

---

# インストール・ガイド

---

このガイドには、任意のプラットフォームでの Oracle Data Integrator、Oracle Data Profiling および Oracle Data Quality for Oracle Data Integrator のインストールおよびアップグレードの手順が記載されています。このガイドの対象読者は、Oracle Data Integrator をインストールまたはアップグレードする管理者です。

## ドキュメントの構成

このドキュメントの構成は、次のとおりです。

- **第1章「インストールの概要」**では、Oracle Data Integrator を設定するための一般的なガイドラインについて説明します。
- **第2章「インストールの前に」**では、システム要件と、Oracle Data Integrator を設定する前に実行する必要がある手順について説明します。
- **第3章から第6章**では、Oracle Data Integrator の各コンポーネントを設定およびアップグレードする方法について説明します。

## インストールの概要

**Oracle Data Integrator** は、パフォーマンスの向上とデータ統合コストの削減を異機種間システムでも実現する比類のない次世代の E-LT (Extract Load and Transform) テクノロジーを提供します。Oracle Data Integrator は、宣言的な設計方法による生産性の向上とともに、バッチとリアルタイムをシームレスに統合するアクティブ統合プラットフォームの利点を備えています。

Data Integrator パッケージには、**Oracle Data Profiling** および **Oracle Data Quality for Data Integrator** が含まれます。

- **Oracle Data Profiling** は、データを調査し品質を監視するツールです。これを使用すると、ビジネス・ユーザーは、メトリックを介してデータの品質を評価し、このデータに基づいてルールを検出または推測し、長期にわたってデータ品質の発展を監視できます。
- **Oracle Data Quality for Data Integrator** は、最も複雑なデータ品質のニーズにも対応する、評価の高い包括的なデータ品質プラットフォームです。強力なルールベース・エンジンと堅固かつスケーラブルなアーキテクチャによって、データ品質および名前と住所のクレンジングが企業のデータ統合戦略の中心となります。

**Oracle Data Profiling** および **Oracle Data Quality for Data Integrator** は、**Oracle Data Quality 製品**とも呼ばれています。

## コンポーネントの詳細

### Oracle Data Integrator

Oracle Data Integrator は、次のコンポーネントで構成されています。

- モジュール型のリポジトリは、1つのマスター・リポジトリと、1つまたは複数の作業リポジトリで構成されます。これらのリポジトリは、ANSI ISO 89 構文をサポートする任意のデータベース・エンジンにインストールできます。
- グラフィカル・インタフェース・モジュール（トポロジ・マネージャ、デザイナー、セキュリティ・マネージャ、オペレータ）と実行エージェント（エージェント）。これらのモジュールは、クライアント/サーバー・モードでリポジトリにアクセス可能な Java コンポーネントですべて構成されています。
- メタデータ・ナビゲータは、Web インタフェースを通じたりポジトリへのアクセスを可能にするサブレット/JSP アプリケーションです。
- 軽量デザイナーは、Web ブラウザを通じてリポジトリのオブジェクトを表示および編集するための Web アプリケーションです。
- Oracle Data Integrator の公開 Web サービスは、サービス指向アーキテクチャ（SOA）において Oracle Data Integrator の機能を最大限に活用できるようにする Web サービスです。

## Oracle Data Profiling

Oracle Data Profiling は、次のコンポーネントで構成されています。

- メタベース・サーバーには、プロファイリング・データとメタデータが含まれます。
- スケジューラ・サーバーは、プロファイリング・ジョブを処理します。
- Oracle Data Quality ユーザー・インタフェースは、Oracle Data Profiling と Oracle Data Quality に共通のクライアント・インタフェースです。
- Oracle Data Quality ODBC アダプタを使用して、ODBC データ・ソースを接続します。このオプション・コンポーネントは、Windows プラットフォームにのみインストールできます。

## Oracle Data Quality for Data Integrator

Oracle Data Quality for Data Integrator が使用するコンポーネントは、Oracle Data Profiling と同じです。これに加えて、次のコンポーネントがあります。

- Data Quality サーバーは、データ品質プロジェクトを実行するランタイム・サーバーです。

## インストール処理

### インストールの前に

以前のリリースの Oracle Data Integrator からアップグレードする場合、「Oracle Data Integrator のアップグレード」を参照してください。

Oracle Data Integrator をインストールする前に、次の点を考慮する必要があります。

- 現在のシステムがシステム要件に準拠していることを確認します。詳細は、「システム要件」および「Java の構成」を参照してください。

- Oracle Data Integrator を使用してデータ・サーバーにアクセスする場合、接続性に関する要件を考慮する必要があります。詳細は、「JDBC および JMS ドライバのインストール」を参照してください。
- Oracle Data Profiling または Oracle Data Quality for Data Integrator をインストールする場合は、「Oracle Data Quality 製品のインストール前の作業」を実行してください。

## インストール手順

「Oracle Data Integrator および Oracle Data Quality 製品のインストール」の手順に従って、Oracle Data Integrator、Oracle Data Profiling および Oracle Data Quality for Data Integrator ソフトウェア・コンポーネントをクライアントおよびサーバー・マシンにインストールします。

- Oracle Data Integrator を使用する場合、リポジトリを作成する必要があります。詳細は、「Oracle Data Integrator のリポジトリ」を参照してください。
- メタデータ・ナビゲータ、軽量デザイナーおよび公開 Web サービスを含むアプリケーション・サーバー・コンポーネントも構成できます。
- Oracle Data Profiling または Oracle Data Quality for Data Integrator を使用するには、メタベースおよびローダー接続を作成する必要があります。

## インストールの前に

### システム要件

#### Oracle Data Integrator の構成

Data Integrator に推奨される最小構成は、次のとおりです。

#### Oracle Data Integrator クライアント

Oracle Data Integrator のグラフィカル・モジュール（デザイナー、トポロジ・マネージャ、セキュリティ・マネージャ、オペレータ）を使用するには、使用中のコンピュータが次の最小システム要件を満たしている必要があります。

##### 最小システム要件:

- 256MB の RAM。
- ハード・ディスク・ドライブ上の 150MB の空きディスク領域。
- グラフィカル機能があり（Microsoft Windows、Linux、HP-UX、Solaris など）、Java 仮想マシン 1.5 をサポートするオペレーティング・システム。
- Java 仮想マシン 1.5。Oracle Data Integrator で Web サービスを使用する場合は、Java Development Kit (SDK) 1.5 が必要です。
- TCP/IP ネットワーク。

ただし、Oracle Data Integrator のすべての拡張機能を利用するために、次の構成を使用することを強くお勧めします。

### 推奨システム要件:

- 512MB の RAM
- ハード・ディスク・ドライブ上の 300MB の空きディスク領域
- グラフィカル機能があり (Microsoft Windows、Linux、HP-UX、Solaris など)、Java 仮想マシン 1.5 をサポートするオペレーティング・システム
- Java Development Kit (SDK) 1.5
- TCP/IP ネットワーク

### Oracle Data Integrator エージェント

エージェントを実行するには、使用中のコンピュータが次の最小システム要件を満たしている必要があります。

#### 最小システム要件:

- 256MB の RAM。
- ハード・ディスク・ドライブ上の 150MB の空きディスク領域。
- Java 仮想マシン 1.3.1 をサポートするオペレーティング・システム。
- Java 仮想マシン 1.3.1。Oracle Data Integrator で Web サービスを使用する場合は、Java Development Kit (SDK) 1.5 が必要です。
- TCP/IP ネットワーク。

ただし、エージェントのすべての拡張機能を利用するために、次の構成を使用することを強くお勧めします。

#### 推奨システム要件:

- 512MB の RAM
- ハード・ディスク・ドライブ上の 200MB の空きディスク領域
- Java 仮想マシン 1.5 をサポートするオペレーティング・システム
- Java Development Kit (SDK) 1.5
- TCP/IP ネットワーク

### Oracle Data Integrator リポジトリ

リポジトリは、ANSI ISO 89 構文をサポートする任意のリレーショナル・データベース・エンジン (Oracle、Microsoft SQL Server、Sybase AS Enterprise、IBM DB2 UDB、IBM DB2/400 など) にインストールできます。

また、評価目的で Oracle Data Integrator に付属のデモ・リポジトリを使用することもできます。このリポジトリは、Hypersonic SQL データベースに基づいており、特別なインストールを必要としません。

リポジトリの記憶域要件は、次のとおりです。

- **マスター・リポジトリ:** 30MB の記憶域。
- **作業リポジトリ:** 40MB の記憶域。ただし、必要となる記憶域は、使用するモジュールおよびプロジェクトのサイズや、リポジトリに格納する実行ログの容量に応じて変化する可能性があります。

## メタデータ・ナビゲータおよび軽量デザイナ

メタデータ・ナビゲータまたは軽量デザイナを実行するには、使用中のコンピュータが次の最小システム要件を満たしている必要があります。

### 最小システム要件:

- 256MB の RAM
- ハード・ディスク・ドライブ上の 50MB の空きディスク領域
- Java Development Kit (SDK) 1.5 をサポートするオペレーティング・システム
- J2EE 1.4 に準拠するアプリケーション・サーバー (OC4J 10.1.3 以上など)
- Java Development Kit (SDK) 1.5
- TCP/IP ネットワーク

ただし、すべての拡張機能を利用するために、次の構成を使用することを強くお勧めします。

### 推奨システム要件:

- 512MB の RAM
- ハード・ディスク・ドライブ上の 100MB の空きディスク領域
- Java Development Kit (SDK) 1.5 をサポートするオペレーティング・システム
- J2EE 1.4 に準拠するアプリケーション・サーバー (OC4J 10.1.3 以上など)
- Java Development Kit (SDK) 1.5
- TCP/IP ネットワーク

## Oracle Data Quality 製品の構成

Oracle Data Profiling および Oracle Data Quality for Data Integrator に推奨される最小構成は、次のとおりです。

### Oracle Data Quality クライアント

Oracle Data Quality を使用するには、使用中のコンピュータが次の最小システム要件を満たしている必要があります。

### 最小システム要件:

- 512MB の RAM
- ハード・ディスク・ドライブ上の 200MB の空きディスク領域
- オペレーティング・システム: Microsoft Windows 2000 Professional/Server (SP4) 、 Microsoft Windows XP Professional (SP2) 、 Microsoft Windows Server 2003 Standard Edition (SP1、32 ビット)
- Internet Explorer バージョン 5.5 以上
- 10Mbps の TCP/IP ネットワーク

ただし、Oracle Data Quality のすべての拡張機能を利用するために、次の構成を使用することを強くお勧めします。

**推奨システム要件:**

- 1GB の RAM
- ハード・ディスク・ドライブ上の 400MB の空きディスク領域

**Oracle Data Profiling および Data Quality サーバー**

Oracle Data Quality 製品を使用するには、使用中のコンピュータが次の最小システム要件を満たしている必要があります。

**最小システム要件:**

これは Oracle Data Profiling/Quality のユーザー数が 2 であり、夜間のデータ品質処理負荷の場合の一般的な構成です。

- 2GB の RAM。
- ディスク領域は、データ処理に応じて異なります。1.5GB に加えて、データ・ソースのサイズの 2~5 倍の領域を確保することをお勧めします。
- オペレーティング・システム:
  - UNIX:
    - AIX 5L (バージョン 5.1、5.2 および 5.3)
    - Solaris 8、9 または 10
    - HP-UX 11.i V1 (PA RISC) 、HP-UX 11.i V2 (IA64)
    - Linux Red Hat Enterprise Linux 2.1 以上
    - Linux Suse Version 9.2 Professional 以上
  - Inetd コンポーネント (Linux の場合は XInetd) を UNIX マシンにインストールする必要があります。
  - Windows:
    - Windows 2000 Server (SP3)
    - Windows Server 2003 Standard Edition (SP1、32 ビット)
    - 100MB の TCP/IP ネットワーク

ただし、Oracle Data Quality 製品のすべての拡張機能を利用するために、次の構成を使用することを強くお勧めします。

**推奨システム要件:**

これは Oracle Data Profiling/Quality のユーザー数が 5 であり、日中の適度なデータ品質処理負荷の場合の一般的な構成です。

- 4GB の RAM (パワー・サーバーの場合は 8GB: ユーザー数が 10 で日中の適度な負荷の場合)
- ディスク領域は、データ処理に応じて異なります。1.5GB に加えて、データ・ソースのサイズの 2~5 倍の領域を確保することをお勧めします。
- 1GB の TCP/IP ネットワーク

## Java の構成

Oracle Data Integrator には、Java 仮想マシン (JRE) または Software Development Kit (SDK) が必要です。Oracle Data Integration Suite のセットアップ・プログラムには、Java Runtime Environment が含まれます。独自の Java マシンを使用する場合は、インストールする必要があります。

インストール作業は使用中のオペレーティング・システムに応じて異なりますが、一般的な手順は次のとおりです。

1. Java 仮想マシンが現在のワークステーションにすでに存在するかどうかと、そのバージョンが Data Integrator でサポートされるかどうかを確認します。コマンドライン・プロンプトを起動し、`java -version` と入力します。
2. オペレーティング・システムのドキュメントを参照して、Java 環境をインストールします。

**注意:** Oracle Data Integrator の実行エージェントは、以前のバージョン (1.3.1 および 1.4) の Java マシンとともに動作します。Oracle Data Integrator のすべてのクライアント・インストールには、JVM 1.5.x を使用することをお勧めします。すべてのクライアントで同じ JVM を使用する必要があります。

**注意:** Oracle Data Integrator で Web サービスを使用する場合は、Java Development Kit (SDK) 1.5 が必要です。SDK はセットアップ・プログラムに含まれないため、個別にインストールする必要があります。Sun 社の Web サイトからダウンロードするか、OS ベンダーに連絡してください。

## JDBC および JMS ドライバのインストール

異なるデータ・サーバーに接続する場合、Oracle Data Integrator では、データベースについては JDBC (Java DataBase Connectivity) 標準が使用され、MOM (Message Oriented Middleware) については JNDI (Java Naming and Directory Interface) および JMS (Java Message Service) が使用されます。

Oracle Data Integrator には、サード・パーティのドライバ・ライセンスは含まれません。

Oracle Data Integrator では、各種のテクノロジーに対応するドライバを `/drivers` サブディレクトリにインストールする必要があります (それらのドライバが Java のクラスパスで参照されていない場合)。

### Oracle Data Integrator 用の新規ドライバを追加する手順:

Oracle Data Integrator 用の新規ドライバを追加するには、次の手順を実行します。

1. ドライバ・パッケージ (.zip または .jar ファイル) を、Oracle Data Integrator のインストール・フォルダにある `/drivers` サブディレクトリにコピーします。Oracle Data Integrator の次回起動時にドライバが自動検出され、各パッケージがクラスパスに追加されます。この作業は、Oracle Data Integrator のコンポーネントを実行するマシンごとに行う必要があります。
2. ドライバを使用してリポジトリに接続する際に、そのリポジトリへの接続にメタデータ・ナビゲータを使用する場合、ドライバのパッケージをメタデータ・ナビゲータのインストール・フォルダにある `/WEB-INF/lib` サブディレクトリにコピーし、アプリケーション・サーバーを再起動します。

個々の設定操作の詳細は、ドライバのドキュメントを参照してください。

Oracle Data Integrator に追加されたドライバは、Oracle Data Integrator の GUI のドライバ・リストや URL に自動的に表示されません。URL とドライバは、ドライバのドキュメントに記載された構文を使用して、適切なフィールドに手動で入力する必要があります。

**注意:** Windows 95 や 98 などの一部のシステムでは、ドライバの自動検出が動作しません。これらのシステムでは、odiparams ファイルの ODI\_ADDITIONAL\_CLASSPATH パス・パラメータにドライバのファイルを追加する必要があります。

**注意:** 使用可能なすべての JDBC ドライバのリストは、Sun 社の Web サイトにあります。

## Oracle Data Quality 製品のインストール前の作業

### 使用可能なポートの確認

インストール時に、Oracle Data Quality から、クライアントからサーバーにアクセス可能なブロックされないポート番号を 2 つ以上入力するよう求められます。これらのポート番号は、次の 2 つのサービスに使用されます。

- Oracle Data Quality スケジューラには、**スケジューラ・ポート**というポートが必要です。
- Oracle Data Profiling および Quality のメタベース・リポジトリには、**リポジトリ・ポート**というポートが必要です。

使用可能なポート番号を検索するには、サーバーで次の操作を行います。

- UNIX の場合、/etc/services にある services ファイルを確認します。
- Windows の場合、netstat -an コマンドを入力して、アプリケーションがリスニングしているポートを確認します。

使用可能なポートを 2 つ選択し、設定手順で使用するために書き留めておきます。

**注意:** ポート番号は、1000 より大きく 65535 以下である必要があります。また、簡単に参照できるように、連続した番号である必要があります。（たとえば、リポジトリ・ポートに 7600、スケジューラ・ポートに 7601。）

**注意:** Windows ベースのサーバーから ODBC データソースを使用する場合は、必要に応じて、Oracle Data Quality ODBC アダプタ用に 3 番目のポートを予約してください。

## インストール手順

### Oracle Data Integrator および Oracle Data Quality 製品のインストール

### Oracle Data Integrator および Oracle Data Quality 製品のインストール



**Oracle Data Integrator** インストール・プログラムには3つの製品が含まれ、インストールするソリューションに応じて4つのインストール・オプションから選択できます。

- 完全インストールには、**Oracle Data Integrator**、**Oracle Data Profiling** および **Oracle Data Quality for Oracle Data Integrator** が含まれます。
  - **Oracle Data Integrator** は、すべてのデータ統合要件を網羅する包括的なデータ統合プラットフォームです。
  - **Oracle Data Profiling** は、データを調査し品質を監視するツールです。
  - **Oracle Data Quality for Oracle Data Integrator** は、最も複雑なデータ品質のニーズにも対応する、評価の高い包括的なデータ品質プラットフォームです。

#### Oracle Data Integrator および Oracle Data Quality 製品をインストールする手順:

1. Oracle Data Integrator セットアップが含まれるディレクトリを開きます。
2. 次の操作を実行して、Oracle Universal Installer を起動します。

プラットフォーム	操作
UNIX および Linux	オペレーティング・システムのプロンプトで、次のコマンドを入力します。 <code>./runInstaller</code>
Windows	<code>setup.bat</code> をダブルクリックします。

「ようこそ」画面が表示されます。

3. 「次へ」をクリックします。  
「インストールする製品の選択」画面が表示されます。
4. インストールする製品を選択します。
5. 「次へ」をクリックします。  
「インストール・タイプの選択」画面が表示されます。
6. インストール・タイプを選択します。
  - **クライアント:** 手順6で指定した製品のクライアント・コンポーネントをインストールします。
  - **サーバー:** 手順6で指定した製品のサーバー・コンポーネントをインストールします。
  - **完全:** 手順6で指定した Oracle Data Integrator および Oracle Data Quality 製品のクライアントおよびサーバー・コンポーネントをインストールします。
7. 「次へ」をクリックします。
8. 「ホームの詳細」画面で、Oracle ホーム名および製品のインストール先ディレクトリのパスを選択します。
9. 「次へ」をクリックします。

**Oracle Data Profiling** または **Oracle Data Quality for Oracle Data Integrator** (あるいはその両方) をインストールする場合、メタベース・サーバーの構成およびクライアントの接続先メタベースの定義を求められます。**Oracle Data Profiling** または **Oracle Data Quality for Oracle Data Integrator** をインストールしない場合は、この手順を無視してください。

手順8で指定したインストール・タイプに応じて、次の画面が表示されます。

- a. クライアント・インストール・タイプを選択した場合、メタベースの接続画面が表示されます。この画面では、特定のメタベースおよびリポジトリへのクライアントの接続を構成します。メタベース接続を次のように指定します。

**ホスト名:** Oracle Data Profiling または Oracle Data Quality for Oracle Data Integrator のサーバー・コンポーネントをインストールするホストの名前。これらのコンポーネントを同じマシンにインストールする場合は、localhost と入力します。

**スケジューラ・ポート:** Oracle Data Quality スケジューラのポート。デフォルトは 7601。

**リポジトリ・ポート:** メタベース・リポジトリのポート。デフォルトは 7600。

**リポジトリ名:** リポジトリの名前。

- b. サーバー・インストール・タイプを選択した場合、メタベースの構成画面が表示されます。メタベースの構成を次のように定義します。

**リポジトリ・ポート:** メタベース・リポジトリのポート。デフォルトは 7600。

**スケジューラ・ポート:** Oracle Data Quality スケジューラのポート。デフォルトは 7601。

**管理者の名前:** メタベース管理者の名前。デフォルトは madmin。

**管理者パスワード:** メタベース管理者のパスワード。

10. 「次へ」をクリックします。

Oracle Data Quality ODBC アダプタの構成画面は、Oracle Data Profiling のサーバー・インストール・タイプまたは完全インストール・タイプを選択した場合のみ表示されます。その場合は、次のようにアダプタの構成を定義します。そうでない場合は、この手順をスキップします。

ODBC アダプタ・ポートを次のように設定します。

ODBC ポート: 7602

11. 「次へ」をクリックします。

「サマリー」画面が表示されます。

12. 「インストール」をクリックします。

インストールの進行状況画面が表示されます。

インストーラにより、各コンフィギュレーション・アシスタントが適切な順序で自動的に実行され、「ステータス」列に進捗状況が表示されます。この画面では、操作は必要ありません。

インストールが完了すると、「インストールの終了」画面に確認用の情報が表示されます。

13. 「終了」をクリックし、確認を求められたら終了の確認を行います。

14. UNIX プラットフォームでは、Oracle Data Integrator をインストールしたユーザーに対する次の環境変数を追加します。

- ODI\_JAVA\_HOME=<ODI\_HOME>/jre/1.4.2
- TS\_QUALITY=<ODI\_HOME>/oracledq/quality\_server/tsq11r0s/Software
- LD\_LIBRARY\_PATH=<ODI\_HOME>/oracledq/quality\_server/tsq11r0s/Software/bin

これでインストール手順は完了です。

**注意:** 特定のケース（セットアップ・プログラムでサポートされないプラットフォーム）では、セットアップ・プログラムを実行せずに Oracle Data Integrator をスタンドアロン・コンポーネントとしてインストールできます。詳細は、「Oracle Data Integrator の手動インストール」および「iSeries および AS/400 への Java エージェントのインストール」を参照してください。

## Oracle Data Integrator および Oracle Data Quality 製品のアンインストール

Oracle Data Integrator および Oracle Data Quality 製品をアンインストールする手順:

1. Oracle Data Integrator セットアップが含まれるディレクトリを開きます。
2. 次の操作を実行して、Oracle Universal Installer を起動します。

プラットフォーム	操作
UNIX および Linux	オペレーティング・システムのプロンプトで、次のコマンドを入力します。 ./runInstaller
Windows	setup.bat をダブルクリックします。

「ようこそ」画面が表示されます。

3. 「インストールされた製品」をクリックします。
4. 「インストールされた製品」ウィンドウで、削除する製品を選択します。
5. 「削除」ボタンを選択します。

### UNIX での Oracle Data Quality サービスの削除

一部の UNIX プラットフォームで、Oracle Data Quality サーバーを正常にアンインストールするには、様々なディレクトリや複数のファイルのエントリの削除が必要な場合があります。

1. メタベース・サーバーにログオンしているユーザーがないことを確認します。
2. UNIX サーバーにルートとしてログオンします。
3. スケジューラ・サーバーを停止します。
  1. 現在のディレクトリを<ODI\_HOME>/oracledq/metabase\_server/metabase/bin に変更します。
  2. コマンド ./scheduler -stop を入力します。
4. データ品質の情報を削除します。
  1. 現在のディレクトリを<ODI\_HOME>/oracledq に変更します。
  2. コマンド rm -R /metabase\_server を入力します。
  3. コマンド rm -R /metabase\_data を入力します。このコマンドにより、すべてのリポジトリおよびメタベースのデータが削除されます。
5. サービスを削除します。
  1. /etc/services ファイルのバックアップ・コピーを作成します。
  2. このファイルを編集用に開き、ファイルの末尾に移動します。

3. メタベース・サーバーおよびスケジューラに対して作成されたサービスに関する 2 行の記述があります。
4. この 2 行を `services` ファイルから削除します。
5. 変更内容を保存してファイルを閉じます。
6. メタベース・サーバーに対応する `inetd` 構成のエントリを削除します。
  1. `/etc/inetd.conf` ファイルのバックアップ・コピーを作成します。このファイルを編集用に開き、ファイル内のメタベース・サーバーに関するエントリを探して削除し、ファイルを保存します。
  2. Linux を使用している場合は、かわりに `/etc/xinetd.d/TSDiscovery` ファイルを削除する必要があります。
7. `inetd` を再起動します。

## メタデータ・ナビゲータのインストール

メタデータ・ナビゲータは、Web インタフェースを通じたリポジトリへのアクセスを可能にするサーブレット/JSP アプリケーションです。このアプリケーションには、Jakarta Tomcat や Oracle Containers for Java (OC4J) などのサーブレット/JSP コンテナを搭載した Web サーバーが必要です。

## メタデータ・ナビゲータのデプロイ

**注意:** メタデータ・ナビゲータをデプロイする場合、使用するアプリケーション・サーバーのドキュメントを参照して、Web アプリケーション・リソース (WAR) ファイルまたは JSP ページのセットから Web アプリケーションをデプロイする際に推奨される方法を確認してください。

### メタデータ・ナビゲータをインストールする手順:

1. Oracle Data Integrator の CD-ROM を CD-ROM ドライブに挿入します。
2. メタデータ・ナビゲータをインストールするには、次のいずれかの方法を使用できます。
  - CD の `/setup/manual/` ディレクトリに含まれる `oracledimn.war` ファイルに、アプリケーション・サーバーの Web アプリケーション・デプロイ機能を使用します。OC4J ではこの方法をお勧めします（「アプリケーション」タブの「配置」ボタンを使用）。
  - アプリケーション・サーバーで対応可能であれば、CD の `/oracledimn` ディレクトリをアプリケーション・サーバーの Web アプリケーション・ディレクトリ (Tomcat では `Webapps`) にコピーします。
3. アプリケーションをホストする Web サーバーを再起動します。

## JDBC ドライバのインストール

メタデータ・ナビゲータは、JDBC 接続を使用して、検索対象のリポジトリをホストするデータベース・サーバーにアクセスします。メタデータ・ナビゲータをインストールするマシンで、JDBC 接続を構成する必要があります。

JDBC ドライバの Java クラスを、メタデータ・ナビゲータのインストール・フォルダに含まれる /WEB-INF/lib サブディレクトリにコピーする必要があります。また、このクラスの拡張子は .jar である必要があります。必要に応じてファイル拡張子を .zip などから .jar に変更してください。詳細は、使用しているデータベースに付属の JDBC ドライバのインストール・ガイドを参照してください。

## メタデータ・ナビゲータ接続の構成

メタデータ・ナビゲータは、デフォルトの接続に基づいてインストールされます。Web アプリケーションで使用するリポジトリに対する接続を構成する必要があります。

**メタデータ・ナビゲータ接続を構成する手順:**

1. デザイナで、作業リポジトリへの接続を構成します。
2. Oracle Data Integrator のインストール・フォルダの /bin サブディレクトリに移動します。
3. このディレクトリの snps\_login\_work.xml ファイルを、メタデータ・ナビゲータのデプロイ・フォルダの /WEB\_INF サブディレクトリにコピーします。

## 軽量デザイナのインストール

軽量デザイナは、Web ブラウザを通じてリポジトリのオブジェクトを表示および編集するための Web アプリケーションです。このアプリケーションでは、JSF (JavaServer Faces) および AJAX (Asynchronous JavaScript and XML) テクノLOGYを使用します。このアプリケーションには、Jakarta Tomcat や Oracle Containers for Java (OC4J) などのサーブレット/JSP コンテナを搭載した Web サーバーが必要です。

## 軽量デザイナのデプロイ

**注意:** 軽量デザイナをデプロイする場合、使用するアプリケーション・サーバーのドキュメントを参照して、Web アプリケーション・リソース (WAR) ファイルまたは JSP ページのセットから Web アプリケーションをデプロイする際に推奨される方法を確認してください。

**軽量デザイナをインストールする手順:**

1. Oracle Data Integrator の CD-ROM を CD-ROM ドライブに挿入します。
2. 軽量デザイナをインストールするには、次のいずれかの方法を使用できます。
  - CD の /setup/manual/ ディレクトリに含まれる oracledilwd.war ファイルに、アプリケーション・サーバーの Web アプリケーション・デプロイ機能を使用します。OC4J ではこの方法をお勧めします (「アプリケーション」タブの「配置」ボタンを使用)。
  - アプリケーション・サーバーで対応可能であれば、CD の /oracledilwd ディレクトリをアプリケーション・サーバーの Web アプリケーション・ディレクトリ (Tomcat では Webapps) にコピーし、後述の構成を実行してから、アプリケーションをホストする Web サーバーを再起動します。

## リポジトリ接続の構成

### データソースの構成

軽量デザイナは、アプリケーション・サーバーで構成された JDBC データソースを通じてリポジトリに接続します。

### Tomcat でデータソースを構成する手順:

次のように、作業リポジトリごとにログインを構成する必要があります。

1. マスター・リポジトリと作業リポジトリをホストする RDBMS に接続するのに必要な JDBC ドライバ (.jar) ファイルを、アプリケーション・サーバーの適切なディレクトリにコピーします。Tomcat の場合、このディレクトリは /common/lib です。
2. マスター・リポジトリと作業リポジトリに対応するデータソースおよび接続情報を META-INF/context.xml ファイルに定義します。Tomcat のデプロイ機能を使用して軽量デザイナをインストールした場合は、かわりに

<TOMCAT\_HOME>/conf/Catalina/localhost/oracledilwd.xml ファイルでデータソースを構成する必要があります。後で必要になるため、リソース名（次の例では緑で表示）を書き留めておいてください。接続情報は、青で表示されています。

Oracle のマスター・リポジトリと作業リポジトリに対応する構成例は、次のとおりです。

```
<Context >
  <Resource
    name="jdbc/ORACLE_MASTER"
    type="javax.sql.DataSource"
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@SRV1:1521:ORA10"
    username="ODIM"
    password="ODIM"
    maxActive="8"
    maxIdle="2"
    maxWait="15000"
    poolPreparedStatements="true"
    removeAbandoned="true"
    initialSize="2"
    removeAbandonedTimeout="1800"
  />
  <Resource
    name="jdbc/ORACLE_WORK"
    type="javax.sql.DataSource"
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@SRV1:1521:ORA10"
    username="ODIW"
    password="ODIW"
    maxActive="8"
    maxIdle="2"
    maxWait="15000"
    poolPreparedStatements="true"
    removeAbandoned="true"
```

```

        initialSize="2"
        removeAbandonedTimeout="1800"
    </Context>

```

パラメータの詳細:

- **name:** データソース名
- **username** および **password:** データベースで使用するユーザー名とパスワード
- **driverClassName:** ドライバの JDBC クラス名
- **url:** JDBC URL
- **maxActive:** 認可された同時接続の最大数
- **maxIdle:** 接続プールでアイドル状態となる同時接続の最大数

詳細は、Apache Tomcat のマニュアルを参照してください。これらのパラメータは、サーバーのワークロードに合わせて調整する必要があります。

#### OC4J でデータソースを構成する手順:

OC4J では、使用するマスター・リポジトリと作業リポジトリごとに、次のようにデータソースを定義する必要があります。

1. マスター・リポジトリと作業リポジトリをホストする RDBMS に接続するのに必要な JDBC ドライバ (.jar) ファイルを、アプリケーション・サーバーの適切なディレクトリにコピーします。OC4J の場合、このディレクトリは ORACLE\_HOME/j2ee/home/applib です。
2. OC4J の管理インタフェースに接続します。
3. 「管理」タブで「サービス」→「JDBC リソース」を選択し、「タスクに移動」をクリックします。
4. 「接続プール」セクションの「作成」ボタンをクリックします。
5. 軽量デザイナ・アプリケーションを選択し、「新規接続プール」→「続行」を選択します。
6. JDBC 接続の各フィールドを入力し、「終了」をクリックします。
7. 「データソース」セクションの「作成」ボタンをクリックします。
8. 軽量アプリケーションを選択し、「マネージド・データソース」→「続行」を選択します。
9. JDBC データソースの各フィールドを入力し、「終了」をクリックします。

OC4J でデータソースを構成する方法の詳細は、アプリケーション・サーバーのドキュメントを参照してください。

#### リポジトリ接続の構成

この構成は、すべてのアプリケーション・サーバーに共通です。

1. Web アプリケーションのデプロイ・ディレクトリにある WEB-INF/web.xml ファイルを編集し、エントリを追加してアプリケーション・コンテキストにデータソースをリンクします。次に例を示します。  
前の手順で定義したデータソースを参照するには、次のようにします。

```

<resource-ref>
    <description>Oracle Datasource for the Master
    Repository</description>
    <res-ref-name>jdbc/ORACLE_MASTER</res-ref-name>

```

```

    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
  </resource-ref>
  <resource-ref>
    <description>Oracle Datasource for the Work
    Repository</description>
    <res-ref-name>jdbc/ORACLE_WORK</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
  </resource-ref>

```

2. これで、データソースが構成されました。次に、作業リポジトリに対する軽量デザイナーのログイン情報を定義する必要があります。repositories.xml ファイルを編集し、接続する作業リポジトリごとに次のようにログインを宣言します。

```

<login name="My Work Repository">
  <master name="jdbc/ORACLE_MASTER"
  masterDriver="oracle.jdbc.OracleDriver"/>
  <work name="jdbc/ORACLE_WORK" workName="WORKREP"/>
</login>

```

この構成項目では、作業リポジトリおよび関連するマスター・リポジトリのデータソース名を指定する必要があります。また、作業リポジトリ名と、マスター・リポジトリに接続するための JDBC ドライバも指定する必要があります。

3. 構成の変更を反映するため、アプリケーションを再起動します。

## Oracle Data Integrator の公開 Web サービスのインストール

Oracle Data Integrator の公開 Web サービスは、サービス指向アーキテクチャ (SOA) において Oracle Data Integrator の機能を最大限に活用できるようにする Web サービスです。公開 Web サービスにより、シナリオの起動などの操作が提供されます。

このコンポーネントは、任意の Web サービス・コンテナ (通常は Java アプリケーション・サーバー) にデプロイできます。コンポーネントの提供形式は、Axis2 アーカイブ・ファイルです。次の例では、アプリケーション・サーバーとして Apache Tomcat 5.5 を使用し、Web サービス・コンテナとして Apache Axis2 を使用します。これらの例は、他の Web サービス・コンテナを使用する場合には適切に調整する必要があります。

**注意:** Oracle Data Integrator の公開 Web サービスとデータ・サービスは、2 つの異なるタイプの Web サービスです。公開 Web サービスでは、Web サービスを介して Oracle Data Integrator の機能にアクセスすることが可能になります。データ・サービスは、Oracle Data Integrator によって生成されるもので、Web サービスを介して独自データにアクセスすることが可能になります。

**Axis2 に Oracle Data Integrator の公開 Web サービスをインストールする手順:**



1. Axis2 で、「**Administration**」 ページに移動します。
2. 「**Upload Service**」 リンクを選択します。
3. Oracle Data Integrator の Web サービスの .aar ファイルを検索します。このファイルは、Oracle Data Integrator のインストール・ディレクトリの /tools/web\_services/ サブディレクトリにあります。
4. 「**Upload**」 ボタンをクリックします。

Axis2 により、Oracle Data Integrator の Web サービスがアップロードされます。Axis2 のサービス・リストに、Data Integrator の公開 Web サービスが表示されます。

## リポジトリの作成

### Oracle Data Integrator のリポジトリ

リポジトリには、次の 2 つのタイプがあります。

- **マスター・リポジトリ:** 企業の IT リソースのトポロジ、セキュリティ、およびプロジェクトやデータ・モデルのバージョン管理に関する情報を保持するデータ構造。このリポジトリは、異なる Oracle Data Integrator モジュールからクライアント/サーバー・モードでアクセス可能なリレーショナル・データベースに格納されます。一般的には、ただ 1 つのマスター・リポジトリが必要です。
- **作業リポジトリ:** データ・モデル、プロジェクトおよびその使用に関する情報を保持するデータ構造。このリポジトリは、異なる Oracle Data Integrator モジュールからクライアント/サーバー・モードでアクセス可能なリレーショナル・データベースに格納されます。必要に応じて、複数のマスター・リポジトリに対して複数の作業リポジトリを指定できます。ただし、バージョン管理上の目的のため、1 つの作業リポジトリは 1 つのマスター・リポジトリにのみリンクできます。

リポジトリの作成手順の詳細は、次を参照してください。

- リポジトリの記憶域の作成
- マスター・リポジトリの作成
- マスター・リポジトリへの接続
- 作業リポジトリの作成
- 作業リポジトリへの接続

**注意:** 作業リポジトリのサイズ削減を目的とする実行ログのページと、リポジトリをホストするデータベースのバックアップをメンテナンス作業として定期的に行うことをお勧めします。

**注意:** Oracle Data Integrator には、スクリプト /bin/startdemo を通じて使用できる、テストおよびデモンストレーション用のリポジトリがデフォルトで付属しています。デモ環境の詳細は、スタート・ガイドを参照してください。

### リポジトリの記憶域の作成

リポジトリは、ANSI ISO 89 構文をサポートする任意のデータベース・エンジンにインストールできます。リポジトリを保持する各データベースでは、最初に記憶域を作成する必要があります。

**注意:** オラクル社のラボでは、リポジトリの記憶域として多くのデータベース・エンジン (Hypersonic SQL、IBM DB2 UDB、IBM DB2/400、Informix、Microsoft SQL Server、Oracle、Sybase AS Anywhere、Sybase AS Enterprise、Sybase ASIQ など) が動作保証済です。使用するデータベース・エンジンがこのリストに含まれていない場合、またはサポートされるエンジンの更新リストが必要な場合は、テクニカル・サポート・チームに連絡してください。

**注意:** メンテナンスおよびバックアップの際に有利であることから、リポジトリはアプリケーション・データの保存場所とは異なる領域に格納することを強くお勧めします (たとえば、Oracle の異なるスキーマに格納するか、Sybase と Microsoft SQL Server などの異なるデータベースに格納します)。

**重要:** Hypersonic SQL、IBM DB2 UDB、IBM DB2/400、Informix、Microsoft SQL Server、Oracle、Sybase AS Anywhere、Sybase AS Enterprise の各データベース・エンジンにインストールされたマスター・リポジトリでは、バージョン管理がサポートされます。

マスター・リポジトリは、作業リポジトリの 1 つとして同じスキーマに格納できます。一方で、同じスキーマに異なる 2 つの作業リポジトリを作成することはできません。

マスター・リポジトリには最低 30MB の領域を、作業リポジトリには最低 40MB の領域を用意する必要があります。次の場合、これらの値は増加する可能性があります。

- マスター・リポジトリでバージョン管理を使用する場合、推奨される領域量は 200MB です。
- 作業リポジトリで実行ログの履歴を保存する場合、推奨される領域量は 200MB です。

手順の例は、次のとおりです。

テクノロジー	手順
Oracle	<p>マスター・リポジトリをホストするスキーマ <b>snpm</b> と、作業リポジトリをホストするスキーマ <b>snpw</b> を作成します。</p> <p>これらのスキーマは、次の SQL コマンドで作成します。</p> <pre>SQL&gt; create user &lt;MY SCHEMA&gt; identified by &lt;MY PASS&gt; default tablespace &lt;MY TBS&gt; temporary tablespace &lt;MY TEMP&gt;; SQL&gt; grant connect, resource to &lt;MY_SCHEMA&gt;;</pre> <p>各変数の意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>&lt;MY_SCHEMA&gt;: 作成するスキーマの名前</li> <li>&lt;MY_PASS&gt;: 指定するパスワード</li> <li>&lt;MY_TBS&gt;: データの格納先となる Oracle 表領域</li> <li>&lt;MY_TEMP&gt;: デフォルト一時表領域</li> </ul> <p><b>注意: Oracle9i より前のリリースにリポジトリをインストールする場合</b></p> <p>リポジトリのインストールと一時オブジェクトの作成では、Oracle の VARCHAR 列の作成に CHAR セマンティック構文を使用します。この構文は、Oracle9i より前のリリースではサポートされません。前のリリースの Oracle にリポジトリをインストールするには、Oracle Data Integrator のインストール・フォルダにある /lib/scripts/xml/サブディレクトリの TECH_Oracle.xml ファイルを編集し、VARCHAR2 (%L CHAR) という文字列を VARCHAR2 (%L) に置き換えます。</p>

<p>Microsoft SQL Server または Sybase ASE</p>	<p>マスター・リポジトリをホストするデータベース <b>db_snpm</b> と、作業リポジトリをホストするデータベース <b>db_snpw</b> を作成します。これらのデータベースをデフォルトで保持する2つのログイン <b>snpm</b> および <b>snpw</b> を作成します。</p> <p>Enterprise Manager を使用して、2つのデータベース <b>db_snpm</b> および <b>db_snpw</b> を作成します（各データベースに対してデータ用に約40MB、ログ用に約20MBを確保します）。</p> <p>クエリ・アナライザまたは I-SQL を使用して、次のコマンドを実行します。</p> <pre>CREATE LOGIN &lt;mylogin&gt;     WITH PASSWORD = '&lt;mypass&gt;',     DEFAULT_DATABASE = &lt;defaultbase&gt;,     DEFAULT_LANGUAGE = us_english; USE &lt;defaultbase&gt;; CREATE USER dbo FOR LOGIN &lt;mylogin&gt;; GO</pre> <p>各変数の意味は次のとおりです。          &lt;mylogin&gt;: snpm または snpw          &lt;mypass&gt;: 各ログインのパスワード          &lt;defaultbase&gt;: それぞれ db_snpm および db_snpw に対応</p>
<p>DB2/400</p>	<p>マスター・リポジトリをホストするライブラリ <b>snpm</b> と、作業リポジトリをホストするライブラリ <b>snpw</b> を作成します。これらのライブラリをデフォルトで保持する2つのユーザー <b>snpm</b> および <b>snpw</b> を作成します。</p> <p><b>注意:</b> 各ライブラリは、SQL コレクションの形式で作成する必要があります。</p>

## マスター・リポジトリの作成

マスター・リポジトリを作成する作業は、表を作成する作業と、各種のテクノロジーの定義を自動的にインポートする作業で構成されます。

### マスター・リポジトリを作成する手順:

1. 「スタート」メニューで「プログラム」→「Oracle Data Integrator」→「Repository Management」→「Master Repository Creation」を選択するか、bin/recreate.bat または bin/recreate.sh を起動します。
2. 次のフィールドを入力します。
  - **ドライバ:** リポジトリをホストするテクノロジーにアクセスするためのドライバ。詳細は、「JDBC URL のサンプル」を参照してください。
  - **URL:** リポジトリをホストするデータ・サーバーの完全なパス。詳細は、「JDBC URL のサンプル」を参照してください。
  - **ユーザー:** 表の所有者のユーザーID またはログイン (**snpm** という名前で作成済)。
  - **パスワード:** このユーザーのパスワード。

- **ID:** デフォルトの 0 とは異なる新規リポジトリの特定の ID。これにより、リポジトリ間のインポートおよびエクスポートが影響を受けます。
- **テクノロジー:** リポジトリの基礎となるテクノロジーをリストから選択します。
- **言語:** マスター・リポジトリの言語を選択します。

3. 「OK」をクリックして設定を適用します。

ディクショナリの作成が開始されます。コンソールに進行状況が表示されます。マスター・リポジトリをテストするには、「マスター・リポジトリへの接続」を参照してください。

## マスター・リポジトリへの接続

マスター・リポジトリに接続する手順:

1. 「スタート」メニューで「プログラム」→「Oracle Data Integrator」→「Topology Manager」を選択するか、トポロジ・マネージャのスクリプト (bin/topology.bat または bin/topology.sh) を起動します。
2. 「新規」ボタン (「ログイン名」フィールドの右側にある最初のボタン) をクリックします。
3. 次のフィールドを入力します。

**Oracle Data Integrator 接続:**

- **ログイン名:** 一般的な別名 (Repository など)
- **ユーザー:** SUPERVISOR (大文字を使用)
- **パスワード:** SUNOPSIS (大文字を使用)

**DBMS 接続(マスター・リポジトリ):**

- **ユーザー:** マスター・リポジトリに対して作成した表の所有者のユーザーID またはログイン。
  - **パスワード:** このユーザーのパスワード。
  - **ドライバ・リスト:** 作成したマスター・リポジトリをサポートする DBMS に接続する際に必要なドライバを選択します。
  - **URL:** リポジトリをホストするデータ・サーバーの完全なパス。詳細は、「JDBC URL のサンプル」を参照してください。
4. 「テスト」をクリックして接続動作を確認します。
  5. 「OK」をクリックして設定を適用し、「OK」をクリックします。トポロジ・マネージャが起動します。

**重要:** SUPERVISOR アカウントのデフォルト・パスワードは、SUNOPSIS です。セキュリティ上問題となるため、このパスワードはできるだけ早く変更してください。

## 作業リポジトリの作成

必要に応じて、複数のマスター・リポジトリに対して複数の作業リポジトリを指定できます。ただし、バージョン管理上の目的のため、1つの作業リポジトリは1つのマスター・リポジトリのみリンクできます。

作業リポジトリの詳細は、リファレンス・マニュアルを参照してください。

**作業リポジトリを作成する手順:**

1. **トポロジ**・モジュールを通じてマスター・リポジトリに接続します。詳細は、「マスター・リポジトリへの接続」を参照してください。
2. アイコン・リストで「トポロジ」→「リポジトリ」→「作業リポジトリ」を選択して右クリックし、「作業リポジトリの挿入」を選択します。作業リポジトリの接続パラメータを指定するためのウィンドウが表示されます。
3. **接続**ウィンドウで、次のパラメータを指定します。
  - **名前:** 作業リポジトリの接続名を入力します。
  - **テクノロジー:** 作業リポジトリをホストするサーバーのテクノロジーを選択します。
  - **ユーザー:** 作業リポジトリに対して作成およびホストする表の所有者のユーザーIDまたはログイン。
  - **パスワード:** このユーザーのパスワード。
  - 「JDBC」タブ→「JDBC ドライバ」: 作業リポジトリをホストする DBMS に接続するためのドライバ。詳細は、「JDBC URL のサンプル」を参照してください。
  - 「JDBC」タブ→「URL JDBC」: 作業リポジトリをホストするデータ・サーバーの完全なパス。詳細は、「JDBC URL のサンプル」を参照してください。
4. 「テスト」をクリックします。

**注意:** 接続テストが適切に完了しなかった場合、「OK」でこのウィンドウを閉じないでください。

5. 「OK」をクリックして、作業リポジトリをホストするサーバーに接続するためのパラメータを適用します。リポジトリに一意の名前とユーザーID コード番号を指定するためのウィンドウが表示されます。
6. **作業リポジトリ**・ウィンドウで、次のパラメータを指定します。
  - **ID:** 1 から 998 までの一意の番号をリポジトリに指定します。
  - **名前:** 作業リポジトリに一意の名前を指定します (WorkRep1 など)。
  - **タイプ:** リストの「デザイナー」を選択します。
7. 「OK」をクリックして設定を適用します。作業リポジトリの作成が開始され、コンソールに様々な作業ステップが表示されます。
8. 作業リポジトリの作成が完了すると、作業リポジトリ・ウィンドウは閉じます。これで、**デザイナー**・モジュールおよび**オペレータ**・モジュールを通じてこのリポジトリにアクセスできます。詳細は、「作業リポジトリへの接続」を参照してください。

## 作業リポジトリへの接続

**作業リポジトリに接続してデザイナー・モジュールを起動する手順:**

1. 「スタート」メニューで「プログラム」→「Oracle Data Integrator」→「Designer」を選択するか、**デザイナー**のスク립ト (bin/designer.bat または bin/designer.sh) を起動します。
2. 「新規」ボタン (「ログイン名」フィールドの右側にある最初のボタン) をクリックします。
3. 次のフィールドを入力します。

**Oracle Data Integrator 接続:**

- ログイン名: 一般的な別名 (Repository など)
- ユーザー: SUPERVISOR (大文字を使用)
- パスワード: SUNOPSIS (大文字を使用)

**DBMS 接続(マスター・リポジトリ):**

- ユーザー: (作業リポジトリではなく) マスター・リポジトリに対して作成した表の所有者のユーザーID またはログイン。
- パスワード: このユーザーのパスワード。
- ドライバ・リスト: 作成した **マスター・リポジトリ** をホストする DBMS に接続する際に必要なドライバを選択します。
- URL: **マスター・リポジトリ** をホストするデータ・サーバーの完全なパス。詳細は、「JDBC URL のサンプル」を参照してください。

**作業リポジトリ:**

- **作業リポジトリ名:** 前の手順で作業リポジトリに指定した名前 (WorkRep1 など)。このフィールドの右側にあるボタンをクリックすると、現在のマスター・リポジトリで使用可能な作業リポジトリのリストを表示できます。
4. 「テスト」をクリックして接続動作を確認します。
  5. 「OK」をクリックします。デザイナー・モジュールが起動します。

**重要:** SUPERVISOR アカウントのデフォルト・パスワードは、SUNOPSIS です。セキュリティ上問題となるため、このパスワードはできるだけ早く変更してください。

## Oracle Data Quality Products の管理

### メタベースおよびリポジトリの管理

この章では、Oracle Data Quality 製品のメタベースを構成および管理する手順について説明します。

#### 概要

Oracle Data Quality 製品 (Oracle Data Profiling および Oracle Data Quality for Data Integrator) のリポジトリは、それぞれが一連のユーザー、データ接続、セキュリティおよびパフォーマンス設定を持つ 1 つ以上のメタベースの集まりです。

Oracle Data Profiling または Data Quality サーバーをインストールするたびに、1 つのリポジトリがインストールされます。したがって、管理者は、インストールされたリポジトリのメタベースを構成する必要があります。メタベースの構成には、メタベース・マネージャを使用します。

メタベース・マネージャは管理ツールです。これを使用して各リポジトリにアクセスして、メタベースの管理、Oracle Data Quality ユーザーの追加、ローダー接続の作成およびセキュリティの構成ができます。

1 つの Oracle Data Quality サーバー・アプリケーションのインストールにつき、許容されるリポジトリは 1 つのみです。Windows サーバーでは、1 サーバーにつき 1 つのリポジトリをインストールできます。UNIX サーバーでは、1 つのサーバーに複数のリポジトリをインストールできま

す。Oracle Data Quality ユーザー・インタフェース（クライアント）は、サイトに定義されたリポジトリにアクセスします。

## メタベース・マネージャの使用

メタベース・マネージャは、メタベース管理者によるリポジトリへの1つ以上のメタベースの追加およびデータ・ソースへの接続の設定に使用される管理インタフェースです。管理者は、メタベース・マネージャを使用して、メタベース内のデータにアクセスできるユーザーを管理できません。

メタベース・マネージャは、Windows の「スタート」メニューからアクセスできます。

**メタベース・マネージャにログオンする手順:**

1. Oracle Data Profiling または Quality クライアントで、Windows の「スタート」メニューをクリックし、「プログラム」→「Oracle」→「Oracle Data Profiling and Quality」→「Metabase Manager」を選択してメタベース・マネージャを開きます。メタベース・マネージャが表示されたら、リストから「リポジトリ」を選択します。
2. Oracle Data Profiling または Quality クライアントのインストール時に、1つのリポジトリが定義されます。
3. サーバーのインストール時に定義したメタベース管理者（madmin）の「ユーザー名」と「パスワード」を入力します。
4. 「OK」をクリックします。

## メタベース・ウィザードの使用

メタベース管理者は、リポジトリにメタベースを追加し、それらを構成して、ユーザーがメタベースにアクセスできるようにする必要があります。少なくとも、（データのインポートのため）メタベースを特定のデータ・ソースに接続し、利用するメタベースに対するアクセス権をユーザーに与える必要があります。また、権限のないユーザーがメタベース内のデータにアクセスできないようにセキュリティを設定する必要があります。

メタベースを初めて設定する場合、メタベース・マネージャのスタート・ウィザードを使用すると役に立つ場合があります。スタート・ウィザードでは、メタベースの構成、ユーザーおよびローダー接続の追加、設定の変更のために最初に実行する必要のある手順について説明します。

**スタート・ウィザードを開く手順:**

1. メタベース・マネージャのメイン・メニューから、「ツール」→「スタート・ウィザード」を選択します。ウィザードの使用手順が右ペインに表示されます。
2. テキストをスクロールし、手順に従って最初のメタベースを構成します。

## リポジトリの管理

この章の大部分では、Oracle Data Quality のリポジトリの管理方法について説明します。すべてのメタベース管理タスクは、メタベース・マネージャを使用して実行します。これに加えて、定期的にバックアップおよびセキュリティ・チェックを実行します。

## メタベースの管理

メタベース・マネージャを使用して、特定のリポジトリのメタベースを管理および構成できます。

**メタベースを追加する手順:**

1. メタベース・マネージャにログオンします。
2. **Control Admin** フォルダを展開します。
3. 「メタベース」を右クリックし、「メタベースの追加...」を選択します。
4. 次の情報を参照して各フィールドに入力します。

フィールド	説明
名前	わかりやすいメタベース名を指定します。
デフォルト・パターン	クライアント・インタフェースに表示されるプロファイリング・パターンを指定します。詳細は、Oracle Data Profiling および Data Quality for Data Integrator のヘルプでパターンの検査に関するトピックを参照してください。
パブリック・キャッシュ・サイズ (MB)	各メタベースが使用するサーバー・メモリーの容量を定義します。この値が大きいほど、ドリルダウンのパフォーマンスが向上します。サイト固有の要件に合わせて、この値を変更してください。 <b>警告:</b> すべてのメタベースのキャッシュの合計が使用可能なサーバー・メモリーを超過すると、パフォーマンスが急速に低下します。

5. 「OK」をクリックします。

**メタベースを編集する手順:**

この手順を実行するには、メタベース・マネージャにログオンする必要があります。

1. メタベース・マネージャにログオンします。
2. エクスプローラで、**Control Admin** というフォルダを展開します。
3. エクスプローラで、「メタベース」という項目をダブルクリックします。リポジトリ内に定義されたすべてのメタベースのリストがリスト・ビューに表示されます。
4. リスト・ビューで、メタベース名を右クリックし、「編集」を選択します。
5. 変更を適用するには、メタベースの作成についてのオプションを参照し、「OK」をクリックします。パブリック・キャッシュ・サイズの設定を変更する場合は、次の手順を実行して有効化する必要があります。この手順を実行しないと、設定の変更は有効になりません。
  - a. **重要:** 編集したメタベースにクライアントがアクセスしていないことを確認してください。
  - b. エクスプローラで、「メタベース」フォルダをダブルクリックします。
  - c. リスト・ビューで、「パブリック・キャッシュ・サイズ」の設定を変更したメタベースを右クリックし、「リカバリ...」を選択します。

**メタベースを削除する手順:**

この手順を実行するには、メタベース・マネージャにログオンする必要があります。

1. メタベース・マネージャにログオンします。
2. エクスプローラで、**Control Admin** というフォルダを展開します。
3. エクスプローラで、「メタベース」という項目をダブルクリックします。リポジトリ内に定義されたすべてのメタベースのリストがリスト・ビューに表示されます。



4. リスト・ビューで、メタベース名を右クリックし、「削除」を選択します。

## ユーザーの管理

リポジトリ内のメタベースにアクセスするには、少なくとも1つのユーザーが定義されている必要があります。メタベース管理者のみがユーザーを追加、編集および削除できます。

### ユーザーを追加する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** フォルダを展開します (左ペイン)。
3. 「ユーザー」を右クリックし、「ユーザーの追加...」を選択します。
4. 次の情報を参照して必須フィールドに入力します。

フィールド	説明
名前	Oracle Data Quality ユーザーのわかりやすい名前。
パスワードの失効	今すぐパスワードを期限切れにするには、このボックスを選択します。次回このユーザーが Oracle Data Quality にログオンするとき、新しいパスワードの指定を求められます。
パスワード	ユーザーのパスワードを指定します。 デフォルトでは、パスワード検証のチェックは行われません。すべての Oracle Data Quality ユーザーのパスワード検証をカスタマイズできます。
パスワードの再入力	パスワードを再入力します。

5. 「OK」をクリックします。

ユーザーのリストでユーザーをダブルクリックして、ユーザーを編集することもできます。ユーザーを削除するには、「削除」コンテキスト・メニューを使用します。

## メタベースのアクセス管理

ユーザーがログオンしてメタベースを開くには、事前にメタベース管理者から権限を付与されている必要があります。

このタスクを実行するには、メタベース管理者としてメタベース・マネージャにログオンする必要があります。

### ユーザーにメタベースへのアクセス権を付与する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** というフォルダを展開します。
3. エクスプローラで、「メタベース・ユーザー」を右クリックし、「ユーザーをメタベースに追加...」を選択します。
4. 次の情報を参照して必須フィールドに入力します。

フィールド	説明
ユーザー名	リストからユーザーを選択します。 ユーザーが表示されない場合は、ユーザーが作成されていることを確認してください。
メタベース	リストからメタベースを選択します。 メタベースが表示されない場合は、メタベースが作成されていることを確認してください。
制限付きユーザー	このユーザーを制限付きで指定するには、このボックスを選択します。 制限付きユーザーは、自分が作成したエンティティまたは他のユーザーから参照権を与えられたエンティティのみ参照できます。エンティティへのアクセス権が付与されていない場合、制限付きユーザーは、そのエンティティに関連するオブジェクト（結合、注釈、ブックマークなど）を表示できません。

5. 「OK」をクリックします。

## その他のリポジトリへの接続の追加

Oracle Data Quality ユーザー・インタフェースをインストールした後、別のリポジトリをエンタープライズに追加することもできます。この新しいリポジトリにアクセスするには、最初に Oracle Data Quality ユーザー・インタフェースからの接続を設定する必要があります。

### 新しいリポジトリへの接続を追加する手順:

1. Oracle Data Quality ユーザー・インタフェースのクライアント・マシンに Windows 管理者としてログオンします。
2. Windows の「コントロールパネル」から、「プログラムの追加と削除」を開きます。
3. 「Oracle Data Quality User Interface」 エントリを探します。
4. 「変更」をクリックします。
5. 「次へ」をクリックします。
6. 「変更」をクリックします。
7. 「次へ」をクリックします。
8. メタベース・リポジトリ・ウィンドウで、「詳細」をクリックし、新しいリポジトリへの接続の追加について次の情報を参照し、「次へ」をクリックします。

フィールド	説明
リポジトリ名	Oracle Data Quality サーバーに格納されたリポジトリのローカルな名前。この名前は、このクライアントのログイン画面に表示されます。リポジトリにアクセスするユーザーにとってわかりやすい任意の名前を選択します。
リポジトリ・ホスト名	リポジトリが格納されている実際のサーバー名。
リポジトリ・ポート	Oracle Data Quality サーバー・アプリケーションのインストール時に定

義したメタベースのログイン・ポート。

9. 「インストール」のプロンプトが表示されるまで「次へ」をクリックし、「インストール」をクリックします。
10. メンテナンスが完了したら「終了」をクリックします。

## ローダー接続の管理

### ローダー接続の管理

データをメタベースにインポートするには、プロファイルするデータ・ソースのタイプごとに、1つ以上のローダー接続が必要です。

ローダー接続を追加、編集および削除するには、メタベース・マネージャにメタベース管理者としてログオンする必要があります。

**重要:** ローダー接続には、リポジトリ内のすべてのメタベースがアクセスできます。したがって、データ・ソースのセキュリティが適切かどうかを確認し、不正なアクセスを確実に防止するように注意してください。

#### ローダー接続を追加する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** フォルダを展開します。
3. メタベースの「ローダー接続」を右クリックし、「ローダー接続の追加...」を選択します。または、メタベース・マネージャのメイン・メニューから、「ツール」→「ロード接続の追加」を選択します。
4. 「名前」および「説明」を指定します。この名前および説明は、クライアント・ユーザー・インタフェースのエンティティ作成ウィザードに再び表示されます。
5. 「タイプ」を選択します。選択した「タイプ」にかかわらず、「デフォルト・フィルタ」を指定できます。このフィルタは、クライアント・インタフェースのエンティティ作成ウィザードに表示されるファイル数を制限します。デフォルトは\*で、指定したディレクトリ内のすべてのファイルが表示されます。
6. 選択したタイプに応じて、タイプ固有のパラメータの入力を求められます。タイプ固有のパラメータを入力してください。

ユーザーのリストでローダー接続をダブルクリックして、ローダー接続を編集することもできます。ローダー接続を削除するには、「削除」コンテキスト・メニューを使用します。

#### 定義されたすべてのローダー接続を表示する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** フォルダを展開します。
3. 「ローダー接続」をダブルクリックします。このリポジトリ（およびこのリポジトリのすべてのメタベース）に定義されたすべてのローダー接続のリストがリスト・ビュー・ペインに表示されます。
4. 行の情報は次のとおりです。

フィールド	説明
名前	ローダー接続の名前。クライアント・アプリケーションのデータのインポート・ウィンドウに表示されます。
タイプ	ローダー接続のタイプ。有効なタイプは、COBOL、DB2、区切り、ODBC および Trillium です。
説明	ローダー接続の説明。クライアント・アプリケーションのデータのインポート・ウィンドウに表示されます。
パラメータ	ファイルの場所またはリレーショナル・ソース情報を指定するパラメータ。タイプごとに異なります。
フィルタ	エンティティ作成ウィザードで使用されるデフォルトのフィルタ。
編集者	接続を最後に編集したユーザー。
編集日	接続が最後に変更された日付。
作成者	接続を作成したユーザー。
作成日	接続が作成された日付。

## パフォーマンスの最適化のためのローダー設定の変更

メタベース・マネージャには、データ・インポートのパフォーマンスに影響を与える可能性がある設定が4つあります。

- **ローダーのキャッシュ・サイズ:** ロード処理に使用可能な最大メモリー
- **ロード時に依存性を実行:** 依存性分析の有効化
- **ロード時にキーを実行:** キー分析の有効化
- **ロード時に重複した行を検出:** 重複したキーの検出の有効化

これらの設定は、**Control Admin** の「ローダー接続」を右クリックした場合に**ローダー設定の編集**オプションで使用可能です。これらの設定については次のトピックで詳しく説明します。

### ローダーのキャッシュ・サイズの変更

ローダーのキャッシュ・サイズでは、ロード処理に許容される最大メモリーを指定します。可能な場合、キャッシュ・サイズの値は、仮想メモリーに依存せずに使用中のハードウェアで許容される最大サイズに構成します。

ローダーのキャッシュ・サイズを変更する手順:

**注意:** これらの設定を変更すると、リポジトリ全体に影響を与えます。

1. メタベース・マネージャにログオンします。
2. **Control Admin** を展開します。
3. 「ローダー接続」を右クリックし、「ローダー設定の編集...」を選択します。

4. 次の情報に基づいて変更します。

設定	説明
ローダーのキャッシュ・サイズ	<p>ロード処理に許容される最大メモリーを指定します。</p> <p><b>**この値は、仮想メモリーに依存せずにサイトのハードウェアで許容されるサイズに変更してください。</b></p> <p>複数の CPU を使用するシステムでは、このプロセスをパラレルで複数回実行できます。この場合、定義した制限を超過しないように、すべてのプロセス間でメモリーを共有します。</p>

5. 「OK」をクリックします。

### ロード時のキー・チェックおよび依存性チェックの無効化

デフォルトでは、Oracle Data Profiling により、データのインポート時にデータのサンプル (10,000 行) に対してキーおよび依存性の分析が自動的に実行され、キーおよび依存性が検出されます。データのインポート時にキーまたは依存性を検出する必要がない場合は、この設定を無効化できます。

キーまたは依存性（あるいはその両方）の分析を無効化すると、データのインポート時のロード・パフォーマンスが向上します。ただし、データのインポート処理を実行する際に依存性分析またはキー分析を実行しない場合は、Oracle Data Quality ユーザー・インタフェースを使用して分析を手動で実行する必要があります。詳細は、Oracle Data Quality ユーザー・インタフェースのヘルプを参照してください。

データのインポート時にキー分析および依存性分析を無効化する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** を展開します。
3. 「ローダー接続」を右クリックし、「ローダー設定の編集...」を選択します。
4. 次の情報を使用して変更します。

無効にする設定	説明
ロード時の依存性分析	<ol style="list-style-type: none"> <li>1. 「ロード時に依存性を実行」をクリックします。</li> <li>2. このボックスの選択を解除します。</li> </ol>
ロード時のキー分析	<ol style="list-style-type: none"> <li>1. 「ロード時にキーを実行」をクリックします。</li> <li>2. このボックスの選択を解除します。</li> </ol>

5. 「OK」をクリックします。

### ロード時の重複した行の設定

デフォルトでは、Oracle Data Profiling は、データをメタベースにインポートする際に重複した行を検出します。この設定を無効化すると、ロードのパフォーマンスを改善できます。

重複した行を検出する必要がない場合は、次の手順に従います。

ロード時の重複した行の検出を無効化する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** フォルダを展開します。
3. 「ローダー接続」を右クリックし、「ローダー設定の編集...」を選択します。
4. 次の情報に基づいて変更します。

無効にする設定	説明
ロード時の重複した行	<ol style="list-style-type: none"> <li>1. 「ロード時に重複した行を検出」タブをクリックします。</li> <li>2. このボックスの選択を解除します。</li> </ol>

5. 「OK」をクリックします。

#### 処理に制限を指定してロード時間を短縮する手順:

多数の重複した行が検出される可能性がある場合、最初のプロセスで検出される重複した行の数を制限できます。

1. 前述の手順 1~4 に従います。
2. 「重複した行の検出」が選択されていることを確認します。
3. 「可能な重複のしきい値」に数値を入力して、最初のプロセスで検出可能な重複した行数に対する制限を指定します。
4. 「OK」をクリックします。

## その他の管理タスク

### 電子メール通知の有効化

Oracle Data Quality for Data Integrator では、次のイベントの発生時に Oracle Data Quality ユーザーに電子メールで通知できます。

- ビジネス・ルールの失敗
- ビジネス・ルールの成功
- 行のエクスポートの完了
- 行のエクスポートの失敗
- ロード・ジョブの完了
- ロード・ジョブの失敗

メタベース・マネージャの電子メール設定を使用して、SMTP サーバーの認証を設定できます。SMTP サーバーの認証では、メールの送信または SMTP サーバーがリスニングするポートの変更を行う際にユーザー名とパスワードが必要です。

電子メール通知を有効化した後、Oracle Data Quality ユーザー・インタフェースを使用して電子メールの構成を設定できます。

#### 認証を使用した電子メール通知を有効化する手順:

1. メタベース・マネージャにログオンします。
2. メイン・メニューから、「ツール」→「電子メール設定の編集」を選択します。
3. 次の情報を入力します。

**送信者:** 電子メールの送信者として表示される名前。メール・サーバーの構成によっては、修飾された電子メール・アドレスの入力が必要な場合もあります。

**電子メール・サーバー:** 電子メールを送信する SMTP サーバーのホスト名。

**電子メール・ログイン:** サーバーにログオンする際に必要なユーザー名。

**電子メール・パスワード:** サーバーにログオンする際に必要なパスワード。

4. 「OK」をクリックします。

## パスワードのセキュリティ管理

これらのタスクを実行するには、メタベース管理者（madmin）としてログオンする必要があります。

すべてのユーザーのパスワードのプロパティを編集する手順:

1. メタベース・マネージャにログオンします。
2. **Control Admin** を展開します。
3. 「ユーザー」を右クリックし、「パスワードのプロパティの編集...」を選択します。
4. 次の情報を参照して変更し、「OK」をクリックします。

フィールド	説明
ユーザー・パスワードの有効期間	新しいパスワードの有効期間を指定します。この期間の終わりに、ユーザーは新しいパスワードの入力を求められ、有効期間が再開します。
古いパスワードの再利用の禁止回数:	古いパスワードを設定できる回数。
有効なパスワードの正規表現:	<p><b>例 1</b> 6 文字以上のパスワードのみを許可する。 入力する正規表現: <code>.{6}</code></p> <p><b>例 2</b> 次の条件に合致するパスワードのみを許可する。</p> <ul style="list-style-type: none"> <li>• 長さが 5~16 文字</li> <li>• 英数字のみ（アンダースコア_文字を含む）</li> </ul> <p>入力する正規表現: <code>^[a-zA-Z0-9]{5,16}\$</code></p> <p><b>説明:</b> メタ文字^（開始）および\$（終了）を使用して、文字列全体を一致させます。次に、メタ文字\wを使用して、英数字（[a-zA-Z0-9]）の一致を指定します。さらに、5 文字以上および 16 文字以下の制約{5,16}を追加します。</p> <p><b>例 3</b> 次の条件に合致するパスワードのみを許可する。</p>

無効なパスワードの正規表現:	<ul style="list-style-type: none"> <li>• 長さが 10 文字以上</li> <li>• 少なくとも 1 つの小文字、1 つの大文字、1 つの数字および 1 つの特殊文字を含む</li> <li>• (構成可能な) 特殊文字 (@#\$\$%^&amp;+=) の使用を許可</li> </ul> <pre>{^.*(?=.{10,})(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@#\$\$%^&amp;+=]).*\$}</pre>
	<p><b>例 1</b></p> <p>正規表現 {9} を入力すると、最大長が 8 文字であるパスワードのみが有効になります。</p> <p><b>例 2</b></p> <p>正規表現 {^[a-zA-Z]*\$} を入力すると、アルファベット文字 (小文字と大文字) のみで構成されたパスワードは無効になります。これは事実上、パスワードに非アルファベット文字を含める必要があることを意味します。</p>

## Oracle Data Integrator のアップグレード

次の手順では、以前のリリースの Oracle Data Integrator からアップグレードするプロセスについて説明します。手順全体をよく確認してからアップグレードを開始してください。

### 重要事項

<install\_dir>は、この手順全体を通じて Oracle Data Integrator の既存のインストール・ディレクトリを表します。<ODI\_HOME>は、新規バージョンのインストール・ディレクトリを表します。

アップグレードを実行する前に、次のことを確認してください。

- 次の手順では、Oracle Data Integrator という用語は Sunopsis 製品と同義です。 **Oracle Data Integrator リリース 10.3** は、Sunopsis v4.1 の次のリリースに相当します。
- アップグレードは、Sunopsis リリース 3.0.00.00 以上、および Oracle Data Integrator リリース 10.1.3 以上でサポートされます。これより前のリリースからのアップグレードを希望する場合は、テクニカル・サポートに連絡してください。
- マスター・リポジトリと作業リポジトリのアップグレードでは、リポジトリの構造が変更されます。この一連の変更作業は、連続した 1 つの手順として実行される必要があるため、元に戻すことはできません。 リポジトリのアップグレード中にエラーが発生した場合は、リポジトリのアップグレード・プロセスを再開する前に、リポジトリ・データベースのバックアップをリストアする必要があります。
- Oracle Data Integrator には、Java 仮想マシン (JVM) 1.5 が必要です。Oracle Data Integrator で Web サービスを使用する場合は、Java Development Kit (SDK) 1.5 が必要です。JVM 1.5 は、セットアップ・プログラムに付属しています。
- メタデータ・ナビゲータのアップグレード作業は、この手順に含まれません。アプリケーション・サーバーの管理者は、メタデータ・ナビゲータをアンデプロイしてから再インストールする必要があります。



## 手順 1: アップグレードの準備

### 1.1: エージェント・サービスの停止

エージェントを Windows サービスとして起動している場合、アップグレードの前にそのサービスを停止してアンインストールし、後で再インストールする必要があります。元のサービス構成は、バックアップ・ディレクトリに格納されます。

### 1.2: バックアップ

アップグレード・プロセスを開始する前に、Oracle Data Integrator のインストール・ディレクトリおよびリポジトリのバックアップ・コピーを必ず実行してください。

1. Oracle Data Integrator のインストール・ディレクトリのバックアップ・コピーを作成します（このディレクトリは、以後、バックアップ・ディレクトリと表記します）。セットアップ・プログラムを使用する場合、自動バックアップが実行されるため、このバックアップ作業はオプションです。
2. リポジトリの格納先であるデータベース、ライブラリまたはスキーマのバックアップを作成します。

### 1.3: アップグレード・プロファイルの準備

#### カスタマイズ済のプロファイル

プロファイルは、アップグレード・プロセスにより自動的に再初期化されます。デフォルト・プロファイルを変更している場合、アップグレード作業を継続する前に、変更済のプロファイルのコピーを作成する必要があります。アップグレード・プロセスの最後に、それらのコピーを参照して、アップグレードでリセットされたプロファイルに変更内容を再適用する必要があります。

独自のプロファイルを設計している場合、アップグレード・プロセスの最後に、そのプロセスで追加または削除されたオブジェクトおよびメソッドを反映するよう、プロファイルを更新する必要があります。変更されたオブジェクトのリストは、次を参照してください。

#### 変更されたオブジェクト

Sunopsis v4.0 で変更されたオブジェクト（機能別）は、次のとおりです。

- マーカー: マーカー、マーカー・グループ。ライフ・サイクル状態は非推奨です。
- バージョニング: ソリューション、ソリューション要素、バージョン。バージョニング・メソッドは各オブジェクトに付属します。

Sunopsis v4.1 で追加されたオブジェクト（機能別）は、次のとおりです。

- 共通フォーマット・デザイナー: アクション・グループ、アクション、アクション行、モデル・フォルダ、ダイアグラム

Data Integrator リリース 10.3.1 で追加または変更されたオブジェクトは、次のとおりです。

- 変更されたオブジェクト・タイプ（新規メソッド）: データストア、モデル・フォルダ、モデル、変数、オブジェクト、アクション・グループ、アクション、アクション行、ステップ・レポート、バージョン、プロシージャ、インタフェース、シナリオ、オブジェクト

- 新規オブジェクト・タイプ: シナリオ・フォルダ、Open Tools (このオブジェクトは、Open Tools の追加および削除権限を処理します。)

### 3.2.03.00 より前のリリースの Sunopsis に対応するプロファイルのアップグレード

3.2.03.00 より前のリリースの Sunopsis からアップグレードする場合、次の「内部 ID」の値でデフォルト・プロファイルがインストールされていることを確認します (この値はプロファイル・ウィンドウの「バージョン」タブに表示されます)。

CONNECT: 24999  
DESIGNER: 25999  
NG DESIGNER: 28999  
METADATA ADMIN: 26999  
NG METADATA ADMIN: 29999  
TOPOLOGY ADMIN: 34999  
SECURITY ADMIN: 33999  
OPERATOR: 31999  
REPOSITORY EXPLORER: 32999  
NG REPOSITORY EXPLORER: 30999

プロファイルにすでにこれらの内部 ID 値が割り当てられている場合、アップグレードを続行できます。アップグレードの最後に、新規オブジェクトおよびメソッドを反映するよう自動的にプロファイルに適用された変更が、セキュリティ・ポリシーに準拠していることを確認します。

1 つ以上のプロファイルが前述のリストに一致しない ID を保持している場合:

1. セキュリティ・マネージャを使用して既存のユーザーおよびプロファイルをすべて削除します。
2. アップグレードを実行します。
3. アップグレード・プロセスの最後に、アップグレード・プロセスでインポートされたプロファイルを使用してセキュリティ・ポリシーを再定義します。

## 手順 2: Oracle Data Integrator モジュールのアップグレード

Oracle Data Integrator モジュールのアップグレードは、Oracle Data Integrator モジュール (デザイナー、オペレータ、トポロジ・マネージャ、セキュリティ・マネージャ、エージェント) がインストールされている各マシンで実行する必要があります。このアップグレードは、セットアップ・プログラムを使用して実行するか (推奨)、または手動で実行できます。

### インストール上の注意事項

インストール・プログラムは、古いインストール・ディレクトリとは別のディレクトリに Oracle Data Integrator 10.1.3 がインストールされることを前提としています。したがって、次の点に留意してください。

- 接続の設定 (/bin/snp\_login\_work.xml ファイルおよび /bin/snp\_login\_security.xml ファイルに格納) は、リカバリされません。

- /bin/odiparams.bat (または/bin/snpparams.bat) に格納された構成パラメータは、リカバリされません。
- デモ・リポジトリおよびファイル (/demo/)、古いドライバ・ファイル (/bin/drivers) は、リカバリされません。

この構成情報を新しいインストール・フォルダに対して実施するための手順を手動で実行する必要があります。

## 2.1: セットアップ・プログラムを使用したアップグレード

1. <install\_dir>のバックアップを作成してあることを確認します。
2. プラットフォームに適したセットアップ・プログラムを実行します。詳細は、このガイドの「Oracle Data Integrator のインストール」を参照してください。セットアップ・プログラムの実行中にインストール・フォルダの入力を求められたら、<install\_dir>フォルダとは異なるフォルダを選択します。この手順では、この新しいインストール・フォルダを<ODI\_HOME>と呼びます。セットアップ・プログラムの指示に従ってください。
3. 古いバージョンの Oracle Data Integrator をアンインストールします。

## 2.2: 手動によるアップグレード

1. <install\_dir>のバックアップを作成してあることを確認します。このバックアップは後で必要になります。
2. Oracle Data Integrator の CD からインストールする場合は、CD の/oracledi ディレクトリの内容を<ODI\_HOME>ディレクトリにコピーします。ダウンロードしたアーカイブ・ファイルからインストールする場合、Oracle Data Integrator のアーカイブ・ファイルを解凍し、/oracledi ディレクトリの内容を<ODI\_HOME>ディレクトリにコピーします。

## 2.3: 構成のリストア

手動またはセットアップ・プログラムを使用してインストールした場合は、次の手順を実行する必要があります。

1. 手順 1 でバックアップ・ディレクトリに保存しておいたドライバなどのすべての追加ファイルを、インストール・ディレクトリに手動でコピーします。
2. テキスト・エディタを使用して、<ODI\_HOME>/oracledi/bin フォルダの新規 odiparams ファイルのパラメータをバックアップ・ファイルからリストアします。Sunopsis からアップグレードする場合、odiparams は snpparams ファイルの新しい名前です。
3. 次のファイルをバックアップ・ディレクトリから<ODI\_HOME>/oracledi/bin ディレクトリにコピーします。
  - 拡張子.xml を持つすべてのファイル
  - 拡張子.layout を持つすべてのファイル

## 手順 3: マスター・リポジトリのアップグレード

**注意:** Oracle9i より前のリリースのリポジトリのアップグレード  
リポジトリのアップグレードでは、Oracle の VARCHAR 列の作成に CHAR セマンティック

構文を使用します。この構文は、Oracle9i より前のリリースではサポートされません。前のリリースの Oracle にリポジトリをインストールするには、/lib/scripts/xml/ディレクトリの TECH\_Oracle.xml ファイルを編集し、VARCHAR2(%L CHAR) という文字列を VARCHAR2 (%L) に置き換えます。

使用しているマスター・リポジトリごとに次の手順を 1 回実行する必要があります。

#### マスター・リポジトリをアップグレードする手順:

1. Windows の場合: 「スタート」メニューで「プログラム」→「Oracle Data Integrator」→「Repository Management」→「Master Repository Upgrade」を選択するか、<ODI\_HOME>/oracledi/bin/mupgrade.bat を実行します。

UNIX の場合: <ODI\_HOME>/oracledi/bin/mupgrade.sh を実行します。

マスター・リポジトリ・アップグレード・ツールが起動します。

2. マスター・リポジトリ接続に対応する「ログイン名」を選択するか、次の各フィールドを入力します。
  - **ドライバ:** リポジトリをホストするテクノロジーにアクセスするためのドライバ。
  - **URL:** リポジトリをホストするデータ・サーバーの完全なパス。
  - **ユーザー:** 表を所有するユーザーの ID またはログイン。
  - **パスワード:** このユーザーのパスワード。
  - **テクノロジー:** リポジトリの基礎となるテクノロジーをリストから選択します。
3. 「OK」をクリックします。

リポジトリのアップグレード・プロセスが開始されます。コンソールに進行状況が表示されます。マスター・リポジトリのアップグレードが完了すると、メッセージが表示されます。

## 手順 4: 作業リポジトリのアップグレード

作業リポジトリごとに次の手順を 1 回実行する必要があります。

#### 作業リポジトリをアップグレードする手順:

1. トポロジ・マネージャを通じてマスター・リポジトリに接続します。
2. 「リポジトリ」ツリー・ビューで「作業リポジトリ」を選択し、アップグレードするリポジトリを右クリックして「アップグレード」を選択します。
3. Oracle Data Integrator により、選択した作業リポジトリがアップグレードされます。作業リポジトリのアップグレードが完了すると、ウィンドウが表示されます。

各作業リポジトリでこれらの手順を繰り返します。

## 手順 5: 新規オブジェクトのインポート

### 5.1: オブジェクト、メソッドおよびプロファイル

今回のリリースの Oracle Data Integrator では、新規のオブジェクトおよびメソッドと、それらのオブジェクトおよびメソッドに対する新規権限が導入されています。オブジェクトとメソッドは、自動的に追加されます。Oracle Data Integrator に組み込まれているデフォルト・プロファイルは、

すべての新規オブジェクトおよびメソッドに対する適切な権限を含むようにアップグレード・プロセスによって自動的に更新されます。

## 5.2: テクノロジ

今回のリリースの Oracle Data Integrator には、新規テクノロジ定義が含まれます。これらの新規テクノロジをインポートする必要があります。

特定のテクノロジで共通フォーマット・デザイナー機能を使用する場合、そのテクノロジの定義を再インポートする必要があります。テクノロジを再インポートすると、そのテクノロジに定義されたアクション・グループが追加され、DDL 生成に必要な個別問合せが設定されます。

また、リリース 3.2.03.01 以下からアップグレードするユーザーは、接続リカバリを有効化する現行日付問合せを利用するために、すべてのテクノロジを更新することをお勧めします。

### 新規テクノロジ

Sunopsis v3.1 に新規導入されたテクノロジ:

- Oracle Data Integrator Engine
- XML

Sunopsis v3.2 に新規導入されたテクノロジ:

- Generic SQL
- Java Beanshell
- JavaScript
- Jython
- NetRexx
- PostGreSQL
- SAP Java Connector

Sunopsis v4.0 に新規導入されたテクノロジ:

- Teradata
- Netezza
- Hyperion Essbase

Sunopsis v4.1 に新規導入されたテクノロジ:

- なし。共通フォーマット・デザイナーをサポートするようすべてのテクノロジを更新。

Oracle Data Integrator リリース 10.1.3 に新規導入されたテクノロジ:

- Axis2
- Derby
- SAS
- Salesforce
- Oracle BAM
- 共通フォーマット・デザイナーで使用できるようすべてのテクノロジを更新

## テクノロジーのインポート

**警告:** テクノロジーをインポートすると、そのテクノロジーに加えたすべての変更が削除されます。変更したテクノロジーをコピーし、インポートしたテクノロジーにその変更を再適用することをお勧めします。

1. トポロジ・マネージャを実行します。「物理アーキテクチャ」ビューを選択します。
2. 「テクノロジー」ノードを開きます。
3. 「テクノロジー」ノードを右クリックし、「テクノロジーのインポート」を選択します。
4. ../impexp/ディレクトリで、現在存在しないテクノロジーまたは更新されたテクノロジーを選択します。
5. 「シノニム Insert-Update」インポート・モードを選択します。
6. 「OK」をクリックして新規テクノロジーをインポートします。

## 5.3: 言語

今回のリリースの Oracle Data Integrator には、新規言語が含まれます。これらの言語は、自動的に更新されます。

**警告:** 言語をインポートすると、その言語に加えたすべての変更が削除されます。変更した言語をコピーし、更新した言語にその変更を再適用することをお勧めします。

### 新規言語

更新された言語ファイルは、<ODI\_HOME>/oracledi/scripts/xml/ディレクトリにあります。

Sunopsis v3.1 に新規導入された言語:

- SQL
- SQL\_FILE

Sunopsis v3.2 に新規導入された言語:

- JYTHON
- SAP

Sunopsis v4.0 および 4.1 に新規導入された言語:

- SQL (4.1 ではマイナー・アップデート)

Oracle Data Integrator リリース 10.1.3 に新規導入された言語:

- なし

## 5.4: 汎用アクション

今回のリリースの Oracle Data Integrator には、共通フォーマット・デザイナー用の DDL スクリプトの生成を可能にするアクションが含まれます。これらのアクションは、マスター・リポジトリにインポートする必要があります。

1. トポロジ・マネージャを実行します。「一般アクション」ビューを選択します。
2. 右クリックし、「アクションのインポート」を選択します。

3. <ODI\_HOME>/oracledi/scripts/xml/ディレクトリですべてのアクションを選択します。
4. 「シノニム Insert-Update」 インポート・モードを選択します。
5. 「OK」 をクリックしてアクションをインポートします。

## 5.5: ナレッジ・モジュール

今回のリリースには、既存のナレッジ・モジュールの新規バージョンと、新規のナレッジ・モジュールが含まれます。

既存のプロジェクトのナレッジ・モジュールを更新する必要はありませんが、新規プロジェクトでは新規ナレッジ・モジュールを使用することをお勧めします。以前のナレッジ・モジュールとともにコンパイルされた既存のインタフェース、パッケージおよびシナリオは、アップグレード後も再生成せずに通常どおり機能します。ただし、**ISO SQL 増分更新 IKM** はこの原則の例外となります（後述の説明を参照）。

ナレッジ・モジュールをインポートまたは更新する前に、次のことを確認してください。

- 新規ナレッジ・モジュールは、「複製」モードでプロジェクトにインポートする必要があります。
- 新規テクノロジーのナレッジ・モジュールは、関連するテクノロジーのインポート後にインポートする必要があります。
- プロジェクト内で使用している既存のナレッジ・モジュールを更新する場合、置換するナレッジ・モジュールのコンテキスト・メニューにある「置換のインポート」オプションを使用して更新する必要があります。このオプションでは、選択したナレッジ・モジュールを使用しているすべてのインタフェースが自動的に更新されます。ナレッジ・モジュールの置換後に、関連するインタフェースを詳細にチェックおよびテストする必要があります。
- Oracle Data Integrator に含まれるデフォルトのナレッジ・モジュールを必要に応じてカスタマイズしている場合、元のナレッジ・モジュールに最新の変更を手動で実装する必要があります。これらの変更の詳細は、後述の説明を参照してください。

### 重要な例外: IKM ISO SQL 増分更新

v3.2.03.15 より前のリリースの Sunopsis からアップグレードする場合、このナレッジ・モジュールを更新する必要があります。このナレッジ・モジュールを使用するプロジェクトは、前述のとおり、今回のリリースで提供される「IKM SQL Incremental Update」を使用するよう更新する必要があります。その後、このナレッジ・モジュールを使用しているシナリオを再生成します。

### 変更の詳細

このナレッジ・モジュールでは、Update Existing Rows コマンドに次の変更が加えられています。変更前の列のメタデータは、次のとおりです。

```
$$CSV_COL_LST <%=odiRef.getColList("", "[COL_NAME]", "", "", "",
"(UPD)")%> $$CSV_COL_LST_END
```

変更後の列のメタデータは、次のとおりです。

```
$$CSV_COL_LST <%=odiRef.getColList("", "[COL_NAME]", "", "", "", "(UPD
AND (NOT UK) AND (NOT TRG))"%><%=odiRef.getColList("", "[COL_NAME]", "",
"", "", "(UK)")%> $$CSV_COL_LST_END
```

この変更は、Sunopsis v3.2.03.10 で実装された新規の\$\$CSV\_COL\_LST メタデータ構文に準拠するために追加されました。この古い構文を使用しているカスタマイズ済のナレッジ・モジュールも、すべて更新する必要があります。

## 手順 6: アップグレードの完了

手順 1.1 で停止したエージェント・サービスを再インストールします。

これで、以前のリリースの Oracle Data Integrator で開発したインタフェースやパッケージなどを実行できます。

これで、Oracle Data Integrator のリリースは最新です。

## 付録

### Oracle Data Integrator のインストール・ディレクトリの内容

Oracle Data Integrator のインストール・ディレクトリには、次のサブディレクトリが含まれます。

ディレクトリ	説明
/bin	Oracle Data Integrator の様々なモジュールおよびツールを起動するためのすべてのスクリプトが含まれます。これらのスクリプトの詳細は、後述の説明を参照してください。
/demo	Oracle Data Integrator に付属するデモンストレーションに必要なすべてのデータセットが含まれます。他に、デモ・リポジトリも含まれます。
/doc	電子形式のドキュメントが含まれます。
/drivers	Oracle Data Integrator に付属するすべてのドライバが含まれます。
/impexp	リポジトリにインポートできるすべてのオブジェクト（ナレッジ・モジュール、テクノロジーなど）が含まれます。
/lib	Oracle Data Integrator で使用されるすべての Java ライブラリが含まれます。
/tools	Oracle Data Integrator に付属する外部ツールが含まれます。

### スクリプトおよびツール

次の表に、/bin ディレクトリに含まれるスクリプトとツールをリストします。これらのスクリプトの拡張子は、.bat (Windows スクリプト) および.sh (UNIX スクリプト) です。

これらのスクリプトを起動するには、Oracle Data Integrator の/bin フォルダに移動してスクリプト・ファイルのアイコンをダブルクリックするか、コマンドラインで起動するモジュールの名



前を入力します。オンライン・ヘルプを表示するには、コマンドラインで<script\_name> - help と入力します。

ファイル	説明
agent	エージェントを起動します。
agentscheduler	スケジューラ・エージェントを起動します。
agentservice	エージェントまたはエージェント・スケジューラをサービスとしてインストールまたは削除します。(Windows 専用)
agentstop	エージェントを停止します。
agentweb	Web エージェントを起動します。
designer	デザイナを起動します。
jython	Jython コンソールを起動します。
mimport	マスター・リポジトリ・インポート・ウィザードを起動します。
mupgrade	マスター・リポジトリ・アップグレード・ウィザードを起動します。
operator	オペレータを起動します。
recreate	マスター・リポジトリ作成ウィザードを起動します。
restartsession	セッションを再開します。
security	セキュリティ・マネージャを起動します。
setpath	ユーティリティ・スクリプト。クラスパスにパラメータとして渡されるファイル名を追加します。(Windows 専用)
odiparams	構成スクリプト。このスクリプトには、Oracle Data Integrator モジュールを起動するためのパラメータが含まれます。これらの変数は、ファイル内で手動で更新できます。各変数の説明は、スクリプトに記載されています。
startcmd	Oracle Data Integrator コマンドを起動します。
startdemo	デモンストレーション環境を起動します。
startscen	シナリオの実行を開始します。
stopdemo	デモンストレーション環境を停止します。
topology	トポロジ・マネージャを起動します。

## Oracle Data Integrator の手動インストール

Oracle Data Integrator のスタンドアロンを手動でインストールできます。この手順は、オペレーティング・システムに使用可能な設定方法がない場合に使用できます。

### Oracle Data Integrator を手動でインストールする手順:

1. Java 環境をインストールします。「Java の構成」を参照してください。
2. Oracle Data Integrator の CD からインストールする場合は、CD の/oracledi ディレクトリをインストール・ディレクトリにコピーします。ダウンロード・ファイルからインストールする場合は、Oracle Data Integrator のアーカイブ・ファイルをインストール・ディレクトリに解凍します。
3. 次の環境変数を設定します。
  - ODI\_JAVA\_HOME: これは、Oracle Data Integrator で使用する Java マシンのインストール・ディレクトリを表します。このディレクトリには、/bin サブディレクトリが含まれる必要があります。この変数を設定しない場合、Oracle Data Integrator ではデフォルトの Java マシンが使用されます。

**注意:** Microsoft Windows 95 または 98 を使用する場合、作業を進める前に「Windows 95/98 での Oracle Data Integrator の実行」に記載されている操作を実行する必要があります。

インストールが完了したら、リポジトリの作成に進みます。または、Oracle Data Integrator のモジュールを起動して既存のリポジトリに接続します。

## Windows サービスとしてのエージェントの実行

エージェントまたはスケジューラ・エージェントは、Windows NT、2000、2003 または XP プラットフォームにサービスとしてインストールできます。

セットアップ・プログラムの実行時に、設定で「ツール」オプションを選択する必要があります。このオプションにより、エージェントをサービスとして実行するための Java ラッパー・ツールがインストールされます。

### エージェントをサービスとしてインストールする手順:

1. 構成に一致するよう /bin/odiparams.bat ファイルを編集します。作成されるサービスは、このファイルで指定されたエージェント起動パラメータを使用します。
2. Windows のコマンドプロンプトを起動します (Windows)。
3. Oracle Data Integrator のインストール・フォルダの /bin サブディレクトリに移動します。
4. 適切なパラメータを使用して agentservice.bat スクリプトを実行し、サービスをインストールします。
5. サービスをインストールしたら、Windows のサービス・マネージャに移動してサービスを起動する必要があります。

## 構文

```
agentservice.bat -i|-r -a|-s [<agent_name> [<agent_port>
[<wrapper_configuration_file>]]]
```

パラメータ	説明
-i -r	-i オプションは、サービスをインストールします。 -r オプションは、サービスをアンインストールします。
-a -s	-a は、リスナー・エージェントをインストールします (snpsagent.conf パラメータ・ファイルが使用されます)。 -s は、スケジューラ・エージェントをインストールします (snpsagentscheduler.conf パラメータ・ファイルが使用されます)。
<agent_name>	トポロジで宣言されている物理エージェントの名前。このパラメータは、スケジューラ・エージェントでは必須で、リスナー・エージェントではオプションです。  <b>注意:</b> このパラメータは、作成されるサービスの定義に影響します。サービスをアンインストールするとき、インストール時に指定した名前と同じ <agent_name> を指定する必要があります。
<agent_port>	エージェントのリスニング・ポート。
<wrapper_configuration_file>	ラッパー構成ファイルの名前。このファイルは、Oracle Data Integrator のインストール・フォルダの tools/wrapper/conf/サブディレクトリに存在する必要があります。このパラメータはオプションです。

## 構成ファイル

エージェントの起動構成（接続リポジトリ、Java パラメータ）は、odiparams.bat ファイルで定義します。

Oracle Data Integrator のインストール・フォルダの tools/wrapper/conf/サブディレクトリに含まれるラッパー構成ファイルでは、サービスの起動方法を定義します。このファイルには、エージェントのパラメータの一部が含まれており、その内容は使用する構成に応じて異なる可能性があります。

たとえば、次の要素が変化する可能性があります。

- 追加ドライバは、wrapper.java.classpath.xx プロパティとしてクラスパスに追加する必要があります。
- ログ・ファイル管理に関連するプロパティ (wrapper.logfile.xx) は、必要とされるログのサイズおよびレベルに応じて変化する可能性があります。
- CPU 負荷の高いマシンでは、ping タイムアウト (wrapper.ping.timeout) を増加させます。

この構成ファイルのリファレンスは、<http://wrapper.tanukisoftware.org/>を参照してください。

**警告:** 既存のサービスで使用されている構成ファイルは変更しないでください。最初にサービスをアンインストールする必要があります。

## 構成ファイルの例

このサンプル・ファイルでは、実際の構成に合わせて調整する主なパラメータを緑で表しています。

```
*****
# Java wrapper Properties properties file for an Agent Scheduler
# Note that most parameters are set in the odiparams.bat file, or on
# the agentservice.bat command line.
# They are displayed in this file for information.
*****
# *** Set the path to reference the jvm used for your Agent
# This parameter is read in odiparams.bat
# wrapper.java.command=java.exe
wrapper.working.dir=../../bin/
# Java Main class.
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
# Java Classpath.Add class path elements as needed starting from 1
wrapper.java.classpath.1=../tools/wrapper/lib/wrapper.jar
wrapper.java.classpath.2=../lib/
wrapper.java.classpath.3=../lib/sunopsis.zip
wrapper.java.classpath.4=../lib/snpshelp.zip
wrapper.java.classpath.5=../lib/jakarta-ant-1.4.1-optional.jar
wrapper.java.classpath.6=../lib/NetComponents.jar
wrapper.java.classpath.7=../lib/sunjce_provider.jar
wrapper.java.classpath.8=../lib/jce1_2_2.jar
wrapper.java.classpath.9=../lib/local_policy.jar
wrapper.java.classpath.10=../lib/US_export_policy.jar
wrapper.java.classpath.11=../lib/scripting/
wrapper.java.classpath.12=../lib/scripting/bsf.jar
wrapper.java.classpath.13=../lib/scripting/bsh-1.2b7.jar
wrapper.java.classpath.14=../lib/scripting/js.jar
wrapper.java.classpath.15=../lib/scripting/jython.jar
wrapper.java.classpath.16=../drivers/
wrapper.java.classpath.17=../drivers/classes12.zip
wrapper.java.classpath.18=../drivers/db2java.zip
wrapper.java.classpath.19=../drivers/jt400.zip
wrapper.java.classpath.20=../drivers/mysql.zip
wrapper.java.classpath.21=../drivers/snpsmq.zip
wrapper.java.classpath.22=../drivers/snpsxmlo.zip
wrapper.java.classpath.23=../drivers/sqlj.zip
wrapper.java.classpath.24=../drivers/common.jar
wrapper.java.classpath.25=../drivers/db2fs.jar
wrapper.java.classpath.26=../drivers/db2jcc.jar
```

---

```
wrapper.java.classpath.27=../drivers/jconn2.jar
wrapper.java.classpath.28=../drivers/xerces.jar
# *** add here the additionnal driver classes that do not appear in the
list above
# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.1=../tools/wrapper/lib/
# Java Additional Parameters
wrapper.java.additional.1=-Djava.security.policy=server.policy
# Initial Java Heap Size (in MB)
# This parameter is read in odiparams.bat
# wrapper.java.initmemory=64
# Maximum Java Heap Size (in MB)
# This parameter is read in odiparams.bat
# wrapper.java.maxmemory=128
# Application parameters.
wrapper.app.parameter.1=oracle.odi.Agent
# The parameters below, related to the repository connection, are read
in odiparams.bat
# update them in odiparams.sh
# wrapper.app.parameter.1=oracle.odi.Agent
# wrapper.app.parameter.2="-SECU_DRIVER=org.hsqldb.jdbcDriver"
# wrapper.app.parameter.3="-SECU_URL=jdbc:hsqldb:hsqldb://localhost"
# wrapper.app.parameter.4="-SECU_USER=sa"
# wrapper.app.parameter.5="-SECU_PASS="
# wrapper.app.parameter.6="-WORK_REPOSITORY=WORKREP2"
#
# Add additionnal parameters below, starting from parameter number 10
#*****
# Wrapper Logging Properties
#*****
# Format of output for the console.(See docs for formats)
wrapper.console.format=PM
# Log Level for console output.(See docs for log levels)
wrapper.console.loglevel=INFO
# Log file to use for wrapper output logging.
wrapper.logfile=agentservice.log
# Format of output for the log file.(See docs for formats)
wrapper.logfile.format=LPTM
# Log Level for log file output.(See docs for log levels)
wrapper.logfile.loglevel=INFO
# Maximum size that the log file will be allowed to grow to before
# the log is rolled.Size is specified in bytes.The default value
# of 0, disables log rolling.May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix.For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=1m
```

```
# Maximum number of rolled log files which will be allowed before old
# files are deleted.The default value of 0 implies no limit.
wrapper logfile.maxfiles=10
# Log Level for sys/event log output.(See docs for log levels)
wrapper syslog.loglevel=FATAL
*****
# Wrapper NT Service Properties
*****
# WARNING - Do not modify any of these properties when an application
# using this configuration file has been installed as a service.
# Please uninstall the service before modifying this section.The
# service can then be reinstalled.
# Name of the service
# This parameter is set depending on the agent Name
wrapper.ntservice.name=SnpsAgent
# Display name of the service
wrapper.ntservice.displayname=Oracle Data Integrator Agent
# Description of the service
wrapper.ntservice.description=Executes Oracle Data Integrator Sessions
# Service dependencies.Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=
# Mode in which the service is installed.AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START
# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false
*****
# Wrapper JVM Timeouts in seconds (0 = No timeout limit)
*****
# wrapper.ping.timeout=10
# wrapper.cpu.timeout=30
```

## Windows 95/98 での Oracle Data Integrator の実行

Oracle Data Integrator を Windows 95 または 98 で実行するには、Oracle Data Integrator アプリケーションに割り当てるデフォルト・メモリーを増加させる必要があります。

**Oracle Data Integrator アプリケーションに割り当てるデフォルト・メモリーを増加させる手順:**

1. 次の各ファイルのショートカットを作成します。ショートカットを作成するには、ファイル名を右クリックして「ショートカットの作成」を選択するか、Oracle Data Integrator のセットアップ・プログラムにより作成されたショートカットを変更します。
  - designer.bat
  - topology.bat
  - startdemo.bat
2. 各ショートカットを右クリックし、「プロパティ」を選択します。
3. 「メモリ」タブに移動します。

- 「コンベンショナルメモリ」の「合計」を 640 に（可能であればそれ以上に）設定します。
  - 「コンベンショナルメモリ」の「環境変数の初期サイズ」を 4096 に（可能であればそれ以上に）設定します。
  - 「XMS メモリ」の「合計」を 16384 に（可能であればそれ以上に）設定します。
  - 「MS-DOS プロテクトモード (DPMI) メモリ」を 16384 に（可能であればそれ以上に）設定します。
4. 「適用」をクリックします。
  5. プロパティ・ウィンドウを閉じます。
- これ以降、各コンポーネントを実行するにはこれらのショートカットのみを使用します。

## iSeries および AS/400 への Java エージェントのインストール

一部のデータベースは、iSeries および AS400 のマシンにインストールされています。Oracle Data Integrator の実行エージェントをこれらのマシンにインストールし、次のことを実行できます。

- AS/400 でのロード・プロセスの実行
- OS400 のシステム・コマンドの実行
- ネットワーク・フローの削減（AS/400 にソースとターゲットの両方が存在する場合）

AS/400 マシンに Oracle Data Integrator の実行エージェントを実装するためのインストール手順は、次のとおりです。

### システムの準備

#### 技術上の前提条件

- 次の PTF が適用された AS/400、AS/400 V5R1 または V4R4M0:
  - SF61800
  - SF55849
  - SF54922
  - インストールするプログラム:
- IBM Toolbox for Java
- Java 仮想マシン 1.3.1
- AS/400 マシンで構成および起動された TCP/IP サービス

#### コンポーネント

使用する Java および JDBC コンポーネントは、次のとおりです。

- Oracle Data Integrator 実行エージェント
- Java 仮想マシン 1.3.1

- データベース・アクセス用の JDBC ドライバ

**ドライバに関する注意:** AS/400 から DB2/400 に接続する場合、AS/400 用のドライバ・ラッパーの使用を検討する必要があります。詳細は、「DB2/400 データ・サーバーの作成」を参照してください。

## 実行エージェントのインストール

### ファイルのインストール

1. AS/400 に Oracle Data Integrator ファイルを格納するためのディレクトリ・ツリーを作成します。

```
MKDIR DIR('/odi')
MKDIR DIR('/odi/bin')
MKDIR DIR('/odi/lib')
MKDIR DIR('/odi/lib/scripting')
MKDIR DIR('/odi/drivers')
```

2. Oracle Data Integrator の CD の /oracledi ディレクトリから、手順 1 で作成した AS/400 のディレクトリに (FTP などを使用して) ファイルを転送します。コピーするファイルの場所は、次のとおりです。

```
/oracledi/bin/
/oracledi/lib/
/oracledi/lib/scripting (Oracle Data Integrator のスクリプト機能を使用する
場合)
/oracledi/drivers (使用するドライバのみコピーが必要)
```

### Java プログラムの作成

AS/400 でのパフォーマンス上の理由から、Java パッケージ (.class、.jar または .zip ファイル) は **Java プログラム** に変換する必要があります。

#### .class、.jar または .zip ファイルから Java プログラムを作成する手順:

1. 次のように CL コマンドを実行します。

```
CRTJVAPGM CLSF('<.class, .zip or .jar file
location>') OPTIMIZE(40)
```

たとえば、odi.zip ファイル (Java エージェント) から Java プログラムを作成するには、次のコマンドを実行します。

```
CRTJVAPGM CLSF('/odi/lib/odi.zip') OPTIMIZE(40)
```

**注意:** Java プログラムの作成は、Java パッケージの内容によっては長い時間がかかります。実際に使用するクラスのプログラムのみを作成することをお勧めします。

## 実行エージェントの起動

実行エージェントは、次の 2 つの方法で起動できます。



- **シェル・インタプリタ (QSH または STRQSH) の使用:** これは、UNIX 互換シェルを実装する OS/400 のオプションです。多くの UNIX コマンド (ls、chmod、chown など) に加え、java コマンドも使用できます。Oracle Data Integrator の/bin に含まれる UNIX スクリプト (.sh 拡張子) で標準の構文を使用することで、Oracle Data Integrator を起動できます。エージェントを実行する前に、odiparams.sh ファイルを構成する必要があります。AS/400 では、グラフィカル・モジュール (トポロジやデザイナなど) を使用できます。
- **OS/400 コマンド (CL) の使用:** RUNJAVA または JAVA CL コマンドで Java アプリケーションを実行します。エージェントやシナリオを起動する場合、CL プログラムを使用すると便利です。これらの操作用のプログラム・テンプレートを次に示します。

**JVM バージョンに関する注意:** 複数の Java マシンが AS/400 にインストールされている場合、状況によっては Java コマンドで使用される Java のバージョンを強制的に指定する必要があります。

- **QSH** の場合: バージョンを強制するには、Java コマンドのフラグ `-Djava.version=<java version>` (`-Djava.version=1.3.1` など) を使用します。

- **OS/400 コマンド** の場合: JAVA コマンド・パラメータとして `-PROP((<property> <value>) (<property> <value>))` のようにプロパティを渡します。たとえば、`-PROP((java.version 1.3.1))` のように指定します。

**エージェントの名前付けに関する注意:** 通常、AS/400 では、エージェントをリスナーとして実行する際にそのエージェントの名前を明示的に指定します。そのため、エージェントの実行時には、エージェントのフラグ `-name=<agent name>` を使用する必要があります。

## エージェントの実行

```

                PGM                PARM(&NAME &PORT &VERB)
/* Command AGENT */
/* Parameters: */
/*   &NAME: physical name of the agent */
/*   &PORT: port number */
/*   &VERB: verbose mode -V=[1..5] */
/* Example of call: */
/*   CALL PGM(<myLib/myPGM>) PARM('-NAME=myAgt' '-PORT=20910' '-V=5') */
                DCL                VAR(&NAME) TYPE(*CHAR) LEN(128)
                DCL                VAR(&PORT) TYPE(*CHAR) LEN(30)
                DCL                VAR(&VERB) TYPE(*CHAR) LEN(30)
/* All classes below should be compiled */
/* with the CRTJVAPGM command */
/* with optimize 40. */
                DCL                VAR(&PROJ) TYPE(*CHAR) LEN(512) +
                VALUE('/odi/lib/odi.zip:+
                /odi/lib/sunjce_provider.jar:+
                /odi/lib/commons-net.jar:+
                /odi/lib/local_policy.jar:+
                /odi/lib/jakarta-ant-optional.jar:+
                /odi/lib/US_export_policy.jar:+
                /odi/lib/jce1_2_2.jar')
```

```

/* Replace the drivers below with your own drivers. */
        DCL          VAR(&JDBC) TYPE(*CHAR) LEN(512) +
                    VALUE('/odi/drivers/jt400Native.jar:+
                        /odi/drivers/snpsdb2.jar:+
                        /odi/drivers/ojdbc14.jar')

/* Build the Java CLASSPATH */
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
        CHGVAR       &PATH (&PROJ *tcat ':' *tcat &JDBC)

/* Start the Agent */
        SBMJOB       CMD(JAVA CLASS(oracle.odi.Agent)
CLASSPATH(&PATH) +
                    PARM('&NAME &PORT &VERB') +
                    OPTIMIZE(40) +
                    OUTPUT(*PRINT))

        ENDPGM

```

## シナリオの実行

```

        PGM          PARM(&SCEN &VERS &CTX &VERB)

/* Command STARTSCEN */
/* Parameters: */
/*   &SCEN: scenario name */
/*   &VERS: scenario version */
/*   &CTX: context */
/*   &VERB: verbose mode -V=[1..5] */
/* Example of call: */
/*   CALL PGM(<myLib/myPGM>) PARM('myScen' 'myVers' 'GLOBAL' '-V=5') */
        DCL          VAR(&SCEN) TYPE(*CHAR) LEN(30)
        DCL          VAR(&VERS) TYPE(*CHAR) LEN(30)
        DCL          VAR(&CTX) TYPE(*CHAR) LEN(30)
        DCL          VAR(&VERB) TYPE(*CHAR) LEN(30)

/* All classes below should be compiled */
/* with the CRTJVAPGM command */
/* with optimize 40. */
        DCL          VAR(&PROJ) TYPE(*CHAR) LEN(512) +
                    VALUE('/odi/lib/odi.zip:+
                        /odi/lib/sunjce_provider.jar:+
                        /odi/lib/commons-net.jar:+
                        /odi/lib/local_policy.jar:+
                        /odi/lib/jakarta-ant-optional.jar:+
                        /odi/lib/US_export_policy.jar:+
                        /odi/lib/jce1_2_2.jar')

/* Replace the drivers below with your own drivers. */
        DCL          VAR(&JDBC) TYPE(*CHAR) LEN(512) +
                    VALUE('/odi/drivers/jt400Native.jar:+

```

```

                                /odi/drivers/snpsdb2.jar:+
                                /odi/drivers/ojdbc14.jar')
/* Adapt all parameters below to your environment before use. */
        DCL          VAR(&DRV) TYPE(*CHAR) LEN(128) +
                                VALUE('-
SECU_DRIVER=com.ibm.as400.access.AS400JDBCdriver')
        DCL          VAR(&URL) TYPE(*CHAR) LEN(128) +
                                VALUE('-
SECU_URL=jdbc:as400://195.10.10.13;libraries=ODI')
        DCL          VAR(&USER) TYPE(*CHAR) LEN(30) +
                                VALUE('-SECU_USER=QSECOFR')
        DCL          VAR(&PASS) TYPE(*CHAR) LEN(128) +
                                VALUE('-SECU_PASS=XYZ')
        DCL          VAR(&WREP) TYPE(*CHAR) LEN(30) +
                                VALUE('-WORK_REPOSITORY=WORKREP1')
        DCL          VAR(&SUSER) TYPE(*CHAR) LEN(30) +
                                VALUE('-ODI_USER=SUPERVISOR')
        DCL          VAR(&SPASS) TYPE(*CHAR) LEN(128) +
                                VALUE('-ODI_PASS=XYZ')
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
/* Build the Java CLASSPATH */
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
        CHGVAR      &PATH (&PROJ *tcat ':' *tcat &JDBC)
/* Execute the Scenario */
        SBMJOB      CMD(JAVA CLASS(oracle.odi.Agent)
CLASSPATH(&PATH) +
                                PARM(&DRV &URL &USER &PASS &WREP &SUSER &SPASS
+
                                SCEN &SCEN &VERS &CTX &VERB))
        ENDPGM

```

**注意:** 指定するパスワードは、コマンド agent ENCODE <password>を使用して暗号化する必要があります。