 サーバー・エージェントの追加

サーバー・エージェントを追加する手順は次のとおりです。

**関連項目：** サーバー・エージェントの追加の詳細は、『Oracle Web Services Manager デプロイメント・ガイド』の第 6 章「Oracle WSM Agent のインストール」

1. Web Services Manager Control (<http://localhost:8888/ccore> など) を使用してサーバー・エージェントを作成し、リストから次の値を選択します。
  - コンポーネント・タイプ: **サーバー・エージェント**。
  - コンテナ・タイプ: Oracle Application Server の場合は、「**OC4J**」を選択します。IBM WebSphere Application Server の場合は、「**OTHER**」を選択します。
2. 「**登録**」を選択します。コンポーネント ID を使用してサーバー・エージェント・コンポーネントが生成されます。

### 12.4.1.2 サーバー・エージェントのポリシーの定義

Web Services Manager Control を使用して、サーバー・エージェントに関連付けるポリシーを定義します。ポリシーのデフォルトの実装は、次の場所にあります。

`OIM_HOME/SPMLWS/OWSMPolicy`

**関連項目：** サーバー・エージェントのポリシーの定義の詳細は、『Oracle Web Services Manager 管理者ガイド』の第 5 章「Oracle Web Services Manager のポリシー管理」

URL パターンとポリシーを関連付ける必要があります。これを行うには、「**ポリシー管理**」→「**ポリシーの管理**」→「**ポリシー**」と選択します。その後、「**マッピングの編集**」ボタンをクリックし、ポリシーに関連付ける URL パターンとして次のように入力します。

`/spmlws/HttpSoap11`

カスタム・ポリシーを構成するには、`OIM_HOME/SPMLWS/OWSMPolicy` のクラス・ファイル `com.oracle.xml.spmlws.ws.security.owsm.CustomPolicyStep` を次のファイルに含める必要があります。

`ORACLE_HOME/owsm/lib/extlib/coresvagent.jar`

サーバー・エージェント・コンポーネントの作成後、カスタム・ステップをそのコンポーネントに追加する必要があります。これを行うには、登録したコンポーネントの「ステップ」リンクをクリックします。適切な場所から CustomPolicyStep.xml を選択し、「アップロード」をクリックします。XML ファイルは、`OIM_HOME/SPMLWS/OWSMPolicy/com/oracle/xl/spmlws/ws/security/owsm` にあります。

この後、カスタム・ポリシー・ステップ名が使用可能なポリシーのリストに追加されます。

---

**注意：** カスタム・ポリシーを作成した場合は、そのクラス・ファイルを Web サービス EAR ファイルにある `coresvagent.jar` ファイルに含める必要があります。

---

### 12.4.1.3 サーバー・エージェントのインジェクト

サーバー・エージェントをインジェクトするには、次の手順を行う必要があります。

#### Oracle Application Server の場合

サーバー・エージェントをインジェクトする手順は次のとおりです。

1. `ORACLE_HOME/owsm/bin/agent.properties` ファイルの属性を次の値に変更します。
  - `agent.componentType`: ServerAgent
  - `agent.containerType`: OC4J
  - `agent.containerVersion`: Oracle Application Server の場合、10.1.3
  - `agent.component.id`: エージェントを作成し、Web Services Manager Control を使用して登録した際に生成されたコンポーネント ID を入力します。
2. `agent.properties` ファイルの次のプロパティを編集します。
  - `webservice.application.input`: EAR ファイルのフルパスと名前を入力します。
  - `webservice.application.webapp.name`: コメント解除し、WAR ファイル名 `spmlws.war` を入力します。

---

**注意：** WAR ファイルは `OIMSpmlWS.ear` ファイルにバンドルされています。

---

- `webservice.application.contexturi`: コンテキスト・ルート `/spmlws` を入力します。
3. `wsmadmin installAgent` コマンドを実行します。

#### IBM WebSphere Application Server の場合

サーバー・エージェントをインジェクトする手順は次のとおりです。

1. SOA 10.1.3.1 リリースに対するサーバー・エージェントとしては Oracle Application Server のみがサポートされているため、WebSphere の場合は、Oracle Technology Network の次の場所から必要な ZIP ファイルをまずダウンロードする必要があります。

UNIX の場合：

[http://download.oracle.com/otn/linux/ias/101310/soa\\_linux\\_x86\\_ws\\_agent101310.zip](http://download.oracle.com/otn/linux/ias/101310/soa_linux_x86_ws_agent101310.zip)

Microsoft Windows の場合：

[http://download.oracle.com/otn/nt/ias/101310/soa\\_windows\\_x86\\_ws\\_agent101310.zip](http://download.oracle.com/otn/nt/ias/101310/soa_windows_x86_ws_agent101310.zip)



- ZIP ファイルを任意の場所（/owsm など）に展開します。

**ヒント：**サーバー・エージェントのインジェクトの詳細は、展開した ZIP ファイルの `Readme_Agentinstall.pdf` ファイル

- `bin` ディレクトリを参照し、`agent.properties` ファイルを開いて次のプロパティを設定します。
  - `agent.componentType: serveragent`。
  - `agent.containerType: AXIS、WEBLOGIC、WEBSPHERE、TIBCO-BW、OC4J` など。
  - `agent.containerVersion: SPML Web サービスをデプロイする WebSphere のバージョン`。
  - `oc4j.home: /owsm/oc4j`（/owsm が ZIP ファイルを展開した場所の場合）。
  - `oc4j2ee.home: /owsm/oc4j/j2ee/home`（ここで、/owsm は ZIP ファイルを展開した場所）。
  - `webservice.application.input: Web アプリケーション入力ファイル名とパス`（WAR または EAR ファイルの場所）。たとえば、`/owsm/wars/HelloWorldImpl.war` です。
  - `webservice.application.webapp.name: WAR ファイルの場合は該当しません`。
  - `webservice.application.contexturi: WAR ファイルの場合は該当しません`。
  - `agent.component.id: エージェントを作成し、Oracle Web Services Manager Control を使用して登録した際に生成されたコンポーネント ID を入力します`。
  - `agent.policymanagerURL`（`http://hostname:8888/policymanager` など。ローカルホストではなく、システム名を指定。）
- `bin/coresv.properties` ファイルを開き、次のプロパティを設定します。
  - `coresv.home: /owsm`（/owsm が ZIP ファイルを展開した場所の場合）
  - `ant.home: ANT インストールのホーム・ディレクトリを設定`
  - `java.home: Java インストールのホーム・ディレクトリを設定`
  - `lib.dir: /owsm/lib`（/owsm が ZIP ファイルを展開した場所の場合）
  - `oc4j2ee.home: /owsm/oc4j/j2ee/home`（プロパティがある場合はオプション）
  - `external.oc4j.home: /owsm/oc4j`（プロパティがある場合はオプション）
- カスタム・ポリシーを構成するには、`OIM_HOME/SPMLWS/OWSMPolicy` のクラス・ファイル `com.oracle.xl.spmlws.ws.security.owsm.CustomPolicyStep` を次のファイルに含める必要があります。
 

```
/owsm/lib/extlib/coresvagent.jar
```

（/owsm が ZIP ファイルを展開した場所の場合）
- コマンド・ウィンドウで `bin` ディレクトリに移動します。injectAgent コマンドを実行します。このコマンドで、すべての JAR ファイルが、指定した WAR または EAR ファイルにインジェクトされます。（実行前に、`ant bin` ディレクトリへのパスを設定します。）

#### 12.4.1.4 SPML Web サービスのデプロイ

クライアント・エージェントのインストール後、SPML Web サービスをデプロイします。SPML Web サービスのデプロイ方法は、12-7 ページの「[SPML Web サービスのデプロイ](#)」を参照してください。

## 12.4.2 Oracle WSM Gateway の構成

SPML Web サービスをセキュアにするために Oracle WSM Gateway をデプロイするには、次の処理が必要です。

**関連項目：** Oracle WSM の特定の構成の詳細は、『Oracle Web Services Manager クイック・スタート・ガイド』

1. [Oracle WSM Gateway の登録](#)
2. [SPML Web サービスのゲートウェイへの登録](#)
3. [カスタム・ポリシーのゲートウェイへの追加](#)
4. [SPML Web サービスのデプロイ](#)
5. [WSDL ファイルの表示](#)

### 12.4.2.1 Oracle WSM Gateway の登録

ゲートウェイを登録する手順は次のとおりです。

1. Oracle Web Services Manager Control で、「**新規コンポーネントの追加**」をクリックします。
2. 「**新規コンポーネントの追加**」ページで、次の値を入力します。
  - コンポーネント名 : MyGateway など
  - コンポーネント・タイプ : ゲートウェイ (デフォルト値)
  - コンテナ・タイプ : Oracle Web Services Manager (デフォルト値)
  - コンポーネント URL:  
http://fully\_qualified\_host\_name:http\_port/gateway と入力  
(fully\_qualified\_host\_name は Oracle WSM の URL で、http\_port は Oracle WSM がホストされるポート)
  - コンポーネント・グループ : コンポーネント・グループのデフォルト値を受入れ
3. 「登録」をクリックします。
4. 「OK」をクリックします。

### 12.4.2.2 SPML Web サービスのゲートウェイへの登録

SPML Web サービスをゲートウェイに登録する手順は次のとおりです。

1. Oracle Web Services Manager Control のナビゲーション・ペインから、「**ポリシー管理**」をクリックします。
2. 「**サービスの登録**」をクリックします。
3. 「**サービス**」リンクをクリックします。
4. 「**新規サービスの追加**」をクリックします。「**新規サービスの追加**」ページが表示されます。このページで、次のようにサービスの詳細を入力します。
  - サービス名 : SPMLService
  - サービスのバージョン : 1.0
  - サービスの説明 : SPML リクエストの処理
  - WSDL URL: 次のような WSDL の場所  
http://host:port/spmlws/.../HttpSoap11?wsdl

5. 「次へ」をクリックします。「新規サービスのメッセンジャ・ステップを構成」ページが表示されます。このページで、URL が前述のページで指定した URL と一致することを確認します。「終了」をクリックし、残りのフィールドについてはデフォルト値を受け入れます。
6. 「ポリシーのコミット」をクリックします。

### 12.4.2.3 カスタム・ポリシーのゲートウェイへの追加

Web Services Manager Control を使用して、ゲートウェイに関連付けるポリシーを定義します。ポリシーのデフォルトの実装は、次の場所にあります。

`OIM_HOME/SPMLWS/OWSPolicy`

**関連項目：**ゲートウェイのポリシーの定義の詳細は、『Oracle Web Services Manager 管理者ガイド』の第 5 章「Oracle Web Services Manager のポリシー管理」

ポリシーでは、ユーザーの資格証明が WSSE セキュリティ・タグから抽出され、SOAP ヘッダーの次のカスタム・タグに追加されます。

```
<wsa1:OIMUser soap:mustUnderstand="0"
xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">
<wsa1:OIMUserId
xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">user1</wsa1:OIMUserId>
<wsa1:OIMUserPassword
xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">password1</wsa1:OIMUserPassword>
</wsa1:OIMUser>
```

SPML Web サービスでこのタグが解釈および処理されます。

カスタム・ポリシーを構成するには、`OIM_HOME/SPMLWS/OWSPolicy` のクラス・ファイル `com.oracle.xml.spmlws.ws.security.owsm.CustomPolicyStep` を次のファイルに含める必要があります。

`ORACLE_HOME/owsm/lib/coresv-4.0.jar`

`coresv-4.0.jar` にはポリシーに固有のすべてのクラス・ファイルが含まれるため、カスタム・ポリシー・ステップ・ファイルを含める必要があります。その後、Oracle Application Server を再起動します。

ゲートウェイの登録後、カスタム・ステップを追加する必要があります。カスタム・ステップを追加する手順は次のとおりです。

1. 「ステップの追加」ページで、カスタム・ポリシーの `CustomPolicy.xml` ファイルをアップロードします。この XML ファイルは、次のパスにあります。

`OIM_HOME/SPMLWS/OWSPolicy/com/oracle/xml/spmlws/ws/security/owsm`

2. Oracle Web Services Manager Control で「ステップの追加」ページを開くには、「**ポリシー管理**」を展開し、「**ポリシーの管理**」→「**ステップ**」→「**ステップの追加**」とクリックします。適切な場所から `CustomPolicy.xml` を選択し、「**アップロード**」をクリックします。カスタム・ポリシー・ステップの名前が使用可能なポリシーのリストに追加されます。
3. カスタム・ポリシーをリクエスト・ブロックのパイプラインに追加するには、「**ポリシー**」ページを参照します。「**ポリシー**」ページを開くには、「**ポリシー管理**」を展開し、「**ポリシーの管理**」→「**ポリシー**」とクリックし、最後に「**ポリシー**」をクリックします。

リクエスト・ブロックで、SPML Web サービスへ送られた有効な受信リクエストに対し、構成されたカスタム・ポリシーが実行されます。

#### 12.4.2.4 SPML Web サービスのデプロイ

クライアント・エージェントのインストール後、SPML Web サービスをデプロイします。SPML Web サービスのデプロイ方法は、12-7 ページの「[SPML Web サービスのデプロイ](#)」を参照してください。

#### 12.4.2.5 WSDL ファイルの表示

Web サービスの Web Service Description Language (WSDL) ファイルを表示する手順は次のとおりです。

1. Web Services Manager Control のナビゲーション・ペインから、「**ポリシー管理**」をクリックします。
2. 「**サービスの登録**」をクリックします。
3. ゲートウェイ (MyGateway) にアクセスするには、「**サービス**」をクリックします。
4. 必要なサービスについて、サービスのリストから「**編集**」をクリックします。
5. 「サービスの編集」ページで、「サービスの WSDL URL」フィールドに表示された URL をコピーします。

SPML クライアントでこの URL を使用し、SPML Web サービスにアクセスします。

## 12.5 デプロイ後のタスク

JBoss Application Server、Oracle WebLogic Server または IBM WebSphere Application Server を使用している場合、デプロイ後に実行するステップはありません。

IBM WebSphere Application Server 6.1 を使用している場合、xlDataObjectBeans.jar ファイルを抽出し、それを SPML Web サービスの WAR ファイルの WEB-INF/lib ディレクトリにコピーします。このファイルをコピーした後、WebSphere を再起動する必要があります。

**関連項目：**『Oracle Identity Manager IBM WebSphere Application Server 用インストールレーションおよび構成ガイド』の「xlDataObjectBeans.jar の抽出」

## 12.6 SSL 通信の有効化

この項では、SPML Web サービス用に Secure Sockets Layer (SSL) 通信を有効にする情報を提供します。この項で説明する手順を実行することをお勧めします。

---

---

**注意：**詳細は、アプリケーション・サーバー固有の SSL 構成のドキュメントを参照することをお勧めします。この項では、SPML Web サービスがサポートされているアプリケーション・サーバー別に、SSL 通信を有効にするために必要な最小限の情報を提供します。

この項ではアプリケーション・サーバーの特定のリリースの情報を提供しますが、異なるリリースを使用している場合は、手順の一部が異なる場合があります。

---

---

## 12.6.1 JBoss Application Server

次の項では、JBoss Application Server 4.2.3 GA にインストールされた SPML Web サービスで SSL 通信を有効にするために必要な情報を提供します。

### 12.6.1.1 前提条件

SSL 通信を有効にするための前提条件は次のとおりです。

- JBoss Application Server がインストールされ、Oracle Identity Manager と SPML Web サービスがデプロイされています。
- JBoss Application Server のホーム・ディレクトリは、E:¥jboss-4.2.3.GA です。
- アイデンティティ・ストアは jbosserver.jks で、パスワードは welcome です。
- 証明書のリクエストはローカルホストに対して行われます。
- 自己署名証明書の名前は jbosserver.cert です。
- プライベート・キー・エイリアスは serverjboss で、パスワードは welcome です。

### 12.6.1.2 SSL 証明書の設定

この項では、SSL を設定する手順を説明します。

**ヒント:** 保護機能を高めるには、新たに証明書（自己署名または CA 証明書）を作成し、クライアントおよびサーバー用の個別のキーストアとトラストストアを異なるパスワードで作成することをお勧めします。詳細は、アプリケーション・サーバーの SSL 構成のドキュメントを参照してください。

1. キーの生成
2. 証明書への署名
3. 証明書のエクスポート
4. server.xml ファイルの構成

#### キーの生成

keytool コマンドを使用してキーを生成します。次の keytool コマンドでは、アイデンティティ・ストア jbosserver.jks が生成されます。

```
keytool -genkey -alias serverjboss -keyalg RSA -keysize 1024 -dname
"CN=localhost,OU=Identity,O=Oracle,C=US" -keypass welcome -keystore
E:¥jboss-4.2.3.GA¥server¥jbosserver.jks -storepass welcome -storetype jks
```

#### 証明書への署名

次の keytool コマンドを使用して、作成した証明書に署名します。

```
keytool -selfcert -alias serverjboss -sigalg MD5withRSA -validity 2000 -keypass welcome
-keystore E:¥jboss-4.2.3.GA¥server¥jbosserver.jks -storepass welcome
```

#### 証明書のエクスポート

次の keytool コマンドを使用して、証明書をアイデンティティ・キーストアからファイル (jbosserver.cert など) にエクスポートします。

```
keytool -export -alias serverjboss -file E:¥jboss-4.2.3.GA¥server¥jbosserver.cert
-keypass welcome -keystore E:¥jboss-4.2.3.GA¥server¥jbosserver.jks -storepass welcome
-storetype jks -provider sun.security.provider.Sun
```

### server.xml ファイルの構成

次のエントリを server.xml ファイルに作成します。

```
<Connector port="8443" address="\${jboss.bind.address}"
    maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
    emptySessionPath="true"
    scheme="https" secure="true" clientAuth="false"
    sslProtocol="TLS"
    keystoreFile="E:\jboss-4.2.3.GA\server\jbossserver.jks"
    keystorePass="welcome"
    truststoreFile="E:\jboss-4.2.3.GA\server\jbossserver.jks"
    truststorePass="welcome"/>
```

前述の手順を行った後、サーバーを再起動して変更を有効にします。

---

**注意:** 「[証明書のエクスポート](#)」でエクスポートした証明書を使用して、クライアント側の SSL 通信用トラストストアにインポートできます。

---

## 12.6.2 Oracle WebLogic Server

次の項では、Oracle WebLogic Server にインストールされた SPML Web サービスで SSL 通信を有効にするために必要な情報を提供します。

### 12.6.2.1 前提条件

SSL 通信を有効にするための前提条件は次のとおりです。

- Oracle WebLogic Server がインストールされています。
- WebLogic ドメイン・ディレクトリは、C:\jboss\user\_projects\domains\oim です。
- Oracle WebLogic Server ホーム (WL\_HOME) ディレクトリは、C:\jboss\wlserver\_10.3 です。
- アイデンティティ・ストアは support.jks で、パスワードは support です。
- 証明書のリクエストは xellerate.oracle.com ホストおよび Oracle Identity Management Group に対して行われます。
- 自己署名証明書の名前は、supportcert.pem です。
- プライベート・キー・エイリアスは support で、パスワードは weblogic です。
- setEnv.cmd または setEnv.sh スクリプトを実行して PATH、CLASSPATH およびその他の変数を設定します。

### 12.6.2.2 SSL 証明書の設定

この項では、SSL を設定する手順を説明します。

**ヒント:** 保護機能を高めるには、新たに証明書（自己署名または CA 証明書）を作成し、クライアントおよびサーバー用の個別のキーストアとトラストストアを異なるパスワードで作成することをお勧めします。詳細は、アプリケーション・サーバーの SSL 構成のドキュメントを参照してください。

1. [キーの生成](#)
2. [証明書への署名](#)
3. [証明書のエクスポート](#)
4. [Oracle WebLogic Server の構成](#)

### キーの生成

提供された `keytool` コマンドを使用して、プライベート / パブリック証明書ペアを生成します。次のコマンドは、アイデンティティ・キーストア (`support.jks`) を作成します。必要に応じて、`keytool` コマンドに渡すパラメータの値を変更してください。keytool 引数に改行がないことを確認してください。

```
keytool -genkey
        -alias support
        -keyalg RSA
        -keysize 1024
        -dname "CN=xellerate.oracle.com, OU=Identity, O=Oracle Corporation,
L=RedwoodShores, S=California, C=US"
        -keypass weblogic
        -keystore C:\bea\user_projects\domains\oim\support.jks
        -storepass support
```

---

**注意:** ホスト名は、`xlconfig.xml` ファイルで使用する名前と同じものを使用します。たとえば、`xlconfig.xml` ファイルで `https://xellerate.oracle.com:7002` および `t3s://xellerate.oracle.com:7002` を使用する場合は、`keytool` コマンドの CN の値は `xellerate.oracle.com` である必要があります。SSL 証明書の生成には、IP アドレスではなく、ドメイン名 (たとえば、`xellerate.oracle.com`) を使用することをお勧めします。

---

### 証明書への署名

次のコマンドを使用して、作成した証明書に署名します。

```
keytool -selfcert -alias support
        -sigalg MD5withRSA
        -validity 2000
        -keypass weblogic
        -keystore C:\bea\user_projects\domains\oim\support.jks
        -storepass support
```

---

**注意:** 証明書の署名には、信頼できる認証局 (VeriSign や Thawte など) の使用をお勧めします。

---

### 証明書のエクスポート

次のコマンドを使用して、証明書をアイデンティティ・キーストアからファイル (`supportcert.pem` など) にエクスポートします。

```
keytool -export -alias support
        -file C:\bea\user_projects\domains\oim\supportcert.pem
        -keypass weblogic
        -keystore C:\bea\user_projects\domains\oim\support.jks
        -storepass support
```

**Oracle WebLogic Server の構成**

Oracle WebLogic Server を構成する手順は次のとおりです。

1. WebLogic Server 管理コンソールで、「環境」 → 「サーバー」 → *Server\_Name* → 「コンフィグレーション」をクリックし、「一般」をクリックします。
2. 「ロックして編集」をクリックします。
3. 「SSL リスン・ポートを有効化」を選択します。デフォルト・ポートは 7002 です。
4. 「キーストア」タブをクリックします。
5. キーストア・リストで「カスタム ID と Java 標準信頼」を選択します。
6. カスタム ID キーストア・フィールドに、カスタム ID キーストアのファイル名として `C:\%bea%\user_projects\domains\%oim%\support.jks` を指定します。
7. カスタム ID キーストア・タイプとして **JKS** を指定します。
8. カスタム ID キーストアのパスフレーズおよびカスタム ID キーストアのパスフレーズの確認のフィールドにパスワードを入力します。
9. 「保存」をクリックします。
10. 「SSL」タブをクリックします。
11. プライベート・キー・エイリアスとして、`support` と入力します。
12. パスワード（たとえば `support`）をプライベート・キーのパスフレーズおよびプライベート・キーのパスフレーズの確認のフィールドに入力します。
13. 「保存」をクリックします。
14. 「変更のアクティブ化」をクリックします。
15. サーバーを再起動し、変更を有効にします。

---

**注意:** 「証明書のエクスポート」でエクスポートした証明書を、クライアント側の SSL 通信用トラストストアにインポートできます。

次の `keytool` コマンドを使用して、証明書を SPML クライアント・トラストストアにインポートします。

```
keytool -import -alias serverwl -trustcacerts -file
D:\%bea%\user_projects\domains\%mydomain%\wlservercert.pem -keystore
<client-trust store> -storepass <client-trust-store password>
```

---

**12.6.3 IBM WebSphere Application Server**

次の項では、IBM WebSphere Application Server にインストールされた SPML Web サービスで SSL 通信を有効にするために必要な情報を提供します。

**12.6.3.1 前提条件**

SSL 通信を有効にするための前提条件は次のとおりです。

- IBM WebSphere Application Server がインストールされ、Oracle Identity Manager と SPML Web サービスがデプロイされています。
- IBM WebSphere Application Server を構成し、Oracle Identity Manager と SPML Web サービスをデプロイすると、SSL および非 SSL ポートを使用してアプリケーションにアクセスできます。
- SSL を使用してアプリケーションにセキュアにアクセスするには、ポート 9443 または `WC_defaultthost_secure` を使用します。次に例を示します。

```
https://localhost:9443/spmlws/HttpSoap11
```

- デフォルト・アイデンティティ・ストアは `key.p12` で、パスワードは `WebAS` です。



- デフォルト・トラストストアは `trust.p12` で、パスワードは `WebAS` です。

---

**注意:** SSL 通信の場合、デフォルトの証明書を `key.p12` からエクスポートします。

---

### 12.6.3.2 SSL 証明書の設定

この項の手順では次のことが行えます。

- 証明書のファイルへのエクスポート
- 証明書ファイルのインポート

#### 証明書のファイルへのエクスポート

IBM WebSphere Application Server では、IBM WebSphere のデフォルト・キーストア (`key.p12`) およびこのデフォルト証明書が使用されます。このデフォルト証明書をファイルにエクスポートする必要があります。次の `keytool` コマンドを使用してこれを行います。

```
IBM_JDK_HOME/jre/bin/keytool -export -alias default -file <Exported Certificate file>
-keypass WebAS -keystore FULL_PATH_OF_IBM_WEBSHERE "key.p12" -storepass WebAS
-storetype pkcs12 -provider com.ibm.crypto.provider.IBMJCE
```

前述のコマンドで、次の項目が適切な場所を指すよう置き換えます。

- IBM WebSphere Application Server の `key.p12` (デフォルト IBM キーストア) のフルパス
- `IBM_JDK_HOME`: IBM WebSphere Application Server Java フォルダ
- エクスポートされる証明書場所

#### 証明書ファイルのインポート

次の `keytool` コマンドを使用して、証明書ファイルを SPML Web サービス・クライアントのトラストストアにインポートします。

```
keytool -import -alias serverws -trustcacerts -file <Exported Certificate file>
-keystore E:¥SPMLTest¥mykeystore -storepass mypass -storetype jks
```

**ヒント:** 保護機能を高めるには、新たに証明書 (自己署名または CA 証明書) を作成し、クライアントおよびサーバー用の個別のキーストアとトラストストアを異なるパスワードで作成することをお勧めします。詳細は、アプリケーション・サーバーの SSL 構成のドキュメントを参照してください。

## 12.6.4 Oracle Application Server

次の項では、Oracle Application Server にインストールされた SPML Web サービスで SSL 通信を有効にするために必要な情報を提供します。

### 12.6.4.1 前提条件

SSL 通信を有効にするための前提条件は次のとおりです。

- Oracle Application Server がインストールされ、Oracle Identity Manager と SPML Web サービスがデプロイされています。
- Oracle Application Server 10.1.3 のインストール・ディレクトリは、`E:¥product¥10.1.3.1¥OracleAS_1` です。
- アイデンティティ・ストアは `oc4jserver.jks` で、パスワードは `welcome` です。
- 証明書のリクエストはローカルホストに対して行われます。
- 自己署名証明書の名前は `oc4jserver.cert` です。
- プライベート・キー・エイリアスは `serveroc4j` で、パスワードは `welcome` です。

## 12.6.4.2 SSL 証明書の設定

この項の手順では次のことが行えます。

**ヒント:** 保護機能を高めるには、新たに証明書（自己署名または CA 証明書）を作成し、クライアントおよびサーバー用の個別のキーストアとトラストストアを異なるパスワードで作成することをお勧めします。詳細は、アプリケーション・サーバーの SSL 構成のドキュメントを参照してください。

1. キーの生成
2. 証明書への署名
3. 証明書のエクスポート
4. Oracle Application Server の構成

### キーの生成

次の keytool コマンドを使用してキーを生成します。次の keytool コマンドでは、アイデンティティ・キーストア oc4jserver.jks が作成されます。

```
keytool -genkey -alias serveroc4j -keyalg RSA -keysize 1024 -dname
"CN=localhost,OU=Identity,O=Oracle,C=US" -keypass welcome -keystore
E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks -storepass welcome -storetype jks
```

### 証明書への署名

次の keytool コマンドを使用して、作成した証明書に署名します。

```
keytool -selfcert -alias serveroc4j -sigalg MD5withRSA -validity 2000 -keypass welcome
-keystore E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks -storepass welcome
```

### 証明書のエクスポート

次の keytool コマンドを使用して、証明書をアイデンティティ・キーストアからファイルにエクスポートします。

```
keytool -export -alias serveroc4j -file E:\product\10.1.3.1\OracleAS_1\oc4jserver.cert
-keypass welcome -keystore E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks -storepass
welcome -storetype jks -provider sun.security.provider.Sun
```

### Oracle Application Server の構成

Oracle Application Server を構成する手順は次のとおりです。

1. E:\product\10.1.3.1\OracleAS\_1\j2ee\home\config\default-web-site.xml ファイルのコピーを同じ場所に作成し、secure-web-site.xml という名前に変更します。
2. secure-web-site.xml ファイルで、次の変更を行います。
  - port 属性: 4443
  - secure: true
  - protocol: https

次に例を示します。

```
web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/
web-site-10_0.xsd" port="4443" secure="true" protocol="https"
display-name="OC4J 10g (10.1.3) Default Web Site" schema-major-version="10"
schema-minor-version="0">
```

3. 同じファイルの `web-site` ノードの下に、次の例のようなキーストアを指す新規要素 `ssl-config` を追加します。

```
<ssl-config keystore="E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks"
keystore-password="welcome" />
```

4. `E:\product\10.1.3.1\OracleAS_1\j2ee\home\config\server.xml` ファイルに次のエントリを追加します。

```
<web-site path="./secure-web-site.xml"/>
```

5. `E:\product\10.1.3.1\OracleAS_1\opmn\conf\opmn.xml` ファイルに次の追加を行います。

```
<port id="secure-web-site" range="4443" protocol="https"> under
<ias-component id="default_group">
```

6. `opmn.xml` ファイルの `<ias-component id="default_group">` タグの下に `<data id="java-options" value="-Xrs"` に、次の追加を行います。

```
-Djavax.net.ssl.trustStore=E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks
-Djavax.net.ssl.trustStorePassword=welcome
-Djavax.net.ssl.keyStore=E:\product\10.1.3.1\OracleAS_1\oc4jserver.jks
-Djavax.net.ssl.keyStorePassword=welcome"
```

7. サーバーを再起動し、変更を有効にします。

## 12.6.5 Oracle HTTP Server への HTTP 通信に対する SSL の有効化

次の項では、Oracle HTTP Server への HTTP 通信での SSL の有効化に関する情報を提供します。

デフォルトでは、Oracle HTTP Server は、SSL および SSL 証明書ストアを使用するよう構成されています。証明書ストアは、`ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default/` にあります。`ORACLE_HOME/Apache/Apache/conf/ssl.conf` ファイルの `listen` パラメータは、Oracle HTTP Server で使用される SSL ポートを指します。

インストールに付属のデフォルト証明書ストアを使用する場合、構成は変更しません。

**ヒント：** 保護機能を高めるには、新たに証明書（自己署名または CA 証明書）を作成し、クライアントおよびサーバー用の個別のキーストアとトラストストアを異なるパスワードで作成することをお勧めします。詳細は、アプリケーション・サーバーの SSL 構成のドキュメントを参照してください。

### 12.6.5.1 証明書のエクスポート

証明書をデフォルト Oracle ウォレット

`ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default/ewallet.p12` からエクスポートする必要があります。この証明書は、Design Console で Oracle Application Server を信頼するために使用されます。証明書をエクスポートする手順は次のとおりです。

1. Oracle Wallet Manager Console を使用して、`ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default/ewallet.p12` ファイルを開きます。これを行うには、「ウォレットを開く」をクリックし、ウォレットの場所を参照します。
2. ストア・パスワードを要求されたら、**welcome** を入力します。
3. 「証明書（待機中）」を右クリックし、「ユーザー証明書のエクスポート」をクリックします。
4. ファイルを `server.cert` の名前で保存します。

**関連項目：** Oracle Wallet Manager の詳細は、『Oracle Application Server 管理者ガイド』の Secure Sockets Layer に関する項を参照してください。

## 12.7 SPML Web サービスのクライアントの開発

この項では、SPML Web サービスのクライアントを作成する際に利用できる情報およびガイドラインを提供します。

---

**注意：**この章では、SPML Web サービスのクライアントを SPML クライアントと呼びます。

---

SPML クライアントを開発するには、クライアントを開発する各アプリケーション・サーバーの WSDL ファイルを参照する必要があります。各アプリケーション・サーバーの WSDL ファイルは、`OIM_HOME/SPMLWS/SampleHttpClient/wsd1` ディレクトリにあります。

SPML クライアントのサンプルのコード・ファイルは次の場所にあります。

`OIM_HOME/SPMLWS/SampleHttpClient`

コード例を理解しやすいようにしておくには、この SPML クライアントでは HTTP/S 接続のかわりに SPML Web サービスへの HTTP 接続を使用します。

この SPML クライアントのサンプルでは、SOAP リクエストを含む XML ファイルを参照し、SPML Web サービスへの HTTP POST を行います。

このリリースの Oracle Identity Manager には、SOAP リクエストの一連のサンプルが含まれています。この SOAP リクエストのサンプル・ファイルへアクセスするパスは次のとおりです。

`SampleHttpClient/sampleRequests/Application Server`

`sampleRequests` ディレクトリには、サポートされているアプリケーション・サーバーごとに SOAP リクエストが含まれています。

---

**注意：**前述のとおり、SPML Web サービスでは、UTF-8 エンコーディングのリクエストのみがサポートされます。クライアント実装では、`content-type` ヘッダーの `charset` に UTF-8 を指定して、HTTP での SPML リクエストの受渡しを行うことをお勧めします。サンプル・クライアントでの `sendSOAPRequest` 機能の実装については、`OIM_HOME/SPMLWS/SampleSPMLClient/src/testspml/HttpConnect.java` ファイルを参照してください。また、リクエストの送信前に、データを `byte []` にシリアルライズする必要があります。`SendSPMLRequest.java` ファイルの `main` 関数では、ファイルが `byte []` に直接コピーされ、HTTP 接続を介して送信されます。

---

次の項で、SPML クライアントの開発時に利用できる情報を提供します。

- [サポートされている SPML の処理](#)
- [認証](#)
- [SPML リクエストに含まれるフィールド](#)
- [SOAP ヘッダーの構造](#)
- [SOAP SPML メッセージの例](#)

## 12.7.1 サポートされている SPML の処理

前述のとおり、SPML Web サービスでは次の処理がサポートされています。

- 追加処理
- 変更処理  
変更処理の SPML リクエストに必ず `psoid` が含まれているようにします。
- 削除処理  
削除処理の SPML リクエストに必ず `psoid` が含まれているようにします。
- 参照の追加、置換または削除
- ルックアップ処理
- 検索処理
- パスワード処理  
パスワード処理の SPML リクエストに必ず `psoid` が含まれているようにします。
- ユーザーの一時停止、再開またはアクティブ処理  
一時停止機能には、`suspendRequest`、`resumeRequest` および `activeUser` 処理があります。`resumeRequest` および `activeUser` 処理の SPML リクエストに必ず `psoid` が含まれているようにします。
- ListTargets 処理

---

**注意：** `psoid` の詳細は、12-4 ページの「[SPML Web サービスでサポートされているプロビジョニング処理](#)」を参照してください。

---

## 12.7.2 認証

SPML クライアントは、送信される SPML リクエストごとに認証される必要があります。これは、認可されていないユーザーの SPML Web サービスの使用を防ぐためです。

SPML クライアントは、次の方法で認証されます。SPML クライアントでは SOAP ヘッダーを使用してユーザー資格証明を Web サービスへ送信します。これは、次の方法で行われます。

- 資格証明がカスタム・タグのヘッダー情報として指定されます。
- SPML Web サービスをセキュアにするよう Oracle WSM を構成します。この場合、資格証明は標準の WSSE タグで送信されます。デフォルトのポリシー実装でこの資格証明が処理されます。製品に同梱のデフォルト・ポリシー・ファイルは次のパスにあります。

`OIM_HOME/SPMLWS/OWSMPolicy`

---

**注意：** Oracle WSM を SPML Web サービスとの組合せで構成する詳細は、12-10 ページの「[Oracle Web Services Manager を使用し、SPML Web サービスをデプロイするセキュリティの有効化](#)」を参照してください。

---

### 12.7.3 SPML リクエストに含まれるフィールド

表 12-1 に、SPML リクエストに含まれる必須フィールドおよび必須でないフィールドを示します。

**注意：** 次の表では、必須フィールド列に `psoid` を示す行があります。これらのリクエストでは、Oracle Identity Manager 属性は必要ありません。

表 12-1 SPML リクエストの必須フィールドおよび必須でないフィールド

SPML リクエスト	必須フィールド	必須でないフィールド
ユーザーに対する <code>addRequest</code>	Users.User ID Users.First Name Users.Last Name Organizations.Organization Name Users.Xellerate Type Users.Role Users.Password	ユーザーの作成に関連する他の OIM User フィールドは必須ではありません。
グループに対する <code>addRequest</code>	Groups.Group Name	グループの作成に関連する他の OIM User フィールドは必須ではありません。
組織に対する <code>addRequest</code>	Organizations.Organization Name	Organizations.Type Organizations.Parent Name
ユーザーに対する <code>deleteRequest</code>	<code>psoid ID="Users:7"</code>	
グループに対する <code>deleteRequest</code>	<code>psoid ID="Groups:7"</code>	
組織に対する <code>deleteRequest</code>	<code>psoid ID="Organizations:9"</code>	
ユーザーに対する <code>modifyRequest</code>	<code>psoid ID="Users:5"</code> 変更する属性に対応する 1 つ以上の変更要素	
グループに対する <code>modifyRequest</code>	<code>psoid ID="Groups:5"</code> 変更する属性に対応する 1 つ以上の変更要素	
組織に対する <code>modifyRequest</code>	<code>psoid ID="Organizations:3"</code> 変更する属性に対応する 1 つ以上の変更要素	
ユーザーに対する <code>lookupRequest</code>	<code>psoid ID="Users:7"</code>	
グループに対する <code>lookupRequest</code>	<code>psoid ID="Groups:7"</code>	
組織に対する <code>lookupRequest</code>	<code>psoid ID="Organizations:9"</code>	
ユーザーに対する <code>suspendRequest</code>	<code>psoid ID="Users:7"</code>	
ユーザーに対する <code>resumeRequest</code>	<code>psoid ID="Users:7"</code>	
ユーザーに対する <code>activeRequest</code>	<code>psoid ID="Users:7"</code>	
ユーザーに対する <code>setPasswordRequest</code>	<code>psoid ID="Users:7"</code>	

表 12-1 SPML リクエストの必須フィールドおよび必須でないフィールド (続き)

SPML リクエスト	必須フィールド	必須でないフィールド
ユーザーに対する resetPasswordRequest	psoID ID="Users:7"	
ユーザーに対する searchRequest	<p>検索が行われる basePsoID は、Organization または空の文字列である必要があります。</p> <p>例: &lt;basePsoID ID="Organization:7"/&gt;, &lt;basePsoID ID="" /&gt;</p> <p>検索を行う基となる属性について等価比較を指定する一連の dsml:filter 条件。</p> <p><b>注意:</b> オブジェクト・クラスの情報が入力フィルタの 1 つの一部として送信され、検索が行われるコンテナを SPML Web サービスに指定します。次に例を示します。</p> <pre>&lt;dsml:filter&gt;&lt;dsml:equalityMatch name="Object Class"&gt; &lt;dsml:values&gt;Users&lt;/dsml:values&gt; &lt;/dsml:equalityMatch&gt;&lt;/dsml:filter&gt;</pre>	
グループに対する searchRequest	<p>検索が行われる basePsoID は、空の文字列である必要があります。</p> <p>検索を行う基となる属性について等価比較を指定する一連の dsml:filter 条件。</p> <p><b>注意:</b> オブジェクト・クラスの情報が入力フィルタの 1 つの一部として送信され、検索が行われるコンテナを SPML Web サービスに指定します。次に例を示します。</p> <pre>&lt;dsml:filter&gt;&lt;dsml:equalityMatch name="Object Class"&gt; &lt;dsml:values&gt;Groups&lt;/dsml:values&gt; &lt;/dsml:equalityMatch&gt;&lt;/dsml:filter&gt;</pre>	
組織に対する searchRequest	<p>検索が行われる basePsoID は、Organization または空の文字列である必要があります。</p> <p>検索を行う基となる属性について等価比較を指定する一連の dsml:filter 条件。</p> <p><b>注意:</b> オブジェクト・クラスの情報が入力フィルタの 1 つの一部として送信され、検索が行われるコンテナを SPML Web サービスに指定します。次に例を示します。</p> <pre>&lt;dsml:filter&gt;&lt;dsml:equalityMatch name="Object Class"&gt;&lt;dsml:values&gt; Organizations&lt;/dsml:values&gt; &lt;/dsml:equalityMatch&gt;&lt;/dsml:filter&gt;</pre>	
listTargetRequest	なし	

## 12.7.4 SOAP ヘッダーの構造

SPML Web サービスには Oracle Identity Manager 資格証明が必要です。これは、SPML をセキュアにするために Oracle WSM が使用されているかどうかに応じて SOAP ヘッダーに指定されます。この情報について、次の項で説明します。

### カスタム・セキュリティ・タグの使用

SOAP リクエストが SPML Web サービスに直接送信される場合、カスタム・セキュリティ・タグ (`wsa1:OIMUser`) が SOAP ヘッダーに使用され、Oracle Identity Manager 資格証明が組み込まれます。SPML Web サービスでこのタグが解釈され、サーバー側のハンドラで、次の SOAP ヘッダー例で示すような資格証明情報が抽出されます。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsa1:OIMUser soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">
<wsa1:OIMUserPassword>password1</wsa1:OIMUserPassword>
<wsa1:OIMUserId>user1</wsa1:OIMUserId>
</wsa1:OIMUser>
</soapenv:Header>
.....</soapenv:Envelope>
```

### WSSE セキュリティ・タグの使用

SPML Web サービスをセキュアにするために Oracle WSM Gateway または Agent が使用されている場合、SPML SOAP メッセージが Gateway または Agent によってインターセプトされます。この場合、Oracle Identity Manager 資格証明は、次の例で示すように標準 wsse セキュリティ・タグで指定されます。

```
soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken>
<wsse:Username>user1</wsse:Username>
<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password1</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
.....</soapenv:Envelope>
```

### wsa1:lang タグ

SPML クライアントでは、Oracle Identity Manager 資格証明情報とともに、ロケール情報も SPML Web サービスに送信されます。これには、SOAP ヘッダーの `wsa1:lang` タグが使用されます。これらのタグは SPML Web サービスによって処理されます。Oracle Web Services Manager が構成されている場合は、SPML Web サービスではこのタグは無視されます。この場合、ヘッダー情報は次のようになります。



カスタム・タグの場合:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsa1:OIMUser soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">
<wsa1:OIMUserPassword>password1</wsa1:OIMUserPassword>
<wsa1:OIMUserId>user1</wsa1:OIMUserId>
</wsa1:OIMUser>
<wsa1:lang soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">
<wsa1:language>en</wsa1:language>
<wsa1:sublanguage>US</wsa1:sublanguage>
</wsa1:lang>
</soapenv:Header>
.....</soapenv:Envelope>
```

WSSE タグの場合:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1
.0.xsd">
<wsse:UsernameToken>
<wsse:Username>user1</wsse:Username>
<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.
0#PasswordText">password1</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
<wsa1:lang soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:wsa1="http://xmlns.oracle.com/OIM/provisioning">
<wsa1:language>en</wsa1:language>
<wsa1:sublanguage>US</wsa1:sublanguage>
</wsa1:lang>

</soapenv:Header>
.....</soapenv:Envelope>
```

## 12.7.5 SOAP SPML メッセージの例

次の SOAP SPML メッセージは、Oracle Application Server 上の SPML Web サービスに対する追加リクエスト処理の例です。

カスタム・セキュリティ・タグを使用：

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>
<wsa:OIMUser soap:mustUnderstand="0"
xmlns:wsa="http://xmlns.oracle.com/OIM/provisioning">
<wsa:OIMUserId
xmlns:wsa="http://xmlns.oracle.com/OIM/provisioning">user1</wsa:OIMUserId>
<wsa:OIMUserPassword
xmlns:wsa="http://xmlns.oracle.com/OIM/provisioning">password1</wsa:OIMUserPassword>
</wsa:OIMUser>
</soap:Header>
<soap:Body>
<SPMLv2Document xmlns="http://xmlns.oracle.com/OIM/provisioning">
<addRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core">
<data>
<dsm1:attr name="objectclass">
<dsm1:value>Users</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.User ID">
<dsm1:value>John Doe</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.First Name">
<dsm1:value>John</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.Last Name">
<dsm1:value>Doe</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Organizations.Organization Name">
<dsm1:value>Xellerate Users</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.Xellerate Type">
<dsm1:value>End-User</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.Role">
<dsm1:value>Full-Time</dsm1:value>
</dsm1:attr>
<dsm1:attr name="Users.Password">
<dsm1:value>welcome</dsm1:value>
</dsm1:attr>
</data>
</addRequest>
</SPMLv2Document>
</soap:Body>
</soap:Envelope>
```

WSSE セキュリティ・タグを使用：

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

```
<wsse:UsernameToken>
  <wsse:Username>user1</wsse:Username>
  <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.
0#PasswordText">password1</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
  <SPMLv2Document xmlns="http://xmlns.oracle.com/OIM/provisioning">
    <addRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core">
      <data>
        <dsm1:attr name="objectclass">
          <dsm1:value>Users</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.User ID">
          <dsm1:value>John Doe</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.First Name">
          <dsm1:value>John</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.Last Name">
          <dsm1:value>Doe</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Organizations.Organization Name">
          <dsm1:value>Xellerate Users</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.Xellerate Type">
          <dsm1:value>End-User</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.Role">
          <dsm1:value>Full-Time</dsm1:value>
        </dsm1:attr>
        <dsm1:attr name="Users.Password">
          <dsm1:value>welcome</dsm1:value>
        </dsm1:attr>
      </data>
    </addRequest>
  </SPMLv2Document>
</soapenv:Body>
</soapenv:Envelope>
```



---

## Design Console ユーザー用リファレンス

以前のリリースのこの付録では、Design Console の処理およびアダプタ・タスク・マッピング情報が説明されていました。このリリース以降、この情報は、『Oracle Identity Manager リファレンス』の「Design Console のアクション」および「アダプタ・マッピング情報」に記載されています。



---

## SPML メッセージの例

この付録では、SPML Web サービスでサポートされている SOAP SPML メッセージの例を示します。SPML メッセージは SOAP リクエストに組み込まれます。

### 追加リクエスト

次の例には、ユーザー ID John Doe を使用してユーザーを作成し、Groups 5 と Groups 6 の 2 つのグループにサブスクライブする追加リクエスト処理が含まれています。

```
<addRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core">
  <data>
    <dsm1:attr name="objectclass">
      <dsm1:value>Users</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.User ID">
      <dsm1:value>John Doe</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.First Name">
      <dsm1:value>John</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.Last Name">
      <dsm1:value>Doe</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Organizations.Organization Name">
      <dsm1:value>Xellerate Users</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.Xellerate Type">
      <dsm1:value>End-User</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.Role">
      <dsm1:value>Full-Time</dsm1:value>
    </dsm1:attr>
    <dsm1:attr name="Users.Password">
      <dsm1:value>welcome</dsm1:value>
    </dsm1:attr>
  </data>
  <capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
    <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
      <toPsoID ID="Groups:5" />
    </reference>
    <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
      <toPsoID ID="Groups:6" />
    </reference>
  </capabilityData>
</addRequest>
```

---

次の例には、前述の追加リクエスト処理に対する成功レスポンスが含まれています。

```
<addResponse status="success">
<ps0>
<ps0ID ID="Users:7"/>
<data>
<attr name="Users.User ID">
<value>John Doe</value>
</attr>
<attr name="Users.Key">
<value>7</value>
</attr>
<attr name="Users.Last Name">
<value>Doe</value>
</attr>
<attr name="Users.First Name">
<value>John</value>
</attr>
<attr name="Users.Xellerate Type">
<value>End-User</value>
</attr>
<attr name="Users.Creation Date">
<value>2007-08-28 12:42:32.147</value>
</attr>
<attr name="Users.Updated By">
<value>1</value>
</attr>
<attr name="Users.Update Date">
<value>2007-08-28 12:42:36.38</value>
</attr>
<attr name="Users.Status">
<value>Active</value>
</attr>
<attr name="Users.Disable User">
<value>0</value>
</attr>
<attr name="Users.Lock User">
<value>0</value>
</attr>
<attr name="Organizations.Key">
<value>1</value>
</attr>
<attr name="Users.Role">
<value>Full-Time</value>
</attr>
<attr name="Organizations.Organization Name">
<value>Xellerate Users</value>
</attr>
<attr name="Users.Provisioned Date">
<value>2007-08-28 12:42:32.147</value>
</attr>
<attr name="Users.Change Password At Next Logon">
<value>1</value>
</attr>
<attr name="objectclass">
<value>Users</value>
</attr>
</data>
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:5"/>
</reference>
</addResponse>
```



```

<reference typeOfReference="memberOf">
<toPsoID ID="Groups:6"/>
</reference>
</capabilityData>
</pso>
</addResponse>

```

次の例には、指定したユーザー ID のユーザーがすでに存在している場合の追加リクエスト処理に対する失敗レスポンスが含まれています。

```

<addResponse status="failure" error="alreadyExists">
<errorMessage>
exception=tcDuplicateUserException;errorMessage=Duplicate_User
</errorMessage>
</addResponse>

```

次の例には、グループを作成し、Groups:4 と Groups:5 の 2 つのグループにサブスクライブする追加リクエスト処理が含まれています。このリクエストでは、グループを Groups:7 と Groups:8 の 2 つの管理者グループへ割り当てることも行われます。

```

<addRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
<data>
<dsml:attr name="objectclass">
<dsml:value>Groups</dsml:value>
</dsml:attr>
<dsml:attr name="Groups.Group Name">
<dsml:value>Add Group40</dsml:value>
</dsml:attr>
</data>
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="memberOf" xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:4" />
</reference>
<reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:5" />
</reference>
<reference typeOfReference="administrator"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:7" />
</reference>
<reference typeOfReference="administrator"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:8" />
</reference>
</capabilityData>
</addRequest>

```

次の例には、グループを作成する、前述の追加リクエスト処理に対する成功レスポンスが含まれています。

```

<addResponse status="success">
<pso>
<psoID ID="Groups:11"/>
<data>
<attr name="Groups.Key">
<value>11</value>
</attr>
<attr name="Groups.Group Name">
<value>Add Group40</value>
</attr>
<attr name="Groups.Updated By">

```

```

<value>1</value>
</attr>
<attr name="Groups.Update Date">
<value>2007-08-28 15:22:04.953</value>
</attr>
<attr name="Groups.Creation Date">
<value>2007-08-28 15:22:04.953</value>
</attr>
<attr name="Groups.Created By">
<value>1</value>
</attr>
<attr name="objectclass">
<value>Groups</value>
</attr>
</data>
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:4"/>
</reference>
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:5"/>
</reference>
<reference typeOfReference="administrator">
<toPsoID ID="Groups:1"/>
</reference>
<reference typeOfReference="administrator">
<toPsoID ID="Groups:7"/>
</reference>
<reference typeOfReference="administrator">
<toPsoID ID="Groups:8"/>
</reference>
</capabilityData>
</pso>
</addResponse>

```

次の例には、グループがすでに存在した場合の前述の処理に対する失敗レスポンスが含まれています。

```

<addResponse status="failure" error="alreadyExists">
<errorMessage>exception=Duplicate_Group</errorMessage>
</addResponse>

```

### 変更リクエスト

次の例には、ユーザーの2つのグループへのサブスクリプションを解除し、Groups:4 と Groups:5 の2つのグループに新たにサブスクリプションする変更リクエスト処理が含まれています。

```

<modifyRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core">
<psoID ID="Users:31"></psoID>
  <modification modificationMode="add">
    <capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
      <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
        <toPsoID ID="Groups:4" />
      </reference>
      <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
        <toPsoID ID="Groups:5"/>
      </reference>
    </capabilityData>
  </modification>

```

```

<modification modificationMode="delete">
  <capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
    <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
      <toPsoID ID="Groups:6"/>
    </reference>
    <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
      <toPsoID ID="Groups:7"/>
    </reference>
  </capabilityData>
</modification>
</modifyRequest>

```

次の例には、前述の変更リクエスト処理に対する成功レスポンスが含まれています。

```

<modifyResponse status="success">
  <psso>
  <pssoID ID="Users:31"/>
  <data>
  <attr name="Users.User ID">
  <value>John Doe</value>
  </attr>
  <attr name="Users.Key">
  <value>7</value>
  </attr>
  <attr name="Users.Last Name">
  <value>Doe</value>
  </attr>
  <attr name="Users.First Name">
  <value>John</value>
  </attr>
  <attr name="Users.Xellerate Type">
  <value>End-User</value>
  </attr>
  <attr name="Users.Creation Date">
  <value>2007-08-28 12:42:32.147</value>
  </attr>
  <attr name="Users.Updated By">
  <value>1</value>
  </attr>
  <attr name="Users.Update Date">
  <value>2007-08-28 12:42:36.38</value>
  </attr>
  <attr name="Users.Status">
  <value>Active</value>
  </attr>
  <attr name="Users.Disable User">
  <value>0</value>
  </attr>
  <attr name="Users.Lock User">
  <value>0</value>
  </attr>
  <attr name="Organizations.Key">
  <value>1</value>
  </attr>
  <attr name="Users.Role">
  <value>Full-Time</value>
  </attr>
  <attr name="Organizations.Organization Name">
  <value>Xellerate Users</value>
  </attr>
  <attr name="Users.Provisioned Date">

```

```

<value>2007-08-28 12:42:32.147</value>
</attr>
<attr name="Users.Change Password At Next Logon">
<value>1</value>
</attr>
</data>
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:3"/>
</reference>
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:4"/>
</reference>
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:5"/>
</reference>
</capabilityData>
</pso>
</modifyResponse>

```

次の例には、ユーザーが使用可能でない場合の前述の処理に対する失敗レスポンスが含まれています。

```

<modifyResponse status="failure" error="noSuchIdentifier">
<errorMessage>
exception=OIMSpmlException;errorMessage=NO_USER_ID_DEFINED
</errorMessage>
</modifyResponse>

```

次の例には、2つのグループ・メンバーシップ (Groups:7 と Groups:8) およびグループ管理者 (Groups:9) を追加する、グループ (Groups:36) とそのグループ参照の変更リクエスト処理が含まれています。ここでは、既存のグループ・メンバーシップ (Groups:10) および管理者参照 (Groups:11) の削除も行われます。

```

<modifyRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core">
<psoID ID="Groups:36"></psoID>
<modification>
<dsm1:modification name="Groups.Group Name" operation="add">
<dsm1:value>Changed</dsm1:value>
</dsm1:modification>
</modification>
<modification modificationMode="add">
<capabilityData
capabilityURI="urn:oasis:names:tc:SPML:2:0:reference" mustUnderstand="true">
<reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:7"/>
</reference>
<reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:8"/>
</reference>
<reference typeOfReference="administrator"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
<toPsoID ID="Groups:9"/>
</reference>
</capabilityData>
</modification>
<modification modificationMode="delete">
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="administrator"

```

```

xmlns="urn:oasis:names:tc:SPML:2:0:reference">
    <toPsoID ID="Groups:10"/>
  </reference>
  <reference typeOfReference="memberOf"
xmlns="urn:oasis:names:tc:SPML:2:0:reference">
    <toPsoID ID="Groups:11"/>
  </reference>
</capabilityData>
</modification>
</modifyRequest>

```

次の例には、前述の変更リクエスト処理に対する成功レスポンスが含まれています。

```

<modifyResponse status="success">
  <pso>
    <psoID ID="Groups:36"/>
    <data>
      <attr name="Groups.Key">
        <value>36</value>
      </attr>
      <attr name="Groups.Group Name">
        <value>Changed</value>
      </attr>
      <attr name="Groups.Updated By">
        <value>1</value>
      </attr>
      <attr name="Groups.Update Date">
        <value>2007-08-29 21:00:51.657</value>
      </attr>
      <attr name="Groups.Creation Date">
        <value>2007-08-27 15:22:43.97</value>
      </attr>
      <attr name="Groups.Created By">
        <value>1</value>
      </attr>
    </data>
    <capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
      <reference typeOfReference="memberOf">
        <toPsoID ID="Groups:7"/>
      </reference>
      <reference typeOfReference="memberOf">
        <toPsoID ID="Groups:8"/>
      </reference>
      <reference typeOfReference="administrator">
        <toPsoID ID="Groups:1"/>
      </reference>
      <reference typeOfReference="administrator">
        <toPsoID ID="Groups:9"/>
      </reference>
    </capabilityData>
  </pso>
</modifyResponse>

```

次の例には、グループ ID が存在しない場合の前述の処理に対する失敗レスポンスが含まれています。

```

<modifyResponse status="failure" error="noSuchIdentifier">
  <errorMessage>
    exception=OIMSpmlException;errorMessage=GROUP_NOT_FOUND
  </errorMessage>
</modifyResponse>

```

---

## 日付書式付き追加リクエスト

次の追加リクエストの例では、開始日および終了日属性を割り当てることでユーザーが追加されます。日付書式は、yyyy-mm-dd HH:MM:SS.sss 形式です。

```
<addRequest returnData="everything" xmlns="urn:oasis:names:tc:SPML:2:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
  <data>
    <dsml:attr name="objectclass">
      <dsml:value>Users</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.User ID">
      <dsml:value>John Doe</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.First Name">
      <dsml:value>John</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Last Name">
      <dsml:value>Doe</dsml:value>
    </dsml:attr>
    <dsml:attr name="Organizations.Organization Name">
      <dsml:value>Xellerate Users</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Xellerate Type">
      <dsml:value>End-User</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Manager Login">
      <dsml:value>Jane Doe</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Role">
      <dsml:value>Full-Time</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Password">
      <dsml:value>welcome</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.Start Date">
      <dsml:value>2007-06-18 00:00:00.000</dsml:value>
    </dsml:attr>
    <dsml:attr name="Users.End Date">
      <dsml:value>2017-06-18 00:00:00.000</dsml:value>
    </dsml:attr>
  </data>
</addRequest>
```

次の例には、前述の追加リクエスト処理に対する成功レスポンスが含まれています。

```
<addResponse status="success">
  <pso>
    <psoID ID="Users:8"/>
    <data>
      <attr name="Users.User ID">
        <value>John Doe</value>
      </attr>
      <attr name="Users.Key">
        <value>8</value>
      </attr>
      <attr name="Users.Last Name">
        <value>Doe</value>
      </attr>
      <attr name="Users.First Name">
        <value>John</value>
      </attr>
      <attr name="Users.Manager Key">
        <value>4</value>
      </attr>
    </data>
  </pso>
</addResponse>
```

```
<attr name="Users.Manager Login">
<value>John</value>
</attr>
<attr name="Users.Manager First Name">
<value>John</value>
</attr>
<attr name="Users.Manager Last Name">
<value>Doe</value>
</attr>
<attr name="Users.Xellerate Type">
<value>End-User</value>
</attr>
<attr name="Users.Creation Date">
<value>2007-08-29 21:27:03.39</value>
</attr>
<attr name="Users.Updated By">
<value>1</value>
</attr>
<attr name="Users.Update Date">
<value>2007-08-29 21:27:06.937</value>
</attr>
<attr name="Users.Status">
<value>Active</value>
</attr>
<attr name="Users.Disable User">
<value>0</value>
</attr>
<attr name="Users.Lock User">
<value>0</value>
</attr>
<attr name="Organizations.Key">
<value>1</value>
</attr>
<attr name="Users.Role">
<value>Full-Time</value>
</attr>
<attr name="Organizations.Organization Name">
<value>Xellerate Users</value>
</attr>
<attr name="Users.Start Date">
<value>2007-06-18 00:00:00.0</value>
</attr>
<attr name="Users.End Date">
<value>2017-06-18 00:00:00.0</value>
</attr>
<attr name="Users.Provisioning Date">
<value>2007-06-18 00:00:00.0</value>
</attr>
<attr name="Users.Provisioned Date">
<value>2007-08-29 21:27:03.39</value>
</attr>
<attr name="Users.Change Password At Next Logon">
<value>1</value>
</attr>
<attr name="objectclass">
<value>Users</value>
</attr>
</data>
<capabilityData capabilityURI="urn:oasis:names:tc:SPML:2:0:reference"
mustUnderstand="true">
<reference typeOfReference="memberOf">
<toPsoID ID="Groups:3"/>
</reference>
```

---

```
</capabilityData>  
</pso>  
</addResponse>
```

次の例には、ユーザーがすでに存在した場合の前述の処理に対する失敗レスポンスが含まれています。

```
<addResponse status="failure" error="alreadyExists">  
<errorMessage>  
exception=tcDuplicateUserException;errorMessage=Duplicate_User  
</errorMessage>  
</addResponse>
```



---

---

# 索引

## A

---

「Adapter Factory」 フォームのタブ, 1-4

## E

---

Execution Schedule, 1-4

## I

---

includeDataForCapability 要素, 12-5

## O

---

Oracle Identity Manager API タスクの作成, 3-16

『Oracle Identity Manager デザイン・コンソール・ガイド』, 1-3

Oracle Identity Manager のソリューション, 1-2

## R

---

Remote Manager のインストール, 2-2

Resources, 1-4

Responses, 1-5

## S

---

SPML リクエスト, B-1

## U

---

Usage Lookup, 1-5

## V

---

Variable List, 1-5

## あ

---

アダプタ, 1-2, 3-2

アダプタ・タスクの削除, 3-24

アダプタ・タスクの作成, 3-6

アダプタ・タスクの変更, 3-22

アダプタのエクスポートおよびインポート, 10-1

アダプタのコンパイル, 9-1

アダプタの無効化, 3-3

アダプタ変数, 3-3

アダプタ変数の変更, 3-5

## い

---

インストール後の構成, 11-2

## え

---

エンティティ・アダプタ, 3-26

エンティティの管理, 8-1

## し

---

事前移入アダプタ, 7-1

事前移入アダプタのアタッチ, 7-3

信頼関係の追加, 2-3

## す

---

ストアド・プロシージャ・タスクの作成, 3-12

## た

---

タスク割当てアダプタのプロセス・タスクからの削除,  
5-6

タスク割当てアダプタのプロセス・タスクへのアタッチ,  
5-3

## と

---

統合の問題, 1-2

## ふ

---

プロセス・タスク・アダプタの動作, 4-3

プロセス・タスク・アダプタのプロセス・タスクからの  
削除, 4-7

プロセス・タスク・アダプタのプロセス・タスクへのア  
タッチ, 4-4

## ゆ

---

ユーティリティ・タスクの作成, 3-15

## り

---

リモート・タスクの作成, 3-11

## る

---

- ルール・ジェネレータのアダプタ変数のマッピング, 6-3
- ルール・ジェネレータのスケジュール設定, 3-26
- ルール・ジェネレータのフォーム・フィールドからの削除, 6-6

## れ

---

- レスポンスの削除, 3-26
- レスポンスの作成, 3-25
- レスポンスの使用, 3-25
- レスポンスの変更, 3-26