

# リファレンス・マニュアル

---

このマニュアルでは、Oracle Data Integrator の Graphical User Interface について説明します。対象読者は、Oracle Data Integrator に関する知識を持っていて、さらに活用しようとしている、高度な開発者および管理者です。

## このマニュアルの構成

このマニュアルでは次の内容を説明しています。

- 第1章から第4章は、Oracle Data Integrator の Graphical User Interface のリファレンスです。

## デザイナ

### デザイナの概要

デザイナ・モジュールで操作できるものは次のとおりです。

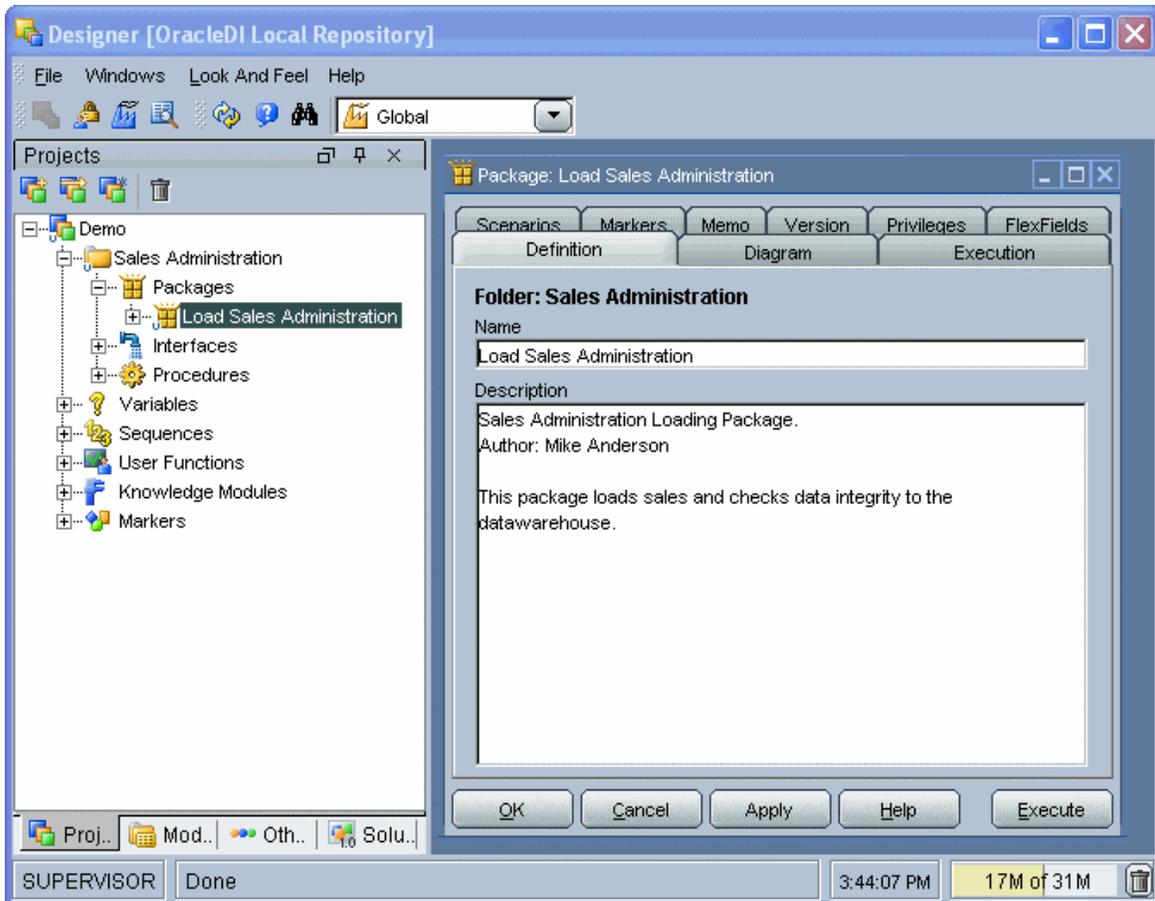
モデル: データとアプリケーションの構造の説明

プロジェクト: デザイナで作成される開発物

デザイナ・モジュールはこの情報を作業リポジトリに格納します。トポロジおよびセキュリティ情報はマスター・リポジトリで定義されているものを使用します。

### デザイナのインタフェース

デザイナの GUI は次のように表示されます。



## メニュー

メニューでは、プルダウン・メニューから次の機能にアクセスできます。

- インポート/エクスポート
- ウィザード
- オプションの表示
- モジュールまたはツリー表示のオープン
- ユーザーのパスワードおよびオプションの変更

## ツールバー

ツールバーからは、次の操作を実行できます。

- 他のモジュールの起動
- ツリー表示のリフレッシュ
- オンライン・ヘルプの起動
- デフォルトのコンテキストを選択します。

選択されたコンテキストが、アプリケーション・ウィンドウで選択されるすべてのコンテキストのデフォルトとして使用されます。また、データが参照されると（データストアでデータを右ク

リック)、データはツールバーで定義されたコンテキストで表示されます。たとえば、コンテキストが「開発」の場合、「開発」コンテキストでデータが参照されます。安全のため、すべてのコンテキストで認証されている場合でも、不適当な操作をしないように、コンテキストを変更する際には常に確認のパスワードを求められます。メニューバーには、許可されているコンテキストのみが表示されます。

## ツリー表示

現行ユーザーが使用できるデザイナー・オブジェクトは、**プロジェクト**、**モデル**、**ソリューション** および**その他** (ユーザー・ファンクション、グローバル変数および順序) の各ツリー表示に整理して表示されます。

各ツリー表示は、メイン・ウィンドウの両側にドッキングできるフローティング・フレームに表示されます。フレームは重ねることもできます。複数のフレームを重ねた場合は、フレーム・ウィンドウの下部に表示されるタブから各フレームにアクセスできます。

ツリー表示フレームは、フレームのタイトルまたはタブを選択してドラッグすることにより、移動したり、ドッキングしたり、重ねたりできます。ツリー表示の位置を固定するには、「**ウィンドウ**」メニューから「**ビューのロック**」を選択します。

ツリー表示フレームがウィンドウに表示されないか、閉じている場合は、「**ウィンドウ**」→「**ビューの表示**」メニューを使用します。

各ツリー表示では、次の操作が可能です。

- ルート・オブジェクトの挿入またはインポート (フレーム・タイトルで該当するボタンをクリック)
- ノードの展開または折りたたみ (ノードをクリック)
- オブジェクトに関連付けられているメソッド (編集、削除など) のアクティブ化 (ポップアップ・メニューを使用)
- オブジェクトの編集 (オブジェクトをダブルクリック、または**ワークベンチ**にドラッグ・アンド・ドロップ)

## ワークベンチ

編集または表示されているオブジェクトのウィンドウは、**ワークベンチ**に表示されます。

## 接続

### 作業リポジトリへの接続

このウィンドウを使用すると、**デザイナー**・モジュールを開く際に、作業リポジトリへの接続を選択できます。既存の接続の作成、変更または削除を開始することもできます。

**注意:** 作業リポジトリ接続の定義は、ローカルの/bin/snps\_login\_work.xml ファイルに格納されています。

### 一般プロパティ

プロパティ	説明
ログイン名	作業リポジトリへの定義された接続のリスト。
ユーザー	選択された作業リポジトリに接続するためのデフォルトの <b>Oracle Data Integrator</b> ユーザー。デフォルト・ユーザーが指定されなかった場合は、ここに入力する必要があります。
パスワード	接続する <b>Oracle Data Integrator</b> ユーザーに対応するパスワード。デフォルトのパスワードが指定されなかった場合は、ここに入力する必要があります。

## ツールバー

ボタン	説明
	作成される作業リポジトリへの新しい接続を許可します。
	「ログイン名」フィールドで選択された接続の変更を許可します。
	「ログイン名」フィールドで選択された接続の削除を許可します。

## 作業リポジトリ接続の詳細

このウィンドウを使用すると、**作業リポジトリ**への接続を定義または変更できます。

**注意:** 各作業リポジトリはマスター・リポジトリにアタッチされているため、作業リポジトリへの物理的接続についての情報は、アタッチ先のマスター・リポジトリに格納されます。作業リポジトリへの接続を定義するとは、マスター・リポジトリへの接続を定義してから、そのマスター・リポジトリにアタッチされた作業リポジトリの1つを選択することです。

**注意:** 作業リポジトリ接続の定義は、ローカルの `/bin/snps_login_work.xml` ファイルに格納されています。

## 一般プロパティ

プロパティ	説明
<b>Oracle Data Integrator 接続</b>	この情報グループを使用して、接続ウィンドウに表示されるプロパティを定義することができます。
ログイン名	Data Integrator モジュールが開いている場合、これがリストに表示される名前です。
ユーザー	作業リポジトリに接続するためのデフォルトの Oracle Data Integrator ユーザー。デフォルト・ユーザーが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。
パスワード	Oracle Data Integrator ユーザーに対応しているパスワード。デフォルトのパスワードが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。
<b>データベース接続(マスター・リポジトリ)</b>	この情報グループは、マスター・リポジトリをホストするリレーショナル・データベースの接続パラメータを示します。
ユーザー	作業リポジトリがアタッチされているマスター・リポジトリが含まれているデータベースのスキーマへの接続を許可するユーザー。
パスワード	そのデータベース・ユーザーのデータ・サーバー・パスワード。
ドライバ・リスト	マスター・リポジトリが基づいているテクノロジーのタイプ。 ボタン  をクリックすると、使用されるテクノロジーとドライバに関する詳細情報を表示できます。詳細は、「ドライバの詳細」を参照してください。
ドライバ名	マスター・リポジトリのデータ・サーバーに接続するために使用する JDBC ドライバの名前。テクノロジーを選択すると、そのテクノロジーで最もよく使用されるドライバが表示されます。他のドライバを使用することもできます。その場合、そのドライバが適切にインストールされている必要があります。詳細は、インストレーション・ガイドの「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。
URL	マスター・リポジトリのデータ・サーバーに接続するための URL。ボタン  をクリックすると、使用するドライバの URL モデルを選択することができます。詳細は、「URL サンプル」を参照してください。 ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。
<b>作業リポジトリ</b>	この情報グループには、接続に使用されるマスター・リポジトリにアタッチされた作業リポジトリが表示されます。
リポジトリ名	作業リポジトリの名前。 マスター・リポジトリ接続を正しく指定すると、ボタン  を使用して、この

	マスター・リポジトリにアタッチされた作業リポジトリの1つを選択するためのリストにアクセスすることができます。
デフォルトの接続	このボックスを選択すると、Oracle Data Integrator グラフィック・モジュールの起動時に、デフォルトで、現在の接続が提案されます。

## ドライバの詳細

このウィンドウには、使用するドライバとテクノロジーに関する情報が表示されます。

### 一般プロパティ

プロパティ	説明
名前	選択されたテクノロジーで使用するドライバ。
説明	ドライバについての情報。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## URL サンプル

このウィンドウには、ドライバの URL モデルが表示されます。

### 一般プロパティ

プロパティ	説明
名前	選択されたドライバ用の有効な URL モデルのリスト。
説明	URL モデルの詳細。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## プロジェクト

### プロジェクトの概要

プロジェクトは、Oracle Data Integrator を使用して開発されたオブジェクトのグループです。

- プロジェクトの管理

### Oracle Data Integrator プロジェクト・コンポーネント

ツリー表示のプロジェクトの配下に、次のコンポーネントが格納されます。

#### フォルダ

---

プロジェクト内の一部のオブジェクトは、**フォルダ**と**サブフォルダ**に整理されます。

### パッケージ

**パッケージ**は、Oracle Data Integrator で最大の実行単位です。パッケージは、実行ダイアグラムに整理された一連の**手順**で構成されています。

- パッケージの詳細情報
- パッケージの作成

### インタフェース

インタフェースは、1つ以上のソース・データストアから、データストアまたは一時ターゲット構造をロードする方法を定義した一連のルールで構成されます。

- インタフェースの詳細情報
- インタフェースの作成

### プロシージャ

固有プロシージャは、インタフェース・フレームワークにそぐわない操作をグループ化した、再利用可能なコンポーネントで、1つ以上のソースからターゲットデータストアをロードします。

プロシージャの例:

- 待機してファイルを **unzip** する
- **FTP** を介してファイルを一括送信する
- 電子メールを受信する
- データベースをパージする

プロシージャは、トポロジで定義された論理スキーマで、コマンドを起動することができますが、OS コマンドや Oracle Data Integrator ツールを使用することもできます。

### 変数

変数の値は Oracle Data Integrator に格納されます。この値は、実行中に変化することもあります。値には次の性質があります。

- 作成時に定義されたデフォルト値を持ちます。
- 変数を使用したシナリオの実行時にパラメータとして渡すことができます。
- 変数のリフレッシュ、設定および増分ステップで変更できます。
- パッケージで条件を作成して評価できます。
- インタフェース、プロシージャ、ステップなどで使用できます。

変数をプロジェクトの外部（グローバル・スコープ）で定義することにより、すべてのプロジェクトで使用されるようにすることもできます。

### 順序

順序は、使用時に自動的に増分される変数です。その値は、1回使用されてから次に使用されるまで持続します。

順序は、インタフェース、プロシージャ、ステップなどで変数と同じように使用できます。

変数と同様に、プロジェクトの外部（グローバル・スコープ）で順序を定義すれば、すべてのプロジェクトで使用されるようにできます。

## ユーザー関数

ユーザー関数を使用すると、テクノロジー依存の実装を定義する場合に、カスタマイズされた関数や関数の別名を定義できます。ユーザー関数は、インタフェースとプロシージャで使用できます。

## ナレッジ・モジュール

Oracle Data Integrator は、指定されたテクノロジーに関連するメソッドを定義するためにナレッジ・モジュールを使用します。これらのモジュールでは、専用の機能に特化したテクノロジー用のプロセスを生成できます。

**注意:** デフォルトのナレッジ・モジュールは Oracle Data Integrator に付属しており、使用前にプロジェクトにインポートする必要があります。

## マーカー

開発の方法論または組織を反映するために、プロジェクトの要素にフラグを設定することがあります。

フラグはマーカーを使用して定義されます。このマーカーはグループに整理され、プロジェクト内で大部分のオブジェクトに適用できます。

## シナリオ

パッケージ、インタフェース、プロシージャまたは変数コンポーネントが終了すると、それはシナリオにコンパイルされます。シナリオは、スケジュール可能な本番環境用の実行単位です。

# プロジェクト

## 定義

プロパティ	説明
名前	Oracle Data Integrator のグラフィカル・インタフェースに表示されるプロジェクトの名前。
コード	プロジェクト・コードは、プロジェクト・タイプの変数の接頭辞として使用されます。したがって、それを変更すると、プロジェクト・タイプの変数の名前も変更されます。

## フォルダ

フォルダは、パッケージ、インタフェースおよび固有プロシージャのグループです。フォルダとサブフォルダを使用すると、プロジェクト特有の基準に応じてオブジェクトをグループ化して整理することができます。作成できるサブフォルダのレベル数に制限はありません。

**注意:** サブフォルダを作成するには、フォルダを作成し、親フォルダの中にドラッグ・アンド・ドロップします。

## 定義

プロパティ	説明
名前	Oracle Data Integrator のグラフィカル・インタフェースに表示されるフォルダまたはサブフォルダの名前

## パッケージ

### パッケージの概要

パッケージは、Oracle Data Integrator で最大の実行単位です。パッケージは、実行ダイアグラム内に編成された一連のステップで構成されます。

- パッケージの作成

### 手順

手順にはいくつかのタイプがあります。それらは、次の手順群にグループ化することができます。

- **フロー (インタフェース)**: インタフェースを実行します。
- **プロシージャ**: プロシージャを実行します。
- **変数**: 変数の値の宣言、設定、リフレッシュ、または評価を行います。
- **Oracle Data Integrator Tools**: ツールボックスから使用可能なこれらのツールは、すべての Oracle Data Integrator の API コマンドへのアクセスや、オペレーティング・システム・コールを可能にします。
- **モデル、サブモデルおよびデータストア**: これらのオブジェクトに対してジャーナル化、静的管理またはリバース・エンジニアリング操作を行います。

たとえば、「Populate Sales Datamart」パッケージを、次のジョブで構成することができます。

1. プロシージャ「System Backup」
2. インタフェース「Customer Group」
3. インタフェース「Customer」
4. インタフェース「Product」
5. 変数「Last Invoice ID」のリフレッシュ
6. インタフェース「Invoice Header」
7. インタフェース「Invoice Lines」

### パッケージ

## 定義

プロパティ	説明
名前	Oracle Data Integrator のグラフィカル・インタフェースに表示されるパッケージの名前。
説明	パッケージの詳細な説明。

## ダイアグラム

パッケージ・ダイアグラムの詳細は、ダイアグラムの項を参照してください。

## 実行

パッケージは、シナリオを生成しないで直接実行することができます。パッケージを実行することによって、パッケージを構成する一連の手順がスムーズに実行されるかどうかテストすることができます。実行するには「**実行**」ボタンを使用します。実行オプションを選択するウィンドウが開きます。

「**実行**」タブに、前回の実行結果が、次の要素を含むグラフで示されます。

プロパティ	説明
日付	パッケージ実行開始の日付と時刻。
コンテキスト	パッケージの実行コンテキスト。
エージェント	パッケージを実行したエージェントの名前。 <b>internal</b> というメモがついている場合、 <b>デザイナー</b> に含まれるエージェントが使用されたことを示しています。
継続時間	パッケージ実行の所要時間。
ログ・レベル	パッケージの実行時に選択されたロギング・レベル。
エラー	パッケージのリターン・コード。

**注意:** パッケージは、**デザイナー**・モジュール・ツリーでパッケージのコンテキスト・メニューから「**実行**」を選択することによっても起動できます。

## シナリオ

このタブには、このコンポーネント用に生成されたシナリオが、スケジュールとともに、表形式で示されます。プロパティを表示するには、シナリオまたはスケジュールをダブルクリックします。

シナリオの作成および削除は、 「**生成**」および  「**削除**」ボタンをクリックすることによって行うことができます。

このタブでは、シナリオおよびスケジュールのポップアップ・メニューも使用できます。

## ダイアグラム

「パッケージ・ダイアグラム」タブは、パッケージに含まれる一連の**手順**をグラフィカルな手段で定義するために使用されます。

「ダイアグラム」タブは、次の3つのペインに分かれます。

- ツールバー
- 手順および手順シーケンスのグラフィカル表示が含まれる、パッケージの**ダイアグラム・ペイン**
- 選択した手順の情報が表示される**プロパティ・ペイン**

パッケージでは、手順の次には常に、成功か失敗かに対応する別の手順（緑または赤のリンクで表示）、またはパッケージ末尾（表示されません）が追隨します。

### ツールバー

ツールバーに含まれるツールでは次の操作を行うことができます。

- リンクまたは手順の選択
- 手順成功時の次の手順の定義
- 手順失敗時の次の手順の定義
- プロパティ・ペインの表示/非表示
- 選択された手順の複製
- 選択された手順やリンクの削除
- ダイアグラムの再編成
- 選択された手順にリンクしたオブジェクトの編集
- 選択された手順の実行
- パッケージ全体の実行

パッケージ組織にエラーがあると（たとえば未リンクの手順）、**「エラー」** ボタンの色が変わります。エラーの詳細を表示するにはボタンをクリックします。ボタンが選択できない場合、パッケージにエラーはありません。

### ダイアグラム・ペイン

パッケージに手順を追加するには、ツリー表示からダイアグラム・ペインへオブジェクト（インタフェース、変数、特定の処理）をドラッグ・アンド・ドロップします。

ダイアグラム・ペインでは、それぞれの手順が手順タイプ固有のアイコンで表示されます。手順シーケンスは、緑の矢印（成功時）と赤い矢印（失敗時）で表示されます。**変数評価**手順では、どちらの矢印も緑で、その条件が真または偽の場合に次の手順が実行されるかどうかを示すマーカーが付けられます。

また、手順やリンクのドラッグ・アンド・ドロップを使用して、手順のコンテキスト・メニューのオプションを並べ替えたり、**「再編成」** ボタンを使用して手順の位置を自動で並べ替えたりすることができます。

手順の**プロパティ・ペイン**を開くには、そのアイコンをクリックします。**リンクされたオブジェクト**（フロー手順の場合はインタフェース、変数評価手順の場合は変数）を開くには、手順のアイコンをダブルクリックします。

手順を右クリックしてコンテキスト・メニューからそれぞれのオプションを選択することで、手順を**実行、複製、編集または削除**できます。

### プロパティ・ペイン

#### 「一般」タブ

このタブには、手順に関する情報が表示されます。このタブのプロパティは、選択された手順タイプにより異なります。

#### 一般的なプロパティ

次のプロパティは、大部分の手順タイプで表示されます。

プロパティ	説明
名前	Oracle Data Integrator のグラフィカル・インタフェースに表示される手順の名前。
タイプ	手順のタイプ。このフィールドは Oracle Data Integrator ツールでは表示されません。
リンクされたオブジェクト	この手順を作成するためにパッケージに挿入されたインタフェース、手順、モデル、サブモデル、データストアまたは変数の名前です。このフィールドは Oracle Data Integrator ツールでは表示されません。
パス	リンクされたオブジェクトへのパス。たとえば、フロー手順のインタフェースを含むプロジェクトとフォルダが含まれます。

#### 固有プロパティ

次に、それぞれのタイプの手順に固有のプロパティについて詳しく説明します。次に示されていない手順には固有プロパティはありません。

#### 変数の設定

変数設定手順では、次のプロパティが表示されます。

プロパティ	説明
値を割当て	このオプションを選択すると、変数は、「変数」タブの「値」フィールドで指定された値をとります。
値を増加させます	このオプションは、数値タイプの変数でのみ表示されます。「変数」値を、「増分」フィールドで定義された値だけ増分します。
値、増分	変数に割り当てられた値（割当ての場合）、または変数に加えられる増分（数値変数の増分の場合）。

#### 変数の評価

変数評価手順では、次のプロパティが表示されます。

プロパティ	説明
オペレータ	変数の値と、「変数」タブの「値」フィールドで定義された値の間のリレーショナル演算子。
値	変数の値と比較する値。

**注釈:** value フィールドに比較先の変数名を入力することで、2つの変数を比較することが可能です。変数名には、GLOBAL か、この変数が含まれるプロジェクトの名前を接頭辞として付ける必要があります。

**注意:** IN 演算子を使用する場合、変数値の検索対象となる値リストは、AB;AC;AA;AE または 66;33;88;11;22;92 などのセミコロンで区切られたリストです。

#### ジャーナル化手順

このプロパティは、**ジャーナル化タイプ**の手順でのみ表示されます。次に示すフィールドは、処理されるオブジェクト（データストア、モデル）やジャーナル化モードのタイプ（一貫性セットまたは簡易）に関係なく表示されます。

プロパティ	説明
開始	データストア、モデルまたはサブモデル用のジャーナル化を開始します。
停止	データストア、モデルまたはサブモデル用のジャーナル化を停止します。
サブスクリイバの追加	サブスクリイバのリスト用のサブスクリプションを作成します。「 <b>追加</b> 」または「 <b>削除</b> 」ボタンを使用して、リストに対してサブスクリイバを追加または削除することができます。
サブスクリイバの削除	サブスクリイバのリストに表示されたサブスクリプションを削除します。
ウィンドウの拡張	CDC セットまたはデータストア用の整合性ウィンドウを拡大します。
ジャーナルのページ	不要のエントリからジャーナルをページします。
サブスクリイバのロック	指定されたサブスクリイバに対して整合性ウィンドウ内のエントリをロックします。
サブスクリイバのロック解除	指定されたサブスクリイバに対して整合性ウィンドウ内のエントリをロック解除します。変更の消費を検証します。

#### チェック手順

このプロパティは、**チェック・タイプ**の手順でのみ表示されます。

プロパティ	説明
チェック済表からのエラーの削除	チェックされたデータストア、モデルまたはサブモデルから、誤っているデータを削除します。

#### OS コマンド実行手順

このプロパティは、**OS コマンド実行**タイプの手順でのみ表示されます。

プロパティ	説明
順序	エージェントにより、OS Order または Oracle Data Integrator Tool が起動されました。Oracle Data Integrator ツールの詳細は、Data Integrator ツールの構文を参照してください。

**注意:** OS コマンドと Oracle Data Integrator ツールは、指定された環境（OS、マシンなど）でエージェントによって起動されるため、環境の特徴を考慮に入れる必要があります（ファイル・アクセス・パス、OS 命令の構文など）。

#### その他の変数

このタブは、SnpsStartScen ツールの使用時に表示されます。このタブでは、このシナリオの起動変数のリストを指定することができます。

#### 「コマンド」タブ

このタブは、Oracle Data Integrator ツールの使用時に表示されます。「一般」タブで指定されたパラメータを使用して呼び出されたツールに対応するコマンドが含まれます。

#### 「拡張」タブ

このタブでは、手順が成功するか、失敗するかどうかに応じて、パッケージの動作を指定することができます。

#### 成功または失敗の条件

- **フロー（インタフェース）、プロシージャ、OS コマンド実行または Oracle Data Integrator ツール実行**タイプの手順が成功するとは、プロシージャ、インタフェースまたはコマンドが適切に実行されることです（リターン・コード 0）。それ以外の場合は失敗です。
- **変数設定**タイプの手順が成功するとは、割当てが正しく実行されることです。割当てができない場合（たとえば、数値に連鎖を割り当てた場合）は、手順は失敗です。
- **変数リフレッシュ**・タイプの手順が成功するとは、変数の値の計算を可能にする選択順位が実行コンテキストで正しく実行され、このタイプの変数に対して値が適切であるということです。それ以外の場合は手順は失敗です。
- **変数評価**タイプの手順が成功するとは、条件<variable> <operator> <value> が真であるということです。それ以外の場合は失敗です。
- **ジャーナル化、リバースまたはチェック**・タイプの手順が成功するとは、操作が適切に実行されることです（リターン・コード 0）。それ以外の場合は失敗です。

#### 成功または失敗の後に続く動作

次のパラメータを指定することによって、成功または失敗の後に続く動作を定義することができます。

プロパティ	説明
成功時	手順が成功した場合に実行する動作を記述します。
終了	パッケージの実行が停止し、すべてのトランザクションがロールバックに従います。

ステップの実行	このフィールドを使用すると、パッケージ手順を選択できます。特殊な手順<<Next Step>>を使用すると、パッケージ内で現在の手順の後に続く手順を実行できます。
失敗時	手順が失敗した場合に実行する動作を記述します。
試行回数	手順が失敗した場合に再試行する回数を定義します。
各試行の間隔	手順の再試行間隔（単位は秒）。
終了	パッケージの実行が停止し、すべてのトランザクションがロールバックに従います。この動作は再試行後に実行されます。
ステップの実行	このフィールドを使用すると、パッケージ手順を選択できます。特殊な手順<<Next Step>>を使用すると、パッケージ内で現在の手順の後に続く手順を実行できます。この動作は再試行後に実行されま
ジャーナルへのステップのログ	手順の実行が完了したときに、手順情報をどのようにジャーナルに記録するかを記載します。
ジャーナルへのステップのログ	<p>手順は、実行前と実行中に実行ログに表示されます。このドロップ・ボックスでは、手順の実行が終了した後も手順をジャーナルに維持するかどうかを示します。</p> <ul style="list-style-type: none"> <li>なし: 手順はジャーナルから削除されます。</li> <li>常時: 常に手順がジャーナルに維持されます。</li> <li>エラー: 失敗した場合にかぎり、手順がジャーナルに維持されま</li> </ul> <p>ず。それ以外の場合は削除されます。</p>

#### デフォルトの動作

パッケージ内の手順のデフォルト動作は次のとおりです。

- 手順が成功した場合は、パッケージ内の分類順序における次の手順に移ります。
- 手順が異常終了すると、パッケージの実行が停止し、すべてのトランザクションがロールバックに従います。

標準シリーズおよび繰り返しループの一部のモデルが「シリーズの例」に記載されています。

#### 「オプション」タブ

このタブを使用すると、プロシージャまたはナレッジ・モジュールのオプションの値（ジャーナル化、チェックまたはリバース・エンジニアリング用の）を設定できます。それはプロシージャタイプの手順またはジャーナル化、リバースまたはチェック手順でのみ使用されます。オプションの詳細は、「オプション」の項を参照してください。

プロパティ	説明
オプションの名前	値が割り当てられるプロシージャ・オプションの名前
値	実行のためにオプションに割り当てられる値

注意: **KM** オプションの値は、**KM** を使用するインタフェースで指定されます。

## 実行パラメータ

パッケージまたは手順の起動時に、実行パラメータを指定する必要があります。この情報はリポジトリに保存されないため、毎回指定する必要があります。

プロパティ	説明
コンテキスト	セッションが起動されるコンテキストを指定します。権限を付与されているコンテキストのみがここに表示されます。
エージェント	パッケージまたは手順を実行するエージェント。「ローカル(エージェントなし)」を選択すると、パッケージまたは手順が直接実行されます。
ログ・レベル	維持するロギング情報のレベル。この値以下のログ・レベルを定義されているすべてのセッション・タスクが、セッション完了時にログに維持されます。ただし、パッケージまたは手順が異常終了した場合は、この設定値にかかわらず、すべてのタスクが維持されます。

## 例

### 繰返しループの定義

10 回繰返すループを作成するには、増分を含む数値変数 `snp_increment` を作成してから、次の 3 つの手順を挿入します。

- 手順 1 (初期化ループ) : `snp_increment` に「0」を割り当てる **変数設定** タイプの手順。
- 手順 2 (増分ループ) : `snp_increment` を「1」だけ増分する **変数設定** タイプの手順。
- 手順 3~n: ループ内で n 回実行するアクション。`snp_increment` 変数は、次のプロシージャで使用できます。
- 手順 n+1 (ループ終了テスト) : `snp_increment <= 10` をテストする **変数評価** タイプの手順。成功した場合に **実行するタスク**: 手順 2。失敗した場合に **実行するタスク**: <<next step>>
- 手順 n+2: .... 次のアクション。

## シナリオ

### シナリオ

シナリオは、ソース・コンポーネント（インタフェース、パッケージ、プロシージャ、変数）を実使用に投入するように設計されます。シナリオは、このコンポーネント用のコード（SQL、シェルなど）生成の結果として生じます。

**注意:** 生成されたシナリオは作業リポジトリに格納されます。シナリオは、エクスポートして別のリポジトリ（リモートまたは非リモート）にインポートし、異なるコンテキストで使用することができます。

**注意:** シナリオ・コード（生成された言語）は変更不能で、シナリオの作成に関係したコンポーネントがそれ以降に変更されても、シナリオは変更されません。

シナリオのプロパティは**デザイナ**・モジュールの**シナリオ**・ウィンドウの「**定義**」タブに表示されるか、**オペレータ**・モジュールを介して表示されます。シナリオの操作は、同一のウィンドウまたはオペレーティング・システムのコマンド・ウィンドウで行われます。デザイナ・モジュールでは、シナリオはそれぞれのソース・コンポーネントの配下にグループ化されます。

### プロパティ

プロパティ	説明
パッケージ、インタフェース、変数、プロシージャ	シナリオを作成するために使用されるコンポーネントの名前。
名前	シナリオの名前。
バージョン	シナリオのバージョン。
説明	シナリオの詳細な説明。

### 実行

このタブでは、「**実行**」ボタンを使用してシナリオを起動することができます。実行オプションを選択するウィンドウが開きます。

### 変数

このタブを使用すると、シナリオ変数を選択できます。シナリオ変数は、**Metadata Navigator** を使用してシナリオを起動する場合やスケジュールを作成する場合にパラメータとして表示されます。ここでパラメータ変数を選択すると、パラメータ化できないユーザー変数を非表示にすることができます。

**すべてを使用**値を選択すると、すべての変数がパラメータとみなされます。**選択的使用**値を選択すると、パラメータとして使用する変数を選択することができます。

### シナリオの操作

シナリオは、それが格納される作業リポジトリから運用されます。同一の作業リポジトリにある同一のシナリオを、いくつかのコンテキストで実行することができます。

シナリオでは次の操作が可能です。

- デザイナまたはオペレータから起動できます。

- Data Integrator のスケジューラでスケジュールできます。
- オペレーティング・システムのコマンド・ライン・インタフェースから、Web サービスまたは HTTP URL を使用して起動できます。
- 任意のスケジューラを使用してスケジュールできます。
- 別の作業リポジトリにエクスポートして、類似の方法で起動またはスケジュールできます。「別のリポジトリでの操作」を参照してください。
- シナリオを実使用でインポートできます。

## スケジュール

「Scheduler」オプションがあると、シナリオの実行をスケジュールできます。

「Scheduler」オプションを使用すると、エージェントは事前に定義されたスケジューラに応じてシナリオを自動的にアクティブ化します。

スケジュールが関係するシナリオは1つだけですが、1つのシナリオを複数の方法でスケジュールしたり、複数のスケジュールを設定することができます。各スケジュールで、開始日付と繰り返しサイクルを指定できます。

例:

- スケジュール 1: 毎週木曜日午後 9 時、1 回のみ
- スケジュール 2: 毎日午前 8 時～12 時、5 秒ごとに繰り返し
- スケジュール 3: 毎日午後 2 時～6 時、5 秒ごとに繰り返し、最大サイクル期間 5 時間

スケジュールは、シナリオのコンテキスト・メニューにある「スケジュールの挿入」オプションを使用して作成します。

### 一般パラメータ

プロパティ	説明
コンテキスト	スケジュールされたシナリオの実行コンテキスト。
エージェント	シナリオの実行を担当するエージェント。
ログ・レベル	ロギング・レベル。この値以下のロギング・レベルが設定されたすべてのタスクは、実行の最後にログに保存されます。
状態	スケジュールのアクティブ化を定義します。
アクティブ	スケジューリングは、エージェントが再起動されるか、物理エージェントのスケジューリングが更新された場合にアクティブになります。アクティブなスケジュールは、ツリー表示に次のアイコン  で表示されます。
非アクティブ	スケジュールをアクティブにできません。非アクティブなスケジュールは、ツリー表示に次のアイコン  で表示されます。
アクティブな期間	スケジュールのアクティビティの範囲。ある期間にアクティブなスケジュールは、ツリー表示にアイコン  で表示されます。

実行	それぞれの実行サイクルの実行頻度を定義します。
実行	実行オプションの頻度（年次、月次...シンプル） このオプションは、メイン・オプションに依存する1組のオプションにより入力されます。

## 実行サイクル

実行サイクルはシナリオの繰り返しモードを示します。

プロパティ	説明
1回	シナリオは1回のみ実行されます。
複数回	シナリオは数回繰り返されます。
最大反復数	サイクルの間にシナリオが繰り返される回数の最大値。
最大サイクル継続時間	最大回数に到達すると、シナリオはそれ以降再開されず、サイクルが停止します。
反復間隔	シナリオの各回の実行の間のダウンタイム。
制約	<b>実行時に問題が発生した場合に、サイクルに対して適用される限界を設定できません。</b>
失敗時の試行回数	連続してエラーを返す実行回数の最大値。
実行停止期限	最大サイクル・タイム。この回数に到達すると、シナリオは自動的に停止します。

**注意:** エージェント・スケジューラが予期せず停止した後、進行中（実行サイクルの途中）であって再起動できるスケジューラは、エージェントが再起動する際に自動的に回復されます。

## 変数

シナリオのスケジューラを作成する場合、スケジューラされた実行のためにパラメータ変数がとる値を定義することができます。

プロパティ	説明
名前	変数の名前。変数名には、グローバル変数の場合 GLOBAL、そうでない場合は変数を含むプロジェクトのコードを接頭辞として付ける必要があります。
デフォルト	このボックスを選択すると、変数は前回の値（一度も設定されることがない場合はデフォルト値）をとります。選択しない場合、変数は値フィールドに示した値をとります。
値	実行のために変数がとる値。

## シナリオ・レポート

シナリオ実行レポートは、シナリオの一般的な実行に関する情報を提供します。

### 定義

プロパティ	説明
実行エージェント	シナリオの実行を担当するエージェント。
状態	シナリオの状態（完了、エラー、実行中、待機中、警告）。
実行コンテキスト	シナリオの実行コンテキスト。
開始	シナリオ実行開始の日付と時刻。
終了	シナリオ実行終了の日付と時刻。
継続時間	シナリオ実行の所要時間。
リターン・コード	シナリオのリターン・コード。
メッセージ	シナリオ実行エラー・メッセージ（必要な場合）。

### 詳細

プロパティ	説明
挿入数	シナリオ実行時に挿入された行数。
削除数	シナリオ実行時に削除された行数。
更新数	シナリオ実行時に更新された行数。
エラー数	シナリオでエラーになった行数。
行数	シナリオによって処理された行の総数。注意: この値は前述の値の合計ではありません。

## 手順レポート

手順実行レポートは、シナリオ実行レポートにリンクされています。

## 定義

プロパティ	説明
名前	実行された手順の名前。
順序番号	このセッションの手順にリンクされた順序番号。

## 実行

プロパティ	説明
実行数	手順（ループ）の実行が成功した際の実行数。
<b>実行</b>	<b>手順実行の詳細な結果を提供します。</b>
開始	手順実行開始の日付と時刻。
終了	手順実行終了の日付と時刻。
継続時間	手順実行の所要時間。
状態	手順の状態（完了、エラー、実行中、待機中、警告）。
リターン・コード	手順のリターン・コード。
メッセージ	手順実行エラーメッセージ（必要な場合）。
挿入数	手順実行時に挿入された行数。
削除数	手順実行時に削除された行数。
更新数	手順実行時に更新された行数。
エラー数	手順でエラーになった行数。

## インタフェース

### インタフェースの概要

インタフェースは、1つ以上のソース・データストアからのデータストアまたは一時ターゲット構造のロードを定義する1組のルールで構成されています。

- インタフェースの作成

### インタフェースのコンポーネント

### ターゲット・データストア

**ターゲット・データストア**は、インタフェースによってロードされる要素です。永続データストア（モデル内で定義される）と、一時データストア（インタフェースによってステージング領域に作成される）があります。

### ソース・データストア

**ソース・データストア**には、ターゲット・データストアをロードするために使用されるデータが含まれます。モデルからのデータストアと、インタフェースの一時データストア・ターゲットの、2タイプのデータストアをインタフェースのソースとして使用することができます。

インタフェースのソース・データストアは、ロード処理時にフィルタでき、結合を使用してリレーション内に置かれる必要があります。**結合**と**フィルタ**はモデル定義から回復でき、またインタフェースで定義することもできます。

### マッピング

**マッピング**はソースにあるトランスフォーメーション・ルールで、ターゲットをロードするためのデータの生成を可能にします。

### フロー

**フロー**は、マップされるデータのロードと統合に関する1組の戦略で、**ナレッジ・モジュール**上にあります。

### コントロール戦略

**フロー・コントロール戦略**では、ターゲットに挿入する前にフローをチェックするために使用されるメソッドを定義できます。コントロール戦略は**チェック・ナレッジ・モジュール**（CKM）によって定義されます。

インタフェースは次のコンポーネントを使用します。

- モデル内で、ソースとターゲット、またはロード処理として定義された**データストア**。
- 適切なプロセスを生成するための**ナレッジ・モジュール**。
- 式の値またはカウンタを格納するための**変数**と**シーケンス**。
- トランスフォーメーション・ルールのコーディングを容易にする**ユーザー・ファンクション**。

## 「定義」タブ

このタブでは、インタフェースの一般的なプロパティを定義します。

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるインタフェースの名前。
最適化コンテキスト	これは、インタフェースの設計時に、ロードするのに必要なデータの決定、式の構文のチェック、およびデータの表示にデザイナーが使用するコンテキストです。 インタフェースの実行コンテキスト（インタフェースの起動時に選択）は、最適化コンテキストと異なっていてもかまいません。ただし、異なるコンテキストを使用して参照される物理アーキテクチャの間に互換性がある必要があります。
ターゲットと異なるステージング領域	一般的なルールとして、ターゲット・データ・サーバーに結合とトランスフォーメーションの機能（もともとのSQLの機能）がある場合、トランスフォーメーションにはこのデータ・サーバーのエンジンを使用することをお勧めします。

一時オブジェクトは、**ステージング領域**と呼ばれるこのサーバー上のスキーマに作成されます。一時ターゲット表を持つインタフェースの場合、一時表はこの領域に作成されます。

そうでない場合（たとえばターゲットが **MOM** またはファイルの場合）は別のトランスフォーメーション・エンジンを選択できます。このエンジンは、別のデータ・サーバーの場合も、メモリー内で動作する内部エンジンの場合もあります。

**注意:** データ・フローの整合性を自動的にチェックするには、ターゲット上のトランスフォーメーション・エンジンを選択する必要があります。

説明 インタフェースの詳細な説明。

## 「ダイアグラム」タブ

### 「ダイアグラム」タブ

ソース、ターゲット、およびトランスフォーメーションとマッピングのルールは「**インタフェース**」ウィンドウの「**ダイアグラム**」タブで定義されます。

このタブを使用して、次のものを定義できます。

- エンティティ・リレーション・ダイアグラム形式のソース・データ・モデル（SQL 問合せエディタと同様）。ウィンドウ左側の**コンポジション・パネル**に表示されます。データ・モデルはソース・データストア（永続的あるいは一時的）、結合およびフィルタで構成されています。
- ターゲット・データストア。
- マッピング、つまりソースとターゲットのデータの間に対応ルールのセット。

#### ダイアグラムのオプション

ダイアグラム上に表示されるオプションは次のとおりです。

プロパティ	説明
列	列から列への結合を表示します（黒い線）。
データストア	表間の結合を表示します（青い線）。多くの結合がある場合に便利です。
セット	結合の結果生じた集合の構築順序を表す緑の矢印を描画します。結合のプロパティに表示される順序番号を使用すると、集合構築順序を変更できます。この表示はそれに応じて変更されます。（1に近い順序番号を持つ結合が先に構築されます）。
エラー	インタフェース上で検出されたエラーを表示できるようにします。エラーが検出されると、ボタンのテキストが赤字になります。警告がある場合に黄色で表示されることもあります。
プロパティ	結合または表のプロパティを表示します（一番下のパネル）。

### ソース・データストア

インタフェースは、次の2つのタイプのデータストアをソースとして使用することができます。

- モデルから生じる永続的データストア
- 別のインタフェースによって作成された一時データストア

エイリアスが異なっていれば、同一のデータストアをソースとして複数回使用することができます。

ソース・データストアのプロパティを編集するには、ダイアグラム内でそれを表しているエンティティを1度だけクリックします。コンポジション・パネルの最下部にプロパティが表示されます。

プロパティ	説明
ジャーナル化されたデータのみ	データのジャーナル化された（変更された）行のみをこのインタフェースのソースにすることを示します。  このボックスを選択すると、使用する行を定義するためのフィルタがソース・データストア上に自動的に作成されます。
別名	エイリアスを使用すると、インタフェースに参加しているデータストアに一意の名前を付けることができます。これはすべての式ですべての列名の前に使用されるため、簡潔な名前にしておきます。
順序	これは、プロシージャが生成される際に、SQL 問合せの FROM 句にデータストアが表示される順序です。この順序は順序付き結合（ISO）の場合は無視されます。
コンテキスト	データストアを体系的に検索する必要があるコンテキスト。  デフォルトでは、Oracle Data Integrator は最適化コンテキスト（設計時）と実行コンテキスト（インタフェース実行時）を選択します。コンテキストを選択する必要があるのは、インタフェースがマルチコンテキストである場合だけです。  <b>例:</b> 「Production」コンテキスト表から「Test」コンテキスト表をロードする場合。

#### ビジュアル・インジケータ

ソース・データストア列の一部のプロパティは、ビジュアル・インジケータで表示されます。

インジケータ	説明
	列はこのデータストアの主キーの一部です。
<b>n</b>	列はマッピング、結合またはフィルタで使用されます。文字は一般に列タイプの最初の文字です。 <b>n</b> は数値、 <b>s</b> は文字列、などです。
<b>n</b>	マッピング、結合またはフィルタで使用されていない列。
<b>*</b>	モデル上で定義された必須の列。

#### 結合

インタフェースに複数のソース・データストアが含まれる場合、それらは結合によって相互リンクされる必要があります。ソース・データストア上の結合の数は、ソース・データストアの数から 1 を引いた数未満にはなれません。

結合のプロパティを編集するには、ダイアグラム内でそれを表しているリレーションをクリックします。

プロパティ	説明
アクティブな句	このチェック・ボックスは、インタフェースの実行時に結合を考慮する必要があることを示します。
実装	通常、結合式は SQL 言語で表現されます。順序付き結合 (ISO) の場合、これは ON 句のテキストです。 例: REG.REGION_ID=CIT.REGION_ID
技術的な説明	このフィールドは編集できません。列名のかわりに列の説明を使用する結合のバージョンが表示されます。
ビジネス・ルール	このフィールドを使用すると、業務説明を入力したり、結合に関する注意やコメントを入力したりできます。
実行:	結合を構築しているトランスフォーメーション・エンジン。 <ul style="list-style-type: none"> <li>• <b>ソース:</b> これは、2 つのソース・データストアをホストするデータ・サーバーです。このサーバーは結合機能を持っている必要があります。</li> <li>• <b>ステージング領域:</b> インタフェースの「定義」タブで選択されているトランスフォーメーション・エンジン。デフォルトでは、これはターゲット・エンジンです。</li> </ul>
オペレータ	リレーショナル・モデルに応じて、異なる演算子を使用してデータセットを結合できます。10 行と 20 行の 2 つの表を結合する場合を考えます。 <ul style="list-style-type: none"> <li>• <b>CROSS:</b> 体系的に 200 行 (10*20) を返す積集合。</li> <li>• <b>INNER JOIN:</b> 結合する 2 つの表のうち、JOIN 句に関係している行、つまり 0 と 20 の間の行のみが維持されます。</li> <li>• <b>LEFT/RIGHT OUTER:</b> 外部結合と呼ばれる結合です。結合演算子の左または右にある表の行のみを返します。この場合、10 行または 20 行を返します。</li> <li>• <b>FULL:</b> 両側についての外部結合と同等です。この場合、この演算子は 20 と 30 の間の結果行を返します。</li> </ul> <p>一部のテクノロジーでは、これらすべての演算子が有効であるわけではありません。その場合、結合記号は打消し線の付いた赤字で示されます。</p>
順序付き結合 (ISO)	結合が順序付きの場合 (ISO 構文とも呼ばれます)、このボックスを選択します。
順序番号	これは、順序付き結合の場合に結合が解決される順序を決定します。 たとえば、A と B の間の結合と A と C の間の結合がある場合、A と B が先に結合され、その結果が C と結合されるのか、それとも A と C が先に結合され、その結果が B と結合されるのか知る必要があります。順序番号を使用する

	と、それを決定できます。順序番号が低い結合が先に解決されます。
データストアの計算	このボタンは、Oracle Data Integrator に結合を強制的に再計算させます。結合のコードを評価し、使用されるデータストアを指定します。特定の場合にエラーが発生することがあります（データストアがない場合や権限が無効にされた）。

## フィルタ

インタフェースがソースとみなすデータを制限するために、インタフェースのソース・データストアにフィルタを設定することができます。

フィルタのプロパティを編集するには、ダイアグラムのフィルタ記号をクリックします。

プロパティ	説明
アクティブ・フィルタ	このチェック・ボックスは、インタフェースの実行時にフィルタを考慮する必要があることを示します。
実装	通常、フィルタ式は SQL 言語で表現されます。
技術的な説明	このフィールドは編集できません。列名のかわりに列の説明を使用するフィルタのバージョンが表示されます。
ビジネス・ルール	このフィールドを使用すると、業務説明を入力したり、フィルタに関する注意やコメントを入力したりできます。
実行	<p>フィルタを実行するトランスフォーメーション・エンジン。</p> <ul style="list-style-type: none"> <li> <b>ソース:</b> これは、ソース・データストアをホストするデータ・サーバーです。このサーバーは、フィルタ操作を実行するのに十分な能力がある必要があります。         </li> <li> <b>ステージング領域:</b> インタフェースの「定義」タブで選択されているトランスフォーメーション・エンジン。デフォルトでは、これはターゲット・エンジンです。         </li> </ul>

## ターゲット・データストア

インタフェースが持てるターゲット・データストアは1つだけです。このデータストアには2つの可能性があります。

- モデル内にすでに存在するデータストアに対応する**永続的データストア**。
- まだモデルに存在せず、インタフェースの「定義」タブで指定された**ステージング領域**の論理スキーマの作業スキーマまたはデータ・スキーマに、インタフェースによって動的に作成される**一時データストア**。

各列にマッピングがあるターゲット・データストアは「**ダイアグラム**」タブの右側に表示されません。

ターゲット・データストアのプロパティを編集するには、データストアのタイトルをクリックします。プロパティ・パネルがウィンドウの一番下に表示されます。

プロパティ	説明
名前	ターゲット表の名前。このフィールドは、一時ターゲット・データストアについてのみ変更できます。
更新キー	更新する行を特定する列のセット。主に次の2つの状況で使用されます。 <ul style="list-style-type: none"> <li>更新キーは、IKMの増分更新で、ソース・データストアの各行に対して、更新するターゲット・データストアの行を特定します。</li> <li>フロー・コントロールでは、エラー表内の行と照合することによって、フローから削除する、誤っている行を特定します。</li> </ul> このフィールドは、永続的データストアでのみ表示されます。カスタムの更新キーを指定するには、<Undefined>を選択してから、各列に対応する「キー」チェック・ボックスを選択します。
一時データストアの場所	このフィールドは、一時データストアでのみ表示されます。一時データストアは常にステージング領域に作成されます。正確に言えば、この一時データストアに対して指定されたコンテキストで、ステージング領域に対応している物理スキーマ上に作成されます。 <ul style="list-style-type: none"> <li><b>データ・スキーマ</b>: 一時データストアはこの物理スキーマの通常（データ）スキーマに作成されます。</li> <li><b>「作業スキーマ」</b>: 一時データストアはこの物理スキーマの作業スキーマに作成されます。</li> </ul>
コンテキスト	指定すると、データストアの実行コンテキストをオーバーライドします。つまり、データストアが検索されるか（永続的データストア）、作成される（一時データストア）コンテキストを指定します。 <p>デフォルトでは、Oracle Data Integrator は最適化コンテキスト（設計時）と実行コンテキスト（インタフェース実行時）を選択します。</p>

#### ビジュアル・インジケータ

ターゲット・データストア列の一部のプロパティは、ビジュアル・インジケータで表示されます。

インジケータ	説明
	列は更新キーの一部です。
n	データ型: 文字は一般に列タイプの最初の文字です。n は数値、s は文字列、などです。
	ソース上でマッピングが実行されました。
	ステージング領域上でマッピングが実行されました。
	ターゲット上でマッピングが実行されました。
	マッピングに関連付けられた警告。問題を参照するにはマウス・カーソルを警告

	に重ねます。
	マッピングに関連付けられたエラー。エラーのあるインタフェースを実行することはできません。問題を参照するにはマウス・カーソルをエラーに重ねます。
X	モデルで定義された必須列。ただし <b>Not Null</b> チェックは設定されていません。
+	モデルで定義されていない必須列。ただし <b>Not Null</b> チェックは設定されていません。
*	モデルで定義された必須列で、 <b>Not Null</b> チェックは設定されていません。

## マッピング

マッピングは、ターゲット・データストア列と、それをソース・データストア列にリンクするトランスフォーメーション・ルールのセットです。

ターゲット列とそのマッピングのプロパティを編集するには、ターゲット・データストアでターゲット列名（「ダイアグラム」タブの右側）をクリックします。列とマッピングのプロパティが合成パネルの一番下に表示されます。

プロパティ	説明
アクティブ・マッピング	このチェック・ボックスは、インタフェースの実行時にマッピングを考慮する必要があることを示します。
実装	マッピングのテキスト。これは、マップされる列に挿入する値を決定するトランスフォーメーション・ルールです。通常、SQL 言語で書かれます。コンポジション・パネルのエンティティ・リレーション・モデルの列を、このフィールドに直接追加することができます。 このフィールドから、式エディタを呼び出すこともできます。
技術的な説明	このフィールドは編集できません。列名のかわりに列の説明を使用するマッピングのバージョンが表示されます。
ビジネス・ルール	このフィールドを使用すると、業務説明を入力したり、マッピングに関する注意やコメントを入力したりできます。
実行	変換を行うトランスフォーメーション・エンジン。このエンジンには、これらのトランスフォーメーションを実行するのに十分な機能（関数、SQL など）がある必要があります。 <ul style="list-style-type: none"> <li> <b>ソース:</b> これは、ソース・データストア（複数可）をホストするデータ・サーバーです。複数のソース上で実行される可能性があるトランスフォーメーションのソース実行の場合、コンボ・ボックスを使用してマッピングが実行されるソースを選択することができます。 たとえば、2つの Oracle サーバーにある2つのソース表を使用して、SYSDATE マッピングを入力すると、ソース上の実行時に、どちらかの Oracle サーバーの日付を使用するかどうか指定できます。 </li> <li> <b>ステージング領域:</b> インタフェースの「定義」タブで選択されているトランスフォーメーション・エンジン。デフォルトでは、これはターゲット・エン </li> </ul>

	<p>ジンです。</p> <ul style="list-style-type: none"> <li>● <b>ターゲット</b>: これはターゲット・データストアをホストするデータ・サーバーです。ターゲット上のマッピングはソース列を参照できず、集約関数を含むこともできません。</li> </ul>
更新	<p>ナレッジ・モジュールに対して、挿入および更新オプションとユーザー固有のオプション（ユーザー定義）を定義します。</p> <ul style="list-style-type: none"> <li>● 挿入: このマッピング列がターゲット表での挿入操作に関与することを示します。</li> <li>● 更新: このマッピング列がターゲット表での更新操作に関与することを示します。</li> <li>● UD1 - UD5: このマッピング列が、ユーザー定義 1~5 にリンクされている操作に関与していることを示します。詳細は、使用する KM を参照してください。</li> </ul>
ターゲット列	この項ではターゲット列を説明します（一時ターゲット・データストアのみ）。
名前	ターゲット列の名前（一時ターゲット・データストアでのみ変更可能）。
データ型	ターゲット列のタイプ（一時ターゲット・データストアでのみ変更可能）。
長さ	ターゲット列の長さ（一時ターゲット・データストアでのみ変更可能）。
スケール	ターゲット列の小数点以下の桁数（一時ターゲット・データストアでのみ変更可能）。
キー	列がインタフェースの更新キーに関与していることを示します。データストアが永続的である場合、ターゲット・データストアのプロパティ・レベルでは更新キーを未選択のままにしておく必要があります。
Not Null チェック	フロー・コントロール時に、この列のすべての非 NULL 値を検証する必要があることを示します。

## 「フロー」タブ

インタフェースの「フロー」タブでは、インタフェースからコードを生成するために使用するナレッジ・モジュールを指定します。

「フロー」タブには、処理が行われる様々なスペース（ソース、ステージング領域およびターゲット）、含まれるデータストア、およびそこに作成される一時表が表示されます。

これらのスペースは、情報システムの物理サーバーを表します。

KM を指定する必要があるスペースは、左上隅に赤い十字が表示されます。

物理サーバーを表すウィンドウのタイトルをクリックすることによって、適切な KM を指定することができます。

## LKM のプロパティ

プロパティ	説明
名前	選択されたソース・スペースの名前。
ロード KM	選択されたスペースから、ステージング領域またはターゲット・スペースにデータを抽出するために使用されるロード・ナレッジ・モジュール (LKM)。どちらのテクノロジーにも適切なプロジェクトにインポートされた LKM からのみ選択できます。
オプション	LKM オプションと、表内での名前および値のリスト。オプションには、KM 作成時に指定されたデフォルト値があります。
LKM の説明	KM 開発者によって書かれた KM の詳細な説明。

### IKM のプロパティ

プロパティ	説明
重複なしの行	ターゲットに挿入する前に、統合により重複の可能性が排除されることを示すことができます。
IKM	統合ナレッジ・モジュール (IKM) は、選択されたスペースにデータを統合します。プロジェクトにインポートされ、このテクノロジーに適切な IKM のみ選択できます。
オプション	LKM オプションと、表内での名前および値のリスト。オプションには、KM 作成時に指定されたデフォルト値があります。
IKM の説明	KM 開発者によって書かれた KM の詳細な説明。

### 「制御」タブ

インタフェース・フロー・コントロール戦略は、「**制御**」タブで定義されます。この戦略は CKM (チェック・ナレッジ・モジュール) に集中します。

**注意:** インタフェースのフロー・コントロールを実行できない場合 (ターゲットが一時データストアの場合など)、このタブには最大許容エラー数と関係があるフィールドのみが含まれます。

プロパティ	説明
許容されているエラーの最大数	このオプションは、ソース・ファイル・データストアからデータを抽出する際に検出されたエラーと、フロー・コントロール処理によって検出されたエラーにのみ適用されます。エラーがこの数または割合を超えると、インタフェースはエラー状態になります。  「%」ボックスを選択しない場合、次のいずれかの条件が真だと、インタフェースはエラーになります。 <ul style="list-style-type: none"> <li>1つのファイルからデータを抽出する際に検出されたエラーの数が、許容エラ</li> </ul>

一数を超えています。

- フロー・コントロール時に検出されたエラーの数が、許容エラー数を超えています。

「%」ボックスを選択した場合、次のいずれかの条件が真だと、インタフェースはエラーになります。

- 1つのファイルからデータを抽出する際に検出されたエラーの数が、許容エラー率を超えています。割合は、ファイルから抽出された行数から計算されます。
- フロー・コントロール時に検出されたエラーの数が、許容エラー割合を超えています。この割合は、ターゲット・データストアに統合されたレコードの件数から計算されます（挿入および更新）。

**注意:** このフィールドが空のままの場合、許容エラー数は無限です。

**注意:** エラーの総数がインタフェースの行数を超えることがあります。これは、1行が複数の制約を侵害したり、複数のエラーを含んだりすることがあるためです。この場合、エラーは合計されます。

CKM	<b>フロー・コントロール戦略を定義します。</b>
CKM	データをチェックするために使用されるチェック・ナレッジ・モジュール (CKM)。チェックが行われるテクノロジ (ステージング領域) に適切なプロジェクト CKM のみが選択できます。
オプション	CKM オプションと、表内での名前および値のリスト。
CKM の説明	KM 開発者によって書かれた KM の詳細な説明。
制約	<b>チェックする制約を表示します。</b>
制約の名前	関係している制約の名前。
値	指定された制約がチェック対象かどうかを示します (yes/no)。

## 実行パラメータ

インタフェースの起動時に、実行パラメータを指定する必要があります。この情報はリポジトリに保存されないため、毎回指定する必要があります。

プロパティ	説明
コンテキスト	セッションが起動されるコンテキストを指定します。権限を付与されているコンテキストのみがここに表示されます。
エージェント	インタフェースを実行するエージェント。「ローカル(エージェントなし)」を選択すると、インタフェースが直接実行されるようになります。
ログ・レベル	維持するロギング情報のレベル。この値以下のログ・レベルを定義されているすべてのセッション・タスクが、セッション完了時にログに維持されます。ただし、インタフェースが異常終了した場合は、この設定値にかかわらず、すべての

タスクが維持されます。

## 「シナリオ」タブ

インタフェース上のこのタブには、このコンポーネント用に生成されたシナリオが、スケジュールとともに、表形式で示されます。プロパティを表示するには、シナリオまたはスケジュールをダブルクリックします。

シナリオの作成および削除は、 「生成」 および  「削除」 ボタンをクリックすることによって行うことができます。

このタブでは、シナリオおよびスケジュールのポップアップ・メニューも使用できます。

## プロシージャ

### プロシージャ

プロシージャとは、エージェントが実行できるコマンドのセットです。これらのコマンドは、Oracle Data Integrator でアクセスできるすべてのテクノロジーに関係します (OS、JDBC、JMS コマンドなど)。

プロシージャのプロパティは、**デザイナー**・モジュールの**プロシージャ**・ウィンドウの「定義」タブに表示されます。テストの実行も同じウィンドウから実行されます。

プロシージャを暗号化して、貴重な開発物をユーザーが読めないように保護することができます。

プロシージャはプロジェクト内のフォルダにあります。別のプロジェクトからはアクセスできません。

### 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるプロシージャの名前。
説明	プロシージャの詳細な説明。
複数接続	プロシージャが、ペアになるデータサーバーにアクセスするかどうかを示します。このボックスを選択すると、必要に応じて、テクノロジーの異なる2つのデータサーバーの間でデータ・ロード・コマンドを交換することが可能になります。たとえば、プロシージャ・コマンド上で、片方のデータベース上でコマンド <b>Select</b> を、もう一方のデータベース上でコマンド <b>Insert</b> を指定して、結果全体を転送することが可能です。
ソース・テクノロジー	プロシージャ・コマンドでデフォルトで使用されるソース・テクノロジー。この情報を使用できるのは、プロシージャがマルチ接続タイプの場合だけです。
ターゲット・テクノ	このプロシージャのコマンドの対象になるデフォルトのテクノロジー。

## ロジ

## プロシージャの詳細

プロシージャはいくつかのコマンドで構成されています。「詳細」タブにはそのコマンドのリストが表示されます。このリストは、グリッドの横や下のボタンを使用して、作成、削除および配置することができます。既存のコマンド行を編集するには、編集する行をダブルクリックします。

## オプション

オプションを使用すると、実行時にプロシージャをパラメータ化することができます。「オプション」タブには、すべてのオブジェクトとそのデフォルト値が表示されます。このタブで指定されるオプション値は、プロシージャの実行時のみ使用されます。

## プロシージャの実行

「実行」タブでは、コンテキストとエージェントを指定してプロシージャを実行するテストを行います。

## シナリオ

このタブには、このコンポーネント用に生成されたシナリオが、スケジュールとともに、表形式で示されます。プロパティを表示するには、シナリオまたはスケジュールをダブルクリックします。

シナリオの作成および削除は、 「生成」 および  「削除」 ボタンをクリックすることによって行うことができます。

このタブでは、シナリオおよびスケジュールのポップアップ・メニューも使用できます。

## プロシージャ・コマンド

プロシージャ・コマンドは、プロシージャの実行単位です。

## 定義

このタブを使用すると、コマンド行を定義できます。

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるコマンドの名前。
ログ・カウンタ	このコマンドによって処理される行の数をどのカウンタ（Insert、Update、Delete または Errors）が記録するかを示します。
ログ・レベル	コマンドの重要性レベル。実行ログの参照はこのレベルに応じて行われます。
エラーの無視	無効なリターン・コードの場合にもプロシージャを中断しないことを示します。このボックスを選択すると、プロシージャ・コマンドは error でなく warning に数えられ、プロシージャが停止しません。

コマンド	データ・サーバー上で起動されたコマンド。類似した2つのタブ（「コマンド・ソース」および「コマンド宛先」）は、マルチ接続プロシージャの場合に表示されます。
テクノロジー	<p>コマンドが実行されるテクノロジー。</p> <p><b>注意:</b> 1 つまたは複数の KM プロシージャ・コマンドで Oracle Data Integrator Tools コマンドを使用するには、テクノロジーを Sunopsis API に設定する必要があります。</p> <p>OS コマンドを使用したり外部プログラムを実行したりするには、オペレーティング・システムにテクノロジーを設定する必要があります。</p>
トランザクション分離	<p>Select 命令のトランザクション分離レベル。表示される分離レベルは SQL ISO のものです。すべてのデータ・サーバーによってサポートされているわけではありません。</p> <p>分離レベルは次のとおりです。</p> <ul style="list-style-type: none"> <li>● デフォルト: トランザクションは接続の分離レベルを保持します。</li> <li>● 非コミット読取り: トランザクションは、別のトランザクションがコミットしていないデータを読み取れます。</li> <li>● コミット読取り: トランザクションは、他のトランザクションがコミットしたデータのみを読み取れます（一般には、これが多くのデータ・サーバーのデフォルトのモードです）。</li> <li>● 反復可能読取り: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更およびコミットされていても、必ず同一の情報を読み取ります。</li> <li>● シリアル化可能: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更、削除、作成およびコミットされていても、必ず同一の情報を読み取ります。</li> </ul>
コンテキスト	問合せの実行コンテキスト。「実行コンテキスト」になっている場合、コマンドは、起動時またはその手順で選択された実行コンテキストで実行されます。
スキーマ	実行の論理スキーマ。コンテキストにリンクされた論理スキーマを使用すると、物理実行スキーマを定義できます。
トランザクション	複数の同時トランザクション（0 から 9 まで番号が付けられます）上でコマンドを実行したり、オプション「自動コミット」を選択しておき、トランザクション外で作業したりすることができます。後者の方が効率的です。
コミット	<p>トランザクション（0 から 9 までのいずれかの番号が付けられています）でコマンドが実行される場合に、継続するか現在のトランザクションをコミットするかを、次のモードに応じて決定できます。</p> <ul style="list-style-type: none"> <li>● コミットなし: トランザクションはコミットされません。この場合、後のコマンドでコミットすることができます。セッションが正常終了すると、デフォルトで、すべてのトランザクションがコミットされます。</li> <li>● コミット: トランザクションはコミットされます。</li> <li>● 1000 行をコミット: 1000 件のレコードを処理するたびに、Oracle Data</li> </ul>

**Integrator** によってコミットされます。これを選択できるのは、ローディング・プロシージャ・コマンド、つまり結果セットを返すソース・コマンドと、**Insert** を実行する宛先コマンドが含まれているコマンドに対してだけです。

**コマンド** ネイティブ言語または標準言語 (SQL、PL/SQL、Transact-SQL、シェルなど) で表現されたコマンドのテキスト。テキスト右側のボタンを使用すると、式エディタにアクセスして使用することができます。

**警告:** コマンドがランダムな文字から作成されているように表示されている場合、プロシージャが暗号化されています。暗号化と復号化の詳細は、KM やプロシージャの暗号化を参照してください。また、プロシージャの提供者にも問い合せてください。

## オプション

このタブを使用すると、プロシージャ・オプションの値に応じて、プロシージャ・コマンドを実行するかどうかを指定できます。

プロパティ	説明
常に実行	プロシージャ・コマンドはオプション値にかかわらず実行されます。
オプション	このチャートにはすべてのオプションが表示されます。各オプションの前にあるチェック・ボックスは、(チェック・ボックス・タイプの) オプションの値が「yes」である場合にコマンドが実行されることを示しています。

## オプション

オプションはプロシージャまたはナレッジ・モジュール (KM) にアタッチされていて、2つの機能があります。

- 使用時に指定される、プロシージャまたは KM の内部のテキスト変数 (**text** 型または **value** 型) を定義します。
- プロシージャまたは KM のタスクを実行するかどうかを決定する、**checkbox** 型のオプションを定義します。

たとえば KM の場合、オプションは、「UPDATE」 (レコード更新に関連するアクションを考慮するかどうかを決定)、「TRUNCATE」 (ターゲット・データをアンロードするかどうか)、「INSERT」 (データをターゲットに挿入するかどうか) のいずれかです。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるオプションの名前。
タイプ	オプションのタイプ。次の3つのタイプがあります。 <ul style="list-style-type: none"> <li>• <b>Check Box:</b> オプションはブール型 (Yes = 1/No = 0) です。これはプロシ</li> </ul>

	<p>ージャと KM でのみ使用され、そのタスクを実行するかどうかを決定するオプションです。</p> <ul style="list-style-type: none"> <li>• <b>Default Value:</b> 英数字型のオプションです。最大 250 文字です。</li> <li>• <b>Text:</b> 英数字型のオプションです。サイズは無制限です。このタイプのオプションへのアクセスは、値型のオプションより低速です。</li> </ul>
説明	オプションの概略。Check Box オプションの場合はコマンド・ウィンドウに表示されます。ここで、コマンドの実行をトリガーするオプションを選択します。
位置	プロシージャまたは KM のオプション・リストが表示される場合、オプションの表示順を決定します。
ヘルプ	オプションに関する使用上の説明。KM の場合、インタフェースで KM を選択すると、プロパティ・パネルに表示されます。
デフォルト値	値がプロシージャまたは KM のユーザーによって指定されなかった場合にオプションがとる値。

## 変数

変数は、任意の式 (SQL またはそれ以外)、およびリポジトリのメタデータ内で使用できます。変数は、それが含まれているコマンドがエージェントまたはグラフィカル・インタフェースによって実行される際に解決されます。

変数のプロパティはデザイナー・モジュールの変数ウィンドウの「定義」タブおよび「リフレッシュ」タブに表示されます。グローバル変数はプロジェクトと同じレベルのツリーにあり、プロジェクト変数はプロジェクト内にあります。

## 定義

プロパティ	説明
名前	使用される形式の変数名。この名前には、変数が使用されるテクノロジーでセパレータ (スペースなど) と解釈される文字は使用できません。変数名では大文字と小文字が区別されます。したがって、「YEAR」と「year」は2つの異なる変数とみなされます。
変数の有効範囲	変数の有効範囲。「Global」と「Project」のいずれか。変数には、すべてのプロジェクトで有効な変数 (グローバル変数) と、現在のプロジェクトのみでは有効な変数があります。Oracle Data Integrator では、先に Project レベルの変数を評価してから Global レベルの変数を評価します。このため、プロジェクト・レベルとグローバル・レベルの両方に変数 YEAR が存在する場合、プロジェクト・レベルの変数のみが評価されます。
説明	変数の詳細な説明。
データ型	変数のタイプ: 英数字 (255 文字)、日付、数値 (最大 10 桁) またはテキスト (長さ無制限)。

アクション	このパラメータは変数の値が保持される時間の長さを示します。 <ul style="list-style-type: none"> <li>● <b>非永続:</b> 変数の値はセッションを通じてメモリーに保持されます。</li> <li>● <b>最後の値:</b> 変数が最後にとった値が Oracle Data Integrator のリポジトリに格納されます。</li> <li>● <b>履歴化:</b> この変数の履歴と、変数がとったすべての値が保持されます。</li> </ul>
デフォルト値	デフォルトで変数に割り当てられた値。

## リフレッシュ

変数をリフレッシュすると、SQL タイプの問合せの結果に基づいて新しい値を変数に割り当てることができます。新しいリフレッシュが実行されるまで、この変数を使用するすべての式が新しい値で実行されます。

プロパティ	説明
スキーマ	SQL 問合せが実行される接続を識別するための論理スキーマ。
問合せの選択	ある行と列について、結果配列を返すことを可能にする Select タイプの SQL 問合せまたは任意のコマンド（ストアド・プロシージャの実行）。 例: <code>Select max(order_no) from crm.order</code>  <b>注意:</b> 接続パラメータは正しい物理スキーマに配置できないので、表名は完全名（カタログ、スキーマなど）で指定する必要があります。
リフレッシュ	上で入力された問合せを実行している間に変数の最新の値を計算するには「リフレッシュ」ボタンをクリックします。コンテキストおよび実行エージェントを入力するウィンドウが表示されます。実行はログでチェックします。  <b>注意:</b> この変数が永続的でない場合、計算された値を表示できません。

## 履歴

このウィンドウには、変数の値の履歴が、log アクションの変数のコンテキスト、または last value アクションの変数の最後の値とともに表示されます。

**注意:** それぞれの値はセッションが実行されたコンテキストにアタッチされています。

-  ボタンを使用すると、履歴を 1 行削除できます。
- 「コンテキスト・フィルタ」フィールドを使用すると、指定されたコンテキストでエントリをフィルタできます。

## シナリオ

このタブには、このコンポーネント用に生成されたシナリオが、スケジュールとともに、表形式で示されます。プロパティを表示するには、シナリオまたはスケジュールをダブルクリックします。

シナリオの作成および削除は、 「生成」 および  「削除」 ボタンをクリックすることによって行うことができます。

このタブでは、シナリオおよびスケジュールのポップアップ・メニューも使用できます。

**注意:** 変数用に生成されたシナリオには、変数のリフレッシュ操作を実行する単一のステップが含まれます。

## シーケンス

シーケンスは、使用されるたびに自分自身を増分する変数です。使用と使用の間、値はリポジトリに格納されるか、外部 RDBMS 表内で管理されます。

Oracle Data Integrator では、2 つのタイプのシーケンスがサポートされています。

- **標準シーケンス。**最後の値がリポジトリに格納されます。
- **固有シーケンス。**最後の値が RDBMS 表のセルに格納されます。Oracle Data Integrator は、値の読取り、行のロック（同時更新時）、最後の増分後の行の更新を行います。

### 一般プロパティ

プロパティ	説明
名前	使用される形式のシーケンス名。この名前には、シーケンスが使用されるテクノロジーでセパレータ（スペースなど）と解釈される文字は使用できません。大文字と小文字は区別されません。 例: ORDER_NO
有効範囲	シーケンスの有効範囲。「Global」と「Project」のいずれか。シーケンスには、すべてのプロジェクトで有効なシーケンス（グローバル・シーケンス）と、現在のプロジェクトのみでは有効なシーケンスがあります。Oracle Data Integrator では、先に Project レベルのシーケンスを評価してから Global レベルのシーケンスを評価します。このため、プロジェクト・レベルとグローバル・レベルの両方にシーケンス ORDER_NO が存在する場合、プロジェクト・レベルのシーケンス値のみが評価されます。
増分	増分の値。正または負です。
<b>標準順序</b>	
位置	標準シーケンスに最後に割り当てられた値。
<b>特定の順序</b>	
スキーマ	シーケンス表が含まれている論理スキーマ。
表	シーケンス値が含まれる表。
列	シーケンス値が含まれているシーケンスの名前。
単一行を取得	後の表に複数の行が含まれる場合、表でのこの行の位置を定めるにはフィルタを指定する必要があります。このフィルタはデータ・サーバーの SQL 構文を取

するフィルタ   り出します。

例: CODE\_TAB = '3'

## ユーザー関数

ユーザー関数を使用すると、カスタマイズされた関数を定義して、インタフェースまたはプロシージャで使用できます。

これらの関数は1つ以上のテクノロジーで実装されます。

### ユーザー関数ウィンドウ

#### 定義

このタブでは、ユーザー関数の一般プロパティを宣言できます。

プロパティ	説明
名前	ユーザー関数の名前。例: NullValue
グループ	ユーザー関数のグループ。存在しないグループ名を入力すると、関数の保存時にそのグループ名で新規グループが作成されます。
構文	<p>式エディタに表示されるユーザー関数の構文。関数の引数はこの構文で指定する必要があります。関数の構文は、次のいずれかです。</p> <pre>function name (\$ (arg name) arg type, \$(arg_name) arg_type, ... ) function_name (\$ (arg_name), \$(arg_name), ... ) function_name () function_name</pre> <p>arg_type は、引数の型を指定するために使用します。必須ではありません。s (文字列)、n (数) または d (日付) のいずれかです。</p> <p>たとえば、関数 NullValue を宣言する場合、次の構文を使用します。</p> <pre>NullValue (\$ (variable), \$(default))</pre>
説明	関数の詳細な説明。

#### 実装

このタブでは、各テクノロジーまたはテクノロジーのグループに対する関数の実装を定義します。

プロパティ	説明
実装	関数が実装されるテクノロジーまたはテクノロジー・グループ。
追加-編集-削除	これらのボタンを使用すると、 <b>実装</b> ウィンドウで、新しい実装を作成したり、既存の実装を変更したりすることができます。

実装のないテクノロジー 関数が実装されていないテクノロジー。  
テクノロジー

## 実装ウィンドウ

このウィンドウは、実装を変更または作成する場合に表示されます。

プロパティ	説明
実装構文	<p>選択されたテクノロジーの言語で記述された関数の構文。関数の引数は、<code>\$(arg_name)</code> という構文で使用されます。</p> <p>たとえば、関数 <code>NullValue \$(variable), \$(default)</code> の2つの実装は次のようになります。</p> <pre>Oracle:nvl \$(variable), \$(default) Microsoft SQL Server:case when \$(variable) is null then \$(default) else \$(variable) end</pre>
関連付けられているテクノロジー	この実装に関連付けられているテクノロジーのリスト。1つのテクノロジーに対して実装は1つのみ可能です。
新しいテクノロジーを自動的に含む	このボックスを選択すると、新しいテクノロジーが自動的にこの実装に追加されます。このボックスを選択できる実装は1つのみです。

## ナレッジ・モジュール

### ナレッジ・モジュール

Oracle Data Integrator は、指定されたテクノロジーに関連するメソッドを定義するためにナレッジ・モジュールを使用します。これらのモジュールでは、専用の機能に特化したテクノロジー用のプロセスを生成できます。

ナレッジ・モジュール (KM) を暗号化して、貴重な開発物をユーザーが読めないように保護することができます。

### LKM - ローディング

LKM (ロード・ナレッジ・モジュール) は、1つのデータ・サーバーから別のデータ・サーバーにデータをロード (または抽出) します。

次のような命名規則に従います。

- **LKM <source server technology> to <target server technology> (<module specificity>)**

**注意:** 一部の KM は、汎用の ISO (SQL または JMS) を使用してデータ・サーバー (たとえばソース、ターゲットまたはステージング領域では) に到達します。ISO テクノロジーで示された KM は、標準をサポートするデータ・サーバーとともに動作します。

**注意:** ナレッジ・モジュールの固有性は、多くの場合、テクノロジーに固有なメソッドまたは機能を示します。これはモジュールによって管理されて高速なデータ処理を可能にし、モジュールで頻繁に使用されます。

## JKM - ジャーナリ化

JKM (ジャーナリ化ナレッジ・モジュール) は、データ・モデル内のデータストアにジャーナリ化を設定します。

次のような命名規則に従います。

- JKM <ジャーナリ化するテクノロジー>

**注意:** ジャーナリ化はトリガーとビューを使用します。このため、これらの関数をサポートしないテクノロジーには、一般に、関連付けられた JKM がありません。

## IKM - 統合

IKM (統合ナレッジ・モジュール) は、ステージング領域からターゲット・データ・サーバーにデータを統合します。通常、ステージング領域とターゲット領域は、同一データ・サーバー上にあります。IKM では、データ挿入 (**追加**) や更新 (**増分更新**) などのオプションを管理できます。

次のような命名規則に従います。

- IKM <ステージング領域データ・サーバーのテクノロジー> to <ターゲット・データ・サーバーのテクノロジー> <モジュール内で管理されるオプション> (<モジュール固有性>)
- IKM <ステージング領域およびターゲットのデータ・サーバーのテクノロジー> <モジュールによって管理されるテクノロジー> (<モジュール固有性>)

**注意:** モジュールによって管理されるオプションは、多くの場合、このナレッジ・モジュールでサポートされている、より複合的なオプションを示します。たとえば、モジュール **Incremental Update** (更新) は、データ挿入のオプション **append** や、ターゲット内のデータの抑止も管理できます。

## CKM - チェック

CKM (チェック・ナレッジ・モジュール) は、静的データおよびデータ・フロー上の、データの品質とクレンジングを管理します。

次のような命名規則に従います。

- CKM <チェックされるテクノロジー> (<モジュール固有性>)

## RKM - リバース・エンジニアリング

RKM (リバース・ナレッジ・モジュール) は、指定されたテクノロジーに格納されたデータ・モデルをリバース・エンジニアリングします。デフォルトでは、**標準リバース**を使用すると、大部分のモデルで、すべての情報とメタデータをリバースできます。一部のテクノロジー (ファイルなど) では、**パーソナライズされたリバース**を使用する必要があります。そのため RKM も使用します。それらのテクノロジーには特別の名前が付けられます。

## SKM - サービス

SKM（サービス・ナレッジ・モジュール）は、データ・サービスを作成するために必要なコードを生成するために使用されます。このコード（通常は Java）は、自動的にコンパイルして、Web サービス・コンテナにデプロイすることができます。

## KM ウィンドウ

### 「定義」タブ

IKM、RKM、JKM と CKM では、「定義」タブには次のフィールドが表示されます。

プロパティ	説明
名前	グラフィカル・インタフェースに表示される KM の名前。
タイプ	KM のタイプ。
一貫性セット・ジャーナル化、単純ジャーナル化	このフィールドはこのナレッジ・モジュールがどのジャーナル化モードを管理するかを定義するもので、ジャーナル化 KM でのみ表示されます。
説明	KM の詳細な説明。
このテクノロジーのペアのデフォルト KM	「ソース・テクノロジー」と「ターゲット・テクノロジー」に表示されているテクノロジーを使用したデータ・サーバーについては、この KM がデフォルトとして選択されることを示します。
複数接続	KM が、ペアになるデータ・サーバーへのアクセス権を持っているかどうかを示します。このボックスを選択すると、必要に応じて、テクノロジーの異なる 2 つのデータ・サーバーの間でデータ・ロード・コマンドを交換することが可能になります。たとえば、プロシージャ・コマンド上で、片方のデータベース上でコマンド <code>Select</code> を、もう一方のデータベース上でコマンド <code>Insert</code> を指定して、結果全体を転送することが可能です。
ソース・テクノロジー	プロシージャ・コマンドでデフォルトで使用されるソース・テクノロジー。この情報を使用できるのは、プロシージャがマルチ接続タイプの場合だけです。
ターゲット・テクノロジー	このプロシージャのコマンドの対象になるデフォルトのテクノロジー。

SKM では、タブには次のフィールドが表示されます。

プロパティ	説明
名前	グラフィカル・インタフェースに表示される KM の名前。
コマンド・テキスト	SKM のコード。
説明	KM の詳細な説明。

SKM の場合、「詳細」タブと「オプション」タブはありません。

## 「詳細」タブ

KM はいくつかのプロシージャ・コマンドで構成されています。「詳細」タブには次のコマンドのリストが表示されます。このリストは、グリッドの横のボタンを使用して、作成、削除および整理することができます。既存のプロシージャ・コマンドを編集するには、編集する行をダブルクリックします。

## 「オプション」タブ

オプションを使用すると、実行時に KM をパラメータ化することができます。「オプション」タブには、すべてのオブジェクトとそのデフォルト値が表示されます。このタブで指定されるオプション値は、KM の実行時のみ使用されます。

## KM コマンド

KM コマンドはナレッジ・モジュールの実行単位です。

### 定義

このタブを使用すると、コマンド行を定義できます。

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるコマンドの名前。
ログ・カウンタ	このコマンドによって処理される行の数をどのカウンタ（Insert、Update、Delete または Errors）が記録するかを示します。
ログ・レベル	コマンドの重要性レベル。実行ログの参照はこのレベルに応じて行われま す。
エラーの無視	無効なリターン・コードの場合にも KM を中断しないことを示します。この ボックスを選択すると、コマンドは error でなく warning に数えられ、KM が停止しません。
ジャーナル化 (IKM)	これらのオプションは IKM コマンドの場合のみ表示され、ジャーナル化さ れたデータストアからデータを処理する際のコマンドの動作を定義します。
ステージング領 域のジャーナル 化された表	ジャーナル化された表がステージング領域にある場合にコマンドを実行しま す。
現行インタフェ ースのジャーナ ル化表ソース	ジャーナル化された表が現在のインタフェースのソースである場合にコマ ンドを実行します。
ジャーナル化 (LKM)	これらのオプションは LKM コマンドの場合のみ表示され、ジャーナル化さ れたデータストアからデータを処理する際のコマンドの動作を定義します。
現在の領域でジ	現在のソース・セットにジャーナル化された表が含まれる場合にコマンドを

ジャーナル化された表	実行します。
<b>ジャーナル化 (JKM)</b>	これらのオプションは JKM コマンドの場合のみ表示されます。ジャーナル化を設定する際のコマンドの動作を定義します。
モデル作成	モデルのジャーナルを起動する際にコマンドを実行します。
モデル削除	モデルのジャーナルを停止する際にコマンドを実行します。
表作成	データストアのジャーナルを起動する際にコマンドを実行します。
表削除	データストアのジャーナルを停止する際にコマンドを実行します。
サブスクリライブ	サブスクリプションを作成する際にコマンドを実行します。
サブスクリライブ解除	サブスクリプションを削除する際にコマンドを実行します。
<b>消費 (JKM)</b>	これらのオプションは JKM コマンドの場合のみ表示されます。ジャーナルから消費する際のコマンドの動作を定義します。
ウィンドウの拡張	CDC セットでウィンドウの拡張操作を実行する際にコマンドを実行します。
ジャーナルのページ	CDC セットでジャーナルのページ操作を実行する際にコマンドを実行します。
サブスクリライブのロック	CDC セットでサブスクリライブのロック操作を実行する際にコマンドを実行します。
サブスクリライブのロック解除	CDC セットでサブスクリライブのロック解除操作を実行する際にコマンドを実行します。
<b>表ごとに反復 (JKM)</b>	このオプション・グループは、データストアのグループ（モデルまたは CDC セット）上で CDC 関連の操作を処理する際に、すべてのデータストアに対してコマンドを反復するかどうかを定義します。
いいえ	コマンドは繰り返されません。
昇順	コマンドは各データストアについて繰り返されます。繰り返しは、CDC セットでのデータストアの <b>順序</b> の値の昇順に実行されます。
降順	コマンドは各データストアについて繰り返されます。繰り返しは、CDC セットでのデータストアの <b>順序</b> の値の降順に実行されます。
<b>チェック (CKM)</b>	これらのオプションは CKM コマンドの場合のみ表示されます。このコマンドが <b>必須で、実行の必要があるアクションのタイプ</b> を示します。
主キー	主キー・チェック用のコマンドを実行します。
代替キー	代替キー・チェック用のコマンドを実行します。

結合	参照または結合のチェック用のコマンドを実行します。
条件	条件チェック用のコマンドを実行します。
必須	必須列 (Not Null) のチェック用のコマンドを実行します。
エラーの削除	エラー削除用のコマンドを実行します。
ロード(LKM)	これらのオプションは LKM コマンドの場合のみ表示されます。このコマンドが必須で、実行の必要がある場合を示します。
統合前(IKM)	統合の前にコマンドを実行します。
統合後(IKM)	統合の後にコマンドを実行します。
コマンド	データ・サーバー上で起動されたコマンド。マルチ接続の KM を扱う場合、類似した 2 つのタブ (「ソース・コマンド」および「宛先コマンド」) が表示されます。
テクノロジー	<p>コマンドが実行されるテクノロジー。</p> <p><b>注意:</b> KM コマンドまたはプロシージャ・コマンドで Oracle Data Integrator ツールを使用するには、テクノロジーを Sunopsis API に設定する必要があります。</p> <p>OS コマンドを使用したり外部プログラムを実行したりするには、オペレーティング・システムにテクノロジーを設定する必要があります。</p>
トランザクション分離	<p>Select 命令のトランザクション分離レベル。表示される分離レベルは SQL ISO のものです。すべてのデータ・サーバーによってサポートされているわけではありません。</p> <p>分離レベルは次のとおりです。</p> <ul style="list-style-type: none"> <li>• 非コミット読取り: トランザクションは、別のトランザクションがコミットしていないデータを読み取れます。</li> <li>• コミット読取り: トランザクションは、他のトランザクションがコミットしたデータのみを読み取れます (一般には、これが多くのデータ・サーバーのデフォルトのモードです)。</li> <li>• 反復可能読取り: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更およびコミットされていても、必ず同一の情報を読み取ります。</li> <li>• シリアル化可能: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更、削除、作成およびコミットされていても、必ず同一の情報を読み取ります。</li> </ul>
コンテキスト	問合せの実行コンテキスト。「実行コンテキスト」になっている場合、コマンドは、起動時またはその手順で選択された実行コンテキストで実行されます。
スキーマ	実行の論理スキーマ。コンテキストにリンクされた論理スキーマを使用すると、物理実行スキーマを定義できます。

トランザクション	複数の同時トランザクション（0 から 9 まで番号が付けられます）上でコマンドを実行したり、オプション「自動コミット」を選択しておき、トランザクション外で作業したりすることができます。後者の方が効率的です。
コミット	トランザクション（0 から 9 までのいずれかの番号が付けられています）でコマンドが実行される場合に、継続するか現在のトランザクションをコミットするかを、次のモードに応じて決定できます。 <ul style="list-style-type: none"> <li>• コミットなし: トランザクションはコミットされません。この場合、後のコマンドでコミットすることができます。セッションが正常終了すると、デフォルトで、すべてのトランザクションがコミットされます。</li> <li>• コミット: トランザクションはコミットされます。</li> <li>• 1000 行をコミット: 1000 件のレコードを処理するたびに、Oracle Data Integrator によってコミットされます。これを選択できるのは、ローディング・コマンド、つまり結果セットを返すソース・コマンドと、Insert を実行する宛先コマンドが含まれているコマンドに対してだけです。</li> </ul>
コマンド	ネイティブ言語または標準言語（SQL、PL/SQL、Transact-SQL、シェルなど）で表現されたコマンドのテキスト。テキスト右側のボタンを使用すると、式エディタにアクセスして使用することができます。 <p><b>警告:</b> コマンドがランダムな文字から作成されているように表示されている場合は、KM が暗号化されています。暗号化と復号化の詳細は、KM やプロセスの暗号化を参照してください。また、KM のプロバイダにも問い合わせてください。</p>

## オプション

このタブを使用すると、KM オプションの値に応じて、コマンドを実行するかどうかを指定できます。

プロパティ	説明
常に実行	コマンドはオプション値にかかわらず実行されます。
オプション	このチャートにはすべての KM オプションが表示されます。各オプションの前にあるチェック・ボックスは、（チェック・ボックス・タイプの）オプションの値が「yes」である場合にコマンドが実行されることを示しています。

## オプション

オプションはプロセスまたはナレッジ・モジュール（KM）にアタッチされていて、2 つの機能があります。

- 使用時に指定される、プロセスまたは KM の内部のテキスト変数（text 型または value 型）を定義します。
- プロセスまたは KM のタスクを実行するかどうかを決定する、checkbox 型のオプションを定義します。

たとえば KM の場合、オプションは、「UPDATE」（レコード更新に関連するアクションを考慮するかどうかを決定）、「TRUNCATE」（ターゲット・データをアンロードするかどうか）、「INSERT」（データをターゲットに挿入するかどうか）のいずれかです。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるオプションの名前。
タイプ	オプションのタイプ。次の 3 つのタイプがあります。 <ul style="list-style-type: none"> <li>• <b>Check Box:</b> オプションはブール型（Yes = 1/No = 0）です。これはプロシージャと KM でのみ使用され、そのタスクを実行するかどうかを決定するオプションです。</li> <li>• <b>Default Value:</b> 英数字型のオプションです。最大 250 文字です。</li> <li>• <b>Text:</b> 英数字型のオプションです。サイズは無制限です。このタイプのオプションへのアクセスは、値型のオプションより低速です。</li> </ul>
説明	オプションの概略。 <b>Check Box</b> オプションの場合はコマンド・ウィンドウに表示されます。ここで、コマンドの実行をトリガーするオプションを選択します。
位置	プロシージャまたは KM のオプション・リストが表示される場合、オプションの表示順を決定します。
ヘルプ	オプションに関する使用上の説明。KM の場合、インタフェースで KM を選択すると、プロパティ・パネルに表示されます。
デフォルト値	値がプロシージャまたは KM のユーザーによって指定されなかった場合にオプションがとる値。

## マーカー・グループ

マーカー・グループは、開発の構成のためにプロジェクトまたはモデルの要素に適用されるフラグのセットです。

## 定義

プロパティ	定義
グループ名	マーカー・グループの名前。この名前はユーザー・インタフェースに表示されます。
グループ・コード	マーカー・グループのコード。このコードは Oracle Data Integrator ツールと API からマーカーにアクセスするために使用されます。
表示位置	ツリー表示内のマークが付いたオブジェクトに対するマーカーの相対位置。なしに設定すると、マーカーはオブジェクト・ウィンドウの「マーカー」タ

	ブにだけ表示され、ツリー表示には表示されません。
順序	すべてのマーカーの中での、このグループからのマーカーの相対位置。順位値が最も低いマーカーがリストとツリー表示で先頭に表示されます。
属性	<p><b>複数状態</b> マーカー・グループを使用すると、同一グループの複数のマーカーを同一のオブジェクトに適用できます。このボックスを選択しない場合、このグループのマーカーは一度に1つだけオブジェクトに適用できます。優先順位マーカー・グループは単一状態です。</p> <p><b>自動増分</b> マーカー・グループでは、同一のオブジェクト上に複数のマーカーが存在できません。また、ツリー表示で自動増分マーカーをクリックすると、グループ内の次のマーカーに切り替わります。進行状況フラグは自動増分です。</p>
マーカー	
アイコン	マーカーのアイコン。アイコンを使用できるのは文字列マーカーのみです。マーカーに日付や数値を格納する場合、アイコンは「<なし>」に設定する必要があります。アイコンがないマーカーはツリー表示に表示されません。
名前	マーカーの名前。この名前はユーザー・インタフェースに表示されます。
コード	マーカー・コード。このコードは Oracle Data Integrator ツールと API からマーカーにアクセスするために使用されます。
タイプ	<p>マーカーの型。このフィールドは、マーカーに格納されるデータ型を示します。</p> <ul style="list-style-type: none"> <li>▪ 文字列。この型はアイコンにも使用されます。</li> <li>▪ 数字</li> <li>▪ 日付</li> </ul> <p>数値と日付はツリーに表示されません。その要素のウィンドウの「マーカー」タブで変更できます。</p>
アクティブ	このボックスを選択すると、マーカーはこのグループに対して表示されます。それ以外の場合は常に非表示です。
ツールチップ	マーカーのアイコンにツールチップを表示するにはこのボックスを選択します。

## モデル

### モデルの概要

モデルとは、物理スキーマに含まれているデータ構造に対応するデータストアのセットです。モデルはモデル・フォルダに整理することができます。

- 「モデルの作成およびリバース・エンジニアリング」を参照してください。

## モデル・フォルダ

**モデル・フォルダ**はモデルを集めてグループ化するオブジェクトです。たとえば、特定のテクノロジーに基づくすべてのモデル、特定のサイトにあるすべてのモデルまたは特定のプロジェクトで使用するすべてのモデルをグループ化することができます。

## サブモデル

**サブモデル**は、モデルのデータストアを階層構造に整理および分類するために使用されるオブジェクトです。構造のルートはモデル自身です。

## リバース・エンジニアリング

モデルは、内部にデータストアがない状態で作成されます。モデルを**リバース・エンジニアリング**すると、データ構造を自動的に取得して **Oracle Data Integrator** でモデルのデータストアを定義することができます。モデルには次の 2 種類があります。

- **標準**リバース・エンジニアリングでは、標準 JDBC 機能を使用してメタデータをリクエストします。
- **カスタマイズ**されたリバース・エンジニアリングは、指定されたテクノロジー特有の方法で指定されたテクノロジー特有のリバース・ナレッジ・モジュール (RKM) を使用してメタデータを取得します。

## データストア

**データストア**はデータを表構造として説明します。データストアは**列**で構成されています。

データストアは **Oracle Data Integrator** のリレーショナル・モデル内で定義されています。このため、次の要素をデータストアにアタッチすることが可能です。

### キー

**キー**は、それぞれのデータストア行を一意に識別するデータストア列のセットです。それが索引でもある場合は、行アクセスも最適化されることがあります。一部のドライバはリバース・エンジニアリング時にキーの説明を取得します。リポジトリ内で直接キーを定義することも可能です。

### 参照

**参照**は 2 つのデータストアの間の機能リンクです。リレーショナル・モデルでは外部キーに対応します。たとえば INVOICE データストアが顧客番号を使用して CUSTOMER データストアを参照する場合があります。

### 条件とフィルタ

**条件**または**フィルタ**は、SQL をサポートする RDBMS に基づいてデータストアにアタッチされている WHERE タイプの SQL 式です。データストア内のデータをフィルタまたはチェックします。

### ジャーナル化

ジャーナル化は、データに対する変更を追跡することです。ジャーナル化は、Oracle Data Integrator では変更のないデータを転送しないように使用されます。この機能は、データ同期化とレプリケーションなど、多くの用途があります。

ジャーナル化は、ある特定のタイプのテクノロジーに基づいて、モデル、サブモデルまたはデータストアに適用できます。

## モデル

### 定義

プロパティ	説明
名前	ユーザー・インタフェースで使用するモデル名。
コード	このモデルの一意のコード。
テクノロジー	データ・モデルのテクノロジー。データ・モデルは単一テクノロジーにリンクされますが、これはデータ型がこのテクノロジーにネイティブな物理データ・モデルです。
メタデータ変更をモデル・ツリーで表示します	このオプションを選択すると、データベース内で最後のリバース・エンジニアリング操作以降に削除されたか、変更されたモデルのすべてのツリー内要素にフラグが設定されます。  <b>注意:</b> このオプションを選択すると、ツリーでモデルをブラウズする際にパフォーマンスの低下を招くことがあります。  <b>重要:</b> このオプションは、標準リバース・エンジニアリングを使用するモデルでのみ機能します。
論理スキーマ	このモデルがアタッチされているトポロジの論理スキーマ。
アクション・グループ	このモデル用の DDL スクリプトを生成するために使用するアクション・グループ。アクション・グループが選択されていない場合は、<一般アクション>グループが使用されます。詳細は、「共通フォーマット・デザイナー」を参照してください。
デフォルト・フォルダ	このモデルの DDL スクリプトが生成されるデフォルト・プロジェクト・フォルダ。詳細は、「共通フォーマット・デザイナー」を参照してください。
説明	モデルの詳細な説明。

## リバース

リバース・エンジニアリングとは、アプリケーションのメタデータ（データ構造の説明）を格納するテクノロジーの辞書からその情報を回復し、Oracle Data Integrator リポジトリに格納することです。

プロパティ	説明
リバースのタイプ	<ul style="list-style-type: none"> <li>標準リバースは、ドライバ（JDBC、ODBC など）の機能を使用してメタデータを取得し、リポジトリに格納します。</li> <li>カスタマイズされたリバースは、プロシージャのリバース・ナレッジ・モジュール（RKM）を使用して特定のタイプのアプリケーションのメタデータを抽出し、リポジトリに格納します。</li> </ul>
コンテキスト	リバースが実行されるコンテキスト。Oracle Data Integrator は、コンテキストと論理スキーマの組合せにより、必要なデータ・サーバーに接続してメタデータにアクセスします。
論理エージェント	これはカスタマイズされたリバース・エンジニアリング・モードに使用されるエージェントです。
リバースエンジニアリングするオブジェクトの型	リバース・エンジニアリング処理の際に考慮されるオブジェクトのタイプ。このリストには、表、ビュー、キュー、システム表、表エイリアスおよびシノニムが含まれることがあります。
マスク	マスクはリバースするオブジェクトを事前選択します。このマスクは、SQL 言語の LIKE 構文を使用します。つまり、「%」記号は 0 個以上の文字、「_」記号は 1 個の文字を表します。
表の別名で削除する文字	別名を作成する際に削除する文字。各データストアには、トランスフォーメーション、フィルタおよびチェックの式で使用される別名があります。これは短縮名で、デフォルトでは、データストア名の最初の 3 文字をとって付けられます。同一の接頭辞で始まる表がアプリケーションに多数含まれている場合は、このセクションに値を指定することで、その接頭辞が無視されるようにすることができます。たとえば、DWG_ という値を指定すると、DWG_PROD、DWG_CUS、DW_ITEM という表に対して、PRO、CUS、ITE という別名が付けられます。
表別名最大長	このモデルに対して生成されるエイリアスの最大長。この長さより後の文字は切り捨てられます。

## 選択的リバース

標準リバースに対して選択的リバースを使用すると、「リバース」タブで定義されている、リバース時に考慮されるマスクとオブジェクトのタイプのリストから、リバースするデータストアを選択することができます。

プロパティ	説明
選択的リバース	次回のリバース・エンジニアリング操作を選択的にするよう指定します。このタブのパラメータが使用されます。
既存のデータストア	選択されたモデルにすでに存在するデータストアを再リバース・エンジニアリングするよう指定します。これらのオブジェクトのメタデータはリポジトリ内で更新されます。変更はすべて実行ログに記録されます。
新規データストア	現在のモデルにまだ存在しないデータストアをリバース・エンジニアリングするよう指定します。
リバースするオブジェクト	リバース・エンジニアリングするデータストアを個々に指定します。表示されるリストではマスクとパラメータが考慮されます。リバースしないオブジェクトを選択解除することによって、このリストをさらに短くすることができます。このリストは <b>カスタム</b> ・モードでは使用できません。

## 管理（データ品質監査）

データ品質監査（静的管理とも呼ばれます）は、モデル内のすべてのデータが、データストアで定義されている整合性制約に適合しているかどうかをチェックします。これらの制約には、参照、主キー、代替キー、条件および必須列が含まれます。常にチェック・ナレッジ・モジュール（CKM）が使用されます。

プロパティ	説明
チェック・ナレッジ・モジュール	データ品質管理で使用される戦略。ここでは、モデルのテクノロジーに対して有効な CKM のみを選択することができます。使用可能な CKM は、少なくとも 1 つのプロジェクトにインポートされている必要があります。
オプション	CKM によって指定されたパラメータ。
説明	CKM の詳細な説明。
実行	データ品質エージェントを実行するためにセッションを開始します。エージェントとコンテキストを選択する必要があります。

## ジャーナル化

ジャーナル化は、モデルのデータストアでのデータ変更（挿入、削除、更新）を追跡します。このタブでは、このモデルで使用するジャーナル化方法を定義して、構成できます。

プロパティ	説明
ジャーナル化モード	このモデルをジャーナル化するために使用されたモード。詳細は、「チェンジ・データ・キャプチャ」を参照してください。
ジャーナル化ナレッジ・モジュール	データのジャーナル化で使用される戦略。ここでは、モデルのテクノロジーに対して有効な JKM のみを選択することができます。使用可能な JKM は、少なくとも 1 つのプロジェクトにインポートされている必要があります。
オプション	JKM によって指定されたパラメータ。
説明	JKM の詳細な説明。
サブスクライバ・リスト	このモデルで現在変更を追跡しているサブスクライバのリスト。

## ジャーナル化される表

この表には、CDC に含まれているモデルからのデータストアが記載されます。データストアを CDC セットに整理することができます。詳細は、「チェンジ・データ・キャプチャ」を参照してください。

プロパティ	説明
順序	CDC セット内でジャーナル化されたデータストアの位置。通常、この順序は外部キー・リレーションに依存します。参照先の表は、参照元の表より後の順序にします。
表名	ジャーナル化されるデータストアの名前。

## サービス

このタブには、このモデル用のデータ・サービスを生成、展開するための構成が含まれます。詳細は、「データ・サービスの設定」を参照してください。

プロパティ	説明
アプリケーション・サーバー	生成された Web サービスがデプロイされるアプリケーション・サーバー (Web サービス・コンテナ) に対応する論理スキーマ。
ネームスペース	使用する Web サービス用のネームスペース。WSDL を生成するために使用されます。
パッケージ名	すべての Web サービスを含むように生成した java パッケージの名前。一般に、これは <code>com.&lt;company name&gt;.&lt;project name&gt;</code> という形式に

	なります。
データソースの名前	アプリケーション・サーバー内で定義されている、モデルのデータが含まれているサーバーに対応するデータ・ソースの名前。名前の前に <b>java:/comp/env/</b> を置きます。
データ・サービスの名前	モデル・レベルで (CDC 関連の) 機能を提供するデータサービスの名前。
サービス KM	Web サービスを生成するために使用されるサービス・ナレッジ・モジュール。プロジェクトにインポートされた KM のみがこのリストに表示されます。
配置済データストア	データ・サービスとして展開されるデータストアのリスト。各データストアに対して、 <b>データサービス名</b> と <b>公開エンティティ名</b> を付けることができます。

## モデル・フォルダ

フォルダとは、モデルのグループです。フォルダとサブフォルダを使用すると、プロジェクト特有の基準に応じてモデルをグループ化して整理することができます。作成できるサブフォルダのレベル数に制限はありません。

**注意:** サブフォルダを作成するには、フォルダを作成し、親フォルダの中にドラッグ・アンド・ドロップします。

### 定義

プロパティ	説明
名前	ユーザー・インタフェースに表示されるフォルダまたはサブフォルダの名前
説明	フォルダの詳細な説明

## サブモデル

サブモデルは、モデル内で機能の点で同種のデータストアのグループです。モデルのデータストアは、ドラッグ・アンド・ドロップを使用して、または自動配布によりサブモデルに挿入することができます。

### 定義

プロパティ	説明
名前	デザイナー・モジュールのグラフィカル・インタフェースで使用するサブモデルの名前
コード	サブモデルの一意のコード

## 管理

この手順を使用すると、サブモデル・データストアに格納されたデータをすぐに静的管理することができます。この手順はチェックの開始時に指定されたコンテキストで実行されます。このチェックは、モデル・レベルで表示されるチェック・ナレッジ・モジュールを使用します。

## 配布

配布を使用すると、サブモデル内のデータストアの自動配布を定義できます。

データストアは、それぞれのサブモデルの**データストア配布ルール**に応じて、**自動割当てマスク**と比較されます。データストアがこのパターンと一致すると、このモデル内に移動されます。

2つの分類方法があります。

- サブモデルの「**配布**」ボタンをクリックすることによって、現在の規則がデータストアに適用されます。
- リバース完了時に、「**リバース後のマスク適用順序**」内のすべてのルールが適用されます。

プロパティ	説明
データストア配布ルール	対象となって自動割当てマスクと比較されるデータストアを決定します。 <b>自動配布なし:</b> データストアは処理されません。 <b>すべての未分類のデータソースを自動配布...:</b> サブモデル・ツリーのルート・モデルにあるデータストアが対象になります。 <b>すべてのデータソースを自動配布:</b> モデル（およびサブモデル）内のすべてのデータストアが対象になります。
自動割当てマスク	名前がこのパターンに従うデータストアがこのサブモデルに分類されます。
リバース後のマスク適用順序	リバース完了時に、「 <b>リバース後のマスク適用順序</b> 」内のすべてのルールが適用されます。  したがって、すべてのデータストアのうち、順位の高いルールが優先されます。未分類のデータストアに関する順位の高いルールは、他のルールのパターンが無視するデータストアについてのみ適用されます。  リバースが終了すると、新しいデータストアは未分類とみなされます。サブモデル内に分類されたデータストアは、そのサブモデルにアタッチされた状態にとどまります。

## データストア

### データストア

データストアは、データの格納を可能にする構造体です。Oracle Data Integrator が使用するミドルウェア (JDBC/ODBC、JMS または JNDI) によってアクセスできる、表、ファイル、メッセージ・キューなど、任意のデータ構造を使用できます。

### 定義

プロパティ	定義
名前	<p>データストアの名前。ツリーに表示される名前、プロジェクト（インタフェースなど）からデータストアを参照するために使用されます。リソース名が未知または不明確な場合、この名前はリソース名と異なるものにします。</p> <p>たとえば、モデルに C3556F または C5677D という名前の表が含まれる場合、C3556F や C5677D はリソース名として残し、CUSTOMER または ORDER などの、よりわかりやすい名前を付けることができます。</p>
データストア・タイプ	データストアが表すオブジェクトのタイプ。
OLAP タイプ	<p>Online Analytic Processing (OLAP) 用の多次元モデルでの表タイプ。</p> <ul style="list-style-type: none"> <li>ファクト表</li> <li>ディメンション</li> <li>緩やかに変化するディメンション</li> </ul>
リソース名	オブジェクトを格納するデータ・サーバーが認識する形式のオブジェクト名。表名でもファイル名でもかまいません。
別名	<p>チェックおよびフィルタの式で使用される短い名前です。この名前が体系的に使用されないと、各列名で相互参照がコンパイルできません。</p> <p>たとえば、表 CUSTOMER では、エイリアスに CUS を指定することができます。この場合、条件 CLIENT_TYPE の式は、「CUS.TYPE in ('A', 'S', 'D)」などとなります。</p> <p>エイリアス名は必ずしも特定のモデルのすべての表で一意でなくてもかまいません。その目的は式の作成を容易にすることと、相互参照の管理を可能にすることだけです。</p>
説明	データストアの詳細な説明。
行数	「リフレッシュ」をクリックすることによって取得したデータストアの行数。

## ファイル

このタブは、ファイル・テクノロジーにアタッチされているデータストアについてのみ表示されません。

プロパティ	定義
ファイル形式	<p>ファイル・データストアの形式。</p> <ul style="list-style-type: none"> <li><b>区切り</b>: 1 行にある複数のフィールドが、レコード・セパレータで分割されています。</li> <li><b>固定</b>: 1 行にある複数のフィールドが、区切られず、長さが固定されています。</li> </ul>

ヘッダー(行数)	ファイルの先頭にあるデータでない行数。この行は無視されます。
レコード・セパレータ	ファイル内で行と行（またはレコードとレコード）を区切っている 1 つ以上の文字。 <ul style="list-style-type: none"> <li>• <b>MS-DOS</b>: DOS の改行記号</li> <li>• <b>UNIX</b>: UNIX の改行記号。</li> <li>• <b>その他</b>: 文字または 16 進コードとして入力できる任意のテキスト。</li> </ul>
フィールド・セパレータ	レコード内でフィールドとフィールドを区切っている 1 つ以上の文字。 <ul style="list-style-type: none"> <li>• <b>タブ</b></li> <li>• <b>領域</b></li> <li>• <b>その他</b>: 文字または 16 進コードとして入力できる任意のテキスト。</li> </ul>
テキスト・デリミタ	STRING フィールドを区切る文字のペア。
小数点セパレータ	NUMERIC の整数部分と小数部分を区切る文字。

## 列

このタブを使用すると、データストア列の概要ビューを表示、追加および削除できます。「リバー

- **リバー**: このリバー・エンジニアリングは、現在のデータストアの**標準**リバーを開始します。
- **リバー**: このリバー・エンジニアリングは、**区切り文字で区切られた**ファイルで使用できます。列名は、ファイルのヘッダー行に取得されるか、または自動的に生成されます。列の長さはデフォルト値に設定されるため、手動で再定義する必要があります。ファイルが固定ファイルの場合は、このボタンは、列を定義するためのウィザードを開きます。
- **COBOL コピーブックのリバー**: このリバー・エンジニアリングは、COBOL コピーブック形式の記述ファイルがある場合に、**固定**ファイルで実行できます。詳細は、「COBOL コピーブックのリバー・エンジニアリング」を参照してください。

**自動調整**: このオプションを固定ファイルにのみ選択すると、先頭位置が列幅の関数として自動調整されるため、隙間や重なりが生じません。たとえば、先頭の列の幅を 5 から 16 まで増やすと、他のすべての列の先頭位置が 11 文字分増えます。自動調節を有効にすると、列の先頭位置を手動で指定することはできません。

## 管理

この手順を使用すると、データストアに格納されているデータの静的管理を実行することができます。この手順は管理の開始時に指定されたコンテキストで実行されます。このチェックは、モデル・レベルで表示されるナレッジ・モジュールを使用します。

## ジャーナル化

このタブには、データストアのジャーナル化ステータスと、このデータストア上で変更を追跡するサブスクライバのリストが、サブスクリプション作成日とともに表示されます。

## サービス

このタブには、このデータストアのデータ・サービスを生成、展開するための構成が含まれます。詳細は、「データ・サービスの設定」を参照してください。

プロパティ	説明
データ・サービスとして配置	このデータストアをデータ・サービスとして展開する場合は、このボックスを選択します。
データ・サービス名	このデータストア用に生成された Web サービスの名前。
公開済エンティティ	このデータ・サービスのすべての操作名を生成するために使用される名前。たとえば、公開エンティティとして <code>customer</code> を指定する場合、生成される操作の名前は、 <code>addcustomer</code> 、 <code>getcustomer</code> などになります。

## 列

データストアは、Oracle Data Integrator によってアクセスされるデータを、2 次元表を使用して記述します。Oracle Data Integrator には、列に関連したメタデータが含まれます。アプリケーションに含まれているデータは行に格納されます。

## 定義

プロパティ	説明
名前	データ・サーバーによって認識される列名。
データ型	データ・サーバーによって認識されたタイプに対応するデータ型。
簡単な説明	列ラベルの詳細な説明。このラベルは、列名が明示的でない場合に、グラフィカル・インタフェースを読みやすくします。
物理フォーマット	データの物理的な保管形式の説明。
順序	データストア内の列の順位または物理ランク。
開始	列の物理位置。この情報は固定形式ファイルに対して指定されます。先頭位置は位置 1 です。
長さ	バイト数で表した物理長。先頭位置が「1」で長さが「4」の列は、それぞれの行のバイト 1、2、3、4 を占めます。 <b>注意:</b> デリミタ付きのファイルでは、列が取得した最大長が表示されます。最大長があらかじめわからない場合は、最大長より大きい長さを指定してみてください。この長さは、情報を読み取るために十分なメモリー空間を確保するために使用されます。

レコード・コード	レコード・コードのリストを使用すると、複数のレコード・パターンを持つファイルを処理できます。  たとえば、レコード・コード <code>CODE_REC</code> の値「HEAD」または「LINE」で識別される、異なるヘッダーとインボイス行を含むファイル <code>INVOICE</code> では、これらの概念エンティティそれぞれに対応するデータストアを作成する必要があります。データストア <code>INVOICE</code> の列 <code>CODE_REC</code> のレコード・コード・リストで「HEAD」を指定し、データストア <code>INVOICE_LINE</code> の対応する列に「LINE」を指定します。  「;」で区切って複数の値を指定できます。
レコード・コードを除外します	このボックスを選択すると、ファイルのロード時に、レコード・コードを考慮する行が含まれず、除外されます。
<b>論理フォーマット</b>	<b>データ表現の説明。</b>
長さ	エンド・ユーザーに対して機能的に表示される、列の論理長。数値の場合、これは総桁数（精度）です。  この長さは、物理長が指定されている場合も指定してください。
形式	日付に対して使用する形式。この形式は、マシンの地域パラメータと一致する Java 日付形式です。
精度	数の総桁数。
スケール	小数部分の桁数。
小数点セパレータ	小数点（デフォルトは「.」）。

## 説明

プロパティ	説明
デフォルト値	値が指定されない場合にこの列に挿入される値。
読取り専用	この列を <code>INSERT</code> または <code>UPDATE</code> コマンドで使用できない場合に、このボックスを選択します。これは、たとえば <code>IDENTITY</code> 列など、値が自動設定され、変更できない列の場合です。
緩やかに変化するディメンションの動作	このフィールドは、OLAP 用の、緩やかに変化するディメンションの表をロードする場合の列の動作を定義します。次の動作がサポートされます。 <ul style="list-style-type: none"> <li>■ サロゲート・キー: 列がレコード・バージョンの一意の（技術的）識別子です。通常、この列は自動増分された値（シーケンス）によってロードされ、ファクト表によって参照されます。</li> <li>■ 自然キー: 列は、バージョンにかかわらず 1 件のレコードを識別するキーの一部です。通常、このキーはソース表の主キーに対応します。</li> <li>■ 変更時に上書き: この列の値が変更されると、現在のレコード・バージョン</li> </ul>

	<p>ンが更新され、この列が上書きされます。</p> <ul style="list-style-type: none"> <li>変更時の行の追加: この列の値が変更されると、新しいレコード・バージョンが作成されます。</li> <li>現在のレコード・フラグ: この列は、現在のレコード・バージョンを識別するフラグです。通常、最新バージョンは1、旧バージョンは0に設定されます。</li> <li>開始タイムスタンプ: レコード・バージョンが有効になる日付。</li> <li>終了タイムスタンプ: レコード・バージョンが有効でなくなる日付。</li> </ul> <p>これらの動作は、緩やかに変化するディメンション (SCD) をサポートしている一部のナレッジ・モジュールで考慮されます。</p>
説明	列の詳細な説明。

### 制御

品質管理プロパティは、フロー・コントロール時、静的管理時、またはファイルのロード時に考慮されます。

プロパティ	説明
必須	列の入力が必須かどうかを示します。
制御	<b>必須ステータスが選択されている品質管理のタイプが表示されます。</b>
フロー	このボックスが選択されると、このデータストアに統合化されたデータのフロー・コントロール時に、列の必須ステータスが検証されます。この情報は、インタフェースの設計時に変更できるデフォルト値です。情報がターゲット・テクノロジーでもチェックされる場合であっても、データ品質を保証するためにこのタイプのチェックをアクティブにすることをお勧めします。
静的	このボックスが選択されると、静的管理時、つまりこのデータストアにすでに存在しているデータがチェックされる場合に、この列の必須ステータスが検証されます。
エラー発生時	<b>ファイル内のデータはデータストア定義と一致しないことがあります。エラー発生時オプションは、ファイル読取り時にある行の1つの値が列の説明と一致しない場合に Oracle Data Integrator が実行するアクションを定義します。</b>
エラーの拒否	エラーが含まれている行が拡張子が「BAD」のファイルに移動され、拡張子が「ERROR」のファイルにエラーの説明が挿入されます。 BAD と ERROR は、読み取るファイルと同一のディレクトリにあります。
エラーの場合は Null(非アクティブ・トレース)	行はフローに保持され、誤っている値は null で上書きされます。
エラーの場合は Null(アクティブ)	行はフローに保持され、誤っている値は null で上書きされ、エラーの説明が拡張

イブ・トレース) 張子「ERROR」のファイルに挿入されます。

## サービス

このタブには、この列用のデータ・サービスを生成、展開するための構成が含まれます。詳細は、「データ・サービスの設定」を参照してください。

プロパティ	説明
許可された操作	データ・サービスがこの列で実行するアクションのタイプ (INSERT/UPDATE/SELECT) に対応するボックスを選択します。このタブの重要な用途の1つに、データ・サービスを介する列への書込みをロックすることがあります。

## 参照

参照は2つのデータストアの間の機能リンクです。参照は、リレーショナル・モデルにおける外部キーの概念に対応します。たとえば、INVOICE データストアは、顧客番号を通じて Customer データストアを参照します。

### 参照の概要

参照は様々な場面で有用です。

- 参照を宣言（またはリバース）すると、モデルの可読性が向上します。
- 参照を定義すると、リンク・データの整合性をチェックできます。
- 参照をチェックすることも、モデルを理解しやすくします。アプリケーション・データが、宣言された参照と比較して100%不整合の場合は、相互参照ルールが誤っていると考えられます。
- 相互参照ルールを使用すると、データをこのデータストアに統合する際に、フロー・コントロール手順を生成できます。たとえば、インボイスと顧客の間の参照を宣言（またはリバース）すると、そのインボイスをロードするすべてのインタフェースが、存在しない顧客を参照しているインボイスを除外します。

**注意:** 静的データ整合性チェックをサポートするのは、SQL 言語をサポートするテクノロジーだけです。

**注意:** リポジトリに参照を作成しても、オブジェクトはモデルに基づいているデータ・サービス上に作成されません。

## 定義

プロパティ	説明
名前	参照名。
タイプ	簡易: 列の同一性に基づく、ユーザーによって定義された参照。 複合: 複合式を使用する、2つのデータストアの間の参照。 外部: Oracle Data Integrator 外部の、メタデータ・リポジトリで宣言された参照。
モデル	参照された表、または主キー表、または親として認識されている表のモデル。
表	参照された表、または主キー表、または親として認識されている表。この情報が未入力の場合、または参照された表がリポジトリ内に定義されていない場合、その名前は「外部表」グループに入力される必要があります。
<b>外部表</b>	<b>Oracle Data Integrator で定義されているモデルの外部の表名。</b>
カタログ	外部表が含まれているカタログ。
スキーマ	外部スキーマが含まれているカタログ。
表	外部表の名前。
データベースでアクティブ	参照が、データストアが含まれているデータベース上に存在することを示します。

### 列（非複合参照）

このタブは、単純参照か複合参照についてのみ表示されます。

単純参照または外部参照の場合、外部キー表の列（子表）と主キー表（親または参照されている表）の間の対応を定義する必要があります。リレーショナル・モデルの ISO 標準に従うと、null でないすべての外部キー列を含む外部キー表内の行のみがチェックされます。

### 式（複合参照）

2つの表を結合する基準は複合的であることがあります。この場合、このタブに自由な表現式を入力できます。

### 動作（外部参照）

動作は純粋な指示メタデータで、それは外部の外部キーに関係します。この情報は、データ・サーバーが参照（外部キー）をチェックする（アクティブ）かどうか、および、行が主キー表から削除された場合、または行の主キー値が主キー表で変更された場合に、データ・サーバーがどのようなアクションを起動するかを示します。

### 管理

品質管理プロパティは、フロー・コントロール時、または静的管理時に考慮されます。

プロパティ	説明
品質管理のタイプ	この参照が選択されている品質管理のタイプが表示されます。
フロー	このボックスを選択すると、このデータストアのフロー・コントロール時に、参照が検証されます。この情報は、インタフェースの設計時に変更できるデフォルト値です。情報がターゲット・テクノロジーでも管理される場合であっても、データ品質を保証するためにこのタイプの管理をアクティブにすることをお勧めします。
静的	このボックスが選択されると、静的管理時、つまりこのデータストアにすでに存在しているデータの品質管理の場合に、参照が検証されます。

「**チェック**」ボタンをクリックすると静的管理が実行されますが、目に見える唯一の結果は、誤っている行の番号が「**制御**」タブに表示されることだけです。エラー時に行のトレースを取得するには、静的管理（非同期と呼ばれます）は「**データストア**」または「**モデル**」ウィンドウから起動される必要があります。

## キー

キーとは、パフォーマンスの点で特権的な方法（索引）でデータストア行を識別したりアクセスしたりできるようにするデータストア列のセットです。一部のドライバでは、リバース・エンジニアリングの処理中にキー記述がリカバリされます。リポジトリ内で直接キー定義を入力することも可能です。

### 主キーと代替キーの宣言の概要

主キーや代替キーは様々な場面で有用です。

- モデルの可読性が向上します。
- リンク・データの整合性チェックを可能にし、それにより重複がないことを検証できます。
- また、参照をチェックすると、検証されるモデルを理解できます。アプリケーション・データが、宣言したキーと比較して 100%不整合の場合は、使用されているキー識別が無効であると考えられます。
- 識別ルールを使用すると、このデータストアに、フロー品質管理手順を生成できます。たとえば、データストア「**Invoice**」上で主キーまたは代替キーを宣言（またはリバース・エンジニアリング）すると、インボイスをロードするすべてのインタフェースで、同一のインボイス番号を持つものが除外されます。

**注意:** 静的データ整合性管理をサポートするのは、SQL 言語をサポートするテクノロジーだけです。

**注意:** 一意でない索引は、単に情報を提供するものであり、チェックの対象になりません。

**注意:** リポジトリにキーや索引を作成しても、オブジェクトはモデルの基となるデータ・サーバー上に作成されません。

## 一般プロパティ

プロパティ	説明
名前	キーの名前。
索引のキーまたはタイプ。	<ul style="list-style-type: none"> <li>主キー: 客観的な基準（これらすべての列に入力する必要があります）にも、主観的な基準（簡潔性、パフォーマンスなど）にも適する一意のキー。データストアがインタフェース・ターゲットの場合、主キーがデフォルトの更新キーとして使用されます。</li> <li>代替キー: 主キーとして使用されない一意のキー。</li> <li>一意索引ではありません: アクセス・パフォーマンスを改善するためにのみ使用される索引。</li> </ul>

## 列

このタブを使用すると、キーに関連する表の列を選択できます。

## 管理

品質管理プロパティは、フロー・コントロール時、または静的管理時に考慮されます。

プロパティ	説明
データベースで定義済	このキーがデータベース辞書（または記述されたアプリケーション）で定義されているかどうかを示します。リバース・エンジニアリングされているキーはデータベースで定義されています。
アクティブ	キーがデータベースでアクティブかどうかを示します。一部のテクノロジーでは、アクティブ・キー（有効）または非アクティブのキー（無効）が辞書に含まれることがあります。この場合、データ・サーバーはデータの整合性をチェックしません。
制御	このキーが選択されている品質管理のタイプが表示されます。
フロー	このボックスを選択すると、このデータストアのフロー・コントロール時に、キーが検証されます。この情報は、インタフェースの設計時に変更できるデフォルト値です。情報がターゲット・テクノロジーでも管理される場合であっても、データ品質を保証するためにこのタイプのチェックをアクティブにすることをお勧めします。
静的	このボックスが選択されると、静的管理時、つまりこのデータストアにすでに存在しているデータのチェックの場合に、キーが検証されます。

「**チェック**」ボタンをクリックすると静的管理がグラフィカル・インタフェースによって実行されますが、目に見える唯一の結果は、誤っている行の番号が「**制御**」タブに表示されることです。エラー時に行のトレースを取得するには、静的管理（非同期と呼ばれます）は「データストア」または「モデル」ウィンドウから起動する必要があります。

## 条件

条件は、SQL をサポートする RDBMS に基づいてデータストアにアタッチされている WHERE タイプの SQL 式です。この条件の目的は、問題のデータストアのデータをフィルタまたはチェックすることです。

### Oracle Data Integrator での条件の宣言の概要

Oracle Data Integrator の条件は様々な場面で有用です。

- たとえば、データストア・データの整合性をルールに応じてチェックするようにできます。
- 条件を使用すると、このデータストア内にフロー・コントロールを生成できます。たとえば、「Client」データストア上で顧客の年齢をチェックする条件を宣言すると、「Client」データストアをロードするすべてのインタフェースで、年齢が範囲外の顧客が除外されます。

**注意:** 静的管理をサポートするのは、SQL 言語をサポートするテクノロジーだけです。

### 一般プロパティ

プロパティ	説明
名前	条件の名前。
タイプ	<ul style="list-style-type: none"> <li>• <b>フィルタ</b>は、チェックおよびインタフェース処理で、関係する表内の行をターゲットにする方法の1つです。フィルタは、インタフェースによって、データストアがソースとして使用される場合に使用されます。アクティブな顧客をターゲットにするフィルタの例: <code>CLI.TYPE_CLIENT like 'A%'</code></li> <li>• <b>Oracle Data Integrator コントロール</b>は、データストアにアタッチされたデータ整合性ルールを表します。このルールは、同期または非同期でチェックすることができます。このデータストアをターゲットに持つフローもチェックされます。アプリケーション・ワークスペース上で誤っているデータを分離できます。顧客の年齢をチェックする Oracle Data Integrator 条件の例: <code>CLI.AGE between 0 and 130</code></li> <li>• <b>DBMS 制約</b>は、RDBMS または別のアプリケーション辞書からリバース・エンジニアリングされたチェック制約を表します。</li> </ul>
条件	<p>条件を定義する Where タイプの式。この式は、各列の前に（「データストア」ウィンドウで定義された）表エイリアスを使用する必要があります。</p> <p>エイリアスが CLI であるデータストア CLIENT に関する式の例: <code>CLI.TYPE_CLIENT like 'A%'</code></p>
メッセージ	エラー表にリストされた、エラーを含むデータストア内の行に対応するエラー・メッセージ。この情報はフィルタには適用できません。

### 管理

品質管理プロパティは、フロー・コントロール時、または静的管理時に考慮されます。フィルタはチェックされないため、管理情報はフィルタ条件には使用できません。

プロパティ	説明
データベースで定義済	この条件がデータベース辞書（または記述されたアプリケーション）で定義されているかどうかを示します。リバースされている条件はデータベースで定義されています。
アクティブ	条件がデータベースでアクティブかどうかを示します。一部のテクノロジーでは、アクティブ（有効）または非アクティブ（無効）な条件が辞書に含まれることがあります。この場合、データ・サーバーはデータの整合性をチェックしません。
<b>制御</b>	<b>この条件が選択されている管理のタイプが表示されます。</b>
フロー	このボックスを選択すると、このデータストアのフロー・コントロール時に、条件が検証されます。この情報は、インタフェースの設計時に変更できるデフォルト値です。情報がターゲット・テクノロジーでも管理される場合であっても、データ品質を保証するためにこのタイプのチェックをアクティブにすることを推奨します。
静的	このボックスが選択されると、静的品質管理時、つまりこのデータストアにすでに存在しているデータの品質管理の場合に、条件が検証されます。

「**チェック**」ボタンをクリックすると静的管理がグラフィカル・インタフェースによって実行され、誤っている行の番号が「**制御**」タブに表示されます。エラー時に行のリストを取得するには、静的管理（非同期と呼ばれます）は「**データストア**」または「**モデル**」ウィンドウから起動される必要があります。

## オペレータ

### オペレータの概要

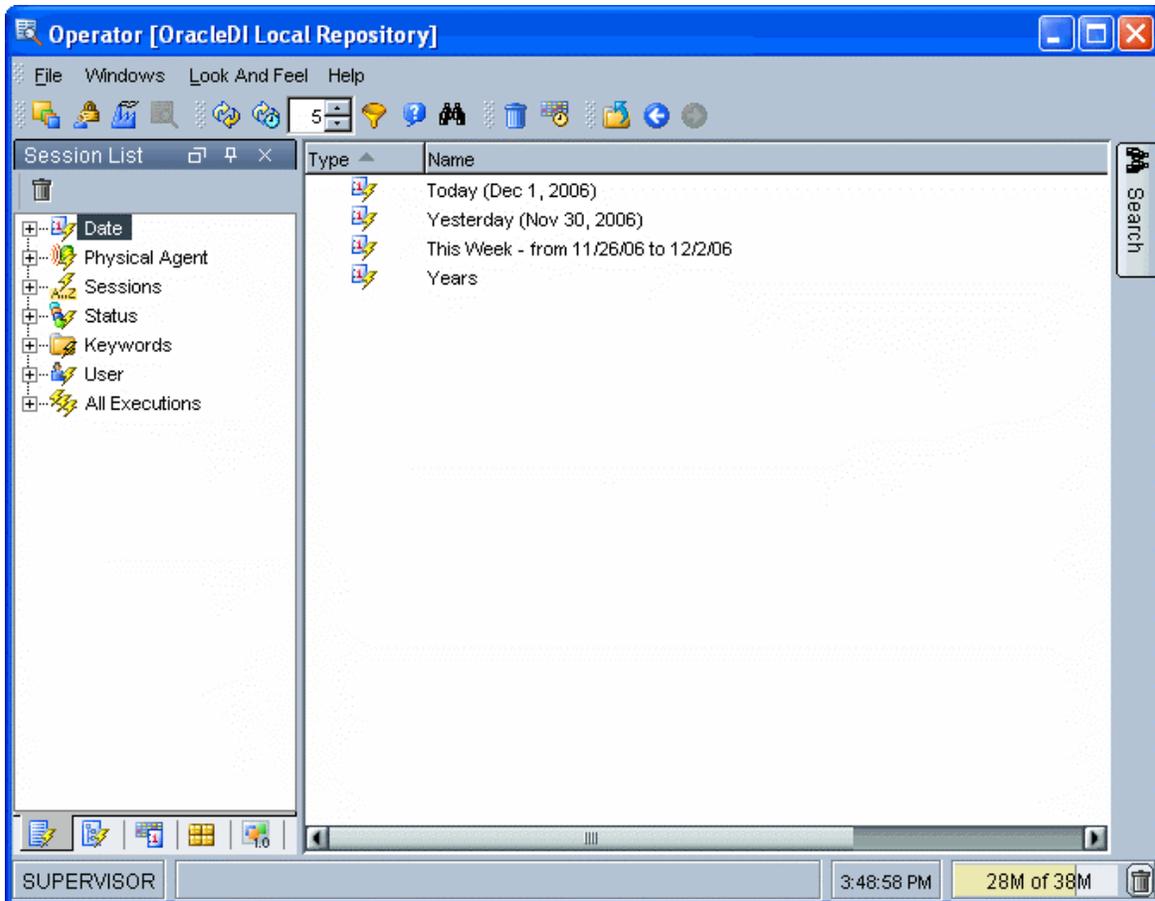
オペレータ・モジュールを使用すると、セッションでのインタフェース実行と、本番でのシナリオを管理することができます。

オペレータ・モジュールは、マスター・リポジトリで定義されているトポロジを使用して、この情報を作業リポジトリに格納します。

- オペレータの操作

### オペレータのインタフェース

オペレータの GUI は次のように表示されます。



## メニュー

メニューでは、プルダウン・メニューから次の機能にアクセスできます。

- インポート
- ログ消去
- スケジューリング
- オプションの表示
- モジュールまたはツリー表示のオープン
- ユーザーのパスワードおよびオプションの変更

## ツールバー

ツールバーからは、次の操作を実行できます。

- 他のモジュールの起動
- ログの参照
- ログの消去
- スケジューリング情報の表示
- 手動または自動でのログのリフレッシュ

- オンライン・ヘルプの起動

## ツリー表示

現行ユーザーが使用できるオペレータ・オブジェクトは、次のようにツリー表示に整理して表示されます。

- **セッション・リスト**は、すべてのセッションを、日付、物理エージェント、状態、キーワードなどを基準に整理して表示します。
- **階層セッション**には、子セッションとともに階層構造に整理された実行セッションが表示されます。
- **スケジューリング**には、物理エージェントとスケジュールのリストが表示されます。
- **シナリオ**には、使用できるシナリオのリストが表示されます。

各ツリー表示は、メイン・ウィンドウの両側にドッキングできるフローティング・フレームに表示されます。フレームは重ねることもできます。複数のフレームを重ねた場合は、フレーム・ウィンドウの下部に表示されるタブから各フレームにアクセスできます。

ツリー表示フレームは、フレームのタイトルまたはタブを選択してドラッグすることにより、移動したり、ドッキングしたり、重ねたりできます。ツリー表示の位置を固定するには、「**ウィンドウ**」メニューから「**ウィンドウ・レイアウトのロック**」を選択します。

ツリー表示フレームがメイン・ウィンドウに表示されないか、閉じている場合は、「**ウィンドウ**」→「**ビューの表示**」メニューを使用します。

各ツリー表示では、次の操作が可能です。

- ノードの展開または折りたたみ（ノードをクリック）
- オブジェクトに関連付けられているメソッド（編集、削除など）のアクティブ化（ポップアップ・メニューを使用）
- オブジェクトの編集（オブジェクトをダブルクリック、または**ワークベンチ**にドラッグ・アンド・ドロップ）

## ワークベンチ

Workbench には、ツリー表示で現在選択されているサブオブジェクトのリストが表示されます。たとえば、ツリー表示で手順が選択されていると、この手順に関するタスクのリストがワークベンチに表示されます。各リストにある列は、列タイトルのポップアップ・メニューを使用してカスタマイズできます。

編集または表示されているオブジェクトのウィンドウは、**ワークベンチ**に表示されます。

## セッション

### セッション

セッションとは、実行エージェントが担当する 1 回の実行（シナリオ、インタフェース、パッケージまたはプロシージャなど）です。セッションは手順で構成されており、手順はタスクで構成されています。

## 定義

プロパティ	説明
名前	このセッションで実行されたシナリオ、パッケージ、インタフェース、プロシージャなどの名前。
バージョン	バージョン（シナリオの）。
実行コンテキスト	実行されたコンテキスト。
実行エージェント	このセッションでタスクの実行を担当するエージェント。
状態	セッション・ウィンドウが開いているときのセッション状態。可能な状態は、完了、エラー、実行中、待機中および警告です。
停止	セッションを停止します。
再起動	セッションを再起動します。
<b>実行</b>	<b>実行情報詳細。</b>
開始	セッション実行開始の日付と時刻。
終了	シナリオ実行終了の日付と時刻。
継続時間	シナリオ実行の所要時間。
リターン・コード	セッションのリターン・コード。
メッセージ	セッション実行エラー・メッセージ（必要な場合）。

## セッション手順

手順とは、タスクとセッションの間に当たる実行単位です。パッケージまたはシナリオ内の単一の処理を表します。たとえば、単一のインタフェースまたは変数を実行する場合、セッションには1つのセッション手順のみが含まれます。

### 定義

プロパティ	説明
セッション	一意のセッション番号。
順序番号	このセッションの手順にリンクされた順序番号。
名前	実行された手順の名前。
タイプ	実行された手順のタイプ。手順のタイプの詳細は、手順を参照してください。
実行コンテキスト	手順が実行されたコンテキスト。

許容されているエラーの最大数	<ul style="list-style-type: none"> <li>「%」ボックスを選択しなかった場合、この値はエラーの最大許容値（エラーの総数）になります。</li> <li>ボックス「%」を選択した場合、この値はエラーの最大許容割合（エラーの総数に基づく）になります。</li> </ul> <p>このフィールドが空のままの場合、許容エラー数は無限です。 エラーがこの数または割合を超えると、手順はエラー状態になります。</p>
----------------	---

## 実行

プロパティ	説明
実行数	手順（ループ）の実行が成功した際の実行数。
<b>実行</b>	<b>手順実行の詳細な結果を提供します。</b>
開始	手順実行開始の日付と時刻。
終了	手順実行終了の日付と時刻。
継続時間	手順実行の所要時間。
状態	手順の状態（完了、エラー、実行中、待機中、警告）。
リターン・コード	手順のリターン・コード。
メッセージ	手順実行エラーメッセージ（必要な場合）。
挿入数	手順実行時に挿入された行数。
削除数	手順実行時に削除された行数。
更新数	手順実行時に更新された行数。
エラー数	手順でエラーになった行数。

## セッション・タスク

タスクは最も小さい実行単位です。タスクは、KM 内のプロシージャ・コマンド、プロシージャ、変数の割当てなどに対応して発生します。

## 定義

プロパティ	説明
説明	<p>タスクのタイプの説明。タスクが関与する操作のタイプが表示されます。インタフェースの場合、インタフェースがどのプロシージャ・フェーズにあるかが示されます（ロード中、統合、チェックなど）。</p> <p>タスクが処理するオブジェクトの名前。</p> <p>タスクの詳細な名前。</p>
順序	これらの2つのフィールドには、セッション内でのタスクの順序（セッションでは一意）、および、タスクの作成に関与した KM またはプロシージャ内でのタスクの順序の詳細が表示されます。
タイプ	実行されたタスクのタイプ。
詳細	タスクのロギング・レベル。
エラーの無視	このタスクがエラー許容範囲を設定しており、それによってこのタスクのエラーがブロックされるかどうかを示します。

## 説明

このタブにはソース接続で投入されるコマンドと、タスク中にデフォルトで投入されるコマンドが含まれます。これらの命令は、テクノロジのネイティブ言語で書かれます。ただし変数は、命令が実際に実行される際に解釈されるとおりの明確な形式で含まれます。

## 実行

プロパティ	説明
デフォルトの接続、接続をロード中	「説明」タブのデフォルト・コマンドまたはローディング・コマンドの起動に使用されている接続。
コンテキスト	タスクの実行コンテキスト。
スキーマ	タスク実行の論理スキーマ。
接続	タスク実行のデータ・サーバー。
トランザクション	コマンドが実行されるトランザクション番号。
トランザクション分離	Select 命令のトランザクション分離レベル。トランザクション分離の詳細は、プロシージャ・コマンドを参照してください。
コミット	トランザクションのコミットを管理します。詳細は、プロシージャ・コマンドを参照してください。

実行	タスク実行の詳細な結果。
開始	タスク実行開始の日付と時刻。
終了	タスク実行終了の日付と時刻。
継続時間	タスク実行の所要時間。
状態	タスクの状態（完了、エラー、実行中、待機中、警告）。
リターン・コード	タスクのリターン・コード。
メッセージ	タスク実行エラー・メッセージ（必要な場合）。
挿入数	タスク実行時に挿入された行数。
削除数	タスク実行時に削除された行数。
更新数	タスク実行時に更新された行数。
エラー数	タスクでエラーになった行数。
行数	このタスク時に処理された行の総数。

## セッション変数

セッション変数には、実行セッションの際に変数がとる値が表示されます。

### 定義

プロパティ	説明
名前	変数の名前。変数名には、「GLOBAL」または変数を含むプロジェクトのコードを接頭辞として付ける必要があります。
説明	変数の詳細な説明。
データ型	変数のタイプ: 英数字、日付または数値（最大 10 桁）。
アクション	このパラメータは変数の値が保持される時間の長さを示します。 <ul style="list-style-type: none"> <li><b>非永続:</b> 変数の値はセッションを通じてメモリーに保持されます。</li> <li><b>最後の値:</b> 変数が最後にとった値が Oracle Data Integrator のリポトリに格納されます。</li> <li><b>履歴化:</b> この変数の履歴と、変数がとったすべての値が保持されます。</li> </ul>
デフォルト値	変数に割り当てられたデフォルト値。

### 履歴

このウィンドウには、変数の値の履歴が、log アクションの変数のコンテキスト、または last value アクションの変数の最後の値とともに表示されます。

**注意:** それぞれの値はセッションが実行されたコンテキストにアタッチされています。

-  ボタンを使用すると、履歴を 1 行削除できます。
- 「コンテキスト・フィルタ」フィールドを使用すると、指定されたコンテキストでエントリをフィルタできます。

## シナリオ

### シナリオ

シナリオは、ソース・コンポーネント（インタフェース、パッケージ、プロシージャ、変数）を実使用に投入するように設計されます。シナリオは、このコンポーネント用のコード（SQL、シェルなど）生成の結果として生じます。

**注意:** 生成されたシナリオは作業リポジトリに格納されます。シナリオは、エクスポートして別のリポジトリ（リモートまたは非リモート）にインポートし、異なるコンテキストで使用することができます。

**注意:** シナリオ・コード（生成された言語）は変更不能で、シナリオの作成に関係したコンポーネントがそれ以降に変更されても、シナリオは変更されません。

シナリオのプロパティは**デザイナ**・モジュールの**シナリオ**・ウィンドウの「**定義**」タブに表示されるか、**オペレータ**・モジュールを介して表示されます。シナリオの操作は、同一のウィンドウまたはオペレーティング・システムのコマンド・ウィンドウで行われます。デザイナ・モジュールでは、シナリオはそれぞれのソース・コンポーネントの配下にグループ化されます。

### プロパティ

プロパティ	説明
パッケージ、インタフェース、変数、プロシージャ	シナリオを作成するために使用されるコンポーネントの名前。
名前	シナリオの名前。
バージョン	シナリオのバージョン。
説明	シナリオの詳細な説明。

### 実行

このタブでは、「**実行**」ボタンを使用してシナリオを起動することができます。実行オプションを選択するウィンドウが開きます。

### 変数

このタブを使用すると、シナリオ変数を選択できます。シナリオ変数は、**Metadata Navigator** を使用してシナリオを起動する場合やスケジュールを作成する場合にパラメータとして表示されます。ここでパラメータ変数を選択すると、パラメータ化できないユーザー変数を非表示にすることができます。

**すべてを使用値**を選択すると、すべての変数がパラメータとみなされます。**選択的使用値**を選択すると、パラメータとして使用する変数を選択することができます。

## シナリオの操作

シナリオは、それが格納される作業リポジトリから運用されます。同一の作業リポジトリにある同一のシナリオを、いくつかのコンテキストで実行することができます。

シナリオでは次の操作が可能です。

- デザインまたはオペレータから起動できます。
- **Data Integrator** のスケジューラでスケジュールできます。
- オペレーティング・システムのコマンド・ライン・インタフェースから、Web サービスまたは HTTP URL を使用して起動できます。
- 任意のスケジューラを使用してスケジュールできます。
- 別の作業リポジトリにエクスポートして、類似の方法で起動またはスケジュールできます。「別のリポジトリでの操作」を参照してください。
- シナリオを実使用でインポートできます。

## シナリオ・レポート

シナリオ実行レポートは、シナリオの一般的な実行に関する情報を提供します。

### 定義

プロパティ	説明
実行エージェント	シナリオの実行を担当するエージェント。
状態	シナリオの状態（完了、エラー、実行中、待機中、警告）。
実行コンテキスト	シナリオの実行コンテキスト。
開始	シナリオ実行開始の日付と時刻。
終了	シナリオ実行終了の日付と時刻。
継続時間	シナリオ実行の所要時間。
リターン・コード	シナリオのリターン・コード。
メッセージ	シナリオ実行エラー・メッセージ（必要な場合）。

## 詳細

プロパティ	説明
挿入数	シナリオ実行時に挿入された行数。
削除数	シナリオ実行時に削除された行数。
更新数	シナリオ実行時に更新された行数。
エラー数	シナリオでエラーになった行数。
行数	シナリオによって処理された行の総数。注意: この値は前述の値の合計ではありません。

## 手順レポート

手順実行レポートは、シナリオ実行レポートにリンクされています。

## 定義

プロパティ	説明
名前	実行された手順の名前。
順序番号	このセッションの手順にリンクされた順序番号。

## 実行

プロパティ	説明
実行数	手順（ループ）の実行が成功した際の実行数。
実行	手順実行の詳細な結果を提供します。
開始	手順実行開始の日付と時刻。
終了	手順実行終了の日付と時刻。
継続時間	手順実行の所要時間。
状態	手順の状態（完了、エラー、実行中、待機中、警告）。
リターン・コード	手順のリターン・コード。
メッセージ	手順実行エラーメッセージ（必要な場合）。

挿入数	手順実行時に挿入された行数。
削除数	手順実行時に削除された行数。
更新数	手順実行時に更新された行数。
エラー数	手順でエラーになった行数。

## スケジューリング情報

スケジュール情報を使用すると、エージェントのスケジュール済タスクを視覚化できます。

**重要:** スケジューリング情報はエージェントのスケジュールから取得されます。正確なスケジューリング情報を表示するには、エージェントを起動してスケジュールをリフレッシュする必要があります。

プロパティ	説明
選択されたエージェント	スケジュールが表示されるエージェント。すべてのエージェントのスケジュールを表示することもできます。
スケジュールの開始と終了	スケジュールが表示される期間。このスケジュールをリフレッシュするには、  「リフレッシュ」ボタンをクリックします。
更新	このボタンを押すと、選択されたエージェントのスケジュールが更新されます。
時間範囲	期間を指定すると（1 時間、2 時間）、現在時刻にこの時間を加え、ダイアグラムをその中央に置くことができます。この機能により、進行中のセッションと到来するセッションを視覚化できます。矢印を使用して、範囲を順方向または逆方向に動かすことができます。
拡大、縮小	Gantt ダイアグラムにズーム・インします。ズームを使用してダイアグラム内の領域を選択することもできます。
ガント・ダイアグラム	このパネルには、選択されたエージェントのスケジュールが Gantt ダイアグラムとして表示されます。
シナリオ詳細	このパネルには、スケジュールされた各シナリオの詳細と実行統計情報が表示されます。

## ダイアグラムの使用

ダイアグラム内の領域を選択すると（マウス・ボタンを押したままにします）、選択したゾーン上に自動的にズームします。

ダイアグラムを右クリックすると、ズーム、ダイアグラムの画像ファイル形式での保存、および表示プロパティの印刷や編集を行うためのメニューがポップアップします。

# トポロジ・マネージャ

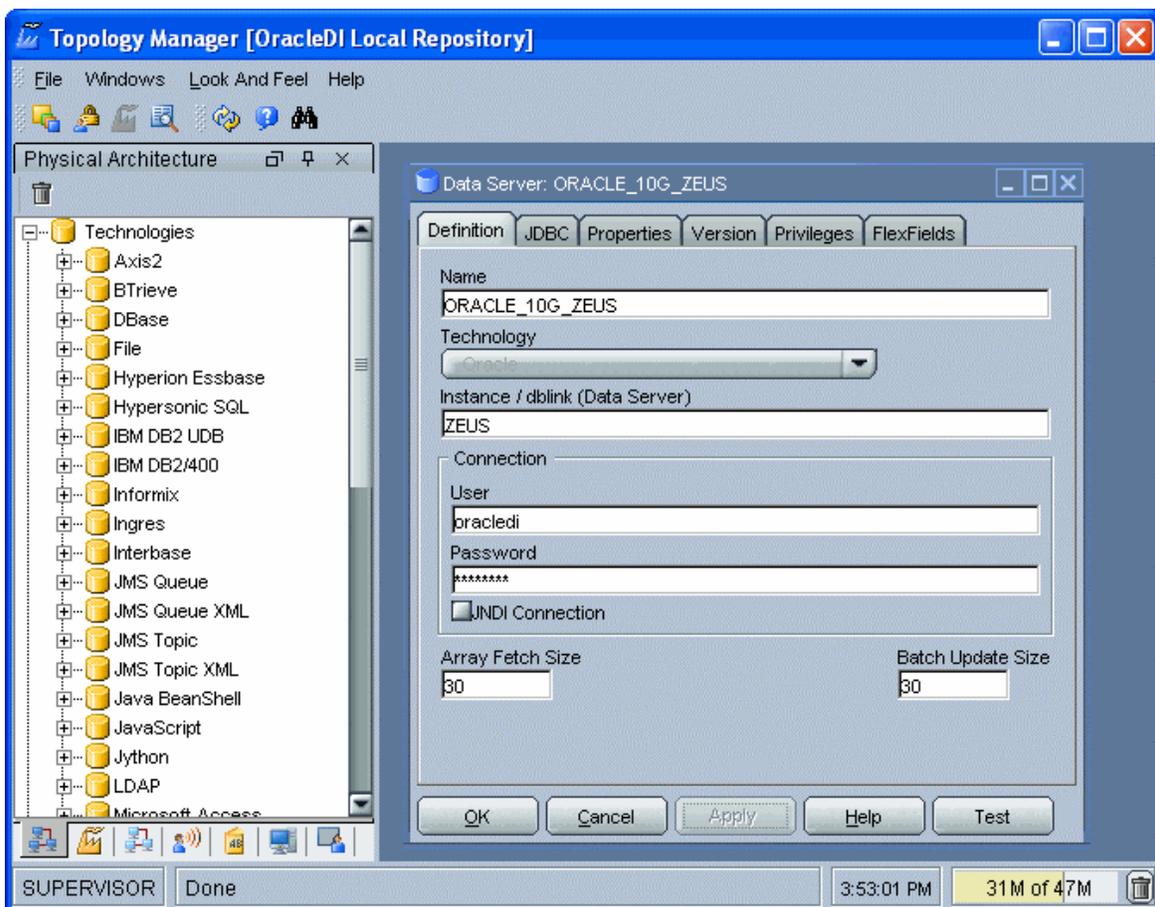
## トポロジ・マネージャの概要

トポロジ・マネージャ・モジュールを使用すると、情報システムのトポロジ、テクノロジーとそのデータ型、そのテクノロジーと含まれるスキーマにリンクされたデータ・サーバー、コンテキスト、言語およびエージェントを管理できます。さらに、トポロジを使用するとリポジトリも管理できます。

トポロジ・モジュールは、この情報をマスター・リポジトリに格納します。この情報は、他のすべてのモジュールで使用できます。

## トポロジ・マネージャのインタフェース

トポロジ・マネージャの GUI は次のように表示されます。



## メニュー

メニューでは、プルダウン・メニューから次の機能にアクセスできます。

- インポート/エクスポート
- ウィザード

- オプションの表示
- モジュールまたはツリー表示のオープン
- ユーザーのパスワードおよびオプションの変更

## ツールバー

ツールバーからは、次の操作を実行できます。

- 他のモジュールの起動
- ツリー表示のリフレッシュ
- オンライン・ヘルプの起動

## ツリー表示

現行ユーザーが使用できるトポロジ・マネージャ・オブジェクトは、次のようにツリー表示に整理して表示されます。

- **物理アーキテクチャ**。テクノロジーとそれに関連付けられたデータ・サーバーと物理スキーマ、および物理エージェントが含まれます。
- **論理アーキテクチャ**。テクノロジーとそれに関連付けられた論理スキーマ、および論理エージェントが含まれます。
- 論理アーキテクチャと物理アーキテクチャをリンクするコンテキスト。
- **言語**。使用できる様々なタイプの言語が記述されます。
- **リポジトリ**。実際のマスター・リポジトリと、アタッチされた作業リポジトリが含まれます。

各ツリー表示は、メイン・ウィンドウの両側にドッキングできるフローティング・フレームに表示されます。フレームは重ねることもできます。複数のフレームを重ねた場合は、フレーム・ウィンドウの下部に表示されるタブから各フレームにアクセスできます。

ツリー表示フレームは、フレームのタイトルまたはタブを選択してドラッグすることにより、移動したり、ドッキングしたり、重ねたりできます。ツリー表示の位置を固定するには、「ウィンドウ」メニューから「ウィンドウ・レイアウトのロック」を選択します。

ツリー表示フレームがメイン・ウィンドウに表示されないか、閉じている場合は、「ウィンドウ」→「ビューの表示」メニューを使用します。

各ツリー表示では、次の操作が可能です。

- ルート・オブジェクトの挿入またはインポート（フレーム・タイトルで該当するボタンをクリック）
- ノードの展開または折りたたみ（ノードをクリック）
- オブジェクトに関連付けられているメソッド（編集、削除など）のアクティブ化（ポップアップ・メニューを使用）
- オブジェクトの編集（オブジェクトをダブルクリック、またはワークベンチにドラッグ・アンド・ドロップ）

## ワークベンチ

編集または表示されているオブジェクトのウィンドウは、**ワークベンチ**に表示されます。

## トポロジの概要

トポロジ・マネージャ・モジュールを使用すると、Oracle Data Integrator を使用して作業するアーキテクチャやコンポーネントの正しい物理的および論理的な見取り図を思いのままに作成することができます。

- トポロジの作成

## 物理アーキテクチャ

物理アーキテクチャは、情報システムの様々な要素と、Oracle Data Integrator で考慮される特性を定義します。

テクノロジーは書式付きデータを処理します。このため、各テクノロジーは1つ以上のデータ型に関連付けられ、Oracle Data Integrator はそれを使用してデータ処理スクリプトを生成します。

**注意:** 各タイプのデータベース (Oracle、DB2 など)、ファイル形式 (XML、ファイル) またはアプリケーション・ソフトウェアは、Oracle Data Integrator ではテクノロジーによって表現されます。

データを格納して返す物理コンポーネントはデータ・サーバーと定義されます。ビジネス・ロジックに応じて異なった情報を格納できるデータ・サーバーは、いくつかの物理スキーマに分割できます。データ・サーバーは、常に単一のテクノロジーにリンクされます。

**注意:** Data Integrator で使用されるすべてのデータベース・サーバー、JMS メッセージ・ファイル、フラット・ファイルのグループなどは、データ・サーバーとして宣言する必要があります。

Oracle Data Integrator で使用されるすべてのスキーマ、データベース、JMS トピックなどは、物理スキーマとして宣言する必要があります。

最後に、物理アーキテクチャには物理エージェントの定義が含まれます。これは、Oracle Data Integrator ジョブをリモート・マシン上で実行できるようにする Java ソフトウェア・コンポーネントです。

## コンテキスト

コンテキストは、情報システムの物理アーキテクチャ (真のアーキテクチャ) のコンポーネントを、Oracle Data Integrator 論理アーキテクチャ (ユーザーが作業するアーキテクチャ) のコンポーネントに統合します。

## 論理アーキテクチャ

論理アーキテクチャは、ユーザーが、構造上同一であるが別個の場所にあるデータストアが含まれている物理スキーマを、テクノロジーごとに構造化されたいくつかの論理スキーマにグループ化することを可能にします。

同じ考え方で、論理アーキテクチャは論理エージェントを定義します。これは、異なったコンテキストで同じ機能を果たすすべての物理エージェントに一意の名前を付けられるようにします。

**例:** 次の2つの物理スキーマに対して、論理スキーマ Accounting を対応付けることができます。

- 開発コンテキストで使用される Accounting Oracle sample  
 - 本番コンテキストで使用される Accounting corporate  
 これら2つの物理スキーマは、構造的には同一（会計データを含む）ですが、物理的には異なります。これらの物理スキーマは、2つの Oracle スキーマ上に存在し、通常は2つの異なる Oracle サーバー（データ・サーバー）に配置されます。

## 言語

このタイプのコンポーネントは、テクノロジーにリンクされ、Oracle Data Integrator によって使用される各言語に固有の特性を定義します。

## リポジトリ

トポロジのこの部分には、マスター・リポジトリと作業リポジトリの2タイプのリポジトリに関連する情報が含まれます。

## ホスト

ホストと用例を使用すると、グラフィック・モジュールへのユーザー・アクセスを管理できます。

## 接続

### マスター・リポジトリへの接続

このウィンドウを使用すると、トポロジ・マネージャまたはセキュリティ・マネージャ・モジュールを開く際に、マスター・リポジトリへの接続を選択できます。既存の接続の作成、変更または削除を開始することもできます。

**注意:** マスター・リポジトリ接続の定義は、ローカルの  
 /bin/snps\_login\_security.xml ファイルに格納されています。

### 一般プロパティ

プロパティ	説明
ログイン名	マスター・リポジトリへの定義された接続のリスト。
ユーザー	選択されたマスター・リポジトリに接続するためのデフォルトの Oracle Data Integrator ユーザー。デフォルト・ユーザーが指定されなかった場合は、ここに入力する必要があります。
パスワード	接続するユーザーに対応するパスワード。デフォルトのパスワードが指定されなかった場合は、ここに入力する必要があります。

## ツールバー

ボタン	説明
	作成されるマスター・リポジトリへの新しい接続を許可します。
	「ログイン名」フィールドで選択された接続の変更を許可します。
	「ログイン名」フィールドで選択された接続の削除を許可します。

## マスター・リポジトリ接続の詳細

このウィンドウを使用すると、マスター・リポジトリへの接続を定義または変更できます。

**注意:** マスター・リポジトリ接続の定義は、ローカルの `/bin/snps_login_secu.xml` ファイルに格納されています。

### 一般プロパティ

プロパティ	説明
<b>Oracle Data Integrator 接続</b>	この情報グループを使用して、接続ウィンドウに表示されるプロパティを定義することができます。
ログイン名	Data Integrator モジュールが開いている場合、これがリストに表示される名前です。
ユーザー	マスター・リポジトリに接続するためのデフォルトの Oracle Data Integrator ユーザー。デフォルト・ユーザーが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。
パスワード	Oracle Data Integrator ユーザーに対応しているパスワード。デフォルトのパスワードが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。
<b>データベース接続 (マスター・リポジトリ)</b>	この情報グループは、マスター・リポジトリをホストするリレーショナル・データベースの接続パラメータを示します。
ユーザー	マスター・リポジトリが含まれているデータベースのスキーマへの接続を許可するユーザー。
パスワード	そのデータベース・ユーザーのデータ・サーバー・パスワード。
ドライバ・リスト	マスター・リポジトリが基づいているテクノロジーのタイプ。 ボタン  をクリックすると、使用されるテクノロジーとドライバに関する詳細情報を表示できます。詳細は、「ドライバの詳細」を参照してください

ドライバ名	い。 マスター・リポジトリのデータ・サーバーに接続するために使用する JDBC ドライバの名前。テクノロジーを選択すると、そのテクノロジーで最もよく使用されるドライバが表示されます。他のドライバを使用することもできます。その場合、そのドライバが適切にインストールされている必要があります。詳細は、インストール・ガイドの「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。
URL	マスター・リポジトリのデータ・サーバーに接続するための URL。ボタン  をクリックすると、使用するドライバの URL モデルを選択することができます。詳細は、「URL サンプル」を参照してください。ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。
デフォルトの接続	このボックスを選択すると、Oracle Data Integrator グラフィック・モジュールの起動時に、デフォルトで、現在の接続が提案されます。

## ドライバの詳細

このウィンドウには、使用するドライバとテクノロジーに関する情報が表示されます。

### 一般プロパティ

プロパティ	説明
名前	選択されたテクノロジーで使用するドライバ。
説明	ドライバについての情報。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## URL サンプル

このウィンドウには、ドライバの URL モデルが表示されます。

### 一般プロパティ

プロパティ	説明
名前	選択されたドライバ用の有効な URL モデルのリスト。
説明	URL モデルの詳細。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## 物理アーキテクチャ

### テクノロジー

#### テクノロジー

Oracle Data Integrator の用語法では、これは JDBC、ODBC、JMS、JNDI、JCA または任意のオペレーティングシステムでアクセス可能な任意のタイプのテクノロジーです。

Oracle Data Integrator では、リポジトリで定義されたテクノロジーと互換性のある言語でスクリプトを生成して実行することができます。任意のタイプのテクノロジーを宣言および定義できます。ただし、Oracle Data Integrator が、適切なミドルウェア (JDBC、ODBC、JMS) またはオペレーティング・システム・コマンドを使用してそれにアクセスできる必要があります。

テクノロジーの例: Oracle、Sybase、Sybase IQ、DB2、ファイルなど

#### 定義

プロパティ	説明
名前	グラフィカル・インタフェースの様々なリストに表示されるテクノロジーの名前。
コード	テクノロジー・コードを使用すると、異なるリポジトリの間でテクノロジーを参照することができます。  <b>注意:</b> このコードは変更しないでください。変更すると、別のリポジトリからテクノロジーを参照する場合、一部の Oracle Data Integrator オブジェクトが不整合になります。
テクノロジー・タイプ	テクノロジーの分類。次の値が可能です。 <ul style="list-style-type: none"> <li>• <b>データベースまたはファイル:</b> ODBC や JDBC からアクセス可能な任意のテクノロジー。Data Integrator にはフラット・ファイル用の JDBC ドライバや XML ファイル用の JDBC ドライバが含まれるため、フラット・ファイルと XML ファイルはこのカテゴリに含まれます。</li> <li>• <b>オペレーティング・システム:</b> Oracle Data Integrator が Java 仮想マシンを介してコマンドを投入できる任意のオペレーティング・システム。</li> <li>• <b>トピック (JMS):</b> JMS 経由でアクセス可能で、トピック管理 (パブリケーションとサブスクリプション) を含む任意の MOM。</li> <li>• <b>キュー (JMS):</b> JMS 経由でアクセス可能な任意の MOM。</li> <li>• <b>Oracle Data Integrator API:</b> Oracle Data Integrator ツールへのコールを可能にしているテクノロジー。</li> <li>• <b>Oracle Data Integrator コネクタ:</b> Java API へのコールを可能にしているテクノロジー。</li> <li>• <b>Bean スクリプト・フレームワーク:</b> スクリプト・インタプリタへのコールを可能にしているテクノロジー。</li> </ul>

論理/物理	<ul style="list-style-type: none"> <li>● <b>Web サービス・コンテナ:</b> Web サービスのデプロイを可能にしているテクノロジー。</li> </ul> <p>テクノロジーが物理的か論理的か、あるいはその両方かを定義します。 3つの場合が可能です。</p> <ul style="list-style-type: none"> <li>● <b>論理的かつ物理的テクノロジー:</b> テクノロジーが論理的かつ物理的である場合、論理スキーマと物理スキーマをサポートできます。論理スキーマは、<u>このテクノロジーのみの物理スキーマ上のコンテキスト</u>に限ってマップされます。</li> <li>● <b>論理的テクノロジー:</b> テクノロジーが論理的であって物理的でない場合、テクノロジーは論理スキーマのみをサポートでき、<u>任意のテクノロジーの物理スキーマ上のコンテキスト</u>でマップされます。 例: 論理的であって物理的でないテクノロジー <code>GENERIC_ISO_SQL</code> を定義する場合、その論理スキーマからは、<b>Oracle</b>、<b>DB2</b> などのデータ・サーバーにある物理スキーマにのみアクセスできます。</li> <li>● <b>物理的テクノロジー:</b> テクノロジーが物理的であって論理的でない場合、物理スキーマのみをサポートでき、そのスキーマには、論理的であって物理的でないテクノロジーの論理スキーマ経由でのみアクセスできます。</li> </ul>
データ処理	この項では、 <b>テクノロジーのデータ処理機能が詳しく説明されます。</b>
選択	問合せの結果としてデータ配列を構築するテクノロジーの機能。
条件	条件に応じてこの配列をフィルタするテクノロジーの機能。
参照	<p>複数の配列（2次元のデータ・セット）を結合して1つにするテクノロジーの機能。この機能にはいくつかの形式があります。</p> <ul style="list-style-type: none"> <li>● <b>なし:</b> テクノロジーは結合を受け入れません。</li> <li>● <b>順序なし:</b> 指定された結合が順序なしで使用されます。これは通常、SQL の Where タイプの句で、<code>where emp.empno = dept.deptno</code> という形式で指定される結合です。</li> <li>● <b>順序付(SQL ISO):</b> 指定される結合が順序付きで使用され、最終結果のデータ配列は、基本集合を順序に従って加えることで構築されます。この構文は、たとえば SQL 言語の From 句で、指定された ISO の構文に合されます。たとえば、<code>FROM emp INNER dept ON (emp.empno = dept.deptno)</code> のようになります。</li> </ul>
命名規則	この項は、このテクノロジーが <b>データ・コンテナ</b> を見つけ、名前を付ける方法を示します。
ファイル	このテクノロジーがファイルの使用に基づくかどうかを示します。
"データ・サーバー"を使用中	<p>テクノロジーが、オブジェクトの命名にその名前を使用できるデータ・サーバーを使用することを示します。ボックスを選択すると、テクノロジーのデータ・サーバーを記述するために使用される用語を指定する必要があります。</p> <p>たとえば、「Oracle」テクノロジーでは、データ・サーバーは</p>

	「Instance/DBLink」によってアクセスされます。
"カタログ"を使用中	<p>テクノロジーがカタログを使用してデータストアに名前を付けることを示します。ボックスを選択すると、テクノロジーでカタログの概念を記述するために使用される用語を指定する必要があります。</p> <p>たとえば、Microsoft SQL Server は「Database」という名前のカタログを使用し、IBM AS/400 は「Library」という名前のカタログを使用します。Oracle はカタログを使用しません。</p>
"スキーマ"を使用中	<p>テクノロジーがスキーマを使用してデータストアに名前を付けることを示します。ボックスを選択すると、テクノロジーでスキーマの概念を記述するために使用される用語を指定する必要があります。</p> <p>たとえば、Microsoft SQL Server は「Owner」という名前のスキーマを使用し、Oracle は「Schema/User」という名前のスキーマを使用します。Microsoft Access はスキーマを使用しません。</p>
ローカル・オブジェクト・マスク	<p>ローカル・オブジェクト・マスクは、オブジェクトがあるデータ・サーバーに接続しているときに、オブジェクトがどのように命名されるかを定義します。</p> <p>たとえば、Oracle では、ユーザーSCOTT に属する表 EMP にアクセスする構文 SCOTT.EMP を表すのに、%SCHEMA.%OBJECT を示す必要があります。</p> <p>Oracle Data Integrator でこのマスクに使用できるタグは次のとおりです。</p> <ul style="list-style-type: none"> <li>• %CATALOG: カタログ名（物理スキーマのプロパティ）を表します。</li> <li>• %SCHEMA: スキーマ名（物理スキーマのプロパティ）を表します。</li> <li>• %OBJECT: データストア名を表します。</li> </ul> <p><b>注意:</b> これらのタグでは大文字と小文字が区別されます。</p>
リモート・オブジェクト・マスク	<p>リモート・オブジェクト・マスクは、オブジェクトがあるのと異なるデータ・サーバーに接続しているときに、オブジェクトがどのように命名されるかを定義します。</p> <p>たとえば、Oracle では、インスタンス（リモート・データ・サーバー）NYORK のユーザーSCOTT の表 EMP にアクセスする構文 SCOTT.EMP@NYORK を表すのに、%SCHEMA.%OBJECT@DSERVER を示す必要があります。</p> <p>このマスクで使用可能なタグはローカル・オブジェクトと同じですが、データ・サーバー（接続のプロパティ）の名前を表す%DSERVER を付加する必要があります。</p>

## SQL

SQL プロパティは、データ・フィルタリング機能や、結合フィルタリング機能（Where）のあるテクノロジーのために確保されています。

プロパティ	説明
-------	----

条件	この項では、フィルタ句で使用する構文について詳しく説明します。データ配列フィルタリング機能のあるテクノロジーでのみ表示されます。
ORDER BY	そのテクノロジーにおいて、SQL で ORDER BY 句に続けて使用される構文のタイプを表示します。次の値が可能です。 <ul style="list-style-type: none"> <li>● <b>複合式:</b> 句の次に複合式または列名が続きます。</li> <li>● <b>列番号:</b> ORDER BY 句の次に列番号が続きます。</li> <li>● <b>別名:</b> ORDER BY 句の次に列エイリアスが続きます。</li> </ul>
GROUP BY	そのテクノロジーにおいて、SQL で GROUP BY 句に続けて使用される構文のタイプを表示します。可能な値は ORDER BY 句の場合と同じです。
HAVING	そのテクノロジーにおいて、SQL で HAVING 句に続けて使用される構文のタイプを表示します。可能な値は ORDER BY 句の場合と同じです。
順序付結合(ISO)	この項では、順序付き結合 (SQL ISO) で使用する構文について詳しく説明します。結合順序付け機能のあるテクノロジーでのみ表示されます。
句の場所	結合句が問合せ構文の FROM レベルにあるか WHERE レベルにあるかを示します。
ON 句でサポートされている大カッコ	テクノロジーが、かっこで区切られたサブセット (SQL ISO 構文の場合) を受け入れるかどうかを示します。
内部	テクノロジーが内部結合をサポートすることを示します。内部結合を指定するキーワード (たとえば「INNER JOIN」) を指定する必要があります。
相互	テクノロジーがデカルト積をサポートすることを示します。積集合を指定するキーワード (たとえば「CROSS JOIN」または「,」) を指定する必要があります。
左外部	テクノロジーが左側外部結合をサポートすることを示します。左側外部結合を指定するキーワード (たとえば「LEFT OUTER JOIN」) を指定する必要があります。
右外部	テクノロジーが右側外部結合をサポートすることを示します。右側外部結合を指定するキーワード (たとえば「RIGHT OUTER JOIN」) を指定する必要があります。
完全外部	テクノロジーが完全外部結合をサポートすることを示します。完全外部結合を指定するキーワード (たとえば「FULL OUTER JOIN」) を指定する必要があります。
順序なし結合	順序なし結合のパラメータを使用すると、外部結合句を WHERE 句で生成することができます。
外部キーワード	要素が外部であることを示す目印として構文に挿入されるキーワードまたは記号。Oracle では、たとえば外部を示すキーワードは(+)です。

外部の場所	<b>外部キーワード</b> が、結合構文の外部（OUTER 側）と内部の（INNER 側）のどちらにあるかを示します。
列の前、列の後	結合に関与しているオブジェクトの名前の前か後に <b>外部キーワード</b> を置くかどうかを示します。
特定の問合せ	<b>特定のタスクを実行するために Oracle Data Integrator が使用する問合せ</b> 。これらの問合せはテクノロジー固有です。
索引リバース	索引のリバース・エンジニアリングに使用する問合せ。この問合せは次の VARCHAR フィールドが含まれているレコードセット（索引列あたり 1 行）を返します。 <ul style="list-style-type: none"> <li>INDEX_NAME: 索引の名前。</li> <li>COLUMN_NAME: 索引列の名前。</li> </ul> レコードセットは索引の INDEX_NAME と、COLUMN_NAME の位置を基準に並べられます。
制約リバースエンジニアリングのチェック	チェック制約のリバース・エンジニアリングに使用する問合せ。この問合せは次の VARCHAR フィールドが含まれているレコードセット（チェック制約あたり 1 行）を返します。 <ul style="list-style-type: none"> <li>CHECK_NAME: チェック制約の名前。</li> <li>CHECK_TXT: 制約の SQL 文。</li> <li>CHECK_STATUS: チェック制約のステータス。1: 有効、0: 無効。</li> <li>CHECK_DESC: チェック制約の説明。</li> </ul>
代替キー・リバース	代替キーのリバース・エンジニアリングに使用する問合せ。この問合せは次の VARCHAR フィールドが含まれているレコードセット（代替キーあたり 1 行）を返します。 <ul style="list-style-type: none"> <li>AK_NAME: 代替キーの名前。</li> <li>COLUMN_NAME: 代替キー列の名前。</li> </ul> レコードセットは代替キーの AK_NAME と、COLUMN_NAME の位置を基準に並べられます。
現在の日付	このテクノロジーのデータ・サーバーにいつでも送信でき、サーバーの現在の日付と時刻を返す問合せ。この問合せは接続を回復するために内部で使用されます。

## その他

プロパティ	説明
列-別名セパレータ	SQL SELECT 句で、列をそのエイリアスから分離するために設計された要素。 ISO 標準ではこのセパレータは「AS」ですが、すべてのデータベースでサポートされているわけではなく、省略可能です。

表-別名セパレータ	SQL FROM 句で、表をそのエイリアスから分離するために設計された要素。この要素は省略可能です。
日付関数	日付と時刻を返すファンクション。 例: Microsoft SQL Server の場合は <code>getdate()</code> 、Oracle では <code>sysdate</code> 。
文字列データ型マスク	DDL 命令 (表作成) で、連鎖タイプのデータを記述するために使用する構文。構文記述では、Oracle Data Integrator のタグ <code>%L</code> (データ長) と <code>%P</code> (データ精度) を使用できます。
日付データ型マスク	DDL 命令 (表作成) で、日付タイプのデータを記述するために使用する構文。
数値データ型マスク	DDL 命令 (表作成) で、数値タイプのデータを記述するために使用する構文。構文記述では、Oracle Data Integrator のタグ <code>%L</code> (データ長) と <code>%P</code> (データ精度) を使用できます。
DDL Null キーワード	空の値を含むことができる列 (NULLABLE と呼ばれる列) を記述するために使用する語。
最大列名長	列名の最大長。単位は文字数。Oracle Data Integrator によって生成されたすべての列名は、このサイズに切り捨てられます。
最大表名長	データストア (表) 名の最大長。単位は文字数。Oracle Data Integrator によって生成されたすべての表名は、このサイズに切り捨てられます。

## 言語

このタブでは、テクノロジーによって実装されている言語が記述され、言語の任意のプロパティを上書きすることができます。

プロパティ	説明
言語	テクノロジーによって、全体または部分が実装されている言語。定義された言語のリストから選択されたものです。
互換性	そのテクノロジーの言語が、 <b>標準</b> とマークされたすべての言語要素を継承していることを示します。詳細は、「言語要素」を参照してください。
デフォルト	これが、このテクノロジーのデフォルト言語であることを示します。
オブジェクト・デリミタ	オブジェクト名を区切るために使用する文字。言語で指定された標準のデリミタに代わって使用されます。 <b>注意:</b> 2つのデリミタ (オブジェクト名の前と後) が異なっている場合は、その両方を使用します。それ以外の場合は、1つのデリミタで十分です。次のようなコードを生成する場合は二重引用符を1つ使用します。 <code>create table MySchema."My_Table" ( ...</code> 次のようなコードを生成する場合は角括弧[]を2つ使用します。

	<code>create table MySchema.[My_Table] ( ...</code>
単語セパレータ	言語の単語（オブジェクト、キーワード、リテラル）を分離するために使用できる文字。
リテラル・デリミタ	リテラル名（値）を区切るために使用する文字。言語で指定された標準のデリミタに代わって使用されます。 <b>注意:</b> 2つのデリミタ（リテラルの前と後）が異なっている場合は、その両方を使用します。それ以外の場合は、1つのデリミタで十分です。
オブジェクトの大/小文字を区別	このテクノロジーのオブジェクト名では、大文字と小文字が区別されます。たとえば、SQL 言語では、デフォルトでオブジェクト名の大文字と小文字は区別されません。このため、大文字と小文字をそのまま使用するために、次のようなコードが生成されます。  <code>create table Snps Temp.dbo."My Table" ( "My Column1" VARCHAR(20) NULL, MY COLUMN2 VARCHAR(20) NULL, )</code> 「オブジェクトの大/小文字を区別」を選択すると、次のようなコードが生成されます。  <code>create table Snps Temp.dbo.My Table ( My Column1 VARCHAR(20) NULL, MY COLUMN2 VARCHAR(20) NULL )</code>
単語の大/小文字を区別	この言語のキーワードでは大文字と小文字が区別されます。

テクノロジーでは次の操作を行うことができます。

- データ型の自動リバース

## データ型

書式付きデータを格納するテクノロジーでは、それぞれのデータに、その特性を定義する型が割り当てられます。例: 数値、文字、日付など

一部のデータ・サーバー・アクセス・ドライバでは、**テクノロジー・ウィンドウの「リバース」** ボタンを使用して、データ型を自動的にリバースすることができます。

## 定義

プロパティ	説明
コード	Oracle Data Integrator で、このデータ型を参照するために使用されるコード。このコードはこのテクノロジーでは一意です。コードは、データ型を参照するためにモデルとフローに格納されます。

名前	グラフィカル・インタフェースに表示されるデータ型の名前。
リバース・コード	ドライバや SQL リバース問合せによって返されるコード。
表作成構文	このタイプの列を作成するために使用される構文。この構文では、タグ%L（長さ）と%P（精度）を使用できます。
書込み可能なデータ型構文	この代替構文は、書込み不可能なデータ型のデータを格納する列を作成するために使用されます。この構文では、タグ%L（長さ）と%P（精度）を使用できます。  型が書込み可能の場合、書込み可能なデータ型構文と表作成構文は同じになります。
アイコンに使用される文字	データ型（このデータ型のグラフィック ID コード）を表すために使用される文字。
書込み可能	このデータ型を INSERT または UPDATE コマンドで使用できる場合に、このボックスを選択します。これは、たとえば IDENTITY 列など、値が自動設定され、変更できないデータ型の場合です。  書込み可能でないデータ型でリバースされた列には、読取り専用のフラグが付けられます。  書込み可能でないデータ型には代替の書込み可能構文が定義され、Oracle Data Integrator でこのタイプのデータを格納するための一時表を作成することができます。

## 変換先

「**変換先**」タブを使用すると、現在の型に対応する、他のテクノロジ用のデータ型を指定できます。このデータ型が使用されるのは、Oracle Data Integrator が、現在のテクノロジにおけるデータストアを反映する作業表を別のテクノロジ上に作成する必要がある場合です。

例:

- Oracle のデータ型 VARCHAR2 は、Microsoft SQL Server データ・サーバー上では VARCHAR に変換されます。

Oracle インスタンスの SQL Server インスタンスへのデータ・フローでは、言語生成エンジンには、両方向の変換機能が必要です（たとえば Oracle から SQL Server に、そして SQL Server から Oracle に）。

## 変換元

このタブ（読出し専用）は、現在のデータ型に対応する他のテクノロジのデータ型を示します。

## データ・サーバー

データ・サーバーは、表形式でデータを格納および再生するデータ処理リソースです。データ・サーバーとして指定できるのは、データベース、MOM、コネクタ、ファイル・サーバーのいずれかです。

データ・サーバーは、ただ1つのテクノロジーと必ずリンクされます。データ・サーバーは物理的に識別可能で、TCP/IP ネットワークでアクセスできる物理マシン上にあります。

**注意:** 各データ・サーバーの定義は1度だけにしよう強くお勧めします。同一のデータ・サーバーを複数回（場合によっては異なる接続パラメータを使用して）宣言すると、Oracle Data Integrator に不必要なデータ・ロード・フェーズが発生し、実行時間を浪費することがあります。

## 定義

定義タブには次のフィールドが含まれます。ただし、Axis2 など、Web サービス・コンテナであるテクノロジーに基づくデータ・サーバーの場合を除きます。

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるデータ・サーバーの名前。  <b>注意:</b> データ・サーバーを命名する際には、<TECHNOLOGY_NAME>_<SERVER_NAME>などの体系的な命名法を使用することをお勧めします。
テクノロジー	データ・サーバーにリンクしているテクノロジー。
(データ・サーバー)	これはデータ・サーバーの物理名です。データ・サーバーをネイティブの方法で相互接続できる場合に、この名前を定義します。  たとえば、Oracle では、別の Oracle データ・サーバーからこのデータ・サーバーにアクセスするために使用されるデータベース・リンクの名前がここに表示されます。
接続	このセクションには、そのデータ・サーバーでの認証方式の詳細が表示されます。
ユーザー	データ・サーバーに接続するために使用するユーザー名。  テクノロジーにより異なりますが、これは「Login」、「User」または「account」のいずれかです。JNDI プロトコルを使用している一部の接続では、ユーザー名とそれに関連付けられたパスワードはオプションです（LDAP ディレクトリで指定されている場合）。
パスワード	ユーザー名とリンクされたパスワード。  <b>注意:</b> このパスワードは暗号化されてリポジトリに格納されます。
JNDI 接続	一部のテクノロジーは、LDAP ディレクトリを介して、接続を受け入れるか、禁じるか、強制します。LDAP を介してデータ・サーバーに接続する、またはすることができる場合に、このボックスを選択します。
配列フェッチ・サイズ	データ・サーバーを使用する各通信で Data Integrator がリクエストする行数（読み取られたレコード）。

バッチ更新サイズ	Oracle Data Integrator の INSERT コマンド 1 つにおける行（書き込まれたレコード）の数。
	<p><b>注意:</b> フェッチ配列パラメータと一括更新パラメータは、JDBC でのみアクセス可能です。ただし、すべての JDBC ドライバが同一の値を受け入れるわけではありません。場合によっては空のままにしておくことをお勧めします。</p> <p><b>フェッチ配列と一括更新に関する注意:</b> それぞれの値に指定される数値が大きくなるほど、データ・サーバーと Oracle Data Integrator の間の交換の数は小さくなります。ただし、1 回の交換で回復されるデータ量が大きくなるため、Oracle Data Integrator マシン上の負荷は大きくなります。</p> <p>一括更新の管理は、フェッチ配列の管理と同様、最適化に含まれます。デフォルト値（30）から始め、パフォーマンスがそれ以上改善されなくなるまで、一度に 10 ずつ値を増やすことをお勧めします。</p>

Web サービス・コンテナの「定義」タブは次のように表示されます。詳細は、「データ・サービスの設定」を参照してください。

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるデータ・サーバーの名前。 <b>注意:</b> データ・サーバーを命名する際には、<TECHNOLOGY_NAME>_<SERVER_NAME>などの体系的な命名法を使用することをお勧めします。
公開されたサービスのベース URL	Web サービスがデプロイされるベース URL。この URL は、WSDL を生成する際に使用されます。
配置オプション	この項では、このコンテナ内の Web サービスのデプロイメント方式が詳しく記述されます。
Web サービスを次のフォルダに保存	これを選択すると、Web サービスは、ファイル・コピーを使用してターゲット・ディレクトリにデプロイされます。ディレクトリは、Web サービス生成を実行するマシンからアクセス可能です。
Axis2 による Web サービスのアップロード	これを選択すると、Web サービスは、Axis2 の Web サービス・アップロード機能を使用してターゲット・ディレクトリにデプロイされます。このアプリケーションに接続するには、ユーザー名とパスワードの他に、Axis2 webspapp のベース URL を指定する必要があります。
Axis2 Web アプリケーションのベース URL	Axis2 アプリケーションの HTTP URL。通常は、http://<Tomcat サーバーの名前>:<HTTP ポート>/axis2/axis2-admin/です。
ユーザー名、ユーザー・パスワード	Axis2 サーバーに接続するためのユーザー名とパスワード。このユーザーは通常、Axis2 管理者です。

ード	
Axis2 による Web サービスのアップロード	これを選択すると、Web サービスが FTP アップロードを使用してデプロイされます。ターゲット FTP URL と、有効な FTP ユーザーを指定する必要があります。
FTP サーバー URL	Oracle Data Integrator が Web サービスをデプロイする FTP URL。
ユーザー名、ユーザー・パスワード	FTP サーバーに接続するためのユーザー名とパスワード。このユーザーは、FTP URL への書き込み権限を持っている必要があります。

## JDBC

JDBC 直接接続を使用すると、LDAP ディレクトリを使用しないでデータ・サーバーにアクセスすることができます。

このタブが表示されるのは、「定義」タブの「JNDI 接続」ボックスが選択されていない場合だけです。

プロパティ	説明
JDBC ドライバ	データ・サーバーに接続するために使用する JDBC ドライバの名前。 ドライバの詳細は、「JDBC URL のサンプル」を参照してください。 JDBC ドライバのインストールの詳細は、インストール・ガイドの「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。
JDBC URL	データ・サーバーに接続するために使用するユーザー名。この URL を使用すると、データ・サーバーを TCP/IP ネットワーク上に置くことができます。 各ドライバに固有のドライバ構文があります。 URL の詳細は、「JDBC URL のサンプル」を参照してください。

## JNDI

JNDI 接続とは、LDAP、Novell Netware NDS、CORBA ネーミング・サービスおよびファイルシステムなどのネーミング・サービスまたはディレクトリ・サービスにある接続情報（ドライバ、URL、必要な場合はユーザーとパスワード）をフェッチすることです。このネーミング・サービスまたはディレクトリ・サービスは、JNDI プロトコル上の URL によってアクセスできます。JNDI 接続は一部のデータベース（JDBC）で使用でき、MOM（JMS）にアクセスするためには必須です。

このタブが表示されるのは、「定義」タブの「JNDI 接続」ボックスが選択されている場合だけです。

プロパティ	説明
JNDI 認証	<ul style="list-style-type: none"> <li>なし: ネーミング・サービスまたはディレクトリ・サービスへの匿名アクセス</li> <li><b>簡易</b>: 暗号化されていない、認証されたアクセス</li> <li><b>CRAM-MD5</b>: MD5 で暗号化された、認証されたアクセス</li> <li><b>&lt;その他の値&gt;</b>: &lt;その他の値&gt;で暗号化された、認証されたアクセス</li> </ul>
JNDI ユーザー	ネーミング・サービスまたはディレクトリ・サービスに接続しているユーザー
パスワード	ネーミング・サービスまたはディレクトリ・サービスに接続しているユーザーのパスワード
JNDI プロトコル	<p>接続で使用するプロトコル</p> <p><b>注意:</b> ここでは最も一般的なプロトコルのみを示しています。これは完全な一覧ではありません。</p> <ul style="list-style-type: none"> <li><b>LDAP</b>: LDAP ディレクトリへのアクセス</li> <li><b>SMQP</b>: SwiftMQ MOM ディレクトリへのアクセス</li> <li><b>&lt;その他の値&gt;</b>: サブプロトコル&lt;その他の値&gt;に従ったアクセス</li> </ul>
JNDI ドライバ	<p>JNDI 接続を実現しているドライバ</p> <p>Sun LDAP ディレクトリ用の例:  <code>com.sun.jndi.ldap.LdapCtxFactory</code></p>
JNDI URL	<p>JNDI 接続を可能にしている URL</p> <p>例: <code>ldap://suse70:389/o=linuxfocus.org</code></p>
JNDI リソース	<p>接続パラメータが含まれているディレクトリ要素</p> <p>例: <code>cn=sampladb</code></p>

## プロパティ

これらのプロパティは、接続の作成時に、オプションの構成パラメータを入力するために渡されます。各プロパティは、キーと値の組合せです。

- JDBC の場合: これらのプロパティは使用するドライバにより異なります。使用可能なプロパティのリストは、ドライバのマニュアルを参照してください。
- JNDI の場合: これらのプロパティは使用するリソースにより異なります。

プロパティ	説明
キー	このプロパティを識別するキー。このキーでは、大文字と小文字が区別されません。
値	プロパティの値。

データ・サーバーでは次の操作を行うことができます。

- データ・サーバー接続のテスト

## 物理スキーマ

物理スキーマは、データ・サーバーを分解したもので、データストア（表、ファイルなど）の分類を可能にします。この分類モードでデータ・サーバーに格納されたオブジェクトには、オブジェクト名にアタッチされたスキーマの名前を指定することによって、アクセスできます。

例:

- Oracle では、表はスキーマ（またはユーザー）によって分類されます。各表はスキーマにリンクされています。したがって、SCOTT.EMP はスキーマが SCOTT の表 EMP を表します。
- Microsoft Access にはスキーマがありません。
- DB2/400 には「ライブラリ」と呼ばれるスキーマがあります。
- Microsoft SQL Server には、各データベースについて、「所有者」と呼ばれるスキーマがあります。この所有者には、デフォルトで、「dbo」という名前が付けられます。HR.dbo.EMP という形式を使用すると、データベース HR 内の、ユーザーdbo に属している表 EMP にアクセスできます。

**注意:** スキーマが本当でないテクノロジーでも、デフォルトの物理スキーマ作成手順を通過する必要があります。

**注意:** データ・サーバーのデータにアクセスするには、プロジェクトで使用されるデータストアを含むすべてのスキーマが宣言されている必要があります。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示される物理スキーマの名前。自動的に計算されます。
(スキーマ)	データ・サーバーのスキーマの名前。要求されるデータが格納されるスキーマ、所有者またはライブラリ。  <b>注意:</b> Oracle Data Integrator では、データ・サーバーにあるすべてのスキーマがリストされます。ただし、このリストを作成できない場合があります。その場合は、適切なスキーマ名を入力します。
(作業スキー	一部のデータ検証またはトランスフォーメーション処理では、作業オブジェクト

マ)	<p>の作成が必要な場合があります。それらのオブジェクトを内部に作成するスキーマを示します。</p> <p><b>注意:</b> Oracle Data Integrator では、データ・サーバーにあるすべてのスキーマがリストされます。ただし、このリストを作成できない場合があります。その場合は、適切なスキーマ名を入力します。</p> <p><b>注意:</b> どんな作業表にも対応できる固有のスキーマを作成しておくくと便利です。すべてのデータ・サーバーで「SAS」または「ODI」という名前のスキーマを作成しておくくと、すべての Oracle Data Integrator アクティビティをアプリケーションから完全に独立した状態に保てます。</p>
デフォルト	このボックスを選択すると、スキーマが指定されなかった場合、物理スキーマがデータ・サーバーのデフォルトのスキーマになります。デフォルトとマークできるのは、1つの物理スキーマだけです。各データ・サーバーのデフォルト・スキーマはアイコン  で示されます。
作業表接頭辞	この項では、この物理スキーマの作業スキーマ内に Oracle Data Integrator が作成する可能性が高い作業表の接頭辞について詳しく記述します。
エラー	誤っているデータが含まれる表を作成するために使用される接頭辞。この表は、データの品質管理の際に作成されたり更新されたりするもので、グラフィック・インタフェースから参照することができます。
ロード中	2つのデータ・サーバーの間のデータ・ローディングを可能にするオブジェクト（表、ビュー、ファイルなど）の作成に使用される接頭辞。
統合	インタフェース実行時のデータ統合専用のオブジェクト（表、ファイルなど）を作成するために使用される接頭辞。
ジャーナル化要素接頭辞	この項では、Data Integrator がこのスキーマでのジャーナル化のための要素を作成するために使用する可能性が高い接頭辞について詳しく記述します。
表	ジャーナル化表（変更マークを含む）を作成するために使用される接頭辞。
ビュー	ジャーナル化表とデータ表をリンクするビューを作成するために使用される接頭辞。
トリガー	ジャーナル化表の更新を可能にするトリガーをデータ表上に作成するために使用される接頭辞。
命名規則	この項は、このテクノロジーがデータ・コンテナを見つけ、名前を付ける方法を示します。
ローカル・オブジェクト・マスク	<p>ローカル・オブジェクト・マスクは、オブジェクトがあるデータ・サーバーに接続しているときに、物理スキーマ内のオブジェクトがどのように命名されるかを示します。</p> <p>たとえば、Oracle では、ユーザーSCOTT に属する表 EMP にアクセスする構文 SCOTT.EMP を表すのに、%SCHEMA.%OBJECT を示す必要があります。</p> <p>Oracle Data Integrator でこのマスクに使用できるタグは次のとおりです。</p>

- %CATALOG: カタログ名（物理スキーマのプロパティ）を表します。
- %SCHEMA: スキーマ名（物理スキーマのプロパティ）を表します。
- %OBJECT: データストア名を表します。

**注意:** これらのタグでは大文字と小文字が区別されます。

リモート・オブジェクト・マスク  
リモート・オブジェクト・マスクは、このオブジェクトがあるのと異なるデータ・サーバーに接続しているときに、物理スキーマ内のオブジェクトがどのように命名されるかを示します。

たとえば、Oracle では、インスタンス（リモート・データ・サーバー）NYORK のユーザーSCOTT の表 EMP にアクセスする構文 SCOTT.EMP@NYORK を表すのに、%SCHEMA.%OBJECT@DSERVER を示す必要があります。

このマスクで使用可能なタグはローカル・オブジェクトと同じですが、データ・サーバー（接続のプロパティ）の名前を表す%DSERVER を付加する必要があります。

## コンテキスト

プロジェクトでは、物理スキーマのデータストアにアクセスする場合は必ず、論理スキーマとコンテキストを指定します。

**注意:** Oracle Data Integrator で物理スキーマを使用できるためには、指定されたコンテキストの物理スキーマと関連付けられている必要があります。

プロパティ	説明
コンテキスト	この物理スキーマが表現されているコンテキストを示します。
論理スキーマ	指定されたコンテキストの物理スキーマにアクセスするために使用する論理スキーマの名前を示します。  <b>注意:</b> リストに適切な論理スキーマ名がない場合、新しい名前を入力すると、新しい論理スキーマが自動的に作成されます。

## アクション

### アクション・グループ

アクション（DDL コマンド用のテンプレート）はアクション・グループに整理されます。アクション・グループは、指定された構文や目的（Oracle、SQL-92 など）に対応しています。

## 定義

プロパティ	説明
アクション・グループ名	グラフィカル・インタフェースに表示されるアクション・グループの名前。
グループ・コード	グループ・コードを使用すると、異なるリポジトリの間でアクション・グループを参照することができます。  <b>注意:</b> このコードは変更しないでください。変更すると、一部のオブジェクトが不整合になります。
デフォルト・グループ	このテクノロジーで作成される新しいモデルに対してこのアクション・グループが選択されるようにするには、「 <b>デフォルト・グループ</b> 」ボックスを選択します。指定されたテクノロジーに対してデフォルトのアクション・グループが選択されていない場合、新しいモデルには汎用のアクションが使用されます。
説明	アクション・グループの詳細な説明。

## アクション

アクションは、データ定義言語（DDL）コマンドのテンプレートです。アクションは、共通フォーマット・デザイナーによって使用されます。具体的には、データ・サーバーにデータ・モデルを実装するためのスクリプトを生成する場合や、Oracle Data Integrator 内に記述されているデータ・モデルと、データ・サーバー内での実装との差異を同期するためのスクリプトを生成する場合に使用されます。

各アクションは、単一の DDL 操作（表の作成、参照の削除など）に対応します。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるアクションの名前。
タイプ	アクションによって実行する操作のタイプ。アクションのタイプのリストは次を参照してください。
説明	アクションの詳細な説明。

### アクション・タイプ

次のアクション・タイプが用意されています。

- <不明>
- 追加
  - 代替キー

- チェック制約
- 列
- 外部キー
- 索引
- 主キー
- 開始
- 変更
  - 列属性
  - キー・タイプ
  - 列のコメント
  - 表のコメント
- 表の作成
- キーの無効化
- 削除
  - 代替キー
  - チェック制約
  - 列
  - 外部キー
  - 索引
  - 主キー
  - 表
- キーの有効化
- 終了
- 名前の変更
  - 列
  - 表

## 詳細

各アクションには、DDL 操作を実行するために必要なコマンドに対応する、いくつかの**アクション行**が含まれます（たとえば、表を削除するには、先にその制約をすべて削除する必要があります）。「詳細」タブにはそのようなアクション行のリストが表示されます。このリストは、グリッドの横のボタンを使用して、作成、削除および整理することができます。既存のアクションを編集するには、グリッドで行をダブルクリックします。アクション行を複製するには、右クリックして、「複製」を選択します。

## アクション行

各アクションは、単一の DDL 操作（表の作成、参照の削除など）に対応します。各アクションには、DDL 操作を実行するために必要なコマンドに対応する、いくつかの**アクション行**が含まれます（たとえば、表を削除するには、先にその制約をすべて削除する必要があります）。

## 定義

プロパティ	説明
名前	アクション行の名前。生成されるプロシージャ・コマンドにはこの名前が付けられます。
エラーの無視	この特定のコマンドがエラーを返してもプロシージャを停止させない場合には、このオプションを選択しておく必要があります。（たとえば、存在しない表を削除すると、処理全体がブロックされます）
コミット	コマンド終了時にコミットを発行するかどうかを示します。
分離レベル	<p>コマンドのトランザクション分離レベルです。</p> <p>分離レベルは次のとおりです。</p> <ul style="list-style-type: none"> <li>• 非コミット読取り: トランザクションは、別のトランザクションがコミットしていないデータを読み取れます。</li> <li>• コミット読取り: トランザクションは、他のトランザクションがコミットしたデータのみを読み取れます（一般には、これが多くのデータ・サーバーのデフォルトのモードです）。</li> <li>• 反復可能読取り: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更およびコミットされていても、必ず同一の情報を読み取ります。</li> <li>• シリアル化可能: トランザクションが同一の SQL 問合せを複数回実行する場合、実行と実行の間にその行が別のトランザクションによって変更、削除、作成およびコミットされていても、必ず同一の情報を読み取ります。</li> </ul>
アクション・テキスト	実行するコマンドのコード。式エディタを呼び出すには、  をクリックします。

## エージェント

## 物理エージェント

このエージェントは、TCP/IP ポートのリスナーとして設定できる Java サービスです。

このサービスでは、次のことが可能です。

- グラフィカル・モジュールからの、オンデマンドでのジョブの実行（モデル・リバース、パッケージ、シナリオ、インタフェースなど）。この場合、リスナー・エージェントを起動する必要があります。
- オンデマンド実行以外の、スケジュールされたシナリオの実行。物理エージェントには、事前定義のスケジュールに応じてシナリオを自動的に起動する、オプションのスケジューラが含まれます。この場合、スケジューラ・エージェントを起動する必要があります。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースで使用するエージェント名。
ホスト	エージェントが起動されたマシンのネットワーク名または IP アドレス。一部の TCP/IP 構成では、マシン名でなく IP アドレス（たとえば 195.10.10.5）を指定する方が適切です。
ポート	エージェントが使用するリスニング・ポート。デフォルトでは、20910 番ポートです。
セッションの最大数	このエージェントで許可されるセッション数の最大値。この値は、ロード・バランシングを使用する場合に使用されます。
スケジューリング情報	エージェントの「スケジューリング情報」ウィンドウを開きます。
スケジューリングの更新	このアクションは、作業リポジトリに含まれたシナリオ上で計画された、異なるエージェント・スケジュールからのエージェントの実行計画をリロードします。
テスト	このアクションを使用して、指定されたエージェントが正しく動作していることを確認できます。

### ロード・バランシング

詳細は、「ロード・バランシング」を参照してください。

プロパティ	説明
リンク	指定された物理エージェントが、委任されたセッションを現在のエージェントから受け取ることができるかどうかを示します。
エージェント	現在のエージェントからセッションを受け取ることができる物理エージェントの名前。

### スケジューリング情報

スケジュール情報を使用すると、エージェントのスケジュール済タスクを視覚化できます。

**重要:** スケジューリング情報はエージェントのスケジュールから取得されます。正確なスケジューリング情報を表示するには、エージェントを起動してスケジュールをリフレッシュする必要があります。

プロパティ	説明
選択されたエージェント	スケジュールが表示されるエージェント。すべてのエージェントのスケジュールを表示することもできます。
スケジュールの開始と終了	スケジュールが表示される期間。このスケジュールをリフレッシュするには、  「リフレッシュ」ボタンをクリックします。
更新	このボタンを押すと、選択されたエージェントのスケジュールが更新されます。
時間範囲	期間を指定すると（1時間、2時間）、現在時刻にこの時間を加え、ダイアグラムをその中央に置くことができます。この機能により、進行中のセッションと到来するセッションを視覚化できます。矢印を使用して、範囲を順方向または逆方向に動かすことができます。
拡大、縮小	Gantt ダイアグラムにズーム・インします。ズームを使用してダイアグラム内の領域を選択することもできます。
ガント・ダイアグラム	このパネルには、選択されたエージェントのスケジュールが Gantt ダイアグラムとして表示されます。
シナリオ詳細	このパネルには、スケジュールされた各シナリオの詳細と実行統計情報が表示されます。

## ダイアグラムの使用

ダイアグラム内の領域を選択すると（マウス・ボタンを押したままにします）、選択したゾーン上に自動的にズームします。

ダイアグラムを右クリックすると、ズーム、ダイアグラムの画像ファイル形式での保存、および表示プロパティの印刷や編集を行うためのメニューがポップアップします。

## ロード・バランシング

Oracle Data Integrator では、物理エージェント間のロード・バランシングを実装しています。

### 概要

各物理エージェントは、次の特性とともに定義されます。

- 同時に実行できるセッションの最大数
- セッションの実行を委任できるリンクされた物理エージェントの数（オプション）

エージェントの負荷は、そのエージェントにおける任意の時点での割合（実行中のセッション数 / 最大セッション数）により決定されます。

### 最大セッション数の決定

最大セッション数は、エージェントが稼働するマシンの性能に応じて設定する必要がある値です。また、この値は、ユーザーが **Data Integrator** エージェントに付与する処理性能の程度に応じて設定することもできます。

#### セッションの委任

リンク・エージェント付きのエージェントでセッションが開始されると、**Oracle Data Integrator** により、負荷のより少ないリンク・エージェントが特定され、そのリンク・エージェントにセッションが委任されます。

ユーザー・パラメータの「新しいロード・バランシングの使用」を使用している場合、1つのセッションが終了するたびに残りのセッションが再分散されます。つまり、処理するセッションがなくなったエージェントには、通常、別のエージェントからセッションが再度割り当てられます。

**注意:** エージェントは、自分自身にリンクできます。自分自身にリンクしないエージェントは、リンク・エージェントにセッションを委任できるだけであり、セッションを実行することはありません。

**注意:** 委任は、カスケード状のリンク・エージェントで機能します。さらに、エージェント・リンクのループを作成することも可能です。このオプションはお薦めしません。

#### 使用不可能なエージェント

特定のエージェントで、実行中のセッション数とその**最大セッション数**に等しくなると、エージェントは、実行中のセッション数とそのエージェントの最大セッション数を下回るまで着信セッションをキュー・ステータスに設定します。

### ロード・バランシングの設定

#### ロード・バランシングを設定する手順:

1. 物理エージェントのセットを定義し、それらをルート・エージェントにリンクします（「物理エージェントの作成」を参照）。
2. ルート・エージェントとリンク・エージェントを起動します。
3. ルート・エージェントで実行を開始します。**Oracle Data Integrator** により、リンク・エージェント間で実行負荷が分散されます。

**注意:** セッションの実行エージェントは、オペレータのセッション・ウィンドウに表示されません。

**注意:** エージェント間で作業をロード・バランシングする場合、エージェントに名前を付ける（つまり、**-NAME** パラメータ付きでエージェントを開始する）必要があります。詳細は、「リスナー・エージェントの起動」を参照してください。

#### 関連項目:

- 物理エージェント
- 物理エージェントの作成
- セッション

## コンテキスト

コンテキストとは、1つ以上のデータ処理アプリケーションで実行する操作またはシミュレーションに必要なリソースの集合です。コンテキストを使用すると、同一のジョブ（リバース、データ品質管理、パッケージなど）を、異なるデータベースやスキーマ上で実行することができます。

Oracle Data Integrator では、コンテキストは、論理オブジェクト（**論理エージェント**、**論理スキーマ**）を物理オブジェクト（**物理エージェント**、**物理スキーマ**）にリンクすることを可能にしています。

例:

- コンテキスト「New York」、「Boston」および「San Francisco」は、営業管理、マーケティング管理およびロジスティクスに関する類似のデータ構造を持つ同一のソフトウェアを運用する、3つのデータ処理サイトを表しています。New York では、会計パッケージとデータウェアハウスも稼働しています。
- 「Development」および「Test」コンテキストは、必要に応じてレプリケートされたデータベース上でプロシージャをシミュレートすることを可能にします。

**デザイナー**および**オペレータ**・モジュールでは、現在のコンテキストはメイン・ウィンドウのツールバーに表示され、ツールバーから変更できます。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるコンテキストの名前。
コード	異なるリポジトリ間でのコンテキストの参照と識別を可能にする、コンテキストのコード。  <b>注意:</b> このコードは、一意であり、かつ可能なかぎり固定的である必要があります。このコードを変更すると、グラフィカル・モジュールのグラフィカル・インタフェースからのコンテキスト参照が再調整されることがあります。
パスワード	ユーザーがこのコンテキストでの作業をリクエストする場合に必要なパスワード。このパスワードにより、アクセス権を制限し、不要なコンテキスト変更を避けることができます。  <b>注意:</b> 本番環境などの非常に重要なコンテキストの場合、コンテキストの作成時にパスワードを入力することを強くお勧めします。  <b>注意:</b> パスワードを指定しない場合、コンテキストの変更時に検証作業が要求されません。
デフォルト	このコンテキストが様々なリストにデフォルトで表示され、 <b>デザイナー</b> または <b>オペレータ</b> を開いたときに選択されることを示します。

## エージェント

このタブを使用すると、このコンテキストでアクセス可能なすべての論理エージェントが表示および更新されます。

リストの左の列には、すべての既存の論理エージェントが含まれます。指定されたコンテキストで論理エージェントを使用できるためには、このコンテキストでこの論理エージェントに対応している物理エージェントを右列で選択する必要があります。

## スキーマ

このタブを使用すると、このコンテキストでアクセス可能なすべての論理スキーマが表示および更新されます。

リストの左の列には、すべての既存の論理スキーマが含まれます。指定されたコンテキストで論理スキーマを使用できるためには、このコンテキストでこの論理スキーマに対応している物理スキーマを右列で選択する必要があります。

## 論理アーキテクチャ

### 論理スキーマ

論理スキーマは、同一のデータストア構造が含まれているすべての**物理スキーマ**に一意の名を与えることを可能にするエイリアスです。

- スキーマの狙いは、異なる物理スキーマ上でのプロシージャとモデルの移植性を確保することです。このため、**デザイナー**におけるすべての開発物は、論理スキーマ上でのみ実行されません。
- 1つの論理スキーマは、別々の物理スキーマ上に1つまたは複数の物理実装を持つことが可能ですが、同一の**テクノロジーのデータ・サーバー**にある必要があります。論理スキーマは、常にテクノロジーに直接リンクされます。
- 論理スキーマを使用するためには、それを**コンテキスト**内で宣言する必要があります。コンテキスト内で論理スキーマを宣言することは、このコンテキストでどの物理スキーマがエイリアス（論理スキーマ）に対応するかを示すことです。

例:

- 論理スキーマ **LEDGER** は、会計アプリケーションの動作に必要な **Sybase** 表の集合です。これらの表は、会計アプリケーションの各インストールごとに、物理スキーマに格納されます。アプリケーションは **Boston** で1回、**Seattle** で2回（本番およびテスト）インストールされました。

論理スキーマの名前。	コンテキスト	物理スキーマ
LEDGER	Boston	Sybase Boston LDG
LEDGER	Seattle Production	Sybase SEATTLE PROD LDG
LEDGER	Seattle Test	Sybase SEATTLE TEST LDG

- デザイナー**または**オペレータ**での作業は、常に論理スキーマ **LEDGER** 上で行われます。このコンテキストでのみ、操作が実際に行われる物理スキーマを決定することができます。このため、ユーザーは1つの物理環境から別の環境に、簡単に変わることができます。

## 定義

プロパティ	説明
名前	<p>論理スキーマの名前。スキーマの機能の内容を示唆する名前を付けることをお勧めします（プログラムまたはアプリケーション名前）。例: LEDGER、CRM、ACCOUNTING</p> <p><b>注意:</b> この名前は、他の Oracle Data Integrator モジュールがアクセスする作業リポジトリに含まれるオブジェクトによって ID コードとして使用されます。この名前をむやみに変更すると、グラフィカル・インタフェースからの参照を手作業で再調整する必要があります。</p>
コンテキスト	Oracle Data Integrator で宣言されたコンテキストのリスト。
物理スキーマ	このコンテキストで論理スキーマに対応する物理スキーマが表示されます。値が未定義の場合、コンテキストに論理スキーマが存在しないことを示します。

## 論理エージェント

論理エージェントは、異なるコンテキストで同一の機能を持つすべての物理エージェントに一意の名を与えることを可能にするエイリアスです。

- 論理エージェントの狙いは、複数のコンテキストでの本番への移行とスケジューリングを簡単にすることです。
- 論理エージェントを使用するためには、それをコンテキスト内で宣言する必要があります。コンテキストで論理エージェントを宣言するとは、指定されたコンテキストでそれに対応する物理エージェントを示すことです。

## 定義

プロパティ	説明
名前	<p>論理エージェントの名前。それが基づいているエージェントまたは OS の機能がわかる名前をすることをお勧めします。</p> <p><b>注意:</b> この名前は、他の Oracle Data Integrator モジュールがアクセスする作業リポジトリに含まれるオブジェクトによって ID コードとして使用されます。この名前をむやみに変更すると、Oracle Data Integrator モジュールのグラフィカル・インタフェースからの参照を手作業で再調整する必要があります。</p>
コンテキスト	Oracle Data Integrator で宣言されたコンテキストのリスト。
物理エージェント	このコンテキストで論理エージェントに対応する物理エージェントが表示されます。値が未定義の場合、コンテキストに論理エージェントが存在しないことを示します。

## 言語

### 言語

Oracle Data Integrator は、コンピュータ言語を使用して**テクノロジー**にアクセスします。これらの言語は、これらのテクノロジー上で実行されるプロシージャを生成するために使用されます。そのため、言語の特性を正しく定義することが、プロシージャの正しい生成と実行に不可欠です。

Oracle Data Integrator では、言語は、サブ言語で組み立てられた言語要素によって記述されます。言語は、次の目的に使用されます。

- **式エディタ**で使用できる言語要素の決定
- プロシージャ生成時の、**単語セパレータ**と**デリミタ**の管理

### 概要

言語は、次の3タイプの**単語**で構成されています。

- **オブジェクト**: 言語によって処理される名前付きのエンティティ (表、スキーマ、列)
- **リテラル**: 言語が処理する値 (たとえば、SQL 言語の場合、表内のレコード列の内容)。
- **予約語**: 言語に固有で、通常、関数やコマンドなどの名前に使用される単語

### 定義

プロパティ	説明
名前	言語の名前。
デリミタ	<b>この項では、その言語でデリミタとして使用する文字を定義できます。</b>
オブジェクト	<p>オブジェクトの名前を区切ることを可能にする文字のペア。一般に、このタイプの文字は、大文字と小文字の違いを保護するためや、オブジェクト名で使用される特殊な語や文字のセパレータになる可能性のあるものを保護するために使用されます。</p> <p>各ペアは開始文字と終了文字で構成されていて、2文字を超えることはできません。</p> <p>例:</p> <ul style="list-style-type: none"> <li>• <code>[]</code>: <code>[Active Customers]</code> 角括弧は、オブジェクト名 <code>Active Customers</code> の大文字と小文字の組合せや、スペースを維持するために必要です。</li> <li>• <code>"</code> : <code>"Account"."Licenses"</code> 引用符は、オブジェクト <code>Account</code> (スキーマ) および <code>License</code> (データストア) の名前におけるアクセントと大文字と小文字の組合せを維持するために使用されます。</li> </ul>
リテラル	<p><b>リテラル (値)</b> を区切るための文字のペア。</p> <p>各ペアは開始文字と終了文字で構成されていて、2文字を超えることはできません。</p>

例:

- `'` : `'Welcome to Paris !'`。引用符は、リテラル内のアクセント、大文字と小文字の組合せ、および特殊文字を維持するために使用されます。

大/小文字区別	言語で大文字と小文字を区別するかどうかを決定します。
オブジェクト	その言語が、列、表、スキーマの名前などの、処理する <b>オブジェクト</b> 名で、大文字と小文字を区別するかどうかを示します。
予約済キーワード	その言語が <b>予約語</b> （たとえば <code>SELECT</code> や <code>INSERT</code> ）で大文字と小文字を区別するかどうかを示します。
単語セパレータ	言語で <b>単語</b> を分離するために使用される文字。

## サブ言語

サブ言語は、言語内でタイプや用例を共有する言語要素のグループです。

例:

- サブ言語「Aggregation」は、SQL 言語のすべての集計演算子をグループ化したものです。

## 定義

プロパティ	説明
名前	サブ言語の名前。
タイプ	このサブ言語内の言語要素のタイプ。次のタイプが可能です。 <ul style="list-style-type: none"> <li>• <b>F</b>: ファンクション、つまりパラメータを受け入れて値を返す要素。</li> <li>• <b>O</b>: 2 つの単語を関係付ける演算子。</li> <li>• <b>KW</b>: 言語のキーワード。</li> </ul>
プロシージャで使用	このサブ言語の要素をプロシージャで使用できることを示します。
マッピングで使用	このサブ言語の要素をマッピングで使用できることを示します。
FROM 句で使用	このサブ言語の要素を FROM 句で使用できることを示します。
フィルタで使用	このサブ言語の要素をフィルタで使用できることを示します。
色	<b>式エディタ</b> で表示される場合のこのサブ言語の要素の色。

## 言語要素

言語要素は演算子、ファンクションまたは言語のキーワードです。この言語要素は、言語が実装されるテクノロジーに応じて、いくつかの実装を持つことがあります。

例:

- SQL 言語の問合せで現在の日付を返すファンクションは、言語要素 CURDATE として宣言されますが、SQL 言語をサポートするテクノロジーでは、TODAY (Sybase、Informix) または CURDATE (Progress、DB2/400 など) のように実装されます。

## 定義

プロパティ	説明
名前	言語要素の名前。
式	言語要素の汎用構文。 テクノロジーの実装構文は、「 <b>実装</b> 」タブに記述されます。ただし、テクノロジーに対して実装がなく、テクノロジーが言語要素 ( <b>Universal</b> または <b>Standard</b> タイプ) へのアクセスを持つ必要がある場合、これは <b>式エディタ</b> が使用するデフォルト構文になります。
グループ関数	この言語要素が要素のグループを扱うことを示します。(例: MAX、MIN など)
ユニバーサル	この要素がユニバーサル・タイプで、この言語をサポートするすべてのテクノロジーによって継承されることを示します。
標準	この要素が標準タイプで、この言語の「 <b>互換性</b> 」ボックスを選択したすべてのテクノロジーによって継承されることを示します。
ヘルプ	要素のヘルプ・テキストと説明。

## 実装

プロパティ	説明
名前	言語要素の実装の名前。
式	この実装での言語要素の構文。この構文は <b>式エディタ</b> に表示される構文です。
テクノロジー	この実装が行われるテクノロジー。
例外	このボックスが選択されている場合、テクノロジーがユニバーサルまたは標準タイプでも、指定されたテクノロジーに対する言語要素は存在しません。

## リポジトリ

### マスター・リポジトリ

**マスター・リポジトリ**は、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です。このリポジトリは、様々な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。

一般に、必要なマスター・リポジトリは1つだけです。

ただし、次のような例外的環境では、状況に応じて複数のマスター・リポジトリを作成する必要があります。

- 高速ネットワークによりリンクされていない複数のサイトにまたがってプロジェクトを構築する場合（オフサイト開発など）。
- インタフェースの運用環境（開発、テスト、本番）を明確に区別する必要がある場合（マスター・リポジトリを格納したデータベースの環境を含む）。これらの環境が複数のサイトに存在する場合、この条件に一致する可能性があります。

## マスター・リポジトリ・ドメイン

マスター・リポジトリには、次の2つの**ファンクショナル・ドメイン**があります。

- **トポロジ**: このドメインは必須です。このドメインには、テクノロジー、データ・サーバーおよびエージェントの記述が含まれます。このドメインの情報は、トポロジ・マネージャ・モジュールを使用して変更することができます。
- **セキュリティ**: このドメインは必須です。ユーザーとその権限に関する情報の、Oracle Data Integrator 内部メタデータと格納構造が含まれます。このドメインの情報は、セキュリティ・マネージャ・モジュールを使用して変更することができます。

## 定義

プロパティ	説明
名前	マスター・リポジトリの名前。
外部 ID コード	リポジトリの一意の ID コード。
成功したインストール	インストールが正常終了したかどうかを示します。この表示がない場合は、インストールが正しく終了しなかった可能性があります。
リポジトリ・バージョン	マスター・リポジトリのバージョン。
接続	このボタンで、マスター・リポジトリに接続するための情報を開くことができます。この接続情報はデータ・サーバーの情報です。マスター・リポジトリへの接続の詳細は、リポジトリの作成に関する項を参照してください。  <b>注意:</b> リポジトリの接続情報を変更する場合は十分注意してください。この操作は Oracle Data Integrator インストールの問題の原因となることがあります。

## 作業リポジトリ

作業リポジトリは、データ・モデル、プロジェクトおよびその操作に関する情報を保持するデータ構造です。このリポジトリは、様々な 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。

必要であれば、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宣言することもできます。ただし、バージョン管理の目的上、1つの作業リポジトリからリンクできるのは1つのマスター・リポジトリに限られます。

作業リポジトリには、次の複数の機能ドメインが含まれます。

- 実行:** このドメインは必須です。このドメインを使用して、ローカルで（またはエージェントにより）実行されるジョブ操作を起動および監視します。このドメインには、通常、**オペレータ**・モジュールを通じてアクセスします。**運用リポジトリ**には、このドメインのみ含まれます。
- プロジェクト/モデル:** このドメインはオプションです。このドメインを使用して、データ・モデルおよびプロジェクト（インタフェースやプロシージャなど）を管理します。このドメインには、通常、**デザイナ**・モジュールを通じてアクセスします。**開発リポジトリ**には、実行ドメインとともにこのドメインが含まれます。

## 一般プロパティ

プロパティ	説明
ID	作業リポジトリの ID コード。 <b>重要:</b> 各作業リポジトリには、作成時に（情報システム全体を通じて）一意の ID コードを指定することを強くお勧めします。この一意の ID を使用すると、次のことが可能です。 <ul style="list-style-type: none"> <li>- 作業リポジトリを複数のマスター・リポジトリから競合なく参照できます。</li> <li>- このリポジトリのオブジェクトを、バージョン管理のためにマスター・リポジトリ上で（企業ドメインによって）参照できます。</li> <li>- オブジェクトを作業リポジトリ間で競合なく転送できます。作業リポジトリの一意の ID コードは、これらのオブジェクトの ID コード構成に組み込まれるため、複数のリポジトリ間でその一意性が保証されます。</li> </ul> セキュリティを向上するため、同じマスター・リポジトリからすべての作業リポジトリを作成することをお勧めします。
名前	リポジトリの名前。
外部 ID コード	リポジトリの一意の ID コード。
タイプ	作業リポジトリのタイプ。使用できるタイプは、 <b>実行</b> または <b>開発</b> です。
接続	このボタンで、作業リポジトリに接続するための情報を開くことができます。この接続情報はデータ・サーバーの情報です。作業リポジトリへの接続の詳細は、リポジトリの作成に関する項を参照してください。 <b>注意:</b> リポジトリの接続情報を変更する場合は十分注意してください。この操作は Oracle Data Integrator インストールの問題の原因となることがあります。

## ホスト

Oracle Data Integrator では、複数のモジュール（デザイナ、トポロジなど）にアクセスできます。ホストと使用方法の指定により、それらのモジュールへのアクセスを管理できます。

### 定義

プロパティ	説明
マシン名	マシンを特定する固有の名前。通常、これはマシンのネットワーク名です。
IP アドレス	マシンとリポジトリ・サーバーに共通のネットワークでの、そのマシンの IP アドレス。

### 使用方法

プロパティ	説明
モジュール	モジュール名。
使用方法	用例のタイプ。 <ul style="list-style-type: none"> <li>● <b>常に許可</b>: ホストは該当のモジュールを常に使用できます。</li> <li>● <b>拒否</b>: ホストは該当のモジュールを使用できません。</li> <li>● <b>自動</b>（デフォルト動作）: ホストは、該当のモジュールを使用してリポジトリに接続する際に、そのモジュールへのアクセス権を自動的に取得します。</li> </ul>
最終使用	このホスト上でのこのモジュールの最後の使用の日付と時刻。

## セキュリティ・マネージャ

### セキュリティ・マネージャの概要

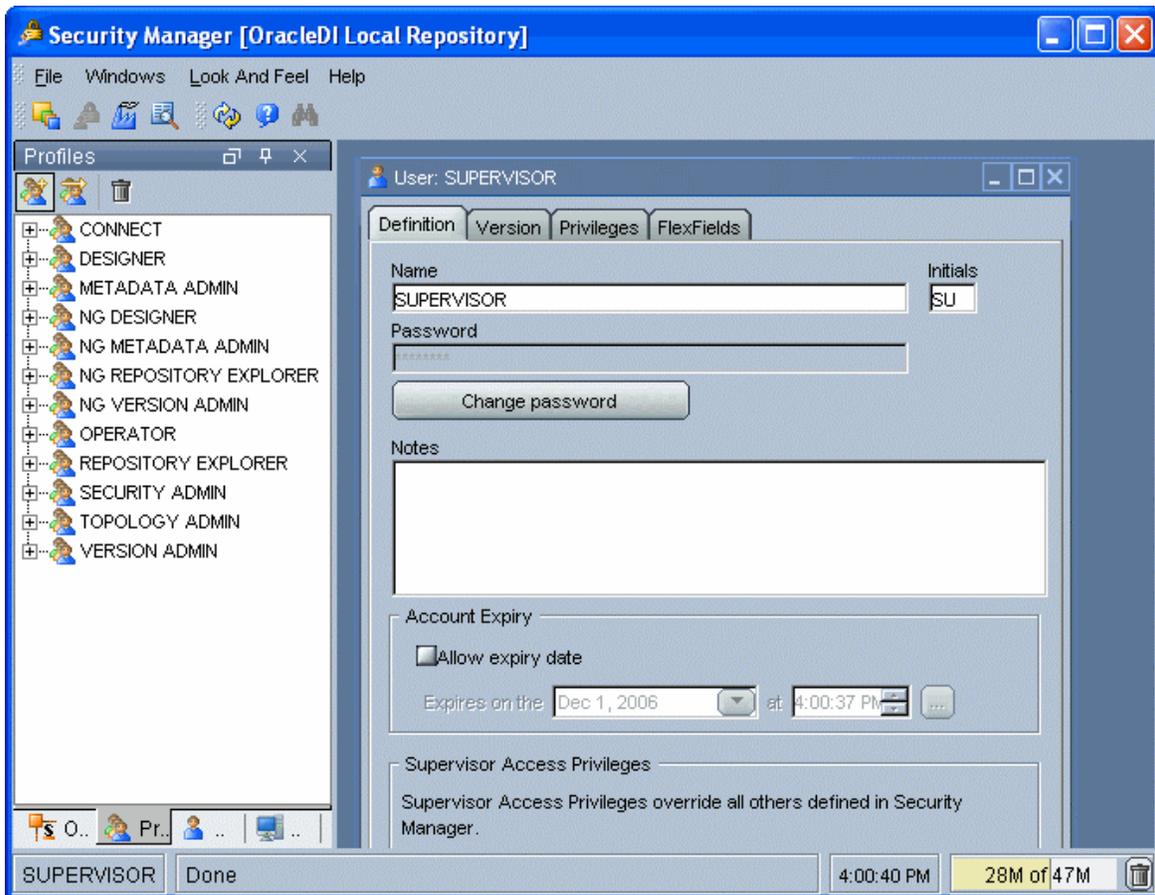
セキュリティ・マネージャ・モジュールを使用すると、Oracle Data Integrator のセキュリティを管理することができます。セキュリティ・マネージャ・モジュールを使用すると、**ユーザー**および**プロファイル**が作成されます。汎用オブジェクト（データ・サーバー、データ型など）で**メソッド**（編集、削除など）にユーザー権限を割り当てるため、およびオブジェクト・**インスタンス**（Server 1、Server 2 など）での権限を微調整するために使用します。

セキュリティ・マネージャ・モジュールは、この情報をマスター・リポジトリに格納します。この情報は、他のすべてのモジュールで使用できます。

- セキュリティ・ポリシーの定義

### セキュリティ・マネージャのインタフェース

セキュリティ・マネージャの GUI は次のように表示されます。



## メニュー

メニューでは、プルダウン・メニューから次の機能にアクセスできます。

- インポート/エクスポート
- ウィザード
- オプションの表示
- モジュールまたはツリー表示のオープン
- ユーザーのパスワードおよびオプションの変更

## ツールバー

ツールバーからは、次の操作を実行できます。

- 他のモジュールの起動
- ツリー表示のリフレッシュ
- オンライン・ヘルプの起動

## ツリー表示

現在のユーザーが使用できるセキュリティ・マネージャ・オブジェクトは、次のように整理してツリー表示に表示されます。

- **オブジェクト**。Oracle Data Integrator の要素タイプ（データストア、モデルなど）が記述されます。
- ユーザーの**プロファイル**と、その**権限**。
- **ユーザー**およびその**権限**。

各ツリー表示は、メイン・ウィンドウの両側にドッキングできるフローティング・フレームに表示されます。フレームは重ねることもできます。複数のフレームを重ねた場合は、フレーム・ウィンドウの下部に表示されるタブから各フレームにアクセスできます。

ツリー表示フレームは、フレームのタイトルまたはタブを選択してドラッグすることにより、移動したり、ドッキングしたり、重ねたりできます。ツリー表示の位置を固定するには、「ウィンドウ」メニューから「ウィンドウ・レイアウトのロック」を選択します。

ツリー表示フレームがメイン・ウィンドウに表示されないか、閉じている場合は、「ウィンドウ」→「ビューの表示」メニューを使用します。

各ツリー表示では、次の操作が可能です。

- ルート・オブジェクトの挿入またはインポート（フレーム・タイトルで該当するボタンをクリック）
- ノードの展開または折りたたみ（ノードをクリック）
- オブジェクトに関連付けられているメソッド（編集、削除など）のアクティブ化（ポップアップ・メニューを使用）
- オブジェクトの編集（オブジェクトをダブルクリック、またはワークベンチにドラッグ・アンド・ドロップ）

## ワークベンチ

編集または表示されているオブジェクトのウィンドウは、**ワークベンチ**に表示されます。

## 接続

### マスター・リポジトリへの接続

このウィンドウを使用すると、トポロジ・マネージャまたはセキュリティ・マネージャ・モジュールを開く際に、マスター・リポジトリへの接続を選択できます。既存の接続の作成、変更または削除を開始することもできます。

**注意:** マスター・リポジトリ接続の定義は、ローカルの  
/bin/snps\_login\_security.xml ファイルに格納されています。

## 一般プロパティ

プロパティ	説明
ログイン名	マスター・リポジトリへの定義された接続のリスト。
ユーザー	選択されたマスター・リポジトリに接続するためのデフォルトの Oracle Data Integrator ユーザー。デフォルト・ユーザーが指定されなかった場合は、ここに入力する必要があります。
パスワード	接続するユーザーに対応するパスワード。デフォルトのパスワードが指定されなかった場合は、ここに入力する必要があります。

## ツールバー

ボタン	説明
	作成されるマスター・リポジトリへの新しい接続を許可します。
	「ログイン名」フィールドで選択された接続の変更を許可します。
	「ログイン名」フィールドで選択された接続の削除を許可します。

## マスター・リポジトリ接続の詳細

このウィンドウを使用すると、マスター・リポジトリへの接続を定義または変更できます。

**注意:** マスター・リポジトリ接続の定義は、ローカルの `/bin/snps_login_secu.xml` ファイルに格納されています。

## 一般プロパティ

プロパティ	説明
Oracle Data Integrator 接続	この情報グループを使用して、接続ウィンドウに表示されるプロパティを定義することができます。
ログイン名	Data Integrator モジュールが開いている場合、これがリストに表示される名前です。
ユーザー	マスター・リポジトリに接続するためのデフォルトの Oracle Data Integrator ユーザー。デフォルト・ユーザーが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。
パスワード	Oracle Data Integrator ユーザーに対応しているパスワード。デフォルトのパスワードが指定されなかった場合、この接続を使用するユーザーは毎回これを入力する必要があります。

データベース接続(マスター・リポジトリ)	この情報グループは、マスター・リポジトリをホストするリレーショナル・データベースの接続パラメータを示します。
ユーザー	マスター・リポジトリが含まれているデータベースのスキーマへの接続を許可するユーザー。
パスワード	そのデータベース・ユーザーのデータ・サーバー・パスワード。
ドライバ・リスト	マスター・リポジトリに基づいているテクノロジーのタイプ。 ボタン  をクリックすると、使用されるテクノロジーとドライバに関する詳細情報を表示できます。詳細は、「ドライバの詳細」を参照してください。
ドライバ名	マスター・リポジトリのデータ・サーバーに接続するために使用する JDBC ドライバの名前。テクノロジーを選択すると、そのテクノロジーで最もよく使用されるドライバが表示されます。他のドライバを使用することもできます。その場合、そのドライバが適切にインストールされている必要があります。詳細は、インストレーション・ガイドの「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。
URL	マスター・リポジトリのデータ・サーバーに接続するための URL。ボタン  をクリックすると、使用するドライバの URL モデルを選択することができます。詳細は、「URL サンプル」を参照してください。 ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。
デフォルトの接続	このボックスを選択すると、Oracle Data Integrator グラフィック・モジュールの起動時に、デフォルトで、現在の接続が提案されます。

## ドライバの詳細

このウィンドウには、使用するドライバとテクノロジーに関する情報が表示されます。

### 一般プロパティ

プロパティ	説明
名前	選択されたテクノロジーで使用するドライバ。
説明	ドライバについての情報。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## URL サンプル

このウィンドウには、ドライバの URL モデルが表示されます。

## 一般プロパティ

プロパティ	説明
名前	選択されたドライバ用の有効な URL モデルのリスト。
説明	URL モデルの詳細。

ドライバと URL の詳細は、「JDBC URL のサンプル」を参照してください。

## オブジェクト

### オブジェクト

オブジェクトは、Oracle Data Integrator（エージェント、モデル、データストアなど）を使用して処理できる要素の表現です。オブジェクトは、Oracle Data Integrator オブジェクト・コンポーネント（Java クラス）の可視部分に相当します。オブジェクトの概念とオブジェクト・インスタンスの概念は統合的に捉える必要があります。これらは、Oracle Data Integrator では、オブジェクト指向プログラミングの概念に類似しています。インスタンス（オブジェクト・インスタンス）は、オブジェクト・タイプ（オブジェクト）にアタッチされています。たとえば、プロジェクト・タイプのオブジェクトから、プロジェクト MY\_PROJ\_1 などといったインスタンスを生成できます。同様に、プロジェクト・タイプのオブジェクトから YOUR\_PROJ\_2 という別のインスタンスを生成することもできます。

### 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるオブジェクトのラベル。

### フレックスフィールド

フレックスフィールドは、Oracle Data Integrator のオブジェクトのプロパティをカスタマイズ可能にするユーザー定義フィールドです。Flexfield は、各オブジェクト・インスタンスに対して、オブジェクト・ウィンドウの「フレックスフィールド」タブを使用して設定されます。値は、各オブジェクト・インスタンスに対して、オブジェクトのインスタンス・ウィンドウの「フレックスフィールド」タブを使用して定義されます。Flexfield の値を使用するには、Oracle Data Integrator の代入メソッドを使用します。

**重要:** フレックスフィールドは、特定のオブジェクト・タイプにのみ存在します。「フレックスフィールド」タブがないオブジェクトではサポートされていません。

プロパティ	説明
名前	オブジェクト・インスタンス・ウィンドウの「フレックスフィールド」タブに表示される flexfield のラベル。
コード	Flexfield のコード。
テクノロジー	flexfield がアクティブ化されるテクノロジー。たとえば、プロセス上の Flexfield は、このテクノロジーを使用してのみ表示されます。
タイプ	flexfield のタイプ（文字列あるいは数値）。
デフォルト	Flexfield のデフォルト値。

## メソッド

メソッドは、オブジェクト上で実行できるアクションのタイプです。各オブジェクトには、そのオブジェクトに固有の一連のメソッドがあります。Oracle Data Integrator のメソッドの概念はオブジェクト指向プログラミングの概念に類似しています。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるメソッドのラベル。
内部 Java 名	このオブジェクト上に実装された Oracle Data Integrator の内部 Java メソッドの名前。
オブジェクト作成メソッド	このメソッドがオブジェクト・インスタンスの作成に関与することを示します。
オブジェクト作成者に付与されたデフォルト・メソッド	このボックスを選択すると、オブジェクト作成者に対して、メソッドに対する権限が自動的に付与されます。 <b>注意:</b> このチェック・ボックスが有効なのは、オブジェクト・インスタンスによる認可がサポートされるオブジェクトだけです。
対象オブジェクト	オブジェクトの内部 ID コード。

## プロファイル

### プロファイル

プロファイルは、Oracle Data Integrator で使用する汎用の権限モデルを表します。1 つ以上のプロファイルをユーザーに割り当てることができます。

## 定義

プロパティ	説明
名前	グラフィカル・インタフェースに表示されるプロファイルのラベル。

## 事前定義のプロファイル・タイプ

プロパティ	説明
CONNECT	Oracle Data Integrator に接続するための最小の汎用プロファイル。すべてのユーザーが、少なくともこのプロファイルを持つ必要があります。
DESIGNER	インタフェース開発者またはインタフェース関連の作業に携わるユーザー用の汎用プロファイル。このプロファイルは、リポジトリに格納されているすべてのプロジェクトおよびプロジェクト・サブ・コンポーネント（フォルダ、インタフェース、ナレッジ・モジュールなど）へのアクセスを提供します。また、ジャーナル化アクション（ジャーナルの起動、サブスクライバの作成など）を実行したり、モデルとデータストアに関する静的管理を実行したりする権限をユーザーに与えます。
METADATA ADMIN	モデルの管理とリバース・エンジニアリングを担当するユーザー用の汎用プロファイル。このプロファイルを持つユーザーは、CKM、RKM または JKM を選択して特定のモデルにアタッチするためにあらゆるプロジェクトを参照することができます。
OPERATOR	オペレータ用の汎用プロファイル。実行ログの参照を許可します。
REPOSITORY EXPLORER	Metadata Navigator を使用してメタデータを参照するための汎用プロファイル。Metadata Navigator からのシナリオの起動も許可します。
SECURITY ADMIN	ユーザー・アカウントおよびプロファイルの管理者用の汎用プロファイル。
TOPOLOGY ADMIN	情報システム・トポロジの管理を担当するユーザー用の汎用プロファイル。このプロファイルを付与されているユーザーは、トポロジ・マネージャ・モジュールを使用して任意のアクションを実行することができます。
VERSION ADMIN	コンポーネントのバージョンとソリューションを管理するための汎用プロファイル。このプロファイルは、DESIGNER および METADATA ADMIN と組み合わせる必要があります。
NG DESIGNER	DESIGNER 用の非汎用プロファイル。
NG METADATA ADMIN	METADATA ADMIN 用の非汎用プロファイル。
NG	Metadata Navigator を使用してメタデータを参照するための非汎用プロファイル

REPOSITORY EXPLORER	ル。
NG VERSION ADMIN	VERSION ADMIN 用の非汎用プロファイル。このプロファイルは、NG DESIGNER および NG METADATA ADMIN と一緒に使用することをお勧めします。

## 汎用プロファイルと非汎用プロファイル

汎用プロファイルでは、すべてのオブジェクトのメソッドに対して「**一般権限**」オプションが選択されています。これは、このようなプロファイルを持っているユーザーが、**デフォルトで**、プロファイルで権限を認められているオブジェクトのすべてのインスタンスのすべてのメソッドに対して権限を与えられることを意味します。

すべてのオブジェクトのメソッドに対して「**一般権限**」オプションが選択されていない場合、**非汎用**プロファイルは、**デフォルトでは**、インスタンスのすべてのメソッドに対する権限を付与されていません。管理者は、ユーザーに、各インスタンスのメソッドについて権限を与える必要があります。

デフォルトではユーザーが権限を持つインスタンスがなく、インスタンスごとに管理者が権限を付与するようにする場合、ユーザーに非汎用プロファイルを付与する必要があります。

あるオブジェクト・タイプのすべてのインスタンスに対してユーザーがデフォルトで権限を持つようにする場合は、ユーザーに汎用プロファイルを付与する必要があります。

プロファイルでは次の操作を行うことができます。

- 新規プロファイルの作成
- ユーザーへのプロファイルの割当て
- プロファイル別の認可の割当て
- プロファイル別の認可の削除
- ユーザーからのプロファイルの削除
- プロファイルの削除

## プロファイルによる認可

プロファイルによる認可は、指定されたプロファイルのオブジェクトのメソッドに対して行われます。このプロファイルを持っているユーザーは、メソッドを介したこのオブジェクトへの権限を、オプションとしてまたは自動的に付与されます。

プロファイル・ツリー内の特定のオブジェクト下に、プロファイル別認可が適用されたメソッドがある場合、そのプロファイルを持つユーザーは、該当するメソッドを通じてそのオブジェクトのインスタンスを操作する権限を（オプションまたは自動で）付与されます。

プロファイルによる認可がない場合、このプロファイルを持っているユーザーは、いかなる場合も、オブジェクトのインスタンス上でメソッドを呼び出すことができません。

## 定義

プロパティ	説明
一般権限	このボックスを選択すると、プロファイルを持っているユーザーは、デフォルトで（自動的に）、このオブジェクトのすべてのインスタンス上でメソッドに対する権限を持ちます。選択しない場合、デフォルトではその権限を持ちません。権限は、オブジェクト・インスタンスによる認可により、事例ごとに表示される必要があります。

プロファイルによる認可では次の操作を行うことができます。

- プロファイル別の認可の割当て
- プロファイル別の認可の削除

## ユーザー

### ユーザー

セキュリティ・マネージャ・モジュール内のユーザーは、Oracle Data Integrator ユーザーを表し、リポジトリへの接続に使用されるログイン名に対応します。

ユーザーは次の権限を継承します。

- すでに持っているプロファイル権限
- オブジェクトに対する権限
- インスタンスに対する権限

## 定義

プロパティ	説明
名前	ログイン名として使用されたユーザー名。
イニシャル	ユーザーの頭文字（リポジトリ内のオブジェクトが変更される場合の識別に使用されます）。
パスワード	ユーザーのパスワード。
パスワードの入力、パスワードの変更	このボタンはユーザーのパスワードを入力および確認するウィンドウを開きます。
メモ	これは、ユーザーに関する情報を入力するための任意のテキストのゾーンです。
アカウント失効	この項では、このアカウントの期限が切れる日付をプログラムすることができます。この日付の後では、管理者によるアカウントの再アクティベーションが必要です。

スーパーバイザ・アクセス権限 「スーパーバイザ」ボックスを使用すると、1人のユーザーに ALL 権限を付与することができます。セキュリティ・マネージャで付与された権限は自動的にすべてオーバーライドされます。**このチェック・ボックスにアクセスできるのは、Supervisor アクセス権限を持っているユーザーだけです。**

**注意:** Supervisor 権限を付与するユーザー数は最小限にとどめてください。Oracle Data Integrator のユーザーSUPERVISORには、デフォルトで付与されます。この権限はこのアカウントだけにとどめておくことをお勧めします。このアカウントは主に保守と管理の目的で使用されます。

ユーザーに対しては次の操作を行うことができます。

- 新規ユーザーの作成
- ユーザーへのプロファイルの割当て
- ユーザー別の認可の割当て
- オブジェクト・インスタンス別の認可の割当て
- オブジェクト・インスタンス別の認可の削除
- ユーザー別の認可の削除
- ユーザーからのプロファイルの削除
- ユーザーの削除

## ユーザーによる認可

ユーザーによる認可は、指定されたユーザーのオブジェクトのメソッドに対して行われます。これを使用すると、該当するメソッドを通じてオブジェクトを操作する権限を（オプションまたは自動で）特定のユーザーに付与できます。

ユーザーのツリー内に、ユーザー別認可を適用されたメソッドがある場合、そのユーザーには、そのメソッドを通じて該当するオブジェクトのインスタンスを操作する権限が（オプションまたは自動で）付与されます。

ユーザー別の認可が指定されていない場合、そのユーザーはいかなる場合にも、該当するオブジェクトのインスタンスに対してそのメソッドを起動できません。

## 定義

プロパティ	説明
一般権限	このボックスを選択すると、ユーザーは、デフォルトで（自動的に）、このオブジェクトのすべてのインスタンス上でメソッドに対する権限を持ちます。選択しない場合、デフォルトではその権限を持ちません。権限は、オブジェクト・インスタンスによる認可により、インスタンスごとに追加される必要があります。

ユーザーによる認可では次の操作を行うことができます。

- ユーザー別の認可の割当て
- ユーザー別の認可の削除

## オブジェクト・インスタンスによる認可

オブジェクト・インスタンスによる認可は、指定されたユーザーのオブジェクト・インスタンスに対して行われます。この方法を使用すると、特定のオブジェクト・インスタンスの特定のメソッドについて、ユーザーに権限を付与できます。

ユーザーのツリー内に、特定のインスタンスに対するオブジェクト・インスタンス認可がある場合、そのユーザーは該当するインスタンスについて、オブジェクトのメソッドに対する特定の権限を持つことができます（権限はウィンドウ内で指定されます）。ユーザーの**インスタンス**・ツリーに含まれないインスタンスについては、そのインスタンスの生成元であるオブジェクトに対するユーザー別またはプロファイル別認可が適用されます。

オブジェクト・インスタンス別の認可は、作業リポジトリごとに定義することもできます。特定のオブジェクト・インスタンスを、複数の作業リポジトリ間でレプリケート（シノニム・モードでのエクスポートまたはインポートを使用して）したり、異なる複数の作業リポジトリで使用できるようにすることもできます（コンテキストなど）。インスタンスを複数のリポジトリで使用できるようにした場合、各権限の付与と拒否は、すべてのリポジトリを対象とすることもできますし、選択したリポジトリのみを対象とすることもできます。

たとえば、開発リポジトリからテスト・リポジトリにプロジェクトをレプリケートすることはよく使われる手法です。ある開発者に対し、担当プロジェクトの編集権限を開発リポジトリについては付与し、テスト・リポジトリについては付与しないということもできます。そして、テスト・リポジトリについてはプロジェクトへの参照権限のみを付与するなどといったことが可能です。

**注意:** オブジェクト・インスタンスによる認可をサポートしているのは、特定のオブジェクト・タイプです。これらのオブジェクト・タイプは各ユーザーの**インスタンス**ノード配下にリストされます。

## セキュリティ

このウィンドウには、オブジェクトと関連付けられているメソッドのリストが含まれます。認可が適用されるインスタンスのメソッドだけでなく、ツリー表示でこのインスタンスの配下にあるオブジェクト・インスタンスのメソッドも含まれます。

リストには、デフォルトで選択されるものはありません。選択するには、表で行をクリックします。項目のグループを選択するには[Shift]キーを押します。複数選択するには、[Ctrl]キーを押したまま選択します。

プロパティ	説明
オブジェクト	権限が適用されるオブジェクト・タイプ。
メソッド	オブジェクト・メソッドの名前。
アクティブ	このフィールドは、メソッドの権限付与について定義します。 <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> <b>常時:</b> これからアタッチされるものも含め、このマスターにアタッチされたすべての作業リポジトリに対して権限が付与されます。</li> <li><input checked="" type="checkbox"/> <b>なし:</b> これからアタッチされるものも含め、このマスターにアタッチされたすべての作業リポジトリに対して権限が拒否されます。</li> <li><input type="checkbox"/> <b>リポジトリ別:</b> 「リポジトリ」フィールドで選択された作業リポジトリに</li> </ul>

対しては権限を付与し、他のリポジトリに対しては拒否します。

リポジトリ このフィールドには、作業リポジトリのリストか、ユーザー未定義のリストが含まれます。これは、この権限が、すべてのリポジトリで付与または拒否（**常時/なし**）されることを意味します。

**たとえば:** 「Model」オブジェクトのインスタンス MODEL1 が挿入される場合、オブジェクト Datastores、Condition などのためのメソッドが表示されます。Datastore オブジェクトの Delete メソッドで、「**アクティブ**」フィールドが「**常時**」に設定されている場合、ユーザーには、すべてのリポジトリの MODEL1 のすべてのデータストアで削除する権限が付与されません。

**注意:** ユーザーが汎用権限を保持するメソッドは、オブジェクト・インスタンス別の認可ウィンドウにリストされません。

オブジェクト・インスタンスによる認可では次の操作を行うことができます。

- オブジェクト・インスタンス別の認可の割当て
- オブジェクト・インスタンス別の認可の削除

## 付録

### 日付書式

Oracle Data Integrator は日付を java 書式で内部的に管理しています。文字列の中の書式のある日付を認識するためや、日付を表示するために、日付書式を指定する必要がある場合は、日付書式 **パターン**を指定する必要があります。

このパターンは、文字列に対応する **記号**（日付の各部を表す文字）と任意のテキストで、最後にデリミタで囲みます。パターン内で指定された英字以外の文字は、デリミタで囲まれていない場合でもそれが表す内容がそのまま再現されます。

各記号には定義済の**表現**が定められています（テキストまたは数値）。たとえば、ある年（2004）が数値として、曜日（月曜日）がテキストとして表現されている場合があります。

### 使用できる記号

記号	説明	表現	例
G	紀元前または紀元後	テキスト	AD
y	年	数値	1996
M	月	テキストまたは数値	7月または 07
d	日	数値	10
h	時 (1~12)	数値	12

H	時 (0~23)	数値	23
m	分	数値	55
s	秒	数値	30
S	ミリ秒	数値	978
E	曜日	テキスト	木曜日
D	年初からの日数	数値	189
F	月初からのその曜日の回数	数値	2 (2 番目の水曜)
w	年初からの週数	数値	27
W	月初からの週数	数値	2
a	AM/PM	テキスト	PM
k	時 (1~24)	数値	24
K	時 (0~11)	数値	0
z	タイムゾーン	テキスト	太平洋標準時間
'	任意のテキストのためのエスケープ文字	なし	
''	任意テキストのデリミタ	なし	

表現によっては、指定する記号の文字数によっても結果は変わります。たとえば、「zz」というパターンの結果が「PDT」の場合、「zzzz」というパターンで生成される結果は「太平洋夏時間」です。このルールについて、次の表に示します。

表現	記号の数	結果
テキスト	1-3	省略形
テキスト	>= 4	完全形
数字	最小限必要な桁数	記号の数が多い場合、余った桁には0が入れられます。記号の数が少ない場合、切り捨てられます。
テキストと数字	1-2	数値表現
テキストと数字	>=3	テキスト表現

## 例

書式	結果（米国の構成の場合）
yyyy.MM.dd G 'at' hh:mm:ss z	1996.07.10 AD at 15:08:56 PDT
EEE, MMM d, 'yy	Wed, July 10, '96
h:mm a	12:08 PM
hh 'o'clock' a, zzzz	12 o'clock PM, Pacific Daylight Time
K:mm a, z	0:00 PM, PST
yyyyyy.MMMMMM.dd GGG hh:mm aaa	1996.July.10 AD 12:08 PM

## JDBC URL のサンプル

Oracle Data Integrator には JDBC ドライバのライセンスは含まれていません。すでに購入したサーバー製品がある場合は、一部のドライバが無料で提供されます。

使用可能なすべての JDBC ドライバのリストは、次の URL にあります。

<http://java.sun.com/products/jdbc/jdbc.drivers.html>

すべてのドライバが Oracle Data Integrator での使用を検証されているわけではありません。

Oracle Data Integrator での使用テストが完了している推奨ドライバのリストを次に示します。

ドライバ	ドライバと URL パラメータ、コメント
Oracle Data Integrator Driver for File (新しいバージョン)	<p>ドライバ: <code>com.sunopsis.jdbc.driver.file.FileDriver</code></p> <p>URL: <code>jdbc:snps:dbfile?&lt;option1&gt;&amp;&lt;option2&gt;&amp;...</code></p> <p>オプション</p> <ul style="list-style-type: none"> <li>ENCODING=&lt;encoding_code&gt;: ファイル・エンコーディング。サポートされているエンコーディングのリストは <a href="http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html">http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html</a> にあります。デフォルトのエンコーディング値は ISO8859_1 です。</li> <li>TRUNC_FIXED_STRINGS=TRUE   FALSE: 固定列幅ファイルからフィールド・サイズに文字列を切り捨てます。デフォルト値は FALSE です。</li> <li>TRUNC_DEL_STRINGS=TRUE   FALSE: デリミタ付きファイルからフィールド・サイズに文字列を切り捨てます。デフォルト値は FALSE です。</li> <li>OPT=TRUE   FALSE: パフォーマンス向上のためにマルチプロセッサ・マシンでファイル・アクセスを最適化します。シングル・プロセッサ・マシンでこのオプションを使用すると、実際の性能が低下することがあります。デフォルト値は FALSE です。</li> </ul> <p><b>警告:</b> このドライバは JDBC 完全準拠ではありませんが、エージェントが使用</p>

Oracle Data Integrator Driver for File (旧バージョン - 廃止予定)	<p>する一部のメソッドをサポートします。</p> <p><b>警告:</b> このドライバ・バージョンは廃止予定です。</p> <p>ドライバ: <code>com.sunopsis.jdbc.driver.FileDriver</code></p> <p>URL: <code>jdbc:snps:file</code></p> <p>他の URL</p> <ul style="list-style-type: none"> <li>• <code>jdbc.snps.file?CHARSET_ENCODING=&lt;encoding&gt;&amp;RAW_HEADERS=true false</code> <ul style="list-style-type: none"> <li>- <code>CHARSET_ENCODING</code> を使用すると、ターゲット・ファイルのエンコーディングが表示されます。サポートされているエンコーディングのリストは、次の URL にあります。 <a href="http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html">http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html</a></li> <li>- <code>RAW_HEADERS</code> を使用すると、ヘッダー行を未処理データとして読み取ることができます。これにより、この行から特殊文字を取得して、リバース・エンジニアリング時に列名を生成することができます。</li> </ul> </li> </ul> <p><b>警告:</b> このドライバは JDBC 完全準拠ではありませんが、エージェントが使用する一部のメソッドをサポートします。</p>
Sun ODBC/JDBC Bridge (JDBC 2 Type 1)	<p>ドライバ: <code>sun.jdbc.odbc.JdbcOdbcDriver</code></p> <p>URL: <code>jdbc:odbc:&lt;odbc datasource name&gt;</code></p> <p><b>注意:</b> タイプ 1 の JDBC Driver を使用する場合は、あらかじめ ODBC Driver をインストールしており、ODBC Administrator を使用して ODBC データソースを作成している必要があります。</p>
Oracle JDBC Driver (JDBC 1 type 4)	<p>ドライバ: <code>oracle.jdbc.driver.OracleDriver</code></p> <p>URL: <code>jdbc:oracle:thin:@&lt;IP address or name&gt;:&lt;listener port&gt;:&lt;SID&gt;</code></p> <p><b>注意:</b> SQL*Net クライアントがクライアント・マシン上にある必要はありません。</p>
Inet Software JDBC Driver for Microsoft SQL/Server (Type 4)	<p>ドライバ: <code>com.inet.tds.TdsDriver</code></p> <p>URL: <code>jdbc:inetdae:&lt;IP address or name&gt;:&lt;listener port&gt;</code></p> <p><b>注意:</b> Open Client と ODBC がクライアント・マシン上にある必要はありません。</p> <p>このドライバは、行をフェッチする前に、クライアントの仮想メモリーに設定されたそれぞれの結果をロードします。</p> <p>他のサブプロトコル</p> <ul style="list-style-type: none"> <li>• <code>jdbc:inetdae: . . . .</code> 旧版</li> <li>• <code>jdbc:inetdae6: . . . .</code> SQL Server 6.5 互換モード (Sprinta ドライバ上)</li> </ul>

<p>Sybase JConnect (Type 4)</p>	<ul style="list-style-type: none"> <li>• jdbc:inetdae7: . . . .SQL Server 7.0 (以上) をサポート (Sprinta ドライバ上)</li> </ul> <p>他の URL</p> <ul style="list-style-type: none"> <li>• jdbc:inetdae:hostname:portnumber</li> <li>• jdbc:inetdae:hostname -&gt; デフォルト・ポートは 1433</li> <li>• jdbc:inetdae:hostname:portnumber?database=MyDb&amp;language=deutsch</li> </ul> <p>-&gt; プロパティ付き</p> <ul style="list-style-type: none"> <li>• jdbc:inetdae://servername/pipe/pipename -&gt; 名前付きパイプ付き</li> </ul> <p>ドライバ: com.sybase.jdbc2.jdbc.SybDriver</p> <p>URL: jdbc:sybase:Tds:&lt;IP address or name&gt;:&lt;listener port&gt;/&lt;database&gt;</p>
	<p><b>注意:</b> DB-LIB、CT_LIB および ODBC がクライアント・マシン上にある必要はありません。</p> <p><b>警告:</b> 表に一意の索引がない場合は配列フェッチを使用できません (Oracle Data Integrator で配列フェッチを空のままにしておきます)。一括更新は使用可能です。</p> <p><b>警告:</b> 使用前に正しい SQL スクリプト sql_server.sql (ドライバ・ディストリビューションに付属) を sql/server 上で実行する必要があります。</p>
<p>Hit JDBC Driver for AS/400 via Client/Access (Type 4)</p>	<p>ドライバ: hit.as400.As400Driver</p> <p>URL: jdbc:as400://&lt;IP address or name&gt;</p> <p><b>注意:</b> 追加のミドルウェアがクライアント・マシン上にある必要はありません。Client/Access サービスはすべて AS/400 上で開始する必要があります (STRHOSTSVR *ALL)。</p> <p>他のオプション</p> <pre>[;user=&lt;user&gt;] [;password=&lt;password&gt;] [;options=[secure,] [http] [&lt;Compressorclass&gt;,] [Cryptographerclass&gt;,] [;license=&lt;serial number&gt;] [;libraries=&lt;lib1,lib2,...,libn&gt;] [;ccsid=&lt;number&gt;] [;use_packages=&lt;YES NO&gt;] [;use_package_library=&lt;package library&gt;] [;use_package_name=&lt;package name&gt;] [;allow_package_update=&lt;YES NO&gt;] [;convert_ccsid_65535=&lt;YES NO&gt;]</pre>

IBM TOOLBOX JDBC Driver for AS/400 (Type 4)	<pre>[;fetch_block_size=&lt;Ksize&gt;] [;naming=&lt;sql system]</pre> <p>ドライバ: <code>com.ibm.as400.access.AS400JDBCdriver</code>  <b>URL:</b> <code>jdbc:as400://&lt;IP address or name&gt;</code></p> <p><b>注意:</b> 追加のミドルウェアがクライアント・マシン上にある必要はありません。</p>
Oracle Data Integrator AS/400 Driver Wrapper	<p>ドライバ:  <code>com.sunopsis.jdbc.driver.wrapper.SnpsDriverWrapper</code></p> <p><b>URL:</b>  <code>jdbc:snps400:&lt;machine name/IP address&gt;[;param1=value1[;param2=vaueur2...]]</code></p> <p>このドライバ・ラッパーは、状況に応じて、JT/400 ドライバのネイティブの使用を可能にします。このドライバ・ラッパーは、ローカルの AS/400 マシン（エージェントが動作しているマシン）への接続が要求されると、Native ドライバを使用します。それ以外の場合、JT/400 ドライバを使用します。このラッパーは次の 3 つの Java プロパティを使用します。</p> <ul style="list-style-type: none"> <li>• <b>HOST_NAME:</b> ローカルの AS /400 マシン用の名前のコマ区切りのリスト。</li> <li>• <b>HOST_NAME:</b> ローカルの AS /400 マシン用の IP アドレス。その IP アドレスでの接続を要求された場合に使用されます。</li> <li>• <b>SNPS_VERBOSE:</b> ドライバ・トレース・レベル。値: 0 (トレースなし) ~5 (フル・トレース)。</li> </ul> <p>詳細は、「DB2/400 データ・サーバーの作成」を参照してください。</p>