

Oracle Data Integrator Tools リファレンス

ユーザー・パラメータ

次のユーザー・パラメータが Oracle Data Integrator の動作を構成します。パラメータの設定には、「ファイル」→「ユーザー・パラメータ...」を使用します。

ユーザー・パラメータは、Oracle Data Integrator インストール・ディレクトリの/bin サブディレクトリにある userpref.xml ファイルに保存されます。

パラメータ	値	説明
同時編集のチェック	0 1	オブジェクトへの変更を保存する場合、他のユーザーによって同じオブジェクトに異なる変更が加えられていないかを確認します。別のユーザーが変更した場合は、オブジェクトを保存できません。
自動マッピング	Yes No Ask	列名の一致が検出されて新しいデータストアがインタフェースに追加されたときに、ソース列をターゲット列に自動的にマップします。
新しいロード・バランシングの使用	Yes No	ロード・バランシングを使用する場合、セッションがなくなったエージェントに、まだ起動していない他のエージェントからセッションを割り当てなおすことができます。それ以外の場合、セッションはそれぞれ1回ずつしか割り当てられません。
インタフェース・ダイアグラムのヘルプ	0 1	1の場合、データストアがアタッチされていないインタフェース・ダイアグラムを編集するとヘルプ・メッセージが表示されます。
ログをパージする前に警告	Yes No	ユーザーまたはセッション名フィルタを指定し、かつシナリオ・レポートをパージするよう選択した場合には、削除されるのはセッションにリンクしたシナリオ・レポートのみで、孤立したシナリオ・レポートは削除されないことを知らせる警告を表示します。
開く時にオブジェクトをロック	0 1 Ask	オブジェクトを編集のために開く際に: <ul style="list-style-type: none"> • 1: 自動的にロックされます。 • 0: ロックされません。 • Ask: オブジェクトをロックするかどうかユーザーに確認します。

閉じる時にオブジェクトをロック解除	0 1 Ask	変更したオブジェクトを閉じる際に: <ul style="list-style-type: none"> • 1: 自動的にロック解除されます。 • 0: ロック解除されません。 • Ask: オブジェクトをロック解除するかどうかユーザーに確認します。
シナリオにリンクされたセッションの削除	Yes No Ask	「はい」に設定した場合、オペレータの「シナリオ」ビューでシナリオを削除すると、リンクしたセッションは自動的に削除されます。
データ・サービス生成用のデフォルト・パス	ディレクトリ	生成されたデータ・サービスの格納先となるパスです。Oracle Data Integrator は、生成されたソース・コードとコンパイルされた Web サービスをこの場所に配置します。このディレクトリは、生成後に削除できる一時的な場所です。
モデル・データストアのみの処理	Yes No Ask	このモデルに存在しないデータストアのための DDL コードを生成するかどうか。Ask の場合、確認メッセージが表示されます。
オペレータの表示制限 (0=制限なし)	数値	オペレータで表示するセッションの数がこの数を超える場合、確認メッセージが表示されます。デフォルト: 100
2 回のリフレッシュ間の遅延(秒)	数値	オペレータでの 2 回のリフレッシュの間に待機する秒数。自動リフレッシュ・モードが有効の場合のみ適用されます。
デフォルトの PDF 生成ディレクトリ	ディレクトリ	レポートを生成する場合に生成された.pdf ファイルを保存するデフォルト・ディレクトリ。
デザイナのデフォルト・コンテキスト	コンテキスト名	デザイナ内で使用されるデフォルトのコンテキストです。ここで指定したコンテキストが、様々なリストにデフォルトで表示され、デザイナを開く際にも選択されます。
デフォルト・エージェント	エージェント名	オブジェクトを実行する際に、「実行」オプションのウィンドウでデフォルトで選択されるエージェント。無効なエージェント名が指定されている場合はローカル・エージェントが使用されます。
PDF ビューア	ファイルへのパス	生成される.pdf ファイルを表示するプログラムの、ファイル名を含む完全なパス。「ファイルを生成後に開きますか。」オプションを使用する場合は必須。
ダイアグラム(PNG)を保存するためのディレクトリ	ディレクトリ	共通フォーマット・デザイナを使用してモデル・ダイアグラムを印刷する場合は、生成される.png ファイルを保存するデフォルト・ディレクトリを指定します。

問合せバッファ・サイズ	数値	リポジトリにアクセスするために Data Integrator によってキャッシュされた、準備済 SQL 文の数。場合によっては、問合せキャッシュを効率的に無効にするために、これを 1 に設定する必要があります。
実行用のデフォルト・コンテキスト	コンテキスト名	オブジェクトを実行する際に、「 実行 」オプションのウィンドウでデフォルトで選択されるコンテキスト。無効なコンテキスト名が指定されている場合はデフォルトのコンテキストが使用されます。
Oracle Data Integrator タイムアウト	数値	データベース・コネクションをあきらめるまでに待機する秒数。タイムアウトの問題を頻繁に経験する場合は、この値を大きくします。デフォルト: 30。
ツリー・ビューにリソース名を表示	Yes No	「 モデル 」ビューでデータストアのリソース名を表示するかどうか。これは、 Oracle Data Integrator で使用されているリソース名が、格納先のデータ・サーバーによって認識されるリソース名と異なる場合に便利です。

このマニュアルは、**Oracle Data Integrator Tools** のリファレンス・マニュアルです。このツールを使用して統合シナリオを設計するアプリケーション開発者を読者として想定しています。

このマニュアルの構成

このマニュアルには次の章が含まれています。

- **第 1 章「Oracle Data Integrator Tools の使用」** では、**Oracle Data Integrator** を使用して統合シナリオを設計する場合の **Oracle Data Integrator Tools** の使用方法について説明します。
- **第 2 章「オープン・ツールの使用」** では、サード・パーティが開発したツールや自作ツールをインストールして使用する方法について説明します。
- **第 3 章「オープン・ツールの開発」** では、独自のオープン・ツールを作成する方法について説明します。
- **第 4 章「Oracle Data Integrator Tools リファレンス」** では、各ツールの構文について詳しく説明します。

Oracle Data Integrator Tools の使用

Oracle Data Integrator Tools (**Oracle Data Integrator** コマンドとも呼ばれます) は、実行時に特定のタスクを行うために用意されているコマンドです。これらのタスクは、一定の時間待機したり音を鳴らしたりする程度の単純なタスクのこともあれば、**ANT** スクリプトを実行したりサーバーから電子メールを読み込んだりする高度なタスクのこともあります。

デザイナーで作業する場合、ツールを追加することができます。それはパッケージ、プロシージャ・コマンドまたはナレッジ・モジュール・コマンドで使用されます。**OS** のコマンドラインから直接使用することもできます。

注意: ODI バージョン 10.1.3 以上では、ツールは **OdiXxx** と名前が変更されています。古い命名規則 **Snpxxx** もまだ有効ですが、廃止予定です。

パッケージ内で Oracle Data Integrator Tool を使用方法:

1. 変更するパッケージを開きます。
2. ツールボックス・ウィンドウが表示されない場合、パッケージ・ウィンドウのメニューバーにある「ツールボックスの表示」ボタンをクリックします。ツールボックス・ウィンドウが表示されます。
3. ツールボックスで、使用するツールのアイコンを選択し、パッケージ・ウィンドウの中をクリックします。ツール・アイコンに対応する手順がパッケージに表示されます。
4. メニューバーの「選択」ボタンをクリックし、続いてこの手順をダブルクリックします。
5. このツールのプロパティ・ウィンドウが表示されます。パラメータの値は、このプロパティ・ウィンドウで設定します。
6. 「適用」をクリックします。

パッケージでは、ツールの各手順に順番を設定し、それぞれが成功したか失敗したかに応じて整理することができます。

ナレッジ・モジュールまたはプロシージャ・コマンド内で Oracle Data Integrator Tool を使用方法:

1. ナレッジ・モジュールまたは固有プロシージャで、既存のコマンドを開きます（または新しいコマンドを追加します）。
2. 「ターゲット」タブを選択し、「テクノロジー」を「Sunopsis API」に設定します。
3. 式エディタを開いて、コマンドを入力します。式エディタには、コマンドの構文でコードのヒントを表示する機能があります。
4. コマンドのログ・レベルを選択します。コマンドが失敗した場合にのみ警告を表示するようにする場合は、「エラーの無視」ボックスを選択します。
5. 「適用」をクリックします。

注意: 「ソース」タブと「ターゲット」タブの両方で Oracle Data Integrator コマンドを使用することができます。

注意: Sunopsis API テクノロジーでは、トランザクションやコミットのオプションはサポートされていません。

コマンドラインから Oracle Data Integrator Tool を使用方法:

1. シェル (UNIX) 、コマンド・プロンプト (Windows) または QSH セッション (AS/400) を起動します。
2. Oracle Data Integrator をインストールしたディレクトリの/bin サブディレクトリに移動します。
3. 次の構文を使用して、startcmd.bat (Windows) または startcmd.sh (UNIX と AS/400) を起動します。

```
startcmd <command_name> [<command_parameters>]*
```

注意: Windows プラットフォームでは、「=」またはスペースを含むコマンド引数を二重引用符で囲む必要があります。コマンド・コールは UNIX のコマンド・コールと異なることがあ

ります。例:

```
startcmd.bat OdiSleep "-DELAY=5000" (Windows)
./call startcmd.sh OdiSleep -DELAY=5000 (UNIX)
```

注意: <command_name>パラメータでは大文字と小文字が区別されます。

ツールの高度な使用方法

ツールのパラメータ値では、変数、順序、Oracle Data Integrator API コールまたは SELECT 文の結果を直接使用することができます。

変数と順序

ツールのパラメータで変数を使用する方法は簡単です。変数は、プロジェクト変数やプロジェクト順序の場合、構文#<project_code>.<変数または順序名>で、グローバル変数やグローバル順序の場合、#GLOBAL.<変数または順序名>で指定することができます。引用符で変数値を囲む必要はありません。

順序または変数をコマンドのコードに挿入する場合、式エディタを使用できます。この場合、変数または順序名の前にプロジェクト・コードまたは GLOBAL を付ける必要はありません。

例:

```
OdiOutFile -FILE=/temp/testsequence.txt
順序値は#SEQ01_NEXTVAL (増分済)、変数値は#VAR01 です。
```

Oracle Data Integrator API コール

Oracle Data Integrator API のメソッドは、リポジトリに含まれるメタデータからコードを生成するために使用されます。コールは API の値の中に対して実行できます。詳細は、メソッドを使用するための構文を参照してください。

注意: 使用できる API は、ツールが使用される場所により異なります (たとえば、ナレッジ・モジュールまたはプロシージャのオプションを取得する `getOption` メソッドは、パッケージ手順で使用されても無意味です)。

例:

```
OdiOutFile -FILE=/temp/testoption.txt
MY_KM_OPTION という名前のナレッジ・モジュール・オプションの値は
<%=odiRef.getOption( "MY_KM_OPTION" )%>です。
```

SELECT 文の結果の使用

ナレッジ・モジュールまたはプロシージャを設計する場合、コマンドの「ソース」タブで SQL の SELECT 文を指定し (SQL に対応するテクノロジーで実行)、「ターゲット」タブで Data Integrator Tool を使用するよう指定する (Sunopsis API テクノロジー上で実行) ことが可能です。

SELECT 文によって返された行それぞれに対して、Oracle Data Integrator ツールのコマンドが実行されます。SELECT 文により返されたフィールドの値は、先頭に文字「#」を付けることで、Oracle Data Integrator ツールのコマンド構文で使用することができます。これらのフィールド名はコマンドの実行前に代入されます。

注意: 式でエイリアスを使用している場合、SQL の SELECT 文によって返されるフィールド名でもエイリアス名を使用してください。

例:

- ソース・コマンド (Oracle) :
`select COUNTRY_NAME, POPULATION/1000 as POP_SMALL from COUNTRY`
- ターゲット・コマンド (Sunopsis API) :
`OdiOutFile -FILE=/temp/testsequence.txt`
`There is #POP_SMALL thousands inhabitants in #COUNTRY_NAME`

オープン・ツールの使用

概要

オープン・ツール機能は、パッケージやプロシージャで使用できるカスタムのサード・パーティ・ツールを開発するための、拡張可能なプラットフォームです。Oracle Data Integrator 付属の標準ツールと同じように、オペレーティング・システムと協調して動作したり、データを操作したりすることができます。

オープン・ツールは Java で書かれています。独自のオープン・ツールの作成方法については「オープン・ツールの開発」で説明しています。

オープン・ツールは、次のいくつかのファイルを含む Java パッケージ (.zip または .jar) の形式で配布されます。

- Java のコンパイル済.class ファイル
- アイコン・ファイルなどその他のリソース

インストール

オープン・ツールをインストールする方法:



1. すべての Oracle Data Integrator ウィンドウを閉じます。新しいオープン・ツールは起動時のみ認識されます。
2. /plugins サブディレクトリに Java パッケージをコピーします (unzip しないでください)。クラスパスにあるディレクトリであれば、他のディレクトリも使用できます。
3. デザイナを起動します。
4. 「ファイル」メニューから、「オープン・ツールの追加/削除...」を選択します。

オープン・ツールの追加/削除ウィンドウが表示されます。

新しいオープン・ツールを追加する方法

新しいツールを追加するには、次のいずれかの操作を行います。

1. 「オープン・ツール・クラス名」編集ボックスにクラスの名前を入力します。
または
1. 「検索」ボタンをクリックし、追加するオープン・ツールの Java クラスの名前を参照します。クラスを名前を検索するには、一番上にあるボックスに名前の一部を入力します。

2. 「OK」をクリックします。
オープン・ツールでないものも含め、Oracle Data Integrator で現在使用可能なすべてのクラスが表示されることに注意してください。クラスを追加するには、そのクラスの名前を知っている必要があります。
3.  「オープン・ツールの追加」 ボタンをクリックします。
4. 目的のオープン・ツールが含まれている行をクリックします。
 - クラスパス上で見つかったら、提供されているアイコン、構文、説明、プロバイダおよびバージョン番号が表示されます。
 -  見つからなかった場合、このアイコンが表示されます。クラスパスを変更するか、オープン・ツールを正しいディレクトリに移動します。

警告: 名前が *Snp* または *Odi* で始まるオープン・ツールは起動できません。名前の先頭がこれらの文字列であるオープン・ツールは無視されます。

オープン・ツールを削除する方法:

1. 削除するオープン・ツールの名前をクリックします。
2. 「削除」をクリックします。

オープン・ツールの使用

パッケージ内のオープン・ツールを使用する方法:

1. ツールの「プラグイン」グループから、インストールされているオープン・ツールを選択します。
2. クリックします。パッケージ・ダイアグラムの上に配置されます。
3. 「プロパティ」ペインで、ツール・コールのパラメータを設定します。

関連項目: ステップの追加

プロシージャ内のオープン・ツールを使用する方法:

1. 新しい手順を作成します。
2. テクノロジには Sunopsis API を選択します。
3. 式エディタを使用してコマンドを作成します。パッケージ内と同一の構文を使用します。

オープン・ツールの開発

概要

オープン・ツールは、インタフェース `oracle.odi.sdk.opentools.IOpenTool` を実装する、コンパイル済 Java クラスが含まれる Java パッケージです。すべてのクラスとメソッドについての詳しい説明は、Open Tools SDK ドキュメント (JavaDoc) を参照してください。

また、オープン・ツール・パッケージには通常、2つのアイコンも含まれています。これらは、オープン・ツールを表す Oracle Data Integrator グラフィカル・インタフェースとして使用されません。

クラス

クラスまたはインタフェース	説明
IOpenTool	すべてのオープン・ツールが実装する必要があるインタフェース。
OpenToolAbstract	いくつかのヘルパー・メソッドを持っているインタフェースの抽象化。インタフェースを直接実装するより、なるべくこのクラスを拡張します。
IOpenToolParameter	オープン・ツールによって使用されるパラメータが実装する必要があるインタフェース。ほとんどの場合は、このインタフェースを実装するのではなく、OpenToolParameter を使用します。
OpenToolParameter	IOpenToolParameter の完全な実装。各 OpenToolParameter が1つのパラメータを持ちます。
OpenToolsExecutionException	オープン・ツールのメソッドによって必要に応じてスローされる例外クラス。
SimpleOpenToolExample	単純なサンプル・オープン・ツール。これをもとに開発を開始できます。

新しいオープン・ツールの開発

次の説明は、基本的なオープン・ツールである、SimpleMessageBox の開発について説明しています。このクラスのソース・コードは、demo/plugins/src ディレクトリにあります。

1. 構文を定義します。この例では、オープン・ツールは次のようにコールされます。
SimpleMessageBox "-TEXT=<text message>" "-TITLE=<window title>"
2. 16x16 および 32x32 のアイコンを作成します（通常は.gif 形式）。
3. クラスを作成し、実装します。次を参照してください。
4. クラスをコンパイルして、2つのアイコン・ファイルを持つパッケージを作成します。
5. 「オープン・ツールの使用」の説明に従って、オープン・ツールをインストールします。

クラスの実装

宣言

パッケージの命名

クラスは適切な名前を持つパッケージに置きます。パッケージ名は、インストール時にオープン・ツールを識別するために使用されます。

```
package com.myCompany.OpenTools;
```

クラスの宣言

オープン・ツールの開発には、2つの基本的な手法があります。

- 既存のクラスを拡張してオープン・ツールに変換する手法。この場合、既存のクラスに直接、インタフェース `IOpenTool` を実装するだけです。
- 新規のクラスを開発する手法。この場合、抽象クラス `OpenToolAbstract` を拡張するのが最も容易です。これにはパラメータで作業するための追加のヘルパー・メソッドが含まれません。

```
public class SimpleMessageBox extends OpenToolAbstract {
```

パッケージ

ほとんどすべてのオープン・ツールで、次の Open Tool SDK パッケージをインポートする必要があります。

```
import oracle.odi.sdk.opentools.IOpenTool; /* これら3つのクラスはすべてのオープン・ツール・クラスに必要です */
import oracle.odi.sdk.opentools.IOpenToolParameter;
import oracle.odi.sdk.opentools.OpenToolExecutionException;
import oracle.odi.sdk.opentools.OpenToolAbstract; /* このオープン・ツール用に拡張する抽象クラス */
import oracle.odi.sdk.opentools.OpenToolParameter; /* パラメータ用に使用するクラス */
```

さらに、この例ではメッセージ・ボックスを作成するための次のパッケージも必要です。

```
import javax.swing.JOptionPane; /* この例で使用されるメッセージ・ボックスのために必要 */
```

パラメータ

`OpenToolParameter` オブジェクトを格納するプロパティを追加します。このプロパティは、構文に対してオブジェクトを定義するためと、実際のユーザーからパラメータの値を取得するための両方の目的で使用されます。このオープン・ツールのパラメータは、次のように静的配列を使用して定義するのが最も簡単です。この配列はプライベートにします。アクセッサ関数を介してアクセスされるからです。

```
private static final IOpenToolParameter[] mParameters = new
    IOpenToolParameter[]
{
    new OpenToolParameter("-TEXT", "Message text", "Text to show in the
        messagebox (Mandatory).", true),
    new OpenToolParameter("-TITLE", "Messagebox title", "Title of the
        messagebox.", false)
};
```

`OpenToolParameter()` コンストラクタに渡される4つのパラメータは次のとおりです。

1. パラメータのコード（先頭のハイフンを含む）。このコードは、必ず `getSyntax()` によって返される構文に対応している必要があります。

2. ユーザーにわかりやすい名前。ユーザーがグラフィカル・インタフェースを使用してパラメータを設定する場合に使用されます。
3. 説明のヘルプ・テキスト。
4. パラメータが必須かどうか。これはユーザーのための目印です。

注意: Oracle Data Integrator では、パラメータに付ける必須フラグはなくてもかまいません。入力されるパラメータの組合せはすべて処理可能である必要があります。

パラメータを取得するにはアクセッサ関数 `getParameters()` を実装する必要があります。

```
public IOpenToolParameter[] getParameters()
{
    return mParameters;
}
```

情報関数

オープン・ツールに関する情報を返す関数 `getDescription()`、`getVersion()`、`getProvider()` を実装します。

```
public String getDescription() { return "This Open Tool displays a
message box when executed."; }
public String getVersion() { return "v1.0"; }
public String getProvider() { return "My Company, Inc."; }
```

`getSyntax()` 関数により、Oracle Data Integrator のグラフィカル・インタフェースに表示されるオープン・ツールの名前と、パラメータの初期値が決定されます。ここでのパラメータ名が、`getParameters()` によって返されたパラメータ名と一致することを確認します。

```
public String getSyntax()
{
    return "SimpleMessageBox \"-TEXT=<テキスト・メッセージ>\\"-TITLE=<ウィンドウ・タイトル>\\"";
}
```

`getIcon()` メソッドが、適切なサイズの 2 つのイメージにパスを返します。次の例のようになります。

```
public String getIcon(int pIconType)
{
    switch (pIconType)
    {
        case IOpenTool.SMALL_ICON:
            return "/com/myCompany/OpenTools/images/SimpleMessageBox_16.gif";
        case IOpenTool.BIG_ICON:
            return "/com/myCompany/OpenTools/images/SimpleMessageBox_32.gif";
        default:
            return "";
    }
}
```

```
}
```

実行

最後は、言うまでもなく、オープン・ツールが提供する機能を実際に実行する `execute()` メソッドです。この場合、メッセージ・ボックスが表示されます。OpenToolAbstract クラスを拡張する場合、`getParameterValue()` メソッドを使用すると、実行時に設定されるパラメータ値を容易に取得できます。

注意: すべての実行を捕捉して、`OpenToolExecutionException` のみが発生するようにする必要があります。

```
public void execute() throws OpenToolExecutionException
{
    try
    {
        if (getParameterValue("-TITLE") == null || getParameterValue("-TITLE").equals("")) /* title was not filled in by user */
        {
            JOptionPane.showMessageDialog(null, (String)
getParameterValue("-TEXT"), (String) "Message",
JOptionPane.INFORMATION_MESSAGE);
        } else
        {
            JOptionPane.showMessageDialog(null, (String)
getParameterValue("-TEXT"),
                (String) getParameterValue("-TITLE"),
                JOptionPane.INFORMATION_MESSAGE);
        }
    }
    /* Traps any exception and throw them as OpenToolExecutionException */
    catch (IllegalArgumentException e)
    {
        throw new OpenToolExecutionException(e);
    }
}
```

実行時のオープン・ツール

一般に、オープン・ツール・クラスは非常に簡潔にインスタンス化できます。次のように使用されます。

インストール

ユーザーがオープン・ツールのインストールを選択すると、Oracle Data Integrator はクラスをインスタンス化して、メソッド `getDescription()`、`getProvider()`、`getIcon()` および `getVersion()` をコールして、クラスに関する情報を取得します。

パッケージでの使用

オープン・ツールがパッケージ内で使用された場合、そのクラスは、メソッド `getDescription()`、`getProvider()`、`getIcon()` および `getVersion()` をコールするために一時的にインスタンス化されます。さらに、オープン・ツールのコード名とそのデフォルト引数を取得するために、`getSyntax()` がコールされます。メソッド `getParameters()` は、引数のリストをユーザーに対して表示するためにコールされます。

実行

Open Tools がパッケージまたはプロシージャ内で実行されるたびに、クラスはその都度インスタンス化されます。実行の完了後までインスタンスが維持されることはありません。`execute()` メソッドは 1 回だけコールされます。

関連資料

オープン・ツールの使用

Open Tools SDK ドキュメント (JavaDoc)

Oracle Data Integrator Tools リファレンス

Oracle Data Integrator Tools のリスト

Oracle Data Integrator Tools のリストを次に示します。

メタデータ

- `OdiReverseGetMetaData`
- `OdiReverseResetTable`
- `OdiReverseSetMetaData`

Oracle Data Integrator オブジェクト

- `OdiDeleteScen`
- `OdiExportAllScen`
- `OdiExportEnvironmentInformation`
- `OdiExportLog`
- `OdiExportMaster`
- `OdiExportObject`

- OdiExportScen
- OdiExportWork
- OdiGenerateAllScen
- OdiImportObject
- OdiImportScen

ユーティリティ

- OdiAnt
- OdiBeep
- OdiDataQuality
- OdiKillAgent
- OdiOSCommand
- OdiPingAgent
- OdiPurgeLog
- OdiReinitializeSeq
- OdiStartScen
- OdiUpdateAgentSchedule

インターネット

- OdiInvokeWebService (OdiExecuteWebService に代わって使用されます)
- OdiReadMail
- OdiSendMail

ファイル

- OdiFileAppend
- OdiFileCopy
- OdiFileDelete
- OdiFileMove
- OdiFileWait
- OdiMkDir
- OdiOutFile
- OdiSqlUnload
- OdiUnZip
- OdiZip

SAP

- OdiSAPALEClient
- OdiSAPALEServer

イベント検出

- OdiFileWait
- OdiReadMail
- OdiSleep
- OdiWaitForChildSession
- OdiWaitForData
- OdiWaitForLogData
- OdiWaitForTable

チェンジ・データ・キャプチャ

- OdiRefreshJournalCount
- OdiRetrieveJournalData
- OdiWaitForData
- OdiWaitForLogData
- OdiWaitForTable

メタデータ

OdiReverseGetMetaData

使用方法

OdiReverseGetMetaData -MODEL=<model_id>

説明

パーソナライズされたリバースでは、次の一時表をロードすることができます。

- SNP_REV_TABLE: データストアの表
- SNP_REV_COL: 列の表
- SNP_REV_KEY: 一意のキーの表
- SNP_REV_KEY_COL: キー列の表
- SNP_REV_JOIN: 外部キーの表
- SNP_REV_JOIN_COL: 外部キー列の表
- SNP_REV_COND: 条件（チェック制約）の表

このときのロードは、特定の処理を使用して、ナレッジ・モジュールまたは Oracle Data Integrator インタフェースのメタデータを伴って行われます。処理は次の 3 つの大きな手順に要約できます。

- リバース・エンジニアリングの一時表のページ
- メタデータからこのリバース・エンジニアリング表を提供する、パーソナライズされた処理
- リポジトリのメタデータの統合の実行

このコマンドを使用すると、一時リバース表で指定されたモデルをリバース・エンジニアリングすることができます。このコマンドは、前に `OdiReverseResetTable`、後に `OdiReverseSetMetaData` が置かれます。

注意: このコマンドは標準のリバース・エンジニアリングと同一のテクニックを使用し、ドライバの機能に依存します。

パラメータ

パラメータ	必須かどうか	説明
<code>-MODEL=<model_id></code>	必須	リバース・エンジニアリングするモデル

例

RKM の現在モデルをリバースします。

```
OdiReverseGetMetaData -MODEL=<%=odiRef.getModel("ID")%>
```

OdiReverseResetTable

使用方法

```
OdiReverseResetTable -MODEL=<model_id>
```

説明

パーソナライズされたリバースでは、次の一時表をロードすることができます。

- `SNP_REV_TABLE`: データストアの表
- `SNP_REV_COL`: 列の表
- `SNP_REV_KEY`: 一意のキーの表
- `SNP_REV_KEY_COL`: キー列の表
- `SNP_REV_JOIN`: 外部キーの表
- `SNP_REV_JOIN_COL`: 外部キー列の表
- `SNP_REV_COND`: 条件（チェック制約）の表

このときのロードは、特定の処理を使用して、ナレッジ・モジュールまたは Oracle Data Integrator インタフェースのメタデータを伴って行われます。処理は次の 3 つの大きな手順に要約できます。

- リバース・エンジニアリングの一時表のページ
- メタデータからこのリバース・エンジニアリング表を提供する、パーソナライズされた処理
- リポジトリのメタデータの統合の実行

コマンド **OdiReverseResetTable** は、指定されたモデルについて、これらの一時表をパーズすることができます。

パラメータ

パラメータ	必須かどうか	説明
-MODEL=<model_id>	必須	リバースする必要があるモデルの内部識別子

例

```
OdiReverseResetTable -MODEL=123001
```

OdiReverseSetMetaData

使用方法

```
OdiReverseSetMetaData -MODEL=<model_id>
```

説明

パーソナライズされたリバース・エンジニアリングでは、次の一時表をロードすることができません。

- SNP_REV_TABLE: データストアの表
- SNP_REV_COL: 列の表
- SNP_REV_KEY: 一意のキーの表
- SNP_REV_KEY_COL: キー列の表
- SNP_REV_JOIN: 外部キーの表
- SNP_REV_JOIN_COL: 外部キー列の表
- SNP_REV_COND: 条件（チェック制約）の表

このときのロードは、特定の処理を使用して、ナレッジ・モジュールまたは **Oracle Data Integrator** インタフェースのメタデータを伴って行われます。処理は次の3つの大きな手順に要約できます。

- リバース・エンジニアリングの一時表のパーズ
- メタデータからこのリバース・エンジニアリング表を提供する、パーソナライズされた処理
- リポジトリのメタデータの統合の実行

コマンド **OdiReverseSetMetaData** は、指定されたデータ・モデルについて、これらの一時表からリポジトリにデータを統合するために使用されます。

パラメータ

パラメータ	必須かどうか	説明
-------	--------	----

`-MODEL=<model_id>`

必須

リバースするモデルの内部識別子

例

```
OdiReverseSetMetaData -MODEL=123001
```

Oracle Data Integrator オブジェクト

OdiDeleteScen

使用方法

```
OdiDeleteScen -SCEN_NAME=<name> -SCEN_VERSION=<version>
```

説明

指定された名前とバージョンのシナリオを削除します。

パラメータ

パラメータ	必須かどうか	説明
<code>-SCEN_NAME=<name></code>	必須	削除するシナリオの名前
<code>-SCEN_VERSION=<version></code>	必須	削除するシナリオのバージョン

例

シナリオ TEST のバージョン 1 を削除します。

```
OdiDeleteScen -SCEN_NAME=TEST -SCEN_VERSION=1
```

OdiExportAllScen

使用方法

```
OdiExportAllScen -TODIR=<Dir> [-FORCE_OVERWRITE=<yes|no>] [-FROM_PROJECT=<I_Project>] [-FROM_FOLDER=<I_Folder>] [-FROM_PACKAGE=<I_Package>] [-RECURSIVE_EXPORT=<yes|no>] [-XML_VERSION=<1.0>] [-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>] [-EXPORT_PACK=<YES|NO>] [-EXPORT_POP=<YES|NO>] [-EXPORT_TRT=<YES|NO>] [-EXPORT_VAR=<YES|NO>]
```

説明

接続しているリポジトリから、`SCEN_<scenario name><scenario version>.xml` という名前の XML エクスポート・ファイルに、シナリオのグループをエクスポートします。このコマンドは、デザイナーおよびオペレータに用意されているエクスポート機能と同等の動作をします。

パラメータ

パラメータ	必須かどうか	説明
-TODIR=<Dir>	必須	エクスポート・ファイルが作成されるディレクトリ
- FORCE OVERWRITE=<yes no>	省略可	yes に設定すると、既存のエクスポート・ファイルが警告なく上書きされます。デフォルトは yes です。
- FROM PROJECT=<I Project>	省略可	エクスポートするシナリオが含まれているプロジェクトの ID。プロジェクト・ウィンドウの「バージョン」タブに表示される 内部識別子 です。このパラメータを設定しない場合、すべてのプロジェクトのシナリオがエクスポート対象になります。
- FROM FOLDER=<I Folder>	省略可	エクスポートするシナリオが含まれているフォルダの ID。フォルダ・ウィンドウの「バージョン」タブに表示される 内部識別子 です。このパラメータを設定しない場合、すべてのフォルダのシナリオがエクスポート対象になります。
- FROM PACKAGE=<I Package>	省略可	エクスポートするシナリオが含まれているソース・パッケージの ID。パッケージ・ウィンドウの「バージョン」タブに表示される 内部識別子 です。このパラメータを設定しない場合、すべてのコンポーネントのシナリオがエクスポート対象になります。
- RECURSIVE EXPORT=<yes no>	省略可	yes に設定すると、すべての子オブジェクト（スケジュール）がシナリオとともにエクスポートされます。デフォルトは yes です。
-XML_VERSION=<1.0>	省略可	XML ヘッダーに表示される XML バージョンを設定します。デフォルトは 1.0 です。
- XML_CHARSET=<charset>	省略可	XML エクスポート・ファイル内に指定されるエンコーディング。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
- JAVA_CHARSET=<charset>	省略可	ターゲット・ファイルのエンコーディング。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-EXPORT_PACK=<YES NO>	省略可	パッケージにアタッチされたシナリオがエクスポートされるかどうかを示します。デフォルト値は YES です。
-EXPORT_POP=<YES NO>	省略可	インタフェースにアタッチされたシナリオがエクスポートされるかどうかを示します。デフォルト値は NO です。

-EXPORT_TRT=<YES NO>	省略可	プロシージャにアタッチされたシナリオがエクスポートされるかどうかを示します。デフォルト値はNOです。
-EXPORT_VAR=<YES NO>	省略可	変数にアタッチされたシナリオがエクスポートされるかどうかを示します。デフォルト値はNOです。

例

内部識別子が 10022 である DW01 プロジェクトから、/temp/ディレクトリへ、すべてのシナリオをすべての依存オブジェクトとともにエクスポートします。

```
OdiExportAllScen -FROM_PROJECT=10022 -TODIR=/temp/ -RECURSIVE_EXPORT=yes
```

OdiExportEnvironmentInformation

使用方法

```
OdiExportEnvironmentInformation -TODIR=<toDir> -FILE_NAME=<FileName> [-CHARSET=<charset>] [-SNP_INFO_REC_CODE=<row_code>] [-MASTER_REC_CODE=<row_code>] [-WORK_REC_CODE=<row_code>] [-AGENT_REC_CODE=<row_code>] [-TECHNO_REC_CODE=<row_code>] [-RECORD_SEPARATOR_HEX=<rec_sep>] [-FIELD_SEPARATOR_HEX=<field_sep>] [-TEXT_SEPARATOR=<text_sep>]
```

説明

技術環境の詳細をカンマ区切り (.csv) ファイルに抽出し、指定したディレクトリにエクスポートします。この情報は、サポート用として役立ちます。

パラメータ

パラメータ	必須かどうか	説明
-TODIR=<toDir>	必須	エクスポートのターゲット・ディレクトリ。
-FILE_NAME=<FileName>	必須	CSV エクスポート・ファイルの名前。デフォルトは snps_tech_inf.csv です。
-CHARSET=<charset>	省略可	エクスポート・ファイルのキャラクタ・セット。XML ファイル・ヘッダーのパラメータ encoding です。 <?xml version="1.0" encoding="ISO-8859-1"?>
-SNP_INFO_REC_CODE=<row_code>	省略可	Oracle Data Integrator の現在のリリースおよび現在のユーザーが記録される行を識別するためのコード。このコードはレコードの最初のフィールドに使用される。デフォルトは SUNOPSIS です。

- MASTER REC CODE=<row code>	省略可	マスター・リポジトリに関する情報を含む行のコード。デフォルトは MASTER です。
- WORK REC CODE=<row code>	省略可	作業リポジトリに関する情報を含む行のコード。デフォルトは WORK です。
- AGENT REC CODE=<row code>	省略可	稼働中の様々なエージェントに関する情報を含む行のコード。デフォルトは AGENT です。
- TECHNO REC CODE=<row code>	省略可	データ・サーバーやそのバージョンなどに関する情報を含む行のコード。デフォルトは TECHNO です。
- RECORD SEPARATOR HEXA=<rec_sep>	省略可	ファイル内で行と行（またはレコードとレコード）を区切る 1 つ以上の 16 進コード文字。デフォルトは 00D0A です。
- FIELD SEPARATOR HEXA=<field_sep>	省略可	レコード内でフィールドとフィールドを区切る 1 つ以上の 16 進コード文字。デフォルトは 2C です。
- TEXT SEPARATOR=<text_sep>	省略可	STRING フィールドを区切る 16 進コード文字。デフォルトは 22 です。

例

技術環境の詳細を /temp/snps_tech_inf.csv エクスポート・ファイルにエクスポートします。

```
OdiExportEnvironmentInformation "-TODIR=/temp/" "-
FILE_NAME=snps_tech_inf.csv" "-CHARSET=ISO8859_1" "-
SNP_INFO_REC_CODE=SUNOPSIS" "-MASTER_REC_CODE=MASTER" "-
WORK_REC_CODE=WORK" "-AGENT_REC_CODE=AGENT" "-TECHNO_REC_CODE=TECHNO" "-
RECORD_SEPARATOR_HEX=0D0A" "-FIELD_SEPARATOR_HEX=2C" "-
TEXT_SEPARATOR_HEX=22"
```

OdiExportLog

使用方法

```
OdiExportLog -TODIR=<toDir> [-ZIPFILE_NAME=<zipFileName>] [-
XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>] [-FROMDATE=<fromdate>]
[-TODATE=<todate>] [-AGENT=<agent>] [-CONTEXT=<context>] [-
STATUS=<status>] [-USER_FILTER=<user>] [-SESS_NAME=<session>]
```

説明

過去のセッション情報を ZIP エクスポート・ファイルにエクスポートします。

パラメータ

パラメータ	必須かどうか	説明
-TODIR=<toDir>	必須	エクスポートのターゲット・ディレクトリ。
-ZIPFILE_NAME=<zipFileName>	省略可	圧縮ファイルの名前。
-XML_CHARSET=<charset>	省略可	エクスポート・ファイル内に指定される XML バージョン。XML ファイル・ヘッダーのパラメータ <code>xml version</code> です。 <?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO-8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-JAVA_CHARSET=<charset>	省略可	結果ファイルの java 文字エンコーディング。デフォルト値は ISO8859_1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-FROMDATE=<fromdate>	省略可	エクスポートの開始日。形式は「yyyy/MM/dd hh:mm:ss」。この日付以降のすべてのセッションがエクスポートされます。
-TODATE=<todate>	省略可	エクスポートの終了日。形式は「yyyy/MM/dd hh:mm:ss」。この日付までのすべてのセッションがエクスポートされます。
-AGENT=<agent>	省略可	指定した名前 (<agent>) のエージェントによって実行されたセッションのみをエクスポートします。
-CONTEXT=<context>	省略可	指定したコンテキスト (<context>) で実行されたセッションのみをエクスポートします。
-STATUS=<status>	省略可	指定した状態のセッションのみをエクスポートします。状態の種類は、完了、エラー、キュー、実行中、待機中および警告です。
-USER_FILTER=<user>	省略可	指定したユーザー (<user>) によって開始されたセッションのみをエクスポートします。
-SESS_NAME=<session>	省略可	エクスポートするセッションの名前。

例

ログ・ファイルをエクスポートして圧縮し、/temp/log2.zip エクスポート・ファイルを生成します。

```
OdiExportLog "-TODIR=/temp/" "-ZIPFILE_NAME=log2.zip" "-XML_CHARSET=ISO-8859-1" "-JAVA_CHARSET=ISO8859_1"
```

OdiExportMaster

使用方法

```
OdiExportMaster -TODIR=<toDir> [-ZIPFILE_NAME=<zipFileName>] [-EXPORT_SOLUTIONS=<yes|no>] [-EXPORT_VERSIONS=<yes|no>] [-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>]
```

説明

マスター・リポジトリを特定のディレクトリまたは ZIP ファイルにエクスポートします。マスター・リポジトリに格納されているバージョンやソリューションをオプションでエクスポートすることもできます。

パラメータ

パラメータ	必須かどうか	説明
-TODIR=<toDir>	必須	エクスポートのターゲット・ディレクトリ。
-ZIPFILE_NAME=<zipFileName>	省略可	圧縮ファイルの名前。
-EXPORT_SOLUTIONS=<yes no>	省略可	リポジトリに格納されているすべてのソリューションをエクスポートします。
-EXPORT_VERSIONS=<yes no>	省略可	リポジトリに格納されているオブジェクトのすべてのバージョンをエクスポートします。
-XML_CHARSET=<charset>	省略可	エクスポート・ファイル内に指定される XML バージョン。XML ファイル・ヘッダーのパラメータ <code>xml version</code> です。 <code><?xml version="1.0" encoding="ISO-8859-1"?></code> のようなタグで指定されます。デフォルト値は ISO-8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-JAVA_CHARSET=<charset>	省略可	結果ファイルの java 文字エンコーディング。デフォルト値は ISO8859_1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。

例

マスター・リポジトリをエクスポートして圧縮し、 /temp/ディレクトリ内に export.zip ファイルを生成します。

```
OdiExportMaster "-TODIR=/temp/" "-ZIPFILE_NAME=export.zip" "-
XML_CHARSET=ISO-8859-1" "-JAVA_CHARSET=ISO8859_1" "-EXPORT_VERSIONS=YES"
```

OdiExportObject

使用方法

```
OdiExportObject -CLASS_NAME=<class_name> -I_OBJECT=<I_Object> -
FILE_NAME=<FileName> [-FORCE_OVERWRITE=<yes|no>] [-
RECURSIVE_EXPORT=<yes|no>] [-XML_VERSION=<1.0>] [-XML_CHARSET=<charset>]
[-JAVA_CHARSET=<charset>]
```

説明

接続しているリポジトリから XML エクスポート・ファイルにオブジェクトをエクスポートします。このコマンドは、ユーザー・インタフェースから使用できるエクスポート機能と同等の動作をします。

パラメータ

パラメータ	必須かどうか	説明
- CLASS_NAME=<class name>	必須	エクスポートするオブジェクトのクラス。クラスのリストは次を参照してください。
- I_OBJECT=<I_Object>	必須	オブジェクト識別子。オブジェクト編集ウィンドウの「バージョン」タブに表示される 内部識別子 です。
- FILE_NAME=<FileName>	必須	エクスポート・ファイルの名前。
- FORCE_OVERWRITE=<yes no>	省略可	yes に設定すると、同一の名前を持つ既存のエクスポート・ファイルが強制的に上書きされます。デフォルトは yes です。
- RECURSIVE_EXPORT=<yes no>	省略可	yes に設定すると、すべての子オブジェクトが現在のオブジェクトとともにエクスポートされます。たとえば、プロジェクトをエクスポートしている場合、このプロジェクト内のすべてのフォルダ、KM などがプロジェクト・エクスポート・ファイルにエクスポートされます。デフォルトは yes です。
-XML_VERSION=<1.0>	省略可	XML ヘッダーに表示される XML バージョンを設定します。デフォルトは 1.0 です。
- XML_CHARSET=<charset>	省略可	XML ファイル内に指定されるエンコーディング。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。

- JAVA CHARSET=<charset>	省略可	ターゲット・ファイルのエンコーディング。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-----------------------------	-----	--

クラスのリスト

オブジェクト	クラス名
列	Snpcol
条件/フィルタ	Snpcond
コンテキスト	Snpcontext
データ・サーバー	Snpcconnect
データストア	Snpcsttable
フォルダ	Snpcfolder
インタフェース	Snpcpop
言語	Snpclang
モデル	Snpcmodel
パッケージ	Snpcpackage
物理スキーマ	Snpcpschema
プロシージャまたは KM	Snpcprt
プロシージャまたは KM のオプション	Snpcuserexit
プロジェクト	Snpcproject
参照	Snpcjoin
シナリオ	Snpcscen
順序	Snpcsequence
手順	Snpcstep
サブモデル	Snpcsubmodel
テクノロジー	Snpctechno
ユーザー関数	Snpcufunc

変数

SnpVar

例

内部識別子が 10022 である DW01 プロジェクトを、/temp/dw1.xml エクスポート・ファイルへ、すべての依存オブジェクトとともにエクスポートします。

```
OdiExportObject -CLASS_NAME=SnpProject -I_OBJECT=10022 -
FILE_NAME=/temp/dw1.xml -FORCE_OVERWRITE=yes -RECURSIVE_EXPORT=yes
```

OdiExportScen

使用方法

```
OdiExportScen -SCEN_NAME=<ScenName> -SCEN_VERSION=<ScenVersion> -
FILE_NAME=<FileName> [-FORCE_OVERWRITE=<yes|no>] [-
RECURSIVE_EXPORT=<yes|no>] [-XML_VERSION=<1.0>] [-
XML_CHARSET=<encoding>] [-JAVA_CHARSET=<encoding>]
```

説明

現在の作業リポジトリに含まれるシナリオを、XML 形式で自動的にエクスポートします。

パラメータ

パラメータ	必須かどうか	説明
-SCEN_NAME=<ScenName>	必須	エクスポートするシナリオの名前。
-SCEN_VERSION=<ScenVersion>	必須	エクスポートするシナリオのバージョン。
-FILE_NAME=<FileName>	必須	エクスポート・ファイルの名前。
-FORCE_OVERWRITE=<yes no>	省略可	Yes の場合、既存のエクスポート・ファイルが上書きされます。
-RECURSIVE_EXPORT=<yes no>	省略可	シナリオ配下のオブジェクトを強制的にエクスポートします。デフォルトは yes です。
-XML_VERSION=<1.0>	省略可	生成された XML エクスポート・ファイル内に指定されるバージョン。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は 1.0 です。
-XML_CHARSET=<encoding>	省略可	XML ファイル内に指定されるエンコーディング。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL

- JAVA CHARSET=<encoding>	省略可	http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。 ターゲット・ファイルのエンコーディング。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
------------------------------	-----	---

OdiExportWork

使用方法

```
OdiExportWork -TODIR=<toDir> [-ZIPFILE_NAME=<zipFileName>] [-XML_CHARSET=<charset>] [-JAVA_CHARSET=<charset>]
```

説明

作業リポジトリを特定のディレクトリまたは ZIP エクスポート・ファイルにエクスポートします。

パラメータ

パラメータ	必須かどうか	説明
-TODIR=<toDir>	必須	エクスポートのターゲット・ディレクトリ。
-ZIPFILE_NAME=<zipFileName>	省略可	圧縮ファイルの名前。
-XML_CHARSET=<charset>	省略可	エクスポート・ファイル内に指定される XML バージョン。XML ファイル・ヘッダーのパラメータ <code>xml version</code> です。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO-8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-JAVA_CHARSET=<charset>	省略可	結果ファイルの java 文字エンコーディング。デフォルト値は ISO8859_1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。

例

作業リポジトリをエクスポートして圧縮し、/temp/workexport.zip エクスポート・ファイルを生成します。

```
OdiExportWork "-TODIR=/temp/" "-ZIPFILE_NAME=workexport.zip"
```

OdiGenerateAllScen

使用方法

```
OdiGenerateAllScen -PROJECT=<Project_id> [-FOLDER=<Folder_id>] [-MODE=<REPLACE|CREATE>] [-GRPMARKER=<marker_group_code> -MARKER=<marker_code>] [-GENERATE_PACK=<YES|NO>] [-GENERATE_POP=<YES|NO>] [-GENERATE_TRT=<YES|NO>] [-GENERATE_VAR=<YES|NO>]
```

説明

フォルダまたはプロジェクトに含まれる、マーカーによってフィルタリングされたコンポーネントから、シナリオのセットを自動的に生成します。

パラメータ

パラメータ	必須かどうか	説明
-PROJECT=<Project_id>	必須	シナリオを生成するコンポーネントが含まれているプロジェクト。
-FOLDER=<Folder_id>	省略可	シナリオを生成するコンポーネントが含まれているフォルダ。
-MODE=<REPLACE CREATE>	省略可	シナリオ生成モード: <ul style="list-style-type: none"> 置換:最後に生成されたシナリオが、新しく生成されたシナリオで上書きされます。このシナリオにリンクされたスケジュールはすべて削除されます。 作成 (デフォルト):最後に生成されたシナリオと名前が同じで、バージョン番号が自動増分された新しいシナリオを作成します。シナリオが未作成の場合、新しいシナリオの名前はコンポーネントと同じになります。 <p>注意:最後のシナリオのバージョンが整数の場合、生成モードに「作成」を選択すると、1だけ自動増分されます。それ以外の場合、バージョンには自動的に現在の日付が設定されます。</p>
-GRPMARKER=<marker_group_code> -MARKER=<marker_code>	省略可	これを指定すると、コードおよびグループ・コードで指定されたマーカーを持つコンポーネントに対してのみシナリオが（再）生成されます。

-GENERATE_PACK=<YES NO>	省略可	パッケージにアタッチされたシナリオを（再）生成するかどうかを指定します。デフォルト値は YES です。
-GENERATE_POP=<YES NO>	省略可	インタフェースにアタッチされたシナリオを（再）生成するかどうかを指定します。デフォルト値は NO です。
-GENERATE_TRT=<YES NO>	省略可	プロシージャにアタッチされたシナリオを（再）生成するかどうかを指定します。デフォルト値は NO です。
-GENERATE_VAR=<YES NO>	省略可	変数にアタッチされたシナリオを（再）生成するかどうかを指定します。デフォルト値は NO です。

例

OdiGenerateAllScen -PROJECT=1003 は、現在リポジトリで ID が 1003 であるプロジェクトに、すべてのシナリオを生成します。

OdilImportObject

使用方法

```
OdiImportObject -FILE_NAME=<FileName> [-
WORK_REP_NAME=<workRepositoryName>] [-IMPORT_MODE=<DUPLICATION |
SYNONYM_INSERT | SYNONYM_UPDATE | SYNONYM_INSERT_UPDATE>]
```

説明

XML エクスポート・ファイルのコンテンツをリポジトリにインポートします。このコマンドは、ユーザー・インタフェースから使用できるインポート機能と同等の動作をします。オブジェクトは、接続しているマスター・リポジトリまたは作業リポジトリ、または接続しているマスター・リポジトリにアタッチされている<workRepositoryName>作業リポジトリにインポートされます。

注意: インポート・モードと、オブジェクトがリポジトリにインポートされる順序は、慎重に指定してください。たとえば、インタフェースを、そのインタフェースで使用されているモデルより先にインポートすると、エラーになることがあります。SYNONYM_INSERT_UPDATE モードでインタフェースをインポートすると、内部識別子が同一のインタフェースまたはサブオブジェクトが消去されることがあります。

注意: この API は、リポジトリに存在しない他のオブジェクトに依存するオブジェクトをインポートする場合に使用すると、正しく機能しないことがあります。この API は、上位レベルのオブジェクト（プロジェクト、モデルなど）をインポートするために使用することをお勧めします。

パラメータ

パラメータ	必須かどうか	説明
-FILE_NAME=<FileName>	必須	インポートする XML エクスポート・ファイルの名前。
-WORK_REP_NAME=<workRepositoryName>	省略可	オブジェクトのインポート先作業リポジトリの名前。この作業リポジトリは、接続しているマスター・リポジトリで定義されている必要があります。このパラメータが指定されない場合、オブジェクトは現在のマスター・リポジトリまたは作業リポジトリにインポートされます。
-IMPORT_MODE=<DUPLICATION SYNONYM INSERT SYNONYM UPDATE SYNONYM_INSERT_UPDATE>	省略可	オブジェクトのインポート・モード。デフォルト値は DUPLICATION です。詳細は、「インポート・モード」を参照してください。

例

DUPLICATION モードを使用して、WORKREP 作業リポジトリに/temp/DW01.xml エクスポート・ファイル（プロジェクト）をインポートします。

```
OdiImportObject -FILE_NAME=/temp/DW01.xml -WORK_REP_NAME=WORKREP -IMPORT_MODE=DUPLICATION
```

OdiImportScen

使用方法

```
OdiImportScen -FILE_NAME=<FileName> [-IMPORT_MODE=<DUPLICATION | SYNONYM_INSERT | SYNONYM_UPDATE | SYNONYM_INSERT_UPDATE>]
```

説明

XML エクスポート・ファイルから現在の作業リポジトリにシナリオを自動的にインポートします。

パラメータ

パラメータ	必須かどうか	説明
-FILE_NAME=<FileName>	必須	エクスポート・ファイルの名前。
-IMPORT_MODE=<DUPLICATION SYNONYM_INSERT SYNONYM_UPDATE SYNONYM_INSERT_UPDATE>	省略可	シナリオのインポート・モード。デフォルト値は DUPLICATION です。インポート・モードの詳細は、「インポート・モー

ド」を参照してください。

ユーティリティ

OdiAnt

使用方法

```
OdiAnt -BUILDFILE=<file> -LOGFILE=<file> [-TARGET=<target>] [-D<property name>=<property value>]* [-PROJECTHELP] [-HELP] [-VERSION] [-QUIET] [-VERBOSE] [-DEBUG] [-EMACS] [-LOGGER=<classname>] [-LISTENER=<classname>] [-FIND=<file>]
```

説明

Apache Software Foundation のビルド・ツール、Ant をコールします。このツールは、アーカイブ、コンパイル、デプロイ、ドキュメント化、EJB などの複数のコマンドの実行、OS プロセスの実行、ファイル、.NET タスク、リモート・タスクなどの操作のために使用できます。

Ant の使用の詳細は、次のオンライン・マニュアルを参照してください。
<http://jakarta.apache.org/ant/manual/index.html>

Ant のライセンス情報

パラメータ

パラメータ	必須かどうか	説明
-BUILDFILE=<file>	必須	Ant ビルド・ファイル。Ant コマンドが含まれている XML ファイル。
-LOGFILE=<file>	必須	指定されたファイルをロギングに使用します。
-TARGET=<target>	省略可	ビルド処理のターゲット。
-D<property name>=<property value>	省略可	プロパティと値のリスト。
-PROJECTHELP	省略可	プロジェクトについてのヘルプを表示します。
-HELP	省略可	Ant のヘルプを表示します。
-VERSION	省略可	Ant のバージョンを表示します。
-QUIET	省略可	非冗長モードで実行します。
-VERBOSE	省略可	冗長モードで実行します。

-DEBUG	省略可	デバッグ情報を印刷します。
-EMACS	省略可	ロギング情報を装飾なしで表示します。
-LOGGER=<classname>	省略可	ロギングを実行する Java クラス。
-LISTENER=<classname>	省略可	クラス・インスタンスをリスナーとして追加します。
-FIND=<file>	省略可	ファイル・システムのルートから Ant Buildfile を検索して、それを使用します。

例

例のリストは、Ant のオンライン・マニュアルを参照してください。

ftp.mycompany.com のディレクトリ /download/public にある *.html ファイルを FTP 経由でディレクトリ C:¥temp¥にダウンロードするには、次の操作を行います。

手順 1: Ant コマンド・ファイルを生成します。

```
OdiOutFile -FILE=c:¥temp¥ant_cmd.xml
<?xml version="1.0"?>
<project name="myproject" default="ftp" basedir="/">
  <target name="ftp">
    <ftp action="get" remotedir="/download/public"
server="ftp.mycompany.com" userid="anonymous"
password="me@mycompany.com">
      <fileset dir="c:¥temp">
        <include name="**/*.html"/>
      </fileset>
    </ftp>
  </target>
</project>
```

手順 2: Ant コマンド・ファイルを実行します。

```
OdiAnt -BUILDFILE=c:¥temp¥ant_cmd.xml -LOGFILE=c:¥temp¥ant_cmd.log
```

OdiBeep

使用方法

```
OdiBeep [-FILE=<sound_file>]
```

説明

エージェントをホスティングしているマシン上のデフォルトのビーブ音またはサウンド・ファイルを再生します。次のファイル形式がデフォルトでサポートされています。

- WAV
- AIF
- AU

注意: 他のファイル形式を再生するには、アプリケーション・クラスパスに、適切な *JavaSound Service Provider Interface* を追加する必要があります。詳細は、<http://java.sun.com> を参照してください。

パラメータ

パラメータ	必須かどうか	説明
-FILE	省略可	再生するサウンド・ファイルのパスとファイル名。指定しない場合、マシンのデフォルト・ビーブ音が使用されます。

例

サウンド・ファイル `c:\wav\alert.wav` を再生します。

```
OdiBeep -FILE=c:\wav\alert.wav
```

OdiDataQuality

使用方法

```
OdiDataQuality -BATCH_FILE=<batch_file> [-OUT_FILE=<stdout_file>] [-ERR_FILE=<stderr_file>] [-SYNCHRONOUS=<yes|no>]
```

説明

データ品質バッチ・ファイルを実行するシステム・コマンド・シェルを呼び出し、データ品質プロジェクトを起動します。

パラメータ

パラメータ	必須かどうか	説明
- BATCH FILE=<batch file>	必須	<p>実行するデータ品質バッチ・ファイルの場所。ファイル名は絶対パスで指定する必要があります。このバッチ・ファイルはオペレーティング・システムに依存するもので、runprojectN と呼ばれます。</p> <p>例: C:\oracle\oracledq\metabase_data\metabase\oracledq\project2\scripts\runproject2.cmd</p>
- OUT FILE=<stdout_file>	省略可	標準出力のリダイレクト先となるファイル。リダイレクトしない場合は空白のままとし、ファイル名は絶対パスで指定します。
- ERR FILE=<stderr_file>	省略可	標準エラーのリダイレクト先となるファイル。リダイレクトしない場合は空白のままとし、ファイル名は絶対パスで指定します。
- SYNCHRONOUS=<yes no>	省略可	YES に設定した場合、このツールは品質処理が完了するまで待機し、その後で終了します（この場合、エラー・コードが返される可能性があります）。NO に設定した場合、このツールは品質処理の完了を待たずにただちに終了します（エラーを返すことはありません）。

例

次のコマンドは、C:\oracle\oracledq\metabase_data\metabase\oracledq ディレクトリにエクスポートされたデータ品質プロジェクトを実行します。

```
OdiDataQuality "-
BATCH_FILE=C:\oracle\oracledq\metabase_data\metabase\oracledq\project2\sc
ripts\runproject2.cmd " "-OUT_FILE=C:\temp\output file" "-
SYNCHRONOUS=YES"
```

OdiKillAgent

使用方法

```
OdiKillAgent -PORT=<TCP/IP Port>
```

説明

エージェントがリスナー・エージェント、スケジューラ・エージェントまたは Web エージェントとして TCP/IP ポート上でリスニングしている場合、OdiKillAgent コマンドでそれを停止することができます。

重要な注意: セキュリティ上の理由で、エージェントの停止は、エージェント・プロセスが起動されたマシンからのみ可能です。
リモート・エージェントを停止することはできません。

パラメータ

パラメータ	必須かどうか	説明
-PORT=<TCP/IP Port>	省略可	停止するエージェントがリスニングしているポート。このパラメータを指定しない場合、デフォルトのポート 20910 上でリスナーとして実行されているエージェントが停止されます。

例

ポート 29666 上でリスニングしているエージェントを停止します。

```
OdiKillAgent -PORT=29666
```

OdiOSCommand

使用方法

```
OdiOSCommand [-OUT_FILE=<stdout_file>] [-ERR_FILE=<stderr_file>] [-
FILE_APPEND=<yes|no>] [-WORKING_DIR_PARAM=<workingdir>] [-
SYNCHRONOUS_PARAM=<yes|no>]] [CR/LF <command> | -COMMAND=<command>]
```

説明

コマンドを実行するシステム・コマンド・シェルを呼び出し、出力をファイルにリダイレクトします。

次のオペレーティング・システムがサポートされます。

- Windows 95、98、ME (command.com 使用)
- Windows NT、2000、XP (cmd 使用)
- POSIX 準拠の OS (sh 使用)

次のオペレーティング・システムはサポートされていません。

- Mac OS

パラメータ

パラメータ	必須かどうか	説明
-COMMAND <command>	必須	実行するコマンド。コマンド・シェルで適切に解釈されるように、スペースを含む引数は引用符で囲みます。複数行のコマンドでは、 OdiOSCommand 行に続いて、コマンド全体を生のテキストとして -COMMAND パラメータなしで渡します。
-OUT_FILE=<stdout_file>	省略可	標準出力をリダイレクトするファイルの絶対名。
-ERR_FILE=<stderr_file>	省略可	標準エラーをリダイレクトするファイルの絶対名。
-FILE_APPEND=<yes no>	省略可	出力ファイルを上書きしないで、ファイル末尾に追加するかどうか。
-WORKING_DIR_PARAM=<workingdir>	省略可	コマンドが実行されるディレクトリ。
-SYNCHRONOUS_PARAM=<yes no>	省略可	yes の場合、セッションはコマンドの終了を待ちます。 no の場合、セッションは即時にエラー・コード 0 で続行します。デフォルトで、「同期」モードで実行されます。

例

次のコマンドは、ファイル `c:\work\load.bat` (Windows マシンの場合) を実行し、出力ストリームをファイルに追加します。

```
OdiOSCommand "-OUT_FILE=c:\work\load-out.txt" "-ERR_FILE=c:\work\load-err.txt" "-FILE_APPEND=YES" "-WORKING_DIR=c:\work"
```

```
c:\work\load.bat
```

OdiPingAgent

使用方法

```
OdiPingAgent -PHYSICAL_AGENT_NAME=<physical_agent>
```

説明

エージェント上で接続テストを実行します。エージェントが起動していない場合にこのコマンドを使用するとエラーになります。

パラメータ

パラメータ	必須かどうか	説明
- PHYSICAL_AGENT_NAME=<physical_agent>	必須	テストする物理エージェントの名前。 注意: 以前の、AGENT_NAME を使用する構文もまだ有効ですが、廃止予定です。

例

物理エージェント AGENT_SOLARIS_DEV をテストします。

```
OdiPingAgent -PHYSICAL_AGENT_NAME=AGENT_SOLARIS_DEV
```

OdiPurgeLog

使用方法

```
OdiPurgeLog [-FROMDATE=<fromdate>] [-TODATE=<todate>] [-  
CONTEXT_CODE=<context_code>] [-USER_NAME=<user_name>] [-  
AGENT_NAME=<agent_name>] [-PURGE_REPORTS=<0|1>] [-  
SESSION_STATUS=<D|E|M>]
```

説明

実行ログのページを許可します。

-FROMDATE を省略すると、一番古いセッションを始点にしてページが実行されます。

-TODATE を省略すると、現在の日付までページが実行されます。

両方のパラメータを省略すると、ログ全体がページされます。

注意: 実行中、待機中あるいはキューされているセッションはページできません。

パラメータ

パラメータ	必須かどうか	説明
-FROMDATE=<fromdate>	省略可	ページの開始日。形式は「yyyy/MM/dd hh:mm:ss」。
-TODATE=<todate>	省略可	ページの終了日。形式は「yyyy/MM/dd hh:mm:ss」。
-CONTEXT_CODE=<context_code>	省略可	<context_code>で実行されたセッションのみをページします。
-USER_NAME=<user_name>	省略可	<user_name>によって開始されたセッションのみをページします。
-AGENT_NAME=<agent_name>	省略可	<agent_name>によって実行されたセッションのみをページします。
-PURGE_REPORTS=<0 1>	省略可	このパラメータの値が1の場合、シナリオ・レポート（各シナリオの 実行 ノードの下に表示）もページされます。
-SESSION_STATUS=<D E M>	省略可	指定された状態のセッションのみをページします。 <ul style="list-style-type: none"> • D: 完了 • E: エラー • M: 警告 このパラメータを指定しない場合、前述のすべての状態でのセッションがページされます。

例

```
OdiPurgeLog "-FROMDATE=2001/03/25 00:00:00" "-TODATE=2001/08/31 21:59:00"
```

```
OdiPurgeLog "-FROMDATE=2001/03/25 00:00:00"
```

```
OdiPurgeLog "-TODATE=2001/08/31 21:59:00"
```

OdiReinitializeSeq

使用方法

```
OdiReinitializeSeq -SEQ_NAME=<seq_name> -CONTEXT=<context> -STD_POS=<std_pos>
```

説明

このコマンドは、Oracle Data Integrator 順序を再初期化するために使用されます。

パラメータ

パラメータ	必須かどうか	説明
- SEQ_NAME=<seq_name>	必須	再初期化する順序の名前。グローバル順序の場合 GLOBAL、そうでない場合はプロジェクト順序の<プロジェクト・コード>を接頭辞として付ける必要があります。
-CONTEXT=<context>	必須	順序を再初期化する必要があるコンテキスト。
-STD_POS=<std_pos>	必須	順序の再初期化先にする必要がある位置。

例

GLOBAL コンテキストでは、グローバル順序 SEQ_I をゼロにリセットします。

```
OdiReinitializeSeq -SEQ_NAME=GLOBAL.SEQ_I -CONTEXT=GLOBAL -STD_POS=0
```

OdiStartScen

使用方法

```
OdiStartScen -SCEN_NAME=<scenario> -SCEN_VERSION=<version> [-CONTEXT=<context>] [-ODI_USER=<odi user> -ODI_PASS=<odi password>] [-SESSION_NAME=<session_name>] [-LOG_LEVEL=<log_level>] [-AGENT_CODE=<logical_agent_name>] [-SYNC_MODE=<1|2>] [-KEYWORDS=<keywords>] [-<PROJECT_CODE>.<VARIABLE>=<var_value>]*
```

説明

シナリオを開始します。

オプションのパラメータ AGENT_CODE は、現在のエージェント以外のエージェントにこのシナリオを適用する場合に使用します。

パラメータ SYNC_MODE を使用すると、シナリオを同期モードまたは非同期モードで開始することができます。

注意: 開始されるシナリオは、コマンドが投入されるリポジトリにある必要があります。シナリオを本番に使用する場合、このコマンドを使用してコールされるすべてのシナリオを確実に対象にしてください。ソリューションは、この目的でのシナリオのグループ化に有用です。

パラメータ

パラメータ	必須かどうか	説明
-------	--------	----

-SCEN_NAME=<scenario>	必須	開始するシナリオの名前。
-SCEN_VERSION=<version>	必須	開始するシナリオのバージョン。指定したバージョンが-1-1の場合、最新バージョンのシナリオが実行されます。
-CONTEXT=<context>	省略可	実行コンテキストのコード。このパラメータを省略すると、シナリオは、コール元セッションの実行コンテキストで実行されます。
-ODI_USER=<odi user>	省略可	シナリオを実行するために使用される Oracle Data Integrator ユーザー。このユーザーの権限が使用されます。このパラメータを省略すると、シナリオは親セッションを起動したユーザーの権限で実行されます。
-ODI_PASS=<odi password>	省略可	Oracle Data Integrator ユーザーのパスワード。このパスワードはエンコードされます。ユーザーが指定されている場合、このパラメータは必須です。
-SESSION NAME=<session name>	省略可	実行ログに表示されるセッションの名前。
-LOG_LEVEL=<log_level>	省略可	実行ログに維持するトレース・レベル (0~5)。デフォルト値は最大 (5) です。
-AGENT CODE=<logical agent_name>	省略可	このシナリオの実行を担当する論理エージェントの名前。このパラメータを省略すると、現在のエージェントはこのシナリオを実行します。
-SYNC_MODE=<1 2>	省略可	シナリオの同期化モード。 1: 同期モード (デフォルト)。シナリオの実行が終わるまで、コール・セッションは実行されません。 2: 非同期モード。コール・セッションの実行は、コールされたシナリオの復帰に関係なく継続します。
-KEYWORDS=<keywords>	省略可	このセッションにアタッチされたキーワードのリスト。これらのキーワードにより、セッションの識別が容易になります。このリストは、各キーワードのカンマ区切りリストです。
-<PROJECT CODE>.<VARIABLE>=<var_value>	省略可	シナリオの実行時に値が強制される変数のリスト。このリストは「PROJECT.VARIABLE=値」という形式です。

例

本番コンテキスト（同期モード）でシナリオ LOAD_DWH のバージョン 2 を開始します。

```
OdiStartScen -SCEN_NAME=LOAD_DWH -SCEN_VERSION=2 -CONTEXT=CTX_PRODUCTION
```

変数 START_DATE（ローカル）および COMPANY_CODE（グローバル）の値を渡して、エージェント「UNIX Agent」上でシナリオ LOAD_DWH のバージョン 2 を現在のコンテキストの非同期モードで実行します。

```
OdiStartScen -SCEN_NAME=LOAD_DWH -SCEN_VERSION=2 -SYNC_MODE=2 "-
AGENT_CODE=UNIX Agent" -MY_PROJECT.START_DATE=10-APR-2002 -
GLOBAL.COMPANY_CODE=SP4356
```

OdiUpdateAgentSchedule

使用方法

```
OdiUpdateAgentSchedule -AGENT_NAME=<scheduling_agent>
```

説明

エージェントにそのタスクのスケジュールを強制的に再計算させます。これは、トポロジ・マネージャで、手動でエージェントのスケジュールを更新するのと同じです。

パラメータ

パラメータ	必須かどうか	説明
-AGENT_NAME	必須	更新する物理エージェントの名前。

指定されたエージェントと通信できない場合、またはスケジュールしているタイプのエージェントでない場合は、エラーになります。

例

この例では、agt_s1 という名前の物理エージェントがスケジュールを更新します。

```
OdiUpdateAgentSchedule -AGENT_NAME=agt_s1
```

インターネット

OdiInvokeWebService

使用方法

```
OdiInvokeWebService -URL=<url> -PORT=<port> -OPERATION=<operation> [-
REQUEST_FILE=<file>] [-RESPONSE_MODE=<NO_FILE|NEW_FILE|APPEND>] [-
RESPONSE_FILE=<file>] [-RESPONSE_XML_ENCODING=<xmlEncoding>] [-
```



```
RESPONSE_FILE_CHARSET=<javaCharset>] [-HTTP_USER=<user>] [-
HTTP_PASS=<password>] [-TIMEOUT=<timeout>]
[<xmlRequest>]
```

説明

HTTP プロトコルを使用して Web サービスをコールし、返されたレスポンスを XML ファイルに書き込みます。

このツールは、記述ファイル (WSDL) の URL が提供されている Web サービスのポート上で、特定の操作を呼び出します。

この操作が SOAP リクエストを必要とする場合、リクエストは、リクエスト・ファイルで提供されるか、あるいはツール・コール (XMLRequest) に直接書き込まれます。Web サービス・リクエストのレスポンスは、Oracle Data Integrator で使用できる XML ファイルに書かれています。

注意: このツールは、OdiExecuteWebService に代わるものです。

注意: このツールは、HTTP と HTTPS の両方のプロトコルをサポートします。

パラメータ

パラメータ	必須かどうか	説明
-URL=<URL>	必須	Web サービスを記述している Web サービス記述ファイル (WSDL) の URL。
-PORT=<port>	必須	使用する Web サービス・ポートの名前。
-OPERATION=<operation>	必須	呼び出す Web サービス操作の名前。
-REQUEST_FILE=<file>	省略可	SOAP (Simple Object Access Protocol) 形式のリクエスト・メッセージが含まれている XML ファイルの場所。リクエストはツール・コール (<xmlRequest>) に直接書き込むことができます。
-RESPONSE_MODE=<NO_FILE NEW_FILE APPEND>	省略可	レスポンス・ファイルの生成モード。このパラメータは次の値をとります。 <ul style="list-style-type: none"> NO_FILE (デフォルト) : レスポンス・ファイルが生成されません。 NEW_FILE: 新しいレスポンス・ファイルが生成されます。ファイルがすでに存在する場合、上書きされます。 APPEND: レスポンスはファイルに追加されます。ファイルが存在しない場合、新しく作成されます。
-RESPONSE_FILE=<file>	場合による	結果を書き込むファイルの名前。-RESPONSE_MODE が NO_FILE の場合は必須。

- RESPONSE FILE CHARSET= <javaCharset>	場合による	レスポンス・ファイルの文字エンコーディング。 次の表を参照してください。-RESPONSE_MODE が NO_FILE の場合は必須。
- RESPONSE XML ENCODING= <xmlEncoding>	場合による	レスポンス・ファイルの XML 宣言ヘッダーに示 される文字エンコーディング。次の表を参照して ください。-RESPONSE_MODE が NO_FILE の場 合は必須。
-HTTP_USER=<user>	省略可	HTTP サーバー上で認証されるユーザー・アカウ ント。
-HTTP_PASS=<password>	省略可	HTTP ユーザーのパスワード。
-TIMEOUT=<timeout>	省略可	Web サービス・リクエストは、この時間だけリプ ライを待機した後、サーバーがレスポンスを返さ ないと判断して、エラーを生成します。値を指定 しない場合、タイムアウトはありません。
<xmlRequest>	必須	SOAP (Simple Object Access Protocol) 形式のリ クエスト・メッセージ。このメッセージは、 OdiInvokeWebService コールのすぐ次の行で提供 する必要があります。

次の表に、最も一般的な XML/Java 文字エンコーディング方式の一部を示します。より包括的なリストは <http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html> にあります。

XML Charset	Java Charset
US-ASCII	ASCII
UTF-8	UTF8
UTF-16	UTF-16
ISO-8859-1	ISO8859_1

例

次の Web サービス・コールは、指定された国 (ISO 国別コードはリクエスト内で送信されます) の首都を返します。リクエストおよびレスポンスの形式と、使用できるポートおよび操作が、URL パラメータで渡された WSDL で定義されていることに注意してください。

```
OdiInvokeWebService -
URL=http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.w
so?WSDL -PORT_TYPE=CountryInfoServiceSoapType -OPERATION=CapitalCity -
RESPONSE_MODE=NEW_FILE -RESPONSE_XML ENCODING=ISO-8859-1 "-
RESPONSE_FILE=/temp/result.xml" -RESPONSE_FILE_CHARSET=ISO8859_1
<CapitalCityRequest>
<sCountryISOCODE>US</sCountryISOCODE>
</CapitalCityRequest>
```

生成される /temp/result.xml ファイルの内容は次のようになります。

```
<CapitalCityResponse>
  <m:CapitalCityResponse>
    <m:CapitalCityResult>Washington</m:CapitalCityResult>
  </m:CapitalCityResponse>
</CapitalCityResponse>
```

パッケージ

Oracle Data Integrator には、パッケージ内で OdiInvokeWebService をコールするための特別なグラフィカル・インタフェースが用意されています。詳細は、「Web サービスの使用法」を参照してください。

OdiReadMail

使用方法

```
OdiReadMail -MAILHOST=<mail_host> -USER=<mail_user> -
PASS=<mail_user_password> -FOLDER=<folder_path> [-PROTOCOL=<pop3|imap>]
[-FOLDER_OPT=<none|sender|subject>] [-KEEP=<no|yes>] [-
EXTRACT_MSG=<yes|no>] [-EXTRACT_ATT=<yes|no>] [-MSG_PRF=<my_prefix>] [-
ATT_PRF=<my_prefix>] [-USE_UCASE=<no|yes>] [-NOMAIL_ERROR=<no|yes>] [-
TIMEOUT=<timeout>] [-POLLINT=<pollint>] [-MAX_MSG=<max_msg>] [-
SUBJECT=<subject_filter>] [-SENDER=<sender_filter>] [-TO=<to_filter>] [-
CC=<cc_filter>] [-BCC=<bcc_filter>]
```

説明

電子メールとその添付ファイルを POP または IMAP インターネット・メール・サービスから読み取るために使用できます。

このコマンドを使用すると、USER と PASS で指定された接続パラメータを使用して、サーバー MAILHOST 上でホスティングされているメールに接続できます。その後、実行エージェントは、MAX_MSG 件のメッセージを受信するか、TIMEOUT で指定された最大待機時間に達するまで待機します。抽出されたメッセージは、パラメータ SUBJECT および SENDER で指定されるようなフィルタに合致する必要があります。メッセージが基準に合致した場合、メッセージの内容と添付ファイルが、パラメータ FOLDER によって指定されたディレクトリに抽出されます。パラメータ KEEP が「no」に設定されている場合、取得したメッセージは、その後メールボックスから削除されます。

パラメータ

パラメータ	必須かどうか	説明
-MAILHOST=<mail_host>	必須	POP または IMAP メール・サーバーの IP アドレス。

-USER=<mail_user>	必須	有効なメール・サーバーのアカウント。
-PASS=<mail_user_password>	必須	メール・サーバー・アカウントのパスワード。
-FOLDER=<folder_path>	必須	添付ファイルおよびメッセージ用ストレージ・フォルダのフルパス。
-PROTOCOL=<pop3 imap>	省略可	メール・アクセスのタイプ (POP3 または IMAP) 。デフォルトは POP3 です。
- FOLDER OPT=<none sender subject>	省略可	<p>次のパラメータに応じてディレクトリ FOLDER にサブディレクトリを作成します。</p> <ul style="list-style-type: none"> • none (デフォルト) : アクションなし。 • sender: 送信者の外部名でサブディレクトリが作成されます。 • subject: メッセージの件名でサブディレクトリが作成されます。 <p>注意: sender および subject フォルダ・オプションでは、スペースと、英数字以外の文字 (@など) は、生成されたフォルダ名ではアンダースコアに置き換えられます。</p>
-KEEP=<no yes>	省略可	<p>yes: フィルタに合致するメッセージを、読んだ後でメールボックスに保存します。</p> <p>no (デフォルト) : フィルタに合致するメッセージを、読んだ後でメールボックスから削除します。</p>
-EXTRACT_MSG=<yes no>	省略可	<p>yes (デフォルト) : メッセージの本文をファイルに抽出します。</p> <p>no: メッセージの本文をファイルに抽出しません。</p>
-EXTRACT_ATT=<yes no>	省略可	<p>はい (デフォルト) : 添付ファイルをファイルに抽出します。</p> <p>no: 添付ファイルを抽出しません。</p>
-MSG_PRF=<my_prefix>	省略可	メッセージ本文を含むファイルの接頭辞。デフォルトは MSG です。
-ATT_PRF=<my_prefix>	省略可	添付ファイルを含むファイルの接頭辞。元のファイル名は維持されます。
-USE_UCASE=<no yes>	省略可	<p>yes: ファイル名を強制的に大文字にします。</p> <p>no (デフォルト) : 大文字と小文字を元のままにします。</p>

-NOMAIL_ERROR=<no yes>	省略可	yes: メールが指定された基準に合致しない場合、エラーを生成します。 no (デフォルト): 指定された基準に対応するメールがない場合、エラーを生成しません。
-TIMEOUT=<timeout>	省略可	最大待機時間 (単位はミリ秒。デフォルトは 0)。 この待機時間に到達すると、コマンドは終了します。 値 0 (デフォルト) は、待機時間が無制限であることを示します (パラメータ MAX_MSG で指定された最大受信メッセージ数に達するまで)。
-POLLINT=<pollint>	省略可	新しいメッセージをスキャンする検索間隔 (単位はミリ秒)。デフォルトは 1000 です (1 秒)。
-MAX_MSG=<max_msg>	省略可	抽出するメッセージ数の最大値 (デフォルトは 1)。この数に到達すると、コマンドは終了します。
-SUBJECT=<subject_filter>	省略可	件名に応じてメッセージをフィルタするために使用されるパラメータ。
-SENDER=<sender_filter>	省略可	送信者に応じてメッセージをフィルタするために使用されるパラメータ。
-TO=<to_filter>	省略可	アドレスに応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。
-CC=<cc_filter>	省略可	CC に応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。
-BCC=<bcc_filter>	省略可	BCC に応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。

例

support のメールを自動受信します。添付ファイルは切り離してエージェントのマシン上のフォルダ C:\support に格納します。最大待機時間を 10 秒として、すべてのメッセージを待機します。

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass -
KEEP=no -FOLDER=c:¥support -TIMEOUT=10000 -MAX_MSG=0 -
SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TEXT -
EXTRACT_ATT=yes
```

1分に1度、新しいメッセージをチェックして、メッセージが10件になるまで時間無制限で待機します。

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass -
KEEP=no -FOLDER=c:¥support -TIMEOUT=0 -MAX_MSG=10 -POLLINT=60000 -
SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TEXT -
EXTRACT_ATT=yes
```

OdiSendMail

使用方法

```
OdiSendMail -MAILHOST=<mail_host> -FROM=<from_user> -TO=<address_list>
[-CC=<address_list>] [-BCC=<address_list>] [-SUBJECT=<subject>] [-
ATTACH=<file_path>]* [-MSGBODY=<message_body> | CR/LF<message_body>]
```

説明

SMTP サーバーを使用して電子メールを送信することができます。

パラメータ

パラメータ	必須かどうか	説明
-MAILHOST=<mail_host>	必須	SMTP サーバーの IP アドレス。
-From=<from_user>	必須	メッセージの送信者のアドレス。 例: support@mycompany.com 注意: 送信者の外部名を送信するために、次の表記法を使用できます。 "-FROM=サポート・センター <support@mycompany.com>"
-TO=<address_list>	必須	受信者の電子メール・アドレスをコンマで区切ったリスト。 例: "-TO=sales@mycompany.com, support@mycompany.com"
-CC=<address_list>	省略可	CC された受信者の電子メール・アドレスをコンマで区切ったリスト。 例:

-BCC=<address_list>	省略可	"-CC=info@mycompany.com" BCC された受信者の電子メール・アドレスをコンマで区切ったリスト。 例: "-BCC=big_boss@mycompany.com"
-SUBJECT=<subject>	省略可	メッセージの目的 (件名)。
-ATTACH=<file_path>	省略可	メッセージに添付するファイルの、実行エージェントに対する相対パス。複数のファイルを添付するには、「-ATTACH=...」を繰り返す必要があります。 例: ファイル profile および cshrc をメールに添付する場合は次のようになります。 -ATTACH=/home/usr/.profile - ATTACH=/home/usr/.cshrc
CR/LF <message_body> または- MSGBODY=<message_body>	省略可	メッセージ本文 (テキスト)。このテキストは、OdiSendMail コマンドの次の行に入力することもできますし (改行記号 CR/LF はメール本文の始まりを示します) -MSGBODY パラメータを使用して定義することもできます。この Oracle Data Integrator コマンドを OS のコマンドラインからコールする場合は、-MSGBODY パラメータを使用してください。

例

管理者に、プロセスが完了し、ファイル job.log と job.bad を添付したことを示すメッセージを送信します。

```
OdiSendMail -MAILHOST=mail.mymail.com "-FROM=Application Oracle Data Integrator<odi@mymail.com>" -TO=admin@mymail.com "-SUBJECT=Execution OK" -ATTACH=C:¥log¥job.log -ATTACH=C:¥log¥job.bad
```

```
Hello Administrator !
```

```
Your process finished successfully.Attached are your files.
```

```
Have a nice day!
```

```
Oracle Data Integrator.
```

ファイル

OdiFileAppend

使用方法

```
OdiFileAppend -FILE=<file> -TOFILE=<dest_file> [-OVERWRITE=<yes|no>] [-CASESENS=<yes|no>] [-HEADER=<n>] [-KEEP_FIRST_HEADER=<yes|no>]
```

説明

複数のファイルを単一ファイルに連結します。

HEADER パラメータは、入力ファイルの最初の n 行を表示しないようにするために使用されます (csv ファイルのヘッダーなど)。KEEP_FIRST_HEADER パラメータは、結果ファイル内の最初のヘッダー行を維持するか、削除するかを指定するために使用されます。

HEADER パラメータを省略すると、連結はバイナリです (ファイルをテキストとして読み取りません)。

パラメータ

パラメータ	必須かどうか	説明
-FILE=<file>	必須	連結するファイルのフルパス。任意の文字を指定するには「*」を使用します。 例: /var/tmp/*.log (フォルダ/var/tmp 内の、拡張子が log のすべてのファイル) arch_*.lst (arch_で始まり、拡張子が lst のすべてのファイル)
-TOFILE=<dest_file>	必須	ターゲット・ファイル。
-OVERWRITE=<yes no>	省略可	ターゲット・ファイルが存在する場合に上書きするかどうかを示します。デフォルトでは、値は no に設定されます。
-CASESENS=<yes no>	省略可	ファイル検索時に、Oracle Data Integrator が大文字と小文字を区別するかどうかを示します。デフォルトでは、ファイルは大文字で検索されます。
-HEADER=<n>	省略可	連結前にソース・ファイルから削除されるヘッダー行の行数。デフォルトでは、行は削除されません。
-KEEP_FIRST_HEADER=<yes no>	省略可	連結時に最初のファイルのヘッダー行を維持するために使用されます。

例

フォルダ/var/tmp のファイル*.log をファイル/home/all_files.log に連結します。

```
OdiFileAppend -FILE=/var/tmp/*.log -TOFILE=/home/all_files.log
```

各店の日次売上のファイルを、最初のファイルのヘッダーを維持して連結します。

```
OdiFileAppend -FILE=/data/store/sales_*.dat -TOFILE=/data/all_stores.dat  
-OVERWRITE=yes -HEADER=1 -KEEP_FIRST_HEADER=yes
```


OdiFileCopy

使用方法

```
OdiFileCopy -DIR=<dir> -TODIR=<dest_dir> [-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

```
OdiFileCopy -FILE=<file> -TOFILE=<dest_file>|-TODIR=<dest_dir> [-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

説明

ファイルまたはフォルダを、実行エージェントのマシン上のファイルまたはフォルダにコピーします。

パラメータ

パラメータ	必須かどうか	説明
-DIR=<dir>	-FILE 省略時は必須	コピーするディレクトリ（またはフォルダ）
-FILE=<file>	-DIR 省略時は必須	コピーするファイルのフルパス。任意の文字を指定するには「*」を使用します。 例: /var/tmp/*.log（フォルダ/var/tmp 内の、拡張子が log のすべてのファイル） arch_*.lst（arch_で始まり、拡張子が lst のすべてのファイル）
-TODIR=<dest_dir>	-DIR 指定時は必須	コピーのターゲット・ディレクトリ。 ディレクトリをコピーする場合（-DIR）、このパラメータは、コピーされるディレクトリの名前を示します。 ファイルをコピーする場合（-FILE）、このパラメータはコピー先ディレクトリを示します。
-TOFILE=<dest_file>	-TODIR 省略時は必須	コピー先ファイル。このパラメータは、パラメータ-DIR と併用できません。 このパラメータには次の内容が含まれます。 <ul style="list-style-type: none"> コピーするファイルが1つのみ場合はコピー先ファイルの名前（任意の文字は使用不可）。 複数のファイルをコピーする場合は、コピー先ファイルの新しい名前のマスク。 <p>注意: TODIR と TOFILE は同時に使用できないパラメータです。両方指定すると、TOFILE は無視され、TODIR</p>

のみが考慮されます。

- OVERWRITE=<yes <u>no</u> >	省略可	フォルダのファイルがすでに存在する場合、上書きするかどうかを示します。デフォルトでは、値は no に設定されます。
-RECURSE=< <u>yes</u> no>	省略可	他のフォルダを含むフォルダの場合、コピーを再帰的に実行するかどうかを示します。値が no の場合、フォルダ内のファイルのみがコピーされ、サブフォルダはコピーされません。
-CASESENS=<yes <u>no</u> >	省略可	ファイル検索時に、Oracle Data Integrator が大文字と小文字を区別するかどうかを示します。デフォルトでは、大文字のファイルが検索されます。

例

ディレクトリ/etc からディレクトリ/home にファイル「host」をコピーします。

```
OdiFileCopy -FILE=/etc/hosts -TOFILE=/home/hosts
```

ディレクトリ/etc からディレクトリ/home に、すべての*.csv ファイルをコピーし、上書きします。

```
OdiFileCopy -FILE=/etc/*.csv -TODIR=/home -OVERWRITE=yes
```

ディレクトリ/etc からディレクトリ/home に、すべての*.csv ファイルをコピーし、拡張子を.txt に変更します。

```
OdiFileCopy -FILE=/etc/*.csv -TOFILE=/home/*.txt -OVERWRITE=yes
```

ディレクトリ C:\odi とそのサブディレクトリを、ディレクトリ C:\Program Files\odi にコピーします。

```
OdiFileCopy -DIR=C:\odi "-TODIR=C:\Program Files\odi" -RECURSE=yes
```

OdiFileDelete

使用方法

```
OdiFileDelete -DIR=<dir> | -FILE=<file> [-RECURSE=<yes | no>] [-CASESENS=<yes | no>] [-NOFILE_ERROR=<yes | no>]
```

説明

エージェントを実行しているマシン上のファイルまたはディレクトリを削除します。

パラメータ

パラメータ	必須かどうか	説明
-DIR=<dir>	-FILE 省略時は必須	<ol style="list-style-type: none"> 1. -FILE 省略時は、削除するディレクトリ（フォルダ）の名前を指定します。 2. -FILE を指定した場合は、ファイルを削除するパスを指定します。
-FILE=<file>	-DIR 省略時は必須	<p>削除するファイルの名前またはマスク。-DIR を指定しない場合は、フルパスを指定してください。ワイルドカード文字を指定するには「*」を使用します。</p> <p>例:</p> <p>/var/tmp/*.log（フォルダ/var/tmp 内のすべての.log ファイル）</p> <p>/bin/arch_*.lst（arch_で始まるすべての.list ファイル）</p>
-RECURSE=<yes no>	省略可	<ol style="list-style-type: none"> 1. -FILE を省略した場合は無効です。すべてのサブディレクトリが無警告で削除されます。 2. -FILE を指定した場合は、このディレクトリとすべてのサブディレクトリからファイルを削除するかどうかを指定します。
-CASESENS=<yes no>	省略可	Oracle Data Integrator がファイル名を照合する際に、大文字と小文字を区別するかどうかを指定します。
-NOFILE_ERROR=<yes no>	省略可	指定されたディレクトリまたはファイルがない場合、エラーを生成するかどうかを指定します。デフォルトでは、エラーを生成しません。

注意: -DIR パラメータと-FILE を組み合わせてファイルとディレクトリを同時に削除することはできません。その場合は、OdiFileDelete を 2 回コールしてください。

例

ディレクトリ c:\data\input からファイル my_data.dat を削除します。ファイルがない場合はエラーを生成します。

```
OdiFileDelete -FILE=c:\data\input\my_data.dat -NOFILE_ERROR=yes
```

すべての.txt ファイルを bin ディレクトリから削除するが、.TXT ファイルは残す場合:

```
OdiFileDelete "-FILE=c:\Program Files\odi\bin\*.txt" -CASESENS=yes
```

この文でも同じ結果になります。

```
OdiFileDelete "-DIR=c:\Program Files\odi\bin" "-FILE=*.txt" -CASESENS=yes
```

ディレクトリ/bin/usr/nothingToDoHere を削除する場合:

```
OdiFileDelete "-DIR=/bin/usr/nothingToDoHere"
```

OdiFileMove

使用方法

```
OdiFileMove -FILE=<file> -TODIR=<dest_dir>|-TOFILE=<dest_file> [-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

```
OdiFileMove -DIR=<dir> -TODIR=<dest_dir> [-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>]
```

説明

ファイルまたはディレクトリを、実行エージェントのマシン上のファイルまたはディレクトリに移動します。

このコマンドは、ファイルまたはディレクトリの名前を変更するために使用することもできます。

パラメータ

パラメータ	必須かどうか	説明
-DIR=<dir>	-FILE 省略時は必須	移動または名前を変更するディレクトリ（またはフォルダ）。
-FILE=<file>	-DIR 省略時は必須	移動または名前を変更するファイルのフルパス。任意の文字を指定するには「*」を使用します。 例: /var/tmp/*.log（ディレクトリ/var/tmp 内の、拡張子が log のすべてのファイル） arch_*.lst（arch_で始まり、拡張子が lst のすべてのファイル）
-TODIR=<dest_dir>	-DIR 指定時は必須	移動のターゲット・ディレクトリ。 ディレクトリを移動する場合（-DIR）、このパラメータは、ディレクトリの新しい名前を示します。 1 つまたは複数のファイルを移動する場合（-FILE）、このパラメータは移動先ディレクトリを示します。
-TOFILE=<dest_file>	-TODIR 省略時は必須	移動先ファイル。このパラメータは、パラメータ-DIR と併用できません。 このパラメータは次のいずれかです。 - 単一ファイルを移動する場合は、ターゲット・ファイルの新しい名前（任意の文字なし）。

		- 複数のファイルを移動する場合は、新しいファイル名のマスク。
- OVERWRITE=<yes <u>no</u> >	省略可	ファイルまたはディレクトリが存在する場合、それを上書きするかどうかを示します。デフォルトでは、値は no です。
- RECURSE=< <u>yes</u> no>	省略可	ディレクトリに別のディレクトリが含まれている場合に、ファイルを再帰的に移動するかどうかを示します。値が no の場合、移動するディレクトリに含まれるファイルのみが移動され、サブフォルダは移動されません。
- CASESENS=<yes <u>no</u> >	省略可	ファイル検索時に、Oracle Data Integrator が大文字と小文字を区別するかどうかを示します。デフォルトでは、大文字のファイルが検索されます。

例

host ファイルの名前を hosts.old に変更します。

```
OdiFileMove -FILE=/etc/hosts -TOFILE=/etc/hosts.old
```

ディレクトリ/etc からディレクトリ/home にファイル「hosts」を移動します。

```
OdiFileMove -FILE=/etc/hosts -TOFILE=/home/odi/hosts
```

ディレクトリ/etc からディレクトリ/home/odi に、すべての*.csv ファイルを移動し、上書きします。

```
OdiFileMove -FILE=/etc/*.csv -TODIR=/home/odi -OVERWRITE=yes
```

ディレクトリ/etc からディレクトリ/home/odi に、すべての*.csv ファイルを移動し、拡張子を.txt に変更します。

```
OdiFileMove -FILE=/etc/*.csv -TOFILE=/home/odi/*.txt -OVERWRITE=yes
```

ディレクトリ C:\odi を C:\odi_is_wonderful に変更します。

```
OdiFileMove -DIR=C:\odi -TODIR=C:\odi_is_wonderful
```

ディレクトリ C:\odi とそのサブフォルダを、ディレクトリ C:\Program Files\odi に移動します。

```
OdiFileMove -DIR=C:\odi "-TODIR=C:\Program Files\odi" -RECURSE=yes
```

OdiFileWait

使用方法

```
OdiFileWait -DIR=<dir> -PATTERN=<pattern> [-ACTION=<DELETE|COPY|MOVE|APPEND|ZIP|NONE>] -TODIR=<dest_dir> -TOFILE=<dest_file> [-OVERWRITE=<yes|no>] [-CASESENS=<yes|no>] [-FILECOUNT=<n>] [-TIMEOUT=<n>] [-POLLINT=<n>] [-HEADER=<n>] [-KEEP_FIRST_HEADER=<yes|no>] [-NOFILE_ERROR=<yes|no>]
```

説明

ディレクトリ DIR をスキャンして、PATTERN で指定されたマスクに合致するファイルを待機します。指定したファイルが見つかったと、パラメータ ACTION で示されたアクションがトリガーされます。

TIMEOUT で指定されたタイムアウト制限に達するまで、実行エージェントはスタンバイ・モードで FILECOUNT 個のファイルの受信を待機します。ディレクトリは POLLINT ミリ秒間隔でスキャンされます。

パラメータ

パラメータ	必須かどうか	説明
-ACTION= <DELETE COPY MOVE APPEND ZIP <u>NONE</u> >	省略可	見つけたファイルに対するアクション。 DELETE: 見つけたファイルを削除します。 COPY: 見つけたファイルをディレクトリ TODIR にコピーします。 MOVE: 見つけたファイルをフォルダ TODIR に移動するか、TOFILE で指定された名前に変更します。 APPEND: 見つけたすべてのファイルを連結し、結果ファイル TOFILE を作成します。ソース・ファイルは削除されます。 ZIP: 見つけたファイルを zip し、zip ファイル TOFILE に格納します。 NONE: アクションを実行しません。これがデフォルトの動作です。
-DIR=<dir>	必須	スキャンするディレクトリ（またはフォルダ）。
-PATTERN=<pattern>	必須	スキャンするファイル名のマスク。任意の文字を指定するには「*」を使用します。 例: *.log（拡張子 log を持つすべてのファイル） arch_*.lst（arch_ で始まり、拡張子が lst のすべてのファイル）
-TODIR=<dest_dir>	省略可	アクションのターゲット・ディレクトリ。アクションにより異なります。 COPY: ファイルがコピーされるディレクトリ MOVE: ファイルが移動されるディレクトリ

-TOFILE=<dest_file>	省略可	宛先ファイル。アクションにより異なります。 MOVE: 移動されたファイルの名前変更マスク。 APPEND: 連結の結果ファイルの名前。 ZIP: ZIP の結果ファイルの名前。
-OVERWRITE=<yes no>	省略可	宛先ファイルが存在する場合、それを上書きするかどうかを示します。デフォルトでは、値は no に設定されます。 このオプションを APPEND とともに使用する場合、宛先ファイルの内容が、最後に処理されたファイルの内容になることに注意してください。
-CASESENS=<yes no>	省略可	ファイル検索で大文字と小文字が区別されるかどうかを示します。デフォルトでは、検索されるファイルは大文字です。
-FILECOUNT=<n>	省略可	待機するファイル数の最大値（デフォルトは 0）。この数に到達すると、コマンドは終了します。 値に 0 を指定すると、タイムアウトするまで、すべてのファイルを待機します。 このパラメータが 0 で、タイムアウトも 0 の場合、このパラメータは警告なく強制的に 1 にされます。
-TIMEOUT=<n>	省略可	最大待機時間（単位はミリ秒。デフォルトは 60000）。 この時間に達すると、パラメータ FILECOUNT の値にかかわらず、コマンドは次のコマンドに制御を渡します。 値 0 は、待機時間が無制限であることを示します（パラメータ FILECOUNT で指定されたメッセージ最大読取り数まで待機します）。
-POLLINT=<n>	省略可	新規ファイルを検索する間隔（単位はミリ秒）。デフォルトは 2000 に設定されます（2 秒）。これは、Oracle Data Integrator が 2 秒間隔で新しいメッセージを検索することを意味します。OdiFileWait 中に作成されたファイルは、この間隔の間、ファイルを閉じた後で対象になります（ファイル・サイズ不変）。

-HEADER=<n>	省略可	このパラメータは APPEND アクションでのみ有効です。 連結前にソース・ファイルから削除されるヘッダー行の行数。デフォルトは処理なしです。
-KEEP_FIRST_HEADER=<yes no>	省略可	このパラメータは APPEND アクションでのみ有効です。 連結時に最初のファイルのヘッダー行を維持するために使用されます。
-NOFILE_ERROR=<yes no>	省略可	ファイルが見つからない場合の動作を示します。 値 no (デフォルト) は、ファイルが見つからなくてもエラーを生成しないことを示します。

例

ディレクトリ c:\events のファイル flag.txt を時間制限なしで待機し、ファイルが検出されたら先に進みます。

```
OdiFileWait -ACTION=NONE -DIR=c:\events -PATTERN=flag.txt -FILECOUNT=1 -TIMEOUT=0 -POLLINT=1000
```

ディレクトリ c:\events のファイル flag.txt を時間制限なしで待機し、ファイルが検出されたら削除します。

```
OdiFileWait -ACTION=DELETE -DIR=c:\events -PATTERN=flag.txt -FILECOUNT=1 -TIMEOUT=0 -POLLINT=1000
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c:\sales_in を 1 秒間隔でスキャンし、ディレクトリ C:\sales_ok にあるファイル sales.dat に連結します。最初のファイルのヘッダーを維持します。

```
OdiFileWait -ACTION=APPEND -DIR=c:\sales_in -PATTERN=*.dat -TOFILE=c:\sales_ok\sales.dat -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -HEADER=1 -KEEP_FIRST_HEADER=yes -OVERWRITE=yes
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c:\sales_in を 1 秒間隔でスキャンし、ディレクトリ C:\sales_ok にファイルをコピーします。上書きはしません。

```
OdiFileWait -ACTION=COPY -DIR=c:\sales_in -PATTERN=*.dat -TODIR=c:\sales_ok -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -OVERWRITE=no
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c:\sales_in を 1 秒間隔でスキャンし、ファイルを zip ファイルにアーカイブします。


```
OdiFileWait -ACTION=ZIP -DIR=c:¥sales_in -PATTERN=*.dat -
TOFILE=c:¥sales_ok¥sales.zip -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -
OVERWRITE=yes
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c: ¥sales_in を 1 秒間隔でスキャンし、ディレクトリ C: ¥sales_ok にファイルを移動します。上書きはしません。ファイル名に.bak を追加します。

```
OdiFileWait -ACTION=MOVE -DIR=c:¥sales_in -PATTERN=*.dat -
TODIR=c:¥sales_ok -TOFILE=*.bak -FILECOUNT=0 -TIMEOUT=350000 -
POLLINT=1000 -OVERWRITE=no
```

OdiMkdir

使用方法

```
OdiMkdir -DIR=<dir>
```

説明

ディレクトリを作成します。

親ディレクトリが存在しない場合、このコマンドは再帰的に親ディレクトリを作成します。

パラメータ

パラメータ	必須かどうか	説明
-DIR=<dir>	必須	作成するディレクトリ（またはフォルダ）。

例

C: ¥Program Files 内にディレクトリ odi を作成します。

```
OdiMkdir "-DIR=C:¥Program Files¥odi"
```

OdiOutFile

使用方法

```
OdiOutFile -FILE=<file_name> [-APPEND] [-CHARSET_ENCODING=<encoding>] [-
XROW_SEP=<hexadecimal_line_break>] [CR/LF <text> | -TEXT=<text>]
```

説明

実行エージェントが実行されているマシン上にテキスト・ファイルを生成します。

ファイルは<file_name>で定義されたパスに格納され、<Text>で指定されたテキストを含みます。オプションのパラメータ-APPEND を指定すると、ファイルにすでにデータが含まれている場

合、ファイル末尾に<Text>を追加します。パラメータ-FILE と-APPEND は同一行に指定する必要があります。

パラメータ

パラメータ	必須かどうか	説明
-FILE=<file_name>	必須	ターゲット・ファイル。パスは、絶対パスでも、実行エージェントの場所に対する相対パスでもかまいません。
-APPEND	省略可	<Text>がファイル末尾に追加されるかどうかを示します。このパラメータが指定されない場合、存在するファイルは上書きされます。
-CHARSET ENCODING=<encoding>	省略可	ターゲット・ファイルのエンコーディング。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-XROW SEP=<hexadecimal line_break>	省略可	行セパレータとして使用する文字（改行）の 16 進コード。0A（UNIX の改行）をデフォルトとします。Windows の改行の場合、値を 0D0A にします。
CR/LF <text>または-TEXT=<text>	必須	ファイルに書き込むテキスト。このテキストは、OdiOutFile コマンドの次の行に入力することもできますし（改行記号 CR/LF はテキストの始まりを示します）-TEXT パラメータを使用して定義することもできます。この Oracle Data Integrator コマンドを OS のコマンドラインからコールする場合は、-TEXT パラメータを使用してください。テキストは、変数または代入メソッドを含むことができます。

例

次のコマンドは、ファイル/var/tmp/my_file.txt を、それを実行するエージェントの UNIX マシン上に生成します。

```
OdiOutFile -FILE=/var/tmp/my_file.txt
Welcome to Oracle Data Integrator
This file has been overwritten by <%=odiRef.getSession("SESS_NAME")%>
```

次のコマンドはエントリ PLUTON を、それを実行するエージェントの NT マシンのファイル hosts に追加します。

```
OdiOutFile -FILE=C:\%winnt%\system32\drivers\etc\hosts -APPEND
195.10.10.6    PLUTON    pluton
```

OdiSqlUnload

使用方法

```
OdiSqlUnload -FILE=<file_name> -DRIVER=<driver> -URL=<url> -USER=<user>
-PASS=<password> [-FILE_FORMAT=<file_format>] [-FIELD_SEP=<field_sep> |
-XFIELD_SEP=<field_sep>] [-ROW_SEP=<row_sep> | -XROW_SEP=<row_sep>] [-
DATE_FORMAT=<date_format>] [-ABS=<yes|no>] [-
CHARSET_ENCODING=<encoding>] [-XML_CHARSET_ENCODING=<encoding>] [-
FETCH_SIZE=<array_fetch_size>] [CR/LF <sql_query> | -QUERY=<sql_query> |
-QUERY_FILE=<sql_query_file> ]
```

説明

接続パラメータが<driver>、<url>、<user>および<encoded_pass>によって与えられるデータ・サーバー上 SQL 問合せ<sql_query>を実行することによって、データ・ファイルを生成します。ファイルは<file_name>によって定義されたパス（エージェントに対する相対パス）に書き込まれます。

パラメータ

パラメータ	必須かどうか	説明
-FILE=<file_name>	必須	実行エージェントに相対的な、出力ファイルへのフルパス。
-DRIVER=<driver>	必須	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。
-CHARSET_ENCODING=<encoding>	省略可	ターゲット・ファイルのエンコーディング。デフォルト値は ISO8859-1 です。サポートされているエンコーディングの完全なリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-XML_CHARSET_ENCODING=<encoding>	省略可	XML ファイル内に指定されるエンコーディング。<?xml version="1.0" encoding="ISO-8859-1"?>のようなタグで指定されます。デフォルト値は ISO8859-1 です。サポートされているエンコーディングのリストは、URL http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html にあります。
-URL=<URL>	必須	データ・サーバーへの JDBC URL。

-USER=<user>	必須	SQL 問合せを実行するために使用されるデータ・サーバー上のユーザーのログイン。
-PASS=<password>	必須	<p>データ・サーバーへのログインのための暗号化されたパスワード。このパスワードは、次のシステム・コマンドを使用して暗号化できます。</p> <pre>agent(.bat or .sh) encode <clear_text_password></pre> <p>注意: エージェント (.bat または.sh) は、Oracle Data Integrator インストール・ディレクトリの /bin サブディレクトリにあります。</p> <p>ファイル形式に、次の3つのうちいずれかの値を指定します。</p> <ul style="list-style-type: none"> • fixed : 固定サイズ記録 • variable: 可変サイズ記録 • xml: XML ファイル <p><file_format>を指定しない場合、形式のデフォルトは variable です。</p> <p><file_format>が xml の場合、生成された XML ノードは次の構造を持ちます。</p> <pre><TABLE> <ROW> <column_name>![CDATA[VALUE]]</column_name> <column_name>![CDATA[VALUE]]</column_name> ... </ROW> </TABLE></pre>
- FILE FORMAT=<file format>	省略可	
-FIELD_SEP=<field_sep>	省略可	FILE_FORMAT=variable の場合の、ASCII 形式のフィールド・セパレータ。デフォルトの <field_sep>はタブ文字です。
- XFIELD_SEP=<field_sep>	省略可	FILE_FORMAT=variable の場合の、16 進形式のフィールド・セパレータ。デフォルトの <field_sep>はタブ文字です。
-ROW_SEP=<row_sep>	省略可	<p>ASCII 形式のレコード・セパレータ文字。デフォルトの<row_sep>は Windows の改行です。たとえば、次の値を使用できます。</p> <ul style="list-style-type: none"> • Unix: -ROW_SEP=\n • Windows: -ROW_SEP=¥r¥n
-XROW_SEP=<row_sep>	省略可	16 進形式のレコード・セパレータ文字。例:

- DATE FORMAT=<date format>	省略可	0A。 DATE データ型で使用する出力形式。詳細は、「日付書式」を参照してください。
-ABS=<yes no>	省略可	ファイル形式 fixed で yes に設定した場合、数値の絶対値のみ。つまり、数の正負符号が出力されません。それ以外の場合、数値の前に正負符号が追加され、フィールドの物理長は論理長に 1 を加えた値に等しくなります。
- FETCH SIZE=<array fetch_size>	省略可	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）。
CR/LF <sql_query>、または- QUERY=<sql_query>、または- QUERY FILE=<sql query file>	必須	データ・サーバー上で実行する SQL 問合せ。問合せは、有効なレコードセットを返す SELECT 文またはストアド・プロシージャのコールである必要があります。この問合せは、 OdiSqlUnload コマンドの次の行に入力できません（改行記号 CR/LF は問合せの先頭を示します）。問合せは、 -QUERY パラメータで定義するか、 -QUERY_FILE パラメータで指定されたファイルに格納できます。このコマンドを OS のコマンドラインからコールする場合は、 -QUERY または -QUERY_FILE パラメータを使用してください。 注意: 以前の、 REQUEST および REQUEST_FILE を使用する構文もまだ有効ですが、廃止予定です。

例

次のコマンドは、**Customers** 表での問合せの結果を含む、コンマで区切られたファイル **C:\temp\clients.csv** を生成します。

```
OdiSqlUnload -FILE=C:\temp\clients.csv -
DRIVER=sun.jdbc.odbc.JdbcOdbcDriver -URL=jdbc:odbc:NORTHWIND_ODBC -
USER=sa -PASS=NFNEKKNKGJHAHBHDHEHJDBGBGFDGGH -FIELD_SEP=; "-
DATE_FORMAT=dd/MM/yyyy hh:mm:ss"
select cust_id, cust_name, cust_creation_date
from Northwind.dbo.Customers
```

OdiUnZip

使用方法

```
OdiUnZip -FILE=<file> -TODIR=<dest_dir> [-OVERWRITE=<yes|no>] [-
ENCODING=<file_name_encoding>]
```

説明

アーカイブ・ファイルをディレクトリに `unzip` するために使用します。

パラメータ

パラメータ	必須かどうか	説明
<code>-FILE=<file></code>	必須	<code>unzip</code> する zip ファイルへのフルパス。
<code>-TODIR=<dest_file></code>	必須	宛先ディレクトリまたはフォルダ。
<code>-OVERWRITE=<yes no></code>	省略可	ターゲット・ディレクトリにすでに存在するファイルを上書きするかどうかを示します。
<code>-ENCODING=<file name encoding></code>	省略可	アーカイブ・ファイル内部でファイル名に使用されている文字エンコーディング。可能な値のリストは、 HTTP://java.sun.com/製品/JDK/1.2/docs/guide/internat/encoding.doc.html を参照してください。デフォルトは、プラットフォームのデフォルト文字エンコーディングです。

例

ファイル `archive_001.zip` を、ディレクトリ `C:¥archive¥` からディレクトリ `C:¥TEMP` に `unzip` します。

```
OdiUnZip "-FILE=C:¥archive¥archive_001.zip" -TODIR=C:¥TEMP¥
```

OdiZip

使用方法

```
OdiZip -DIR=<dir> -FILE=<file> -TOFILE=<dest_file> [-OVERWRITE=<yes|no>] [-RECURSE=<yes|no>] [-CASESENS=<yes|no>] [-ENCODING=<file_name_encoding>]
```

説明

ディレクトリまたは複数のファイルから `zip` ファイルを作成します。

パラメータ

パラメータ	必須かどうか	説明
<code>-DIR=<dir></code>	必須	生成する zip ファイルのルートになるベース・ディレク

		トリ（またはフォルダ）。
-FILE=<file>	必須	ファイルのベース・ディレクトリからアーカイブへのパス。任意の文字を指定するには「*」を使用します。 例: /var/tmp/*.log（ディレクトリ/var/tmp 内の、拡張子が log のすべてのファイル） arch_*.lst（arch_で始まり、拡張子が lst のすべてのファイル）
-TOFILE=<dest_file>	必須	ターゲット zip ファイル。
-OVERWRITE=<yes <u>no</u> >	省略可	ターゲット zip ファイルがすでに存在する場合に、上書きするか（yes）、更新するだけにするか（no）を示します。デフォルトでは、すでに存在する zip ファイルは更新されます。
-RECURSE=< <u>yes</u> no>	省略可	他のディレクトリを含むディレクトリの場合にアーカイブを再帰的に実行するかどうかを示します。値が no の場合、コピーするディレクトリ（サブフォルダなし）に含まれるファイルのみがアーカイブされます。
-CASESENS=<yes <u>no</u> >	省略可	ファイル検索で大文字と小文字が区別されるかどうかを示します。デフォルトでは、大文字のファイルが検索されます。
-ENCODING=<file name encoding>	省略可	アーカイブ・ファイル内部でファイル名に使用する文字エンコーディング。可能な値のリストは、 HTTP://java.sun.com/製品/JDK/1.2/docs/guide/internat/encoding.doc.html を参照してください。デフォルトは、プラットフォームのデフォルト文字エンコーディングです。

例

ディレクトリ C:\Program files\odi のアーカイブを作成します。

```
OdiZip "-DIR=C:\Program Files\odi" -FILE=*. * -
TOFILE=C:\TEMP\odi_archive.zip
```

ディレクトリ C:\Program files\odi のアーカイブを作成します。アーカイブ内に odi ディレクトリが維持されます。

```
OdiZip "-DIR=C:\Program Files" -FILE=odi\*. * -
TOFILE=C:\TEMP\odi_archive.zip
```

SAP

OdiSAPALEClient

使用方法

```
OdiSAPALEClient -USER=<UserName> -ENCODED_PASSWORD=<password> -
GATEWAYHOST=<gatewayhost> -SYSTEMNR=<SYSTEMNR> [-DIR=<SourceDirName>] [-
FILE=<FileName>] [-CASESENS=<yes|no>] [-MOVEDIR=<DirNameToMove>] [-
DELETE=<yes|no>] [-POOL_KEY=<pool_key>] [-LANGUAGE=<language>] [-
CLIENT=<client>] [-MAX_CONNECTIONS=<n>] [-TRACE=<no|yes>]
```

説明

XML ソース・ファイルから SAP Internal Documents (IDoc) を生成し、リモートの tRFC Server (たとえば SAP R/3 Server) へのクライアント接続を作成し、その後 ALE (Application Link Enabling) を使用してこれらの IDoc を転送します。

注意: この API は SAP R/3 用 Open Connector の一部です。

パラメータ

パラメータ	必須かどうか	説明
-USER=<UserName>	必須	SAP ログオン。このユーザーはシステム・ユーザーである場合があります。
-PASSWORD=<password>	廃止予定です。	SAP ログオン・パスワード。このコマンドは廃止予定です。かわりに ENCODED_PASSWORD を使用してください。
-ENCODED_PASSWORD=<password>	必須	暗号化された SAP ログオン・パスワード。OS コマンド agent ENCODE <password> によるパスワードの暗号化が可能です。
-GATEWAYHOST=<gatewayhost>	必須	ゲートウェイ・ホスト。
-SYSTEMNR=<SYSTEMNR>	必須	SAP システム番号。
-DIR=<SourceDirName>	省略可	XML ソース・ファイル・ディレクトリ。このパラメータは、-FILE を指定しない場合に考慮されます。-DIR と -FILE の少なくともいずれかのパラメータを指定する必要があります。
-FILE=<FileName>	省略可	ソース XML ファイルの名前。このパラメータを省略すると、-DIR のすべてのファイルが処理されます。-DIR と -FILE の少なくともいずれかのパラメータを指定する必要があります。
-CASESENS=<yes no>	省略可	ソース・ファイル名で大文字と小文字が区別されるかどうかを示します。デフォルトは NO です。
-MOVEDIR=<DirNameToMove>	省略可	このパラメータを指定すると、処理後にソース・フ

>		ファイルをこのディレクトリに移動します。
-DELETE=<yes no>	省略可	処理後にソース・ファイルを削除します。デフォルトは yes です。
-POOL_KEY=<pool_key>	省略可	接続プールの名前。デフォルトは ODI です。
-LANGUAGE=<language>	省略可	エラー・メッセージで使用する言語コード。デフォルトは EN です。
-CLIENT=<client>	省略可	クライアント識別子。デフォルトは 001 です。
-MAX_CONNECTIONS=<n>	省略可	プールの最大接続数。デフォルトは 3 です。
-TRACE=<no yes>	省略可	生成された IDoc ファイルは、ソース・ファイル・ディレクトリにアーカイブされます。ソース・ファイルが移動されると (-MOVEDIR パラメータ)、生成された IDoc も移動します。デフォルトは no です。

例

/sap ディレクトリのすべてのファイル进行处理して、SAP Server に IDoc として送信します。オリジナルの XML と生成されたファイルは、処理後、/log ディレクトリに格納されます。

```
OdiSAPALEClient -USER=ODI -ENCODED_PASSWORD=xxx -SYSTEMNR=002 -
GATEWAYHOST=GW001 -DIR=/sap -MOVEDIR=/log -TRACE=yes
```

OdiSAPALEServer

使用方法

```
OdiSAPALEServer -USER=<UserName> -ENCODED_PASSWORD=<password> -
GATEWAYHOST=<gatewayhost> -SYSTEMNR=<SYSTEMNR> -
GATEWAYNAME=<gatewayname> -PROGRAMID=<programid> -DIR=<DirName> [-
TIMEOUT=<n>] [-POOL_KEY=<pool_key>] [-LANGUAGE=<Language>] [-
CLIENT=<Client>] [-MAX_CONNECTIONS=<n>] [-INTERREQUESTTIMEOUT=<n>] [-
MAXREQUEST=<n>] [-TRACE=<no|yes>]
```

説明

ALE (Application Link Enabling) を使用して転送された SAP IDoc ファイルを受信できる tRFC リスナーを起動して、指定されたディレクトリのターゲット XML ファイルに変換します。

注意: この API は SAP R/3 用 Open Connector の一部です。

パラメータ

パラメータ	必須かどうか	説明
-------	--------	----

-USER=<UserName>	必須	SAP ログオン。このユーザーはシステム・ユーザーである場合があります。
-PASSWORD=<password>	廃止予定です。	SAP ログオン・パスワード。このコマンドは廃止予定です。かわりに ENCODED_PASSWORD を使用してください。
- ENCODED_PASSWORD=<password>	必須	暗号化された SAP ログオン・パスワード。OS コマンド agent ENCODE <password>によるパスワードの暗号化が可能です。
- GATEWAYHOST=<gatewayhost>	必須	ゲートウェイ・ホスト。
- SYSTEMNR=<SYSTEMNR>	必須	SAP システム番号。
- GATEWAYNAME=<gatewayname>	必須	ゲートウェイ名。
- PROGRAMID=<programid>	必須	プログラム ID。tRFC サーバーによって使用される外部名。
- DIR=<DirName>	必須	ターゲット XML ファイルが格納されるディレクトリ。これらのファイルは<IDOC Number>.xml という名前、IDoc タイプの名前を持つサブディレクトリにあります。デフォルトは ./FromSAP です。
- TIMEOUT=<n>	省略可	サーバーの寿命（単位はミリ秒）。この期間が経過すると、サーバーは自動的に停止します。このタイムアウトがゼロに設定されている場合、サーバーの寿命は無限です。デフォルトは 0 です。
- POOL_KEY=<pool_key>	省略可	接続プールの名前。デフォルトは ODI です。
- LANGUAGE=<Language>	省略可	エラー・メッセージで使用する言語コード。デフォルトは EN です。
- CLIENT=<Client>	省略可	SAP クライアント識別子。デフォルトは 001 です。
- MAX_CONNECTIONS=<n>	省略可	接続プールの最大許容接続数。デフォルトは 3 です。
- INTERREQUESTTIMEOUT=<n>	省略可	n ミリ秒の間に受信した IDOC がない場合、リスナーは停止します。このタイムアウトがゼロに設定されている場合、タイムアウトは無制限です。デフォルトは 0 です。
- MAXREQUEST=<n>	省略可	到達するとリスナーが停止する最大リクエスト数。このパラメータがゼロに設定されている場合、サーバーはリクエスト数が無限だと想定します。デフォ

		ルトは0です。 注意: -TIMEOUT, -INTERREQUESTTIMEOUT および-MAXREQUEST がゼロに設定されるか、空欄のままにしておかれると、-MAXREQUEST は自動的に値1をとります。
-TRACE=<no yes>	省略可	デバッグ・トレースをアクティブにします。デフォルトはnoです。

例

2つのIDocファイルを待機し、/tempディレクトリにターゲットXMLファイルを生成します。

```
OdiSAPALEServer -POOL_KEY=ODI -MAX_CONNECTIONS=3 -CLIENT=001 -USER=ODI -
ENCODED_PASSWORD=xxx -LANGUAGE=EN -GATEWAYHOST=SAP001 -SYSTEMNR=002 -
GATEWAYNAME=GW001 -PROGRAMID=ODI01 -DIR=/tmp -MAXREQUEST=2
```

イベント検出

OdiFileWait

使用方法

```
OdiFileWait -DIR=<dir> -PATTERN=<pattern> [-
ACTION=<DELETE|COPY|MOVE|APPEND|ZIP|NONE>] -TODIR=<dest_dir> -
TOFILE=<dest_file> [-OVERWRITE=<yes|no>] [-CASESENS=<yes|no>] [-
FILECOUNT=<n>] [-TIMEOUT=<n>] [-POLLINT=<n>] [-HEADER=<n>] [-
KEEP_FIRST_HEADER=<yes|no>] [-NOFILE_ERROR=<yes|no>]
```

説明

ディレクトリ DIR をスキャンして、PATTERN で指定されたマスクに合致するファイルを待機します。指定したファイルが見つかったら、パラメータ ACTION で示されたアクションがトリガーされます。

TIMEOUT で指定されたタイムアウト制限に達するまで、実行エージェントはスタンバイ・モードで FILECOUNT 個のファイルの受信を待機します。ディレクトリは POLLINT ミリ秒間隔でスキャンされます。

パラメータ

パラメータ	必須かどうか	説明
-ACTION= <DELETE COPY MOVE APPEND ZIP NONE>	省略可	見つかったファイルに対するアクション。 DELETE: 見つかったファイルを削除します。 COPY: 見つかったファイルをディレクトリ TODIR にコピーします。 MOVE: 見つかったファイルをフォルダ

		<p>TODIR に移動するか、TOFILE で指定された名前に変更します。</p> <p>APPEND: 見つかったすべてのファイルを連結し、結果ファイル TOFILE を作成します。ソース・ファイルは削除されます。</p> <p>ZIP: 見つかったファイルを zip し、zip ファイル TOFILE に格納します。</p> <p>NONE: アクションを実行しません。これがデフォルトの動作です。</p>
-DIR=<dir>	必須	スキャンするディレクトリ（またはフォルダ）。
-PATTERN=<pattern>	必須	<p>スキャンするファイル名のマスク。任意の文字を指定するには「*」を使用します。</p> <p>例:</p> <p>*.log（拡張子 log を持つすべてのファイル）</p> <p>arch_*.lst（arch_ で始まり、拡張子が lst のすべてのファイル）</p>
-TODIR=<dest_dir>	省略可	<p>アクションのターゲット・ディレクトリ。アクションにより異なります。</p> <p>COPY: ファイルがコピーされるディレクトリ</p> <p>MOVE: ファイルが移動されるディレクトリ</p>
-TOFILE=<dest_file>	省略可	<p>宛先ファイル。アクションにより異なります。</p> <p>MOVE: 移動されたファイルの名前変更マスク。</p> <p>APPEND: 連結の結果ファイルの名前。</p> <p>ZIP: ZIP の結果ファイルの名前。</p>
-OVERWRITE=<yes <u>no</u> >	省略可	<p>宛先ファイルが存在する場合、それを上書きするかどうかを示します。デフォルトでは、値は no に設定されます。</p> <p>このオプションを APPEND とともに使用する場合、宛先ファイルの内容が、最後に処理されたファイルの内容になることに注意してください。</p>
-CASESENS=<yes <u>no</u> >	省略可	ファイル検索で大文字と小文字が区別されるかどうかを示します。デフォルトでは、検索されるファイルは大文字です。
-FILECOUNT=<n>	省略可	待機するファイル数の最大値（デフォルトは 0）。この数に到達すると、コマンドは終了

-TIMEOUT=<n>	省略可	<p>します。</p> <p>値に 0 を指定すると、タイムアウトするまで、すべてのファイルを待機します。</p> <p>このパラメータが 0 で、タイムアウトも 0 の場合、このパラメータは警告なく強制的に 1 にされます。</p>
		<p>最大待機時間（単位はミリ秒。デフォルトは 60000）。</p> <p>この時間に達すると、パラメータ FILECOUNT の値にかかわらず、コマンドは次のコマンドに制御を渡します。</p> <p>値 0 は、待機時間が無制限であることを示します（パラメータ FILECOUNT で指定されたメッセージ最大読取り数まで待機します）。</p>
-POLLINT=<n>	省略可	<p>新規ファイルを検索する間隔（単位はミリ秒）。デフォルトは 2000 に設定されます（2 秒）。これは、Oracle Data Integrator が 2 秒間隔で新しいメッセージを検索することを意味します。OdiFileWait 中に作成されたファイルは、この間隔の間、ファイルを閉じた後で対象になります（ファイル・サイズ不変）。</p>
-HEADER=<n>	省略可	<p>このパラメータは APPEND アクションでのみ有効です。</p> <p>連結前にソース・ファイルから削除されるヘッダ行の行数。デフォルトは処理なしです。</p>
-KEEP_FIRST_HEADER=<yes no>	省略可	<p>このパラメータは APPEND アクションでのみ有効です。</p> <p>連結時に最初のファイルのヘッダ行を維持するために使用されます。</p>
-NOFILE_ERROR=<yes no>	省略可	<p>ファイルが見つからない場合の動作を示します。</p> <p>値 no（デフォルト）は、ファイルが見つからなくてもエラーを生成しないことを示します。</p>

例

ディレクトリ c:\events のファイル flag.txt を時間制限なしで待機し、ファイルが検出されたら先に進みます。

```
OdiFileWait -ACTION=NONE -DIR=c:¥events -PATTERN=flag.txt -FILECOUNT=1 -
TIMEOUT=0 -POLLINT=1000
```

ディレクトリ c:¥events のファイル flag.txt を時間制限なしで待機し、ファイルが検出されたら削除します。

```
OdiFileWait -ACTION=DELETE -DIR=c:¥events -PATTERN=flag.txt -FILECOUNT=1
-TIMEOUT=0 -POLLINT=1000
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c: ¥sales_in を 1 秒間隔でスキャンし、ディレクトリ C: ¥sales_ok にあるファイル sales.dat に連結します。最初のファイルのヘッダーを維持します。

```
OdiFileWait -ACTION=APPEND -DIR=c:¥sales_in -PATTERN=*.dat -
TOFILE=c:¥sales_ok¥sales.dat -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -
HEADER=1 -KEEP_FIRST_HEADER=yes -OVERWRITE=yes
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c: ¥sales_in を 1 秒間隔でスキャンし、ディレクトリ C: ¥sales_ok にファイルをコピーします。上書きはしません。

```
OdiFileWait -ACTION=COPY -DIR=c:¥sales_in -PATTERN=*.dat -
TODIR=c:¥sales_ok -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -
OVERWRITE=no
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c: ¥sales_in を 1 秒間隔でスキャンし、ファイルを zip ファイルにアーカイブします。

```
OdiFileWait -ACTION=ZIP -DIR=c:¥sales_in -PATTERN=*.dat -
TOFILE=c:¥sales_ok¥sales.zip -FILECOUNT=0 -TIMEOUT=350000 -POLLINT=1000 -
OVERWRITE=yes
```

売上ファイル*.dat を 5 分間待機し、ディレクトリ c: ¥sales_in を 1 秒間隔でスキャンし、ディレクトリ C: ¥sales_ok にファイルを移動します。上書きはしません。ファイル名に.bak を追加します。

```
OdiFileWait -ACTION=MOVE -DIR=c:¥sales_in -PATTERN=*.dat -
TODIR=c:¥sales_ok -TOFILE=*.bak -FILECOUNT=0 -TIMEOUT=350000 -
POLLINT=1000 -OVERWRITE=no
```

OdiReadMail

使用方法

```
OdiReadMail -MAILHOST=<mail_host> -USER=<mail_user> -
PASS=<mail_user_password> -FOLDER=<folder_path> [-PROTOCOL=<pop3|imap>]
[-FOLDER_OPT=<none|sender|subject>] [-KEEP=<no|yes>] [-
EXTRACT_MSG=<yes|no>] [-EXTRACT_ATT=<yes|no>] [-MSG_PRFX=<my_prefix>] [-
ATT_PRFX=<my_prefix>] [-USE_UCASE=<no|yes>] [-NOMAIL_ERROR=<no|yes>] [-
TIMEOUT=<timeout>] [-POLLINT=<pollint>] [-MAX_MSG=<max_msg>] [-
SUBJECT=<subject_filter>] [-SENDER=<sender_filter>] [-TO=<to_filter>] [-
CC=<cc_filter>] [-BCC=<bcc_filter>]
```

説明

電子メールとその添付ファイルを POP または IMAP インターネット・メール・サービスから読み取るために使用できます。

このコマンドを使用すると、USER と PASS で指定された接続パラメータを使用して、サーバー MAILHOST 上でホスティングされているメールに接続できます。その後、実行エージェントは、MAX_MSG 件のメッセージを受信するか、TIMEOUT で指定された最大待機時間に達するまで待機します。抽出されたメッセージは、パラメータ SUBJECT および SENDER で指定されるようなフィルタに合致する必要があります。メッセージが基準に合致した場合、メッセージの内容と添付ファイルが、パラメータ FOLDER によって指定されたディレクトリに抽出されます。パラメータ KEEP が「no」に設定されている場合、取得したメッセージは、その後メールボックスから削除されます。

パラメータ

パラメータ	必須かどうか	説明
-MAILHOST=<mail_host>	必須	POP または IMAP メール・サーバーの IP アドレス。
-USER=<mail_user>	必須	有効なメール・サーバーのアカウント。
-PASS=<mail_user_password>	必須	メール・サーバー・アカウントのパスワード。
-FOLDER=<folder_path>	必須	添付ファイルおよびメッセージ用ストレージ・フォルダのフルパス。
-PROTOCOL=<pop3 imap>	省略可	メール・アクセスのタイプ (POP3 または IMAP)。デフォルトは POP3 です。
-FOLDER_OPT=<none sender subject>	省略可	次のパラメータに応じてディレクトリ FOLDER にサブディレクトリを作成します。 <ul style="list-style-type: none"> • none (デフォルト) : アクションなし。 • sender: 送信者の外部名でサブディレクトリが作成されます。 • subject: メッセージの件名でサブディレクトリが作成されます。 <p>注意: sender および subject フォルダ・オプションでは、スペースと、英数字以外の文字 (@など) は、生成されたフォルダ名ではアンダースコアに置き換えられます。</p>
-KEEP=<no yes>	省略可	yes: フィルタに合致するメッセージを、読んだ後でメールボックスに保存します。 no (デフォルト) : フィルタに合致するメッセ

		ージを、読んだ後でメールボックスから削除します。
-EXTRACT_MSG=<yes no>	省略可	yes (デフォルト) : メッセージの本文をファイルに抽出します。 no: メッセージの本文をファイルに抽出しません。
-EXTRACT_ATT=<yes no>	省略可	はい (デフォルト) : 添付ファイルをファイルに抽出します。 no: 添付ファイルを抽出しません。
-MSG_PRF=<my_prefix>	省略可	メッセージ本文を含むファイルの接頭辞。デフォルトは MSG です。
-ATT_PRF=<my_prefix>	省略可	添付ファイルを含むファイルの接頭辞。元のファイル名は維持されます。
-USE_UCASE=<no yes>	省略可	yes: ファイル名を強制的に大文字にします。 no (デフォルト) : 大文字と小文字を元のままにします。
-NOMAIL_ERROR=<no yes>	省略可	yes: メールが指定された基準に合致しない場合、エラーを生成します。 no (デフォルト) : 指定された基準に対応するメールがない場合、エラーを生成しません。
-TIMEOUT=<timeout>	省略可	最大待機時間 (単位はミリ秒。デフォルトは 0)。 この待機時間に到達すると、コマンドは終了します。 値 0 (デフォルト) は、待機時間が無制限であることを示します (パラメータ MAX_MSG で指定された最大受信メッセージ数に達するまで)。
-POLLINT=<pollint>	省略可	新しいメッセージをスキャンする検索間隔 (単位はミリ秒)。デフォルトは 1000 です (1 秒)。
-MAX_MSG=<max_msg>	省略可	抽出するメッセージ数の最大値 (デフォルトは 1)。この数に到達すると、コマンドは終了します。
-SUBJECT=<subject_filter>	省略可	件名に応じてメッセージをフィルタするために使用されるパラメータ。
-SENDER=<sender_filter>	省略可	送信者に応じてメッセージをフィルタするために使用されるパラメータ。

-TO=<to_filter>	省略可	アドレスに応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。
-CC=<cc_filter>	省略可	CC に応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。
-BCC=<bcc_filter>	省略可	BCC に応じてメッセージをフィルタするために使用されるパラメータ。このオプションを繰り返し適用して複数のフィルタを作成することができます。

例

support のメールを自動受信します。添付ファイルは切り離してエージェントのマシン上のフォルダ C:\support に格納します。最大待機時間を 10 秒として、すべてのメッセージを待機します。

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass -
KEEP=no -FOLDER=c:\support -TIMEOUT=10000 -MAX_MSG=0 -
SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TEXT -
EXTRACT_ATT=yes
```

1 分に 1 度、新しいメッセージをチェックして、メッセージが 10 件になるまで時間無制限で待機します。

```
OdiReadMail -MAILHOST=mail.mymail.com -USER=myaccount -PASS=mypass -
KEEP=no -FOLDER=c:\support -TIMEOUT=0 -MAX_MSG=10 -POLLINT=60000 -
SENDER=support@mycompany.com -EXTRACT_MSG=yes -MSG_PRF=TEXT -
EXTRACT_ATT=yes
```

OdiSleep

使用方法

```
OdiSleep -DELAY=<delay>
```

説明

<delay> ミリ秒間待機します。

パラメータ

パラメータ	必須かどうか	説明
-DELAY=<delay>	必須	待機するミリ秒数。

例

5 秒間待機します。

```
OdiSleep -DELAY=5000
```

OdiWaitForChildSession

使用方法

```
OdiWaitForChildSession [-PARENT_SESS_NO=<parent_sess_number>] [-
POLL_INT=<polling_interval>] [-
SESSION_NAME_FILTER=<session_name_filter>] [-
SESSION_KEYWORDS=<session_keywords>] [-
MAX_CHILD_ERROR=ALL|<error_number>]
```

説明

<parent_sess_number>で指定されたセッションから起動されたすべてのセッションが終了したかどうかを<polling_interval>秒間隔でチェックします。これらすべての子セッション、場合によっては名前やキーワードでフィルタされたセッションが終了すると（ステータスが Done、Warning または Error）、このコマンドは終了します。

パラメータ

パラメータ	必須かどうか	説明
- PARENT SESS NO=<parent sess number>	省略可	親セッションの ID。このパラメータを指定しない場合、現在のセッション ID が使用されます。
-POLL_INT=<polling_interval>	省略可	一連の子セッション終了テストの間隔（単位は秒）。デフォルト値は 1 です。
- SESSION NAME FILTER=<session name filter>	省略可	名前がこのフィルタに合致する子セッションのみがテストされます。このフィルタには、SQL の LIKE 形式のパターンを使用できます。
- SESSION KEYWORDS=<session keywords>	省略可	このコンマ区切りのリストにすべてのキーワードが合致する項目がある子セッションのみがテストされます。このリストの各要素には、SQL の LIKE 形式のパターンを使用できます。
-MAX CHILD ERROR= ALL <error_number>	省略可	このパラメータを設定すると、一定数の子セッションがエラーにより終了していた場合、

OdiWaitForChildSession がエラーにより終了します。

- ALL: すべての子セッションがエラーで終了した場合にエラーになります。
- <error_number>:
<error_number>以上の数の子セッションがエラーで終了した場合にエラーになります。

このパラメータがゼロ、負または未指定の場合、

OdiWaitForChildSession サーバーは、エラーになった子セッションの数にかかわらず、エラー・ステータスにより終了することはありません。

例

5 秒のポーリング間隔で、名前が「LOADxxx」に類似し、「MANDATORY」および「CRITICAL」をキーワードに持っている現在のセッションのすべての子セッションの終了を待機します。

```
OdiWaitForChildSession -PARENT_SESS_NO=<%=odiRef.getSession("SESS_NO")%>
-POLL_INT=5 -SESSION_NAME_FILTER=LOAD% -
SESSION_KEYWORDS=MANDATORY,CRITICAL
```

OdiWaitForData

使用方法

```
OdiWaitForData -LSHEMA=<LSchema> -TABLE_NAME=<tableName> [-
OBJECT_TYPE=<list of object types>] [-CONTEXT=<context>] [-
RESUME_KEY_VARIABLE=<resumeKeyVariable> -RESUME_KEY_COL=<resumeKeyCol>
[-RESUME_KEY_OPERATOR=<resumeKeyOperator>] |-SQLFILTER=<SQLFilter>] [-
TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-UNIT_ROWCOUNT=<unitRowCount>] [-
TIMEOUT_WITH_ROWS_OK=<yes|no>] [-INCREMENT_DETECTION=<no|yes> [-
INCREMENT_MODE=<M|P|I>] [-
INCREMENT_SEQUENCE_NAME=<incrementSequenceName>]]
```

説明

表または表の集合内の一定数の行を待機します。これは、ビューなど、データが含まれている一定数のオブジェクトにも適用できます。

OdiWaitForData コマンドは、表または表の集合に一定数のレコードが入力されているかどうかをテストします。このテストは一定間隔で (-POLLINT)、目的の数の行が表の 1 つで検出されたか (-UNIT_ROWCOUNT)、すべての表に対して、総計で目的の数の行が検出されたか (-GLOBAL_ROWCOUNT)、またはタイムアウト (-TIMEOUT) に達したかの、いずれかの条件が満たされるまで繰り返されます。

検出された行の集合に対してフィルタを適用できます。フィルタは、明示的な SQL の `where` 句 (`-SQLFILTER`) またはフィールド値演算子句を決定する `-RESUME_KEY_xxx` パラメータ (あるいはその両方) によって指定されます。この 2 つの方式は累積的です (AND)。

行数は絶対数で判定することも (表内の行の総数に対して)、差異によって判定することも (格納された参照値と現在の行数値の差) 可能です。

複数の表を扱う場合は次のようになります。

- `-SQLFILTER` および `-RESUME_KEY_xxx` パラメータは、関係するすべての表に適用されます。
- `UNIT_ROWCOUNT` パラメータは、特定の表の個々に対して想定する行数を決定します。 `-GLOBAL_ROWCOUNT` パラメータは、表の集合に対して累積された行数の総和を決定します。関係している表が 1 つのみの場合、`-UNIT_ROWCOUNT` パラメータと `-GLOBAL_ROWCOUNT` パラメータは同等です。

パラメータ

パラメータ	必須かどうか	説明
<code>-LSHEMA=<LSchema></code>	必須	表が含まれている論理スキーマ。
<code>-TABLE_NAME=<tableName></code>	必須	チェックする表名、マスクまたは表名のリスト。このパラメータには次の 3 つの形式があります。 <ul style="list-style-type: none"> • 表名。 • 表名マスク: このマスクは、ポーリングする表を選択します。マスクは SQL の <code>LIKE</code> 構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号は単一文字のワイルドカードです。 • 表名リスト: 表名のコンマ区切りのリスト。前述のように定義されたマスクが許可されます。
<code>-OBJECT TYPE=<list of object types></code>	省略可	チェックされるオブジェクトのタイプ。デフォルトでは、表のみがチェックされます。他のオブジェクトを考慮するには、オブジェクト・タイプのコンマ区切りリストを指定します。サポートされているオブジェクト・タイプは次のとおりです。 <ul style="list-style-type: none"> • T: 表 • V: ビュー
<code>-CONTEXT=<context></code>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。

-SQLFILTER=<SQLFilter>	省略可	<p>表に適用される明示的な SQL フィルタ。この文は、チェックされた表が含まれているテクノロジーで有効である必要があります。</p> <p>注意: この文は WHERE キーワードを含むことはできません。</p>
- RESUME KEY VARIABLE=<resumeKeyVariable> - RESUME KEY COL=<resumeKeyCol> [- RESUME KEY OPERATOR=<resumeKeyOperator>]	省略可	<p>RESUME_KEY_XXX パラメータを使用すると、ポーリングされた表で、検出された行の集合をフィルタリングできます。</p> <ul style="list-style-type: none"> • <key_column>: チェックされた表の列の名前。 • <operator>: : チェックされた表が含まれているテクノロジーで有効な比較演算子。このパラメータを省略すると、デフォルトで値「>」が使用されます。 • <variable_name>: 値が事前に設定された変数の名前。変数名の先頭には、「:」（バインド）または「#」（代入）を接頭辞として付ける必要があります。変数の有効範囲は、Oracle Data Integrator 構文で明示的に記述してください。GLOBAL.<変数名>（グローバル変数の場合）、<プロジェクト・コード>.<変数名>（プロジェクト変数の場合）。
-TIMEOUT=<timeout>	省略可	<p>データがポーリングされる時間の最大の長さ（単位はミリ秒）。この値がゼロの場合、タイムアウトは無制限です。デフォルトは0です。</p>
-POLLINT=<pollInt>	省略可	<p>データのポーリングを待機する時間（単位はミリ秒）。デフォルトは2000です。</p>
- UNIT ROWCOUNT=<unitRowCount>	省略可	<p>ポーリングされる表で、コマンドを終了させるために想定されている行数。デフォルトは1です。</p>
- GLOBAL ROWCOUNT=<globalRowCount>	省略可	<p>コマンドを終了させるために、表の集合に対して累積的に想定されている行数。デフォルトは1です。</p>
- INCREMENT DETECTION=<no yes>	省略可	<p>コマンドが行数を判定するモードを定義します。行数を絶対数で判定するか（表内の行の総数に対して）、差異によって判定するか（格納された参照値と現在の行数値の差）いずれかです。</p> <ul style="list-style-type: none"> • yes に設定すると、行カウントは差分モ

<p>- INCREMENT_MODE=<M P I></p>	省略可	<p>ードで実行されます。表に追加された行の数が、格納された参照値と比較されます。参照値は INCREMENT_MODE パラメータに依存します。</p> <ul style="list-style-type: none"> no に設定すると、カウントは絶対行数モードで実行されます。 <p>デフォルトは no です。</p> <p>このパラメータは、連続する OdiWaitForData コールの間、参照値の永続モードを指定します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> M: メモリー。参照値は非永続です。OdiWaitForData がコールされる場合、参照値は、ポーリングされる表の行数に等しい値をとります。OdiWaitForData が終了すると、値は失われます。このモードでは、次のコールには、新しい参照値が設定されます。 P: 永続。参照値は永続的です。OdiWaitForData の開始時に増分順序から読み取られ、OdiWaitForData の終了時に増分順序に保存されます。増分順序が設定されていない場合（最初のコール時）、現在の表の行数が使用されます。 I: 初期。参照値は初期化され、永続的です。参照値は、OdiWaitForData の開始時に、ポーリングされる表の行数に等しい値をとります。OdiWaitForData の終了時に、永続モードとして増分順序に保存されます。 <p>デフォルトは M です。</p> <p>注意: マスクまたは表のリストがポーリングされる場合、永続または初期モードはサポートされません。</p>
<p>- INCREMENT_SEQUENCE_NAME=<incrementSequenceName></p>	省略可	<p>このパラメータは、参照値の永続性のために使用される、自動割当てされた記憶域の名前を指定します。この増分順序はリポジトリに格納されます。この名前を指定しない場合、表の名前をとります。</p> <p>警告: この増分順序は Oracle Data Integrator の順序ではなく、OdiWaitForData へのコールの外部では順序として使用できません。</p>

<pre>- TIMEOUT WITH ROWS OK=<yes no></pre>	省略可	このパラメータが Y に設定されている場合、1 つ以上の行が検出され、想定された行数が挿入される前にタイムアウトが起きると、リターン・コード 0 で API が終了します。それ以外の場合、エラーが生成されます。デフォルトは yes です。
---	-----	---

例

ORA_WAITFORDATA スキーマで、DE1P1 表に、フィルタに合致している 200 件のレコードが含まれるのを待機します。

```
OdiWaitForData -LSHEMA=ORA_WAITFORDATA -TABLE_NAME=DE1P1 -
GLOBAL_ROWCOUNT=200 "-SQLFILTER=DATMAJ >
to_date('#MAX_DE1_DATMAJ_ORACLE_CHAR', 'DD/MM/YYYY HH24:MI:SS')
```

SQLSRV_SALES の CITY_SRC または CITY_TRG 表に新しいデータが表示されるのを、最大 4 時間待機します。

```
OdiWaitForData LSCHEMA=SQLSRV_SALES TABLE_NAME=CITY% TIMEOUT=14400000 -
INCREMENT_DETECTION=yes
```

OdiWaitForLogData

使用方法

```
OdiWaitForLogData -LSHEMA=<LSchema> -TABLE_NAME=<tableName> -
CDC_SET_NAME=<cdcSetName> -SUBSCRIBER_NAME=<subscriberName> [-
CONTEXT=<context>] [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-UNIT_ROWCOUNT=<unitRowCount>] [-
OPTIMIZED_WAIT=<yes|no|AUTO>] [-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

説明

ジャーナル化された表またはそのリストで、一定数の変更が行われるのを待機します。

OdiWaitForLogData コマンドは、表または表のグループ上で行が変更されたかどうかを判定します。この変更の検出には、Oracle Data Integrator のチェンジ・データ・キャプチャ (CDC) をシンプル・モード (-TABLE_NAME パラメータを使用) または一貫モード (-CDC_SET_NAME パラメータを使用) が使用されます。テストは、-POLLINT ミリ秒間隔で、目的の数の行変更が表の 1 つで検出されたか (-UNIT_ROWCOUNT)、すべての表に対して、総計で目的の数の行変更が検出されたか (-GLOBAL_ROWCOUNT)、またはタイムアウト (-TIMEOUT) に達したかの、いずれかの条件が満たされるまで繰り返されます。

注意: このコマンドでは、ジャーナル化されたすべての操作 (挿入、更新および削除) が考慮されます。

警告: このコマンドは単純モードまたは一貫モードでジャーナル化された表にのみ適しています。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	ジャーナル化された表が含まれている論理スキーマ。
-TABLE_NAME=<tableName>	必須	<p>チェックする、ジャーナル化された表名、マスクまたはリスト。このパラメータには次の3つの形式があります。</p> <ul style="list-style-type: none"> 表名。 表名マスク: このマスクは、ポーリングする表を選択します。マスクは SQL の LIKE 構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号はワイルドカードです。 表名リスト: 表名のコンマ区切りのリスト。先に定義されたマスクは許可されません。 <p>注意: このオプションは、<u>単純</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、CDC_SET_NAME と併用できません。CDC_SET_NAME が設定されない場合は必須です。</p>
-CDC_SET_NAME=<cdcSetName>	必須	<p>チェックする CDC セットの名前。この CDC セット名は完全修飾のモデル・コードで、通常は PHYSICAL_SCHEMA_NAME.MODEL_CODE です。</p> <p>次に示すように、代入メソッド API コールを使用して、現在のコンテキストで取得できます。</p> <pre><%=odiRef.getObjectName("L", "model code", "logical schema", "D")%></pre> <p>注意: このオプションは、<u>一貫</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、TABLE_NAME と併用できません。TABLE_NAME が設定されない場合は必須です。</p>
-SUBSCRIBER_NAME=<subscriberName>	必須	ジャーナル化情報を取得するために使用されるサブスクライバの名前。
-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト (CONTEXT CODE)。コンテキストを指定しない場合、実行コンテキストが使用されます。

-TIMEOUT=<timeout>	省略可	変更がポーリングされる時間の最大の長さ（単位はミリ秒）。この値がゼロの場合、タイムアウトは無制限です。デフォルトは0です。
-POLLINT=<pollInt>	省略可	ポーリングを待機する時間（単位はミリ秒）。デフォルトは2000です。
-UNIT ROWCOUNT=<unitRowCount>	省略可	ポーリングされる表の1つで、コマンドを終了させるために想定されている変更数。デフォルトは1です。 注意: -UNIT_ROWCOUNT は、-CDC_SET_NAME とともに考慮されません。
-GLOBAL ROWCOUNT=<globalRowCount>	省略可	コマンドが終了するまでに、表またはCDCセットで想定されている変更の総数。デフォルトは1です。
-OPTIMIZED WAIT=<yes no <u>AUTO</u> >	省略可	ジャーナルにアクセスするために使用された方法。 <ul style="list-style-type: none"> • yes: 最適化された方法。この方法は最近のバージョンのジャーナル化で機能します。非最適化モードより高速に実行されます。 • no: 最適化されていない方法。カウントはジャーナル化する表で実行されます。この方法は低パフォーマンスですが、旧バージョンのジャーナル化機能と互換性があります。 • AUTO: 複数の表がチェックされる場合、最適化された方法が使用されます。それ以外の場合は、最適化されていない方法が使用されます。 デフォルトはAUTOです。
-TIMEOUT WITH ROWS OK=< <u>yes</u> no>	省略可	このパラメータがyesに設定されている場合、1つ以上の行が検出され、事前定義された行数がポーリングされる前にタイムアウトが起きると、リターン・コード0でAPIが終了します。それ以外の場合、エラーが生成されます。デフォルトはyesです。

例

SALES_APPLICATION スキーマの CUSTOMERS 表で、SALES_SYNC サブスクリバのために記録されている行変更が 200 になるのを待機します。

```
OdiWaitForLogData -LSHEMA=SALES_APPLICATION -TABLE_NAME=CUSTOMERS -
GLOBAL_ROWCOUNT=200 -SUBSCRIBER_NAME=SALES_SYNC
```

OdiWaitForTable

使用方法

```
OdiWaitForTable -CONTEXT=<context> -LSHEMA=<LSchema> -
TABLE_NAME=<tableName> [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

説明

表が作成され、事前定義された数の行が入力されるのを待機します。

OdiWaitForTable コマンドは、指定された表が作成されたか、その表に一定数のレコードが入力されたかを定期的にテストします。テストは、表が存在し、指定した数の行を含んでいるか (-GLOBAL_ROWCOUNT)、タイムアウトに達したか (-TIMEOUT) の、いずれかの条件が満たされるまで -POLLINT ミリ秒間隔で繰り返されます。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	表が検索される論理スキーマ。
-TABLE_NAME=<tableName>	必須	検索する表の名前。
-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。
-TIMEOUT=<timeout>	省略可	表が検索される時間の最大の長さ (単位はミリ秒)。この値がゼロの場合、タイムアウトは無制限です。デフォルトは 0 です。
-POLLINT=<pollInt>	省略可	テストを待機する時間 (単位はミリ秒)。デフォルトは 2000 です。
-GLOBAL_ROWCOUNT=<globalRowCount>	省略可	表で、コマンドを終了させるために想定されている行数。デフォルトは 1 です。
-TIMEOUT_WITH_ROWS_OK=<yes no>	省略可	このパラメータが yes に設定されている場合、1 つ以上の行が検出され、想定されたレコード数が検出される前にタイムアウトが起きると、リターン・コード 0 で API が終了します。それ以外の場合、エラーが生成されません。デフォルトは yes です。

例

ORA_WAITFORDATA スキーマで、DE1P1 表が存在し、少なくとも 1 件のレコードが含まれるのを待機します。

```
OdiWaitForTable -LSHEMA=ORA_WAITFORDATA -TABLE_NAME=DE1P1 -
GLOBAL_ROWCOUNT=1
```

チェンジ・データ・キャプチャ

OdiRefreshJournalCount

使用方法

```
OdiRefreshJournalCount -LSHEMA=<LSchema> -TABLE_NAME=<tableName> -
SUBSCRIBER_NAME=<subscriberName> [-CONTEXT=<context>] [-
MAX_JRN_DATE=<maxJrnDate>] -CDC_SET_NAME=<cdc set name>
```

説明

このコマンドは、指定されたジャーナル化サブスクリバについて、指定された表リストまたは CDC セットのために消費する行数をリフレッシュします。このリフレッシュは、論理スキーマと指定されたコンテキスト上で実行され、制限されることがあります。

警告: このコマンドは単純モードまたは一貫モードのジャーナル化された表に適しています。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	ジャーナル化された表が含まれている論理スキーマ。
-TABLE_NAME=<tableName>	Simple CDC の作業では必須	<p>チェックする、ジャーナル化された表名、マスクまたはリスト。このパラメータには次の 3 つの形式があります。</p> <ul style="list-style-type: none"> 表名。 表名マスク: このマスクは、ポーリングする表を選択します。マスクは SQL の LIKE 構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号はワイルドカードです。 表名リスト: 表名のコンマ区切りのリスト。先に定義されたマスクは許可されません。 <p>注意: このオプションは、<u>単純</u>モードでジャーナル化されたモデルの表でのみ機能します。</p>

-CDC_SET_NAME=<cdcSetName>	Consistent Set CDC の作業では必須	<p>警告: このパラメータは、CDC_SET_NAME と併用できません。CDC_SET_NAME が設定されない場合は必須です。</p> <p>チェックする CDC セットの名前。</p> <p>注意: このオプションは、<u>一貫モード</u>でジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、TABLE_NAME と併用できません。TABLE_NAME が設定されない場合は必須です。</p>
-SUBSCRIBER_NAME=<subscriberName>	必須	カウントがリフレッシュされるサブスクライバの名前。
-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。
-MAX_JRN_DATE=<maxJrnDate>	省略可	ジャーナル化イベントが考慮される最終の日付（時刻）。

例

SALES_APPLICATION スキーマの CUSTOMERS 表で、SALES_SYNC サブスクライバのために記録されている変更数をリフレッシュします。このデータストアは単純モードでジャーナル化されます。

```
OdiRefreshJournalCount -LSHEMA=SALES_APPLICATION -TABLE_NAME=CUSTOMERS
-SUBSCRIBER_NAME=SALES_SYNC
```

SALES_APPLICATION スキーマの SALES CDC Set のすべての表で、SALES_SYNC サブスクライバのために記録されている変更数をリフレッシュします。このデータストアは一貫セット CDC でジャーナル化されます。

```
OdiRefreshJournalCount -LSHEMA=SALES_APPLICATION -
SUBSCRIBER_NAME=SALES_SYNC -CDC_SET_NAME=SALES
```

OdiRetrieveJournalData

使用方法

```
OdiRetrieveJournalData -LSHEMA=<LSchema> -TABLE_NAME=<tableName> -
SUBSCRIBER_NAME=<subscriberName> [-CONTEXT=<context>] [-
MAX_JRN_DATE=<maxJrnDate>] -CDC_SET_NAME=<cdc set name>
```

説明

このコマンドは、指定されたジャーナル化サブスクリバのために、指定された表リストまたは CDC セットのジャーナル化されたイベントを取得します。取得は、表が含まれているテクノロジー固有の方法で実行されます。この取得は、論理スキーマと指定されたコンテキスト上で実行され、制限されることがあります。

警告: このコマンドは単純モードまたは一貫モードのジャーナル化された表に適しています。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	ジャーナル化された表が含まれている論理スキーマ。
-TABLE_NAME=<tableName>	必須	<p>チェックする、ジャーナル化された表名、マスクまたはリスト。このパラメータには次の3つの形式があります。</p> <ul style="list-style-type: none"> 表名。 表名マスク: このマスクは、ポーリングする表を選択します。マスクは SQL の LIKE 構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号はワイルドカードです。 表名リスト: 表名のコンマ区切りのリスト。先に定義されたマスクは許可されません。 <p>注意: このオプションは、<u>単純</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、CDC_SET_NAME と併用できません。CDC_SET_NAME が設定されない場合は必須です。</p>
-CDC_SET_NAME=<cdcSetName>	必須	<p>更新する CDC セットの名前。</p> <p>注意: このオプションは、<u>一貫</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、TABLE_NAME と併用できません。TABLE_NAME が設定されない場合は必須です。</p>
-SUBSCRIBER_NAME=<subscriberName>	必須	データが取得されるサブスクリバの名前。

-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。
-MAX_JRN_DATE=<maxJrnDate>	省略可	ジャーナル化イベントが考慮される最終の日付（時刻）。

例

SALES_APPLICATION スキーマの CUSTOMERS 表で、SALES_SYNC サブスクライバのジャーナル化イベントを取得します。

```
OdiRetrieveJournalData -LSHEMA=SALES_APPLICATION -TABLE_NAME=CUSTOMERS
-SUBSCRIBER_NAME=SALES_SYNC
```

OdiWaitForData

使用方法

```
OdiWaitForData -LSHEMA=<LSchema> -TABLE_NAME=<tableName> [-
OBJECT_TYPE=<list of object types>] [-CONTEXT=<context>] [-
RESUME_KEY_VARIABLE=<resumeKeyVariable> -RESUME_KEY_COL=<resumeKeyCol>
[-RESUME_KEY_OPERATOR=<resumeKeyOperator>] [-SQLFILTER=<SQLFilter>] [-
TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-UNIT_ROWCOUNT=<unitRowCount>] [-
TIMEOUT_WITH_ROWS_OK=<yes|no>] [-INCREMENT_DETECTION=<no|yes>] [-
INCREMENT_MODE=<M|P|I>] [-
INCREMENT_SEQUENCE_NAME=<incrementSequenceName>]
```

説明

表または表の集合内の一定数の行を待機します。これは、ビューなど、データが含まれている一定数のオブジェクトにも適用できます。

OdiWaitForData コマンドは、表または表の集合に一定数のレコードが入力されているかどうかをテストします。このテストは一定間隔で (-POLLINT)、目的の数の行が表の1つで検出されたか (-UNIT_ROWCOUNT)、すべての表に対して、総計で目的の数の行が検出されたか (-GLOBAL_ROWCOUNT)、またはタイムアウト (-TIMEOUT) に達したかの、いずれかの条件が満たされるまで繰り返されます。

検出された行の集合に対してフィルタを適用できます。フィルタは、明示的な SQL の where 句 (-SQLFILTER) またはフィールド値演算子句を決定する -RESUME_KEY_XXX パラメータ（あるいはその両方）によって指定されます。この2つの方式は累積的です (AND)。

行数は絶対数で判定することも（表内の行の総数に対して）、差異によって判定することも（格納された参照値と現在の行数値の差）可能です。

複数の表を扱う場合は次のようになります。

- -SQLFILTER および -RESUME_KEY_XXX パラメータは、関係するすべての表に適用されません。
- UNIT_ROWCOUNT パラメータは、特定の表の個々に対して想定する行数を決定します。 -GLOBAL_ROWCOUNT パラメータは、表の集合に対して累積された行数の総和を決定します。

関係している表が1つのみの場合、-UNIT_ROWCOUNT パラメータと-GLOBAL_ROWCOUNT パラメータは同等です。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	表が含まれている論理スキーマ。
-TABLE_NAME=<tableName>	必須	<p>チェックする表名、マスクまたは表名のリスト。このパラメータには次の3つの形式があります。</p> <ul style="list-style-type: none"> • 表名。 • 表名マスク: このマスクは、ポーリングする表を選択します。マスクはSQLのLIKE構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号は単一文字のワイルドカードです。 • 表名リスト: 表名のコンマ区切りのリスト。前述のように定義されたマスクが許可されます。
-OBJECT TYPE=<list of object types>	省略可	<p>チェックされるオブジェクトのタイプ。デフォルトでは、表のみがチェックされます。他のオブジェクトを考慮するには、オブジェクト・タイプのコンマ区切りリストを指定します。サポートされているオブジェクト・タイプは次のとおりです。</p> <ul style="list-style-type: none"> • T: 表 • V: ビュー
-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。

<pre>-SQLFILTER=<SQLFilter></pre>	省略可	<p>表に適用される明示的な SQL フィルタ。この文は、チェックされた表が含まれているテクノロジーで有効である必要があります。</p> <p>注意: この文は WHERE キーワードを含むことはできません。</p>
<pre>- RESUME KEY VARIABLE=<resumeKeyVariable> -RESUME_KEY_COL=<resumeKeyCol> [- RESUME KEY OPERATOR=<resumeKeyOperator>]</pre>	省略可	<p>RESUME_KEY_xxx パラメータを使用すると、ポーリングされた表で、検出された行の集合をフィルタリングできます。</p> <ul style="list-style-type: none"> • <key_column>: チェックされた表の列の名前。 • <operator>: : チェックされた表が含まれているテクノロジーで有効な比較演算子。このパラメータを省略すると、デフォルトで値「>」が使用されます。 • <variable name>: 値が事前に設定された変数の名前。変数名の先頭には、「:」（バインド）または「#」（代入）を接頭辞として付ける必要があります。変数の有効範囲は、Oracle Data Integrator 構文で明示的に記述してください。GLOBAL.<変数名>（グローバル変数の場合）、<プロジェクト・コード>.<変数名>（プロジェクト変数の場合）。
<pre>-TIMEOUT=<timeout></pre>	省略可	<p>データがポーリングされる時間の最大の長さ（単位はミリ秒）。この値がゼロの場合、タイムアウトは無制限です。デフォルトは0です。</p>

- POLLINT=<pollInt>	省略可	データのポーリングを待機する時間（単位はミリ秒）。デフォルトは2000です。
- UNIT_ROWCOUNT=<unitRowCount>	省略可	ポーリングされる表で、コマンドを終了させるために想定されている行数。デフォルトは1です。
- GLOBAL_ROWCOUNT=<globalRowCount>	省略可	コマンドを終了させるために、表の集合に対して累積的に想定されている行数。デフォルトは1です。
- INCREMENT_DETECTION=<no yes>	省略可	<p>コマンドが行数を判定するモードを定義します。行数を絶対数で判定するか（表内の行の総数に対して）、差異によって判定するかの（格納された参照値と現在の行数值の差）いずれかです。</p> <ul style="list-style-type: none"> • yes に設定すると、行カウントは差分モードで実行されます。表に追加された行の数が、格納された参照値と比較されます。参照値は INCREMENT_MODE パラメータに依存します。 • no に設定すると、カウントは絶対行数モードで実行されます。 <p>デフォルトは no です。</p>
- INCREMENT_MODE=<M P I>	省略可	<p>このパラメータは、連続する OdiWaitForData コールの間、参照値の永続モードを指定します。</p> <p>可能な値は次のとおりです。</p> <ul style="list-style-type: none"> • M: メモリー。参照値は非永続です。OdiWaitForData がコールされる場合、参照値は、ポーリングされる表の行数に等しい値を

<p>- INCREMENT SEQUENCE NAME=<incrementSequenceName></p>	<p>省略可</p>	<p>とります。 OdiWaitForData が終了すると、値は失われます。このモードでは、次のコールには、新しい参照値が設定されます。</p> <ul style="list-style-type: none"> • P: 永続。参照値は永続的です。 OdiWaitForData の開始時に増分順序から読み取られ、OdiWaitForData の終了時に増分順序に保存されます。増分順序が設定されていない場合（最初のコール時）、現在の表の行数が使用されます。 • I: 初期。参照値は初期化され、永続的です。参照値は、OdiWaitForData の開始時に、ポーリングされる表の行数に等しい値をとります。 OdiWaitForData の終了時に、永続モードとして増分順序に保存されます。 <p>デフォルトは M です。</p> <p>注意: マスクまたは表のリストがポーリングされる場合、永続または初期モードはサポートされません。</p>
		<p>このパラメータは、参照値の永続性のために使用される、自動割当てされた記憶域の名前を指定します。この増分順序はリポジトリに格納されます。この名前を指定しない場合、表の名前をとります。</p> <p>警告: この増分順序は Oracle Data Integrator の</p>

-TIMEOUT_WITH_ROWS_OK=<yes no>	省略可	順序ではなく、 OdiWaitForData へのコールの外部では順序として使用できません。
		このパラメータが Y に設定されている場合、1 つ以上の行が検出され、想定された行数が挿入される前にタイムアウトが起きると、リターン・コード 0 で API が終了します。それ以外の場合、エラーが生成されます。デフォルトは yes です。

例

ORA_WAITFORDATA スキーマで、DE1P1 表に、フィルタに合致している 200 件のレコードが含まれるのを待機します。

```
OdiWaitForData -LSHEMA=ORA_WAITFORDATA -TABLE_NAME=DE1P1 -
GLOBAL_ROWCOUNT=200 "-SQLFILTER=DATMAJ >
to_date('#MAX_DE1_DATMAJ_ORACLE_CHAR', 'DD/MM/YYYY HH24:MI:SS')
```

SQLSRV_SALES の CITY_SRC または CITY_TRG 表に新しいデータが表示されるのを、最大 4 時間待機します。

```
OdiWaitForData LSHEMA=SQLSRV_SALES TABLE_NAME=CITY% TIMEOUT=14400000 -
INCREMENT_DETECTION=yes
```

OdiWaitForLogData

使用方法

```
OdiWaitForLogData -LSHEMA=<LSchema> -TABLE_NAME=<tableName> -
CDC_SET_NAME=<cdcSetName> -SUBSCRIBER_NAME=<subscriberName> [-
CONTEXT=<context>] [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-UNIT_ROWCOUNT=<unitRowCount>] [-
OPTIMIZED_WAIT=<yes|no|AUTO>] [-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

説明

ジャーナル化された表またはそのリストで、一定数の変更が行われるのを待機します。

OdiWaitForLogData コマンドは、表または表のグループ上で行が変更されたかどうかを判定します。この変更の検出には、Oracle Data Integrator のチェンジ・データ・キャプチャ (CDC) をサンプル・モード (-TABLE_NAME パラメータを使用) または一貫モード (-CDC_SET_NAME パラメータを使用) が使用されます。テストは、-POLLINT ミリ秒間隔で、目的の数の行変更が表の 1 つで検出されたか (-UNIT_ROWCOUNT)、すべての表に対して、総計で目的の数の行変更が検出されたか (-GLOBAL_ROWCOUNT)、またはタイムアウト (-TIMEOUT) に達したかの、いずれかの条件が満たされるまで繰り返されます。

注意: このコマンドでは、ジャーナル化されたすべての操作（挿入、更新および削除）が考慮されます。

警告: このコマンドは単純モードまたは一貫モードでジャーナル化された表にのみ適しています。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	ジャーナル化された表が含まれている論理スキーマ。
-TABLE NAME=<tableName>	必須	<p>チェックする、ジャーナル化された表名、マスクまたはリスト。このパラメータには次の3つの形式があります。</p> <ul style="list-style-type: none"> 表名。 表名マスク: このマスクは、ポーリングする表を選択します。マスクはSQLのLIKE構文を使用して指定します。%記号は個数未指定の文字に代わるもので、「_」記号はワイルドカードです。 表名リスト: 表名のコンマ区切りのリスト。先に定義されたマスクは許可されません。 <p>注意: このオプションは、<u>単純</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、CDC_SET_NAME と併用できません。CDC_SET_NAME が設定されない場合は必須です。</p>
-CDC SET NAME=<cdcSetName>	必須	<p>チェックする CDC セットの名前。この CDC セット名は完全修飾のモデル・コードで、通常は PHYSICAL_SCHEMA_NAME.MODEL_CODE です。</p> <p>次に示すように、代入メソッド API コールを使用して、現在のコンテキストで取得できます。</p> <pre><%=odiRef.getObjectName("L", "model_code", "logical_schema", "D")%></pre> <p>注意: このオプションは、<u>一貫</u>モードでジャーナル化されたモデルの表でのみ機能します。</p> <p>警告: このパラメータは、TABLE_NAME と併用できません。TABLE_NAME が設定されない場合は必須です。</p>
-SUBSCRIBER NAME=<subscriberName>	必須	ジャーナル化情報を取得するために使用されるサブスクライバの名前。

-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト (CONTEXT CODE)。コンテキストを指定しない場合、実行コンテキストが使用されます。
-TIMEOUT=<timeout>	省略可	変更がポーリングされる時間の最大の長さ (単位はミリ秒)。この値がゼロの場合、タイムアウトは無制限です。デフォルトは0です。
-POLLINT=<pollInt>	省略可	ポーリングを待機する時間 (単位はミリ秒)。デフォルトは2000です。
-UNIT ROWCOUNT=<unitRowCount>	省略可	ポーリングされる表の1つで、コマンドを終了させるために想定されている変更数。デフォルトは1です。 注意: -UNIT_ROWCOUNT は、-CDC_SET_NAME とともに考慮されません。
-GLOBAL ROWCOUNT=<globalRowCount>	省略可	コマンドが終了するまでに、表または CDC セットで想定されている変更の総数。デフォルトは1です。
-OPTIMIZED WAIT=<yes no <u>AUTO</u> >	省略可	ジャーナルにアクセスするために使用された方法。 <ul style="list-style-type: none"> • yes: 最適化された方法。この方法は最近のバージョンのジャーナル化で機能します。非最適化モードより高速に実行されます。 • no: 最適化されていない方法。カウントはジャーナル化する表で実行されます。この方法は低パフォーマンスですが、旧バージョンのジャーナル化機能と互換性があります。 • AUTO: 複数の表がチェックされる場合、最適化された方法が使用されます。それ以外の場合は、最適化されていない方法が使用されます。 デフォルトは AUTO です。
-TIMEOUT WITH ROWS OK=< <u>yes</u> no>	省略可	このパラメータが yes に設定されている場合、1つ以上の行が検出され、事前定義された行数がポーリングされる前にタイムアウトが起きると、リターン・コード 0 で API が終了します。それ以外の場合、エラーが生成されます。デフォルトは yes です。

例

SALES_APPLICATION スキーマの CUSTOMERS 表で、SALES_SYNC サブスクリバのために記録されている行変更が 200 になるのを待機します。

```
OdiWaitForLogData -LSHEMA=SALES_APPLICATION -TABLE_NAME=CUSTOMERS -
GLOBAL_ROWCOUNT=200 -SUBSCRIBER_NAME=SALES_SYNC
```

OdiWaitForTable

使用方法

```
OdiWaitForTable -CONTEXT=<context> -LSHEMA=<LSchema> -
TABLE_NAME=<tableName> [-TIMEOUT=<timeout>] [-POLLINT=<pollInt>] [-
GLOBAL_ROWCOUNT=<globalRowCount>] [-TIMEOUT_WITH_ROWS_OK=<yes|no>]
```

説明

表が作成され、事前定義された数の行が入力されるのを待機します。

OdiWaitForTable コマンドは、指定された表が作成されたか、その表に一定数のレコードが入力されたかを定期的にテストします。テストは、表が存在し、指定した数の行を含んでいるか (-GLOBAL_ROWCOUNT)、タイムアウトに達したか (-TIMEOUT) の、いずれかの条件が満たされるまで -POLLINT ミリ秒間隔で繰り返されます。

パラメータ

パラメータ	必須かどうか	説明
-LSHEMA=<LSchema>	必須	表が検索される論理スキーマ。
-TABLE_NAME=<tableName>	必須	検索する表の名前。
-CONTEXT=<context>	省略可	論理スキーマが解決されるコンテキスト。コンテキストを指定しない場合、実行コンテキストが使用されます。
-TIMEOUT=<timeout>	省略可	表が検索される時間の最大の長さ（単位はミリ秒）。この値がゼロの場合、タイムアウトは無制限です。デフォルトは 0 です。
-POLLINT=<pollInt>	省略可	テストを待機する時間（単位はミリ秒）。デフォルトは 2000 です。
-GLOBAL_ROWCOUNT=<globalRowCount>	省略可	表で、コマンドを終了させるために想定されている行数。デフォルトは 1 です。
-TIMEOUT_WITH_ROWS_OK=<yes no>	省略可	このパラメータが yes に設定されている場合、1 つ以上の行が検出され、想定されたレコード数が検出される前にタイムアウトが起きると、リターン・コード 0 で API が終了します。それ以外の場合、エラーが生成されません。デフォルトは yes です。

例

ORA_WAITFORDATA スキーマで、DE1P1 表が存在し、少なくとも 1 件のレコードが含まれるのを待機します。

```
OdiWaitForTable -LSHEMA=ORA_WAITFORDATA -TABLE_NAME=DE1P1 -  
GLOBAL_ROWCOUNT=1
```

