

# **Oracle® Provider for OLE DB**

## **開発者ガイド**

### **19c for Microsoft Windows**

F21046-01(原本部品番号:E96615-01)

2019年5月

# タイトルおよび著作権情報

Oracle Provider for OLE DB開発者ガイド 19c for Microsoft Windows

F21046-01

Copyright © 1999, 2019, Oracle and/or its affiliates. All rights reserved.

原著者: Maitreyee Chaliha

原協力者: Janis Greenberg, Alex Keh, Eric Belden, Riaz Ahmed, Kiminari Akiyama, Christian Shay, Valarie Moore, Neeraj Gupta, Sinclair Hsu, Gopal Kirsur, Sunil Mushran, Rajendra Pingte, Helen Slattery, Vikhram Shetty, Sujith Somanathan, Mark Williams

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

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

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

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

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。お客様との間に適切な契約が定められている場合を除いて、オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。お客様との間に適切

な契約が定められている場合を除いて、オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

# 目次

- [表一覧](#)
- [タイトルおよび著作権情報](#)
- [はじめに](#)
  - [対象読者](#)
  - [ドキュメントのアクセシビリティについて](#)
  - [関連ドキュメント](#)
  - [表記規則](#)
- [『Oracle Provider for OLE DB開発者ガイド』のこのリリースでの変更点](#)
  - [12c リリース2 \(12.2.0.1\)内のOracle Provider for OLE DBでの変更点](#)
    - [新機能](#)
  - [ODAC 12c リリース4内のOracle Provider for OLE DBでの変更点](#)
    - [新機能](#)
  - [12c リリース1 \(12.1.0.2\)内のOracle Provider for OLE DBでの変更点](#)
    - [新機能](#)
- [1 Oracle Provider for OLE DBの概要](#)
  - [OLE DBの概要](#)
    - [OLE DBの設計](#)
  - [OraOLEDBの概要](#)
  - [システム要件](#)
  - [OraOLEDBのインストール](#)
  - [コンポーネントの動作保証](#)
  - [Oracle Database Exadata Express Cloud Serviceへの接続](#)
- [2 OraOLEDBの機能](#)
  - [OraOLEDBプロバイダ固有の機能](#)
    - [データ・ソース](#)
      - [OLE DBサービスとの互換性](#)
        - [ADOアプリケーションでのOLE DBサービス](#)
        - [C++/COMアプリケーションでのOLE DBサービス](#)
      - [Oracle Databaseへの接続](#)
      - [OraOLEDB固有の接続文字列属性](#)
      - [デフォルトの属性値](#)
      - [分散トランザクション](#)
      - [拡張フェイルオーバー機能](#)
      - [オペレーティング・システム認証](#)
      - [パスワードの期限](#)
      - [VCharNull](#)
      - [SPPrmDefVal](#)
    - [OraOLEDBセッション](#)
      - [トランザクション](#)
        - [ローカル・トランザクション](#)
        - [分散トランザクション](#)
  - [コマンド](#)

- [ストアド・プロシージャ](#)
- [コマンドの準備](#)
- [コマンド・パラメータ](#)
- [コマンドのためのOraOLEDBカスタム・プロパティ](#)
- [行セットを返すストアド・プロシージャおよびストアド・ファンクション](#)
- [複数の行セット](#)
- [文のキャッシュ](#)
- [メタデータのキャッシュ](#)
  - [接続文字列属性を使用したメタデータのキャッシュの有効化](#)
  - [レジストリを使用したメタデータのキャッシュの有効化](#)
- [コマンド・タイムアウトおよびCancelメソッド](#)
  - [レジストリを使用したCommandTimeoutの有効化](#)
- [行セット](#)
  - [行セットの作成](#)
  - [Updatability](#)
  - [Server Data on Insertプロパティ](#)
  - [IRowsetFind::FindNextによる行の検索](#)
  - [行セットのためのOraOLEDB固有の接続文字列属性](#)
  - [ADOプログラマのためのヒント](#)
  - [スキーマ行セット](#)
  - [日付書式](#)
  - [オブジェクト名の大/小文字](#)
  - [SQL Server Integration Servicesによる高速読込みの実行](#)
- [データ型](#)
  - [バイナリ・データ型](#)
  - [TIMESTAMPデータ型](#)
  - [期間データ型](#)
- [LOBサポート](#)
- [Unicodeサポート](#)
  - [Unicodeエンコーディングのタイプ](#)
  - [Oracle Unicodeサポートの機能](#)
  - [Unicodeサポートの設定](#)
    - [データベースの設定](#)
- [エラー](#)
- [OLEDB.NETデータ・プロバイダの互換性](#)
  - [接続文字列でのOLEDB.NET属性の使用](#)
  - [OraOLEDBカスタム・プロパティの使用](#)
  - [DataTable変更でのOracleの更新](#)
- [Visual BasicでのOraOLEDBの使用](#)
  - [Oracle Databaseの設定](#)
  - [Visual Basicプロジェクトの設定](#)
- [A プロバイダ固有の情報](#)
  - [行セットおよびパラメータのデータ型マッピング](#)
  - [サポートされるプロパティ](#)
    - [データ・ソースのプロパティ](#)

- [DataSourceInfoのプロパティ](#)
- [初期化のプロパティと認可のプロパティ](#)
- [セッションのプロパティ](#)
- [行セットのプロパティ](#)
  - [行セットのプロパティの含意](#)
- [サポートされるインタフェース](#)
  - [データ・ソース](#)
  - [セッション](#)
  - [コマンド](#)
  - [行セット](#)
  - [複数結果](#)
  - [トランザクション・オプション](#)
  - [カスタム・エラー・オブジェクト](#)
- [サポートされるメタデータ列](#)
- [OraOLEDBトレース](#)
  - [コールをトレースするためのレジストリ設定](#)
- [用語集](#)
- [索引](#)

## 表一覧

- [1-1 Oracle Provider for OLE DBファイル](#)
- [2-1 コマンドのためのカスタム・プロパティ](#)
- [2-2 Updatabilityプロパティに設定可能な値](#)
- [A-1 データ型マッピング](#)
- [A-2 DBPROPSET\\_DATASOURCEプロパティ](#)
- [A-3 DBPROPSET\\_DATASOURCEINFOプロパティ](#)
- [A-4 DBPROPSET\\_DBINITプロパティ](#)
- [A-5 DBPROPSET\\_SESSIONプロパティ](#)
- [A-6 DBPROPSET\\_ROWSETプロパティ](#)

# はじめに

Oracle Provider for OLE DB(OraOLEDB)は、オープンで標準的な方法でOracle Databaseにアクセスします。このマニュアルでは、OraOLEDBのプロバイダ固有の機能と特性について説明します。

このマニュアルでは、Windowsオペレーティング・システムに適用されるOracle Database for Windowsの機能について説明します。

## 対象読者

『Oracle Provider for OLE DB開発者ガイド』は、Oracle Provider for OLE DBを使用してOracle Databaseにアクセスするアプリケーションを開発するプログラマを対象としています。また、システム解析を行う方、プロジェクト・マネージャおよびデータベース・アプリケーションの開発に興味のある方にも有用です。

このマニュアルは、読者がOLE DBに精通しており、Microsoft C/C++、Visual BasicまたはActiveX Data Objects(ADO)を使用したアプリケーション・プログラムに対して、実践的な知識を持っていることを前提にしています。Component Object Model(COM)の概念に関する知識も必要です。

読者は、リレーショナル・データベース・システム内の情報にアクセスするためのSQLの使用方法についても理解している必要があります。

## ドキュメントのアクセシビリティについて

Oracleのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートを購入したオラクル社のお客様は、My Oracle Supportを介して電子的なサポートにアクセスできます。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

## 関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- [『Oracle Databaseインストール・ガイドfor Microsoft Windows』](#)
- [『Oracle Databaseリリース・ノート for Microsoft Windows』](#)
- [『Oracle Databaseプラットフォーム・ガイドfor Microsoft Windows』](#)
- [『Oracle Database概要』](#)
- [『Oracle Services for Microsoft Transaction Server開発者ガイド』](#)
- [『Oracle Database Net Services管理者ガイド』](#)
- [『Oracle Database新機能ガイド』](#)
- [『Oracle Databaseリファレンス』](#)
- [『Oracle Databaseグローバル化・サポート・ガイド』](#)
- Oracleエラー・メッセージの詳細は、[Oracle Databaseエラー・メッセージ](#)を参照してください。特定の範囲を表示した



ら、特定のメッセージを検索できます。インターネットに接続している場合、Oracleオンライン・ドキュメントのエラー・メッセージ検索機能を使用して、特定のエラー・メッセージを検索できます。

- スキーマの作成方法および使用方法については、『[Oracle Database サンプル・スキーマ](#)』を参照してください。このマニュアルに含まれる例の多くでは、Oracleのインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用しています。

リリース・ノート、インストール関連ドキュメント、ホワイト・ペーパーまたはその他の関連ドキュメントは、OTN (Oracle Technology Network)から、無償でダウンロードできます。

<http://www.oracle.com/technetwork/index.html>

このマニュアルを含むOracleマニュアルの最新バージョンは、次の場所を参照してください。

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

追加情報は、次の場所を参照してください。

<http://www.microsoft.com>

## 表記規則

このマニュアルでは次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック体	イタリックは、ドキュメントのタイトル、強調またはユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

# 『Oracle Provider for OLE DB開発者ガイド』のこのリリースでの変更点

ここでは、次の内容について説明します。

- [12cリリース2 \(12.2.0.1\)内のOracle Provider for OLE DBでの変更点](#)
- [ODAC 12cリリース4内のOracle Provider for OLE DBでの変更点](#)
- [12cリリース1 \(12.1.0.2\)内のOracle Provider for OLE DBでの変更点](#)

## 12c リリース2 (12.2.0.1)内のOracle Provider for OLE DBでの変更点

Oracle Database 12cリリース2 (12.2.0.1)内の*Oracle Provider for OLE DB*開発者ガイドでは、次の点に変更されました。

### 新機能

このリリースの新機能は次のとおりです。

- 長いスキーマ識別子

Oracle Provider for OLE DBは、128文字までの長さのスキーマ・オブジェクト名(表、列、ビュー、ストアド・プロシージャおよびファンクションなど)をサポートするようになりました。

長いスキーマ・オブジェクト識別子名を使用して、この機能では、データベース・デプロイメントの柔軟性が向上し、Oracleへのデータベース移行が容易になります。

## ODAC 12c リリース4内のOracle Provider for OLE DBでの変更点

ODAC 12cリリース4内の*Oracle Provider for OLE DB*開発者ガイドでは、次の点に変更されました。

### 新機能

このリリースの新機能は次のとおりです。

- トレースの拡張機能

Oracle Provider for OLE DBに新しいトレース機能が含まれました。トレースを、ファイル名ではなく特定のディレクトリに出力できるようになりました。デフォルトのディレクトリはWindowsの一時ファイル・ディレクトリです。

詳細は、"[OraOLEDBトレース](#)を参照してください。

## 12c リリース1 (12.1.0.2)内のOracle Provider for OLE DBでの変更点

Oracle Database 12cリリース1 (12.1.0.2)の『*Oracle Provider for OLE DB*開発者ガイド』では、次の点に変更されました。

## 新機能

このリリースの新機能は次のとおりです。

- IRowsetFastLoadとSQL Server Integration Servicesの高速読み込み

Oracle Provider for OLE DBがIRowsetFastLoadインターフェースをサポートするようになり、Oracle Databaseで高速メモリーベースの一括コピー操作が可能になります。このインターフェースは、SQL Server Integration Servicesの高速読み込み機能とも連携し、プロバイダを使用してOracle Databaseへの大量データの高速な取込みが可能です。

詳細は、[SQL Server Integration Servicesによる高速読み込みの実行](#)を参照してください。

# 1 Oracle Provider for OLE DBの概要

次の各項では、Oracle Provider for OLE DB (OraOLEDB)の概要を示します。

- [OLE DBの概要](#)
- [OraOLEDBの概要](#)
- [システム要件](#)
- [OraOLEDBのインストール](#)
- [コンポーネントの認証](#)
- [Oracle Database Exadata Express Cloud Serviceへの接続](#)

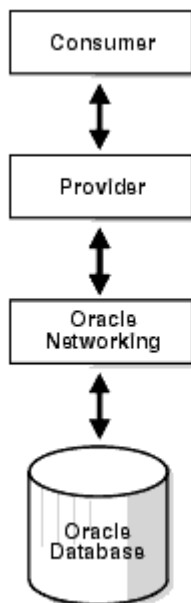
## OLE DBの概要

OLE DBは、オープンで標準的なデータ・アクセス方法であり、様々なタイプのデータに対するアクセスおよび操作において、一連の[Component Object Model\(COM\)](#)インタフェースを使用します。このインタフェースは様々なデータベース開発元から提供されています。

## OLE DBの設計

OLE DBの設計の中心となるのは、[コンシューマ](#)および[プロバイダ](#)の概念です。[図1-1](#)にOLE DBシステムを示します。コンシューマは旧来のクライアントを表します。プロバイダはデータを表形式で格納し、コンシューマにデータを返します。

図1-1 OLE DBの流れ



### OLE DBデータ・プロバイダ

OLE DBデータ・プロバイダは、データソースから**コンシューマ**にデータを転送する一連の**COM**コンポーネントです。OLE DBプロバイダは、コンシューマからのコールに応答して、そのデータを表形式で格納します。プロバイダは、単純にも複雑にもできます。**プロバイダ**は、表を返すこと、コンシューマがその表形式を決定すること、そのデータに対して操作を実行することができます。

各プロバイダはコンシューマからの要求を処理する標準COMセットを実装します。プロバイダは、追加機能を提供するCOMインタフェースをオプションとして実装できます。

標準インタフェースにより、どのプロバイダからのデータに対しても、すべてのOLE DBコンシューマからアクセスできます。COMコン

ポーネントにより、コンシューマはC++、Visual BasicおよびJavaなど、COMをサポートする任意のプログラム言語でアクセスできます。

OLE DBデータ・コンシューマ

OLE DBデータ・コンシューマは、プロバイダのOLE DBインターフェイスを利用して様々なデータにアクセスするためのアプリケーションまたはツールです。

## OraOLEDBの概要

Oracle Provider for OLE DB(OraOLEDB)は、高いパフォーマンスを提供し、OLE DBコンシューマがOracleデータへ効率的にアクセスできるようにするOLE DBデータ・プロバイダです。

通常、この開発者ガイドでは、OLE DBまたはADOを介してOraOLEDBを使用していることを想定しています。

.NETフレームワークの出現により、OraOLEDBでのOLEDB.NETデータ・プロバイダの使用がサポートされるようになりました。適切な接続属性を設定することで、OLEDB.NETデータ・プロバイダでOraOLEDBを利用してOracle Databaseにアクセスできます。

### 関連項目:

OLEDB.NETデータ・プロバイダのサポートの詳細は、[OLEDB.NETデータ・プロバイダの互換性を参照](#)

## システム要件

Oracle Provider for OLE DBを使用するシステムには次の要件が必要です。

- Windowsオペレーティング・システム:
  - 64ビット: Windows 8.1 (ProおよびEnterpriseの各エディション)、Windows 10 x64 (Pro、Enterprise、Educationの各エディション)、Windows Server 2012 R2 x64 (Standard、Datacenter、Essentials、Foundationの各エディション)、Windows Server 2016 x64 (Standard、Datacenter、Essentialsの各エディション)またはWindows Server 2019 x64 (Standard、Datacenter、Essentialsの各エディション)。

Oracleでは、これらのWindows x64オペレーティング・システムで、32-bit Oracle Provider for OLE DBおよび64-bit Oracle Provider for OLE DBがサポートされます。

- Oracle Database(Oracle 11.2以上)へのアクセス
- Oracle ClientおよびOracle Net Services(Oracle Provider for OLE DBのインストールに付属)。
- Microsoft Data Access Components (MDAC) 2.1以上で提供される再配布可能ファイルは、プロバイダが必要です。これらのファイルは、次のMicrosoft社のWebサイトから入手できます。

<http://msdn.microsoft.com/en-us/data/aa937730.aspx>

- Microsoftトランザクション・サーバー用のOracleサービスMicrosoft Transaction Server(MTS)またはCOM+を使用するコンシューマに必要です。

# OraOLEDBインストール

Oracle Provider for OLE DBは、Oracleのインストールに含まれています。これには、この製品を使用して実際の問題を解決する方法を例示する機能およびデモが含まれています。

インストール・プロセス中に、次のファイルがシステムにインストールされます。一部のファイル名にある *ver* は、リリース・バージョンを示しています。

表1-1 Oracle Provider for OLE DBファイル

ファイル	説明	場所
OraOLEDB <i>ver</i> .dll	Oracle Provider for OLE DB	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDBrfc <i>ver</i> .dll	Oracle 行セット・ファイル・キャッシュ・マネージャ	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDBrmc <i>ver</i> .dll	Oracle 行セット・メモリー・キャッシュ・マネージャ	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDBrst <i>ver</i> .dll	Oracle 行セット	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDBgmr <i>ver</i> .dll	Oracle ODBC SQL パーサー	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDB <i>lang</i> <i>ver</i> .dll	言語固有のリソース DLL ( <i>lang</i> は必要な言語)	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDBpus <i>ver</i> .dll	プロパティの説明	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDButil <i>ver</i> .dll	OraOLEDB ユーティリティ DLL	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDB <i>ver</i> .tlb	OraOLEDB タイプ・ライブラリ	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \bin
OraOLEDB.h	OraOLEDB ヘッダー・ファイル	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \oledb\include
OraOLEDB <i>ver</i> .lib	OraOLEDB ライブラリ・ファイル	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \oledb\lib
OraOLEDB <i>lang</i> .msg	言語固有のメッセージ・ファイル ( <i>lang</i> は必要な言語)	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \oledb\msg
Readme	リリース・ノート	<i>ORACLE_BASE</i> \ <i>ORACLE_HOME</i> \oledb

## 関連項目:

インストール手順は、[Oracle Databaseインストール・ガイドfor Microsoft Windows](#)を参照

## コンポーネントの動作保証

オラクル社では、各種プラットフォーム上のコンポーネントに関するサポート情報、互換性のあるクライアントとデータベース・バージョンの一覧、ならびにパッチと対処方法の情報を提供しています。

最新の動作保証情報は、次の場所にあるMy Oracle Support(以前のOracleMetaLink)を参照してください。

<https://support.oracle.com>

## Oracle Database Exadata Express Cloud Serviceへの接続

Oracle Provider for OLE DBでは、Oracle Database Exadata Express Cloud Serviceへの接続がサポートされています。

既知の制限事項

Oracle Provider for OLE DBでは、Oracle Database Exadata Express Cloud Serviceへの接続時には次の機能はサポートされていません。

- ユーザー名およびパスワード以外の認証
- IRowsetFastLoadインタフェース
- クライアントの結果キャッシュ
- データ型:
  - BFILE
  - サイズ制限を32 KBに増やしたVARCHAR2



注意:

4 KB までのサイズの VARCHAR2 はサポートされます

- 分散トランザクション

## 2 OraOLEDBの機能

次の各項では、Oracle Provider for OLE DB (OraOLEDB)のコンポーネントについて、およびこれらのコンポーネントを使用してOLE DBコンシューマ・アプリケーションを開発する方法について説明します。

- [OraOLEDBプロバイダ固有の機能](#)
- [Visual BasicでのOraOLEDBの使用](#)

### OraOLEDBプロバイダ固有の機能

次の各項で、OraOLEDBのプロバイダ固有の機能について説明します。

- [データ・ソース](#)
- [OraOLEDBセッション](#)
- [コマンド](#)
- [行セット](#)
- [データ型](#)
- [LOBサポート](#)
- [Unicodeサポート](#)
- [エラー](#)
- [OLEDB.NETデータ・プロバイダの互換性](#)

この他のプロバイダ固有の情報は、[プロバイダ固有の情報](#)を参照してください。

### データ・ソース

OraOLEDBのデータソース・オブジェクトは、Oracle Databaseへの最初の接続を確立します。初期接続を確立するために、コンシューマはCoCreateInstanceファンクションを使用してデータソース・オブジェクトのインスタンスを作成する必要があります。このファンクションでは、プロバイダに関する重要な情報(プロバイダのクラスIDと実行可能なコンテキスト)が必要です。OraOLEDBのクラスIDは、CLSIDs\_OraOLEDBです。

OraOLEDBはインプロセス・サーバーです。CoCreateInstanceをコールするには、CLSCTX\_INPROC\_SERVERマクロを使用します。次に例を示します。

```
// create an instance of OraOLEDB data source object and
// obtain the IDBInitialize interface
hr = CoCreateInstance(CLSID_OraOLEDB, NULL,
                    CLSCTX_INPROC_SERVER, IID_IDBInitialize,
                    (void**)&pIDBInitialize);
```

上のコード・スニペットでは、データソース・オブジェクトをインスタンス化しても、OLEDBサービスは有効になりません。OLEDBサービスを有効にするには、この後の[OLE DBサービスとの互換性](#)を参照してください。



注意:



OraOLEDB は、永続データソース・オブジェクトをサポートしません。

データソース・オブジェクトのインスタンスが正常に作成されると、コンシューマ・アプリケーションはそのデータソースを初期化してセッションを作成できます。

OraOLEDBは、Oracle Databaseの各リリースとの接続をサポートします。特定のデータベースと接続するために、コンシューマはDBPROPSET\_DBINITプロパティ・セットの次のプロパティを設定する必要があります。

- DBPROP\_AUTH\_USERNAMEにユーザーID(scottなど)を設定します。
- DBPROP\_AUTH\_PASSWORDにパスワード(tigerなど)を設定します。
- DBPROP\_INIT\_DATASOURCEにネット・サービス名(myOraDbなど)を設定します。

また、コンシューマは、DBPROP\_INIT\_PROMPTにDBPROMPT\_PROMPTを使用できます。これを使用すると、**プロバイダ**がログオン・ボックスを表示するので、ユーザーは接続情報を入力できるようになります。

DBPROMPT\_NOPROMPTを使用すると、ログオン・ボックスの表示ができなくなります。この場合は、ログオン情報が不完全なために、プロバイダはログオン・エラーを返します。ただし、このプロパティにDBPROMPT\_COMPLETEまたはDBPROMPT\_COMPLETEREQUIREDが設定されている場合は、ログオン情報が不完全な場合にのみログオン・ボックスが表示されます。

## OLE DBサービスとの互換性

OraOLEDBは、OLE DB Version 2.0以上で使用可能なOLE DBサービスと互換性があります。OLE DBサービスには、自動トランザクション登録、クライアント・カーソル・エンジン(CCE)、接続およびセッション・プールといった、アプリケーションのパフォーマンス向上に役立つ有用なサービスが含まれています。

OLE DBサービスは、C++/COMまたはADOを介してOraOLEDBで使用できます。

デフォルトでは、OraOLEDBのCLSIDの下にあるOraOLEDBのOLEDB\_SERVICESレジストリ・エントリが0xffffffff(-1)に設定され、すべてのサービスが有効になります。DBPROP\_INIT\_OLEDBSERVICESプロパティの設定を使用して、プログラムによって特定のOLE DBサービスを無効または有効にすることもできます。

### 関連項目:

OLE DBサービスの詳細、および特定のサービスを有効または無効にする方法の詳細は、[http://msdn.microsoft.com/en-us/library/ms724518\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms724518(VS.85).aspx)を参照してください。

## ADOアプリケーションでのOLE DBサービス

ADOでは、OLE DBサービスが自動的に有効になります。このため、ADOアプリケーションでOLEDBサービスを使用するために特別なコードは必要ありません。

## C++/COMアプリケーションでのOLE DBサービス

C++/COMアプリケーションでOLE DBサービスを使用するためには、いくつかの追加ステップが必要です。

次のコード・スニペットは、C++/COMアプリケーションでOLE DBサービスを有効にする1つの方法を示しています。このコードは、CoCreateInstance()を使用してCLSID\_MSDAINITIALIZEクラスのインスタンスを作成し、そのオブジェクトからIDataInitializeインタフェースを取得した後、そのインタフェースを介してOLE DBデータソース・オブジェクトを作成するOLE DBコンシューマを示しています。

```

// Instantiate the CLSID_MSDAINITIALIZE class and request for the
// IID_IDataInitialize interface from it
hr = CoCreateInstance(CLSID_MSDAINITIALIZE, NULL, CLSCTX_INPROC_SERVER,
    IID_IDataInitialize, (void**)&pIDataInitialize);

// Set properties, datasource name, userid, and password, etc.
...

// Create an OLEDB data source object using the interface obtained from the
// CLSID_MSDAINITIALIZE class.
hr = pIDataInitialize->CreateDBInstance(CLSID_OraOLEDB, NULL,
    CLSCTX_INPROC_SERVER, NULL, IID_IDBInitialize, (IUnknown**)&pIDBInitialize);
...

// If connection/session pooling was enabled, pIDBInitialize->Release()
// releases the connection/session back to the pool.
// pIDataInitialize->Release() should not be called until the application no
// longer need to use connection/session pooling and the rest of
// the OLE DB Services that were enabled for the application.
//
pIDBInitialize->Release();

```

## Oracle Databaseへの接続

OraOLEDBを使用してOracle Databaseに接続するには、OLE DB接続文字列を次のように設定する必要があります。

```
"Provider=OraOLEDB.Oracle;User ID=user;Password=pwd;Data Source=constr;"
```

リモート・データベースに接続するには、Data Sourceに、適切なネット・サービス名(tnsnames.oraファイル内にある別名)を設定する必要があります。

### 関連項目:

[Oracle Database Net Services管理者ガイド](#)

## OraOLEDB固有の接続文字列属性

OraOLEDBによって、プロバイダ固有の接続文字列属性が提供されますが、これらはプロバイダおよびユーザーIDの設定方法と同じ方法で設定されます。プロバイダ固有の接続文字列属性は、次のとおりです。

- CacheType - クライアント上の行セット・データの格納に使用する、キャッシュのタイプを指定します。[行セットのためのOraOLEDB固有の接続文字列属性](#)を参照してください。
- ChunkSize - プロバイダのキャッシュに格納される、LONG型またはLONG RAW型の列データのサイズを指定します。[行セットのためのOraOLEDB固有の接続文字列属性](#)を参照してください。
- DistribTX - 分散トランザクションの登録機能を有効または無効にします。「[分散トランザクション](#)」を参照してください。
- FetchSize - 取り出す配列のサイズを行数で指定します。[行セットのためのOraOLEDB固有の接続文字列属性](#)を参照してください。
- OLEDB.NET - OLEDB.NETデータ・プロバイダとの互換性を有効または無効にします。[OLEDB.NETデータ・プロバイダの互換性](#)を参照してください。
- OSAuthent - Oracle Databaseに接続するときに、オペレーティング・システム認証が使用されるかどうかを指定しま

す。[オペレーティング・システム認証](#)を参照してください。

- PLSQLRSet - [PL/SQL](#)ストアード・プロシージャから行セットを返す機能を有効または無効にします。[コマンドのためのOraOLEDBカスタム・プロパティ](#)を参照してください。
- PwdChgDlg - パスワードが期限切れになったときに、パスワード変更ダイアログ・ボックスを表示可能または不可にします。「[パスワードの期限](#)」を参照してください。
- UseSessionFormat - セッションの間、NLSセッションのデフォルトの書式を使用するか、OraOLEDBでこれらの書式の一部を上書きするかを指定します。有効値は0(FALSE)と1(TRUE)です。デフォルト値はFALSEで、OraOLEDBはNLSセッションのデフォルトの書式の一部を上書きできます。値がTRUEの場合、OraOLEDBはNLSセッションのデフォルトの書式を使用します。

この接続属性は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下に表示されないので注意してください。

- VCharNull - ストアド・プロシージャでVARCHAR2 OUTパラメータのヌル文字での終了を有効または無効にします。
- SPPrmDefVal - アプリケーションでストアード・プロシージャのパラメータ値が指定されていない場合に、デフォルト値とNULL値のどちらを使用するかを指定します。
- NDataType: コマンドにバインドされているパラメータのいずれかがNデータ型かどうかを指定しますが、これには、NCHAR、NVARCHAR2またはNCLOBが含まれます。「[NDataType](#)」を参照してください。

この接続属性は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下に表示されないので注意してください。

- SPPrmsLOB: ストアド・プロシージャにバインドされている1つ以上のパラメータがLOBデータ型かどうかを指定しますが、これには、CLOB、BLOBまたはNCLOBが含まれます。「[SPPrmsLOB](#)」を参照してください。

この接続属性は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下に表示されないので注意してください。

- StmtCacheSize - キャッシュできる文の最大数を指定します。「[文のキャッシュ](#)」を参照してください。
- MetaDataCacheSize - メタデータをキャッシュできるSELECT文の最大数を指定します。「[メタデータのキャッシュ](#)」を参照してください。
- DeferUpdChk - 読み取り専用の非接続行セットの更新がサポートされるよう、更新可能性チェックを遅らせるかどうかを指定します。[行セットのためのOraOLEDB固有の接続文字列属性](#)のDeferUpdChkを参照してください。
- DBNotifications - 高可用性イベントにサブスクライブするかどうかを指定します。[拡張フェイルオーバー機能](#)を参照してください。
- DBNotificationPort - データベース通知のリスニング用に開かれるポート番号を指定します。[拡張フェイルオーバー機能](#)を参照してください。

## デフォルトの属性値

この属性のデフォルト値は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下にあります。このKEY\_HOMENAMEはOracleホームです。

レジストリにあるすべてのデフォルト値は、プロバイダがメモリーにロードされたときに、OraOLEDBによってそのレジストリから読み込まれます。接続時にOracle固有の接続文字列属性が与えられない場合は、デフォルトのレジストリ値を使用します。ただし、属性が与えられた場合は、新しい値によってデフォルトのレジストリ値が上書きされます。

DBPROPSET\_DBINITプロパティ・セットの中のDBPROP\_INIT\_PROVIDERSTRINGプロパティを設定した場合も、これらの属性を設

定できます。次に例を示します。

```
"FetchSize=100;CacheType=Memory;OSAuthent=0;PLSQLRSet=1;StmtCacheSize=10;"
```

## 分散トランザクション

DistribTX属性は、セッションを分散トランザクションに登録できるかどうかを指定します。有効値は0(使用不可)と1(使用可能)です。デフォルトは1で、セッションを分散トランザクションに登録できることを示します。

Microsoft Distributed Transaction Coordinatorを使用しているアプリケーションでは、DistribTXをデフォルトの1に設定する必要があります。

分散トランザクションに登録できるセッションでは、Oracle Databaseのダイレクト・パス・ロードおよびパラレルDML機能を使用する文を実行できません。このような文は従来型パスのシリアル文として実行されます。

## 拡張フェイルオーバー機能

この機能により、フェイルオーバー機能が強化されます。

これらの接続文字列属性のサポートにより、フェイルオーバー機能が強化されました。

- DBNotifications

DBNotifications属性は、高可用性イベントにサブスクライブするかどうかを指定します。有効値は0(FALSE)と1(TRUE)です。デフォルトはFALSEで、OraOLEDBが高可用性イベントにサブスクライブされないことを示します。この属性が接続時に指定されない場合、デフォルトのレジストリ値が使用されます。

- DBNotificationPort

DBNotificationPort属性は、データベース通知のリスニングに使用されるポート番号を指定します。有効値は符号なしの整数です。

DBNotificationPortは、接続文字列属性またはレジストリ・エントリのいずれかを使用してDBNotifications属性がTRUEに設定されている場合にのみ有効です。DBNotificationPort属性のデフォルトは0で、これはOraOLEDBが有効なポートをランダムに開くことを意味します。OraOLEDBはポート番号を検証しないため、有効なポート番号を指定するのはアプリケーションの役割です。

レジストリ・エントリを使用したフェイルオーバー機能の有効化

- DBNotifications

DBNotificationsレジストリ・エントリは、高可用性イベントにサブスクライブするかどうかを指定します。有効値は0(FALSE)と1(TRUE)です。デフォルト値はFALSEで、OraOLEDBはサブスクライブされません。このレジストリ・エントリ値は、DBNotifications接続文字列属性が設定されていない場合に使用されます。この値は、  
¥¥HKEY\_LOCAL\_MACHINE¥SOFTWARE¥ORACLE¥KEY\_HOMENAME¥OLEDBレジストリ・キーの下にあります。

## オペレーティング・システム認証

OSAuthent属性は、Oracle Databaseに接続するときに、オペレーティング・システム認証が使用されるかどうかを指定します。有効値は0(使用不可)と1(使用可能)です。デフォルトは0で、オペレーティング・システム認証が使用されないことを示します。

オペレーティング・システム認証という機能によって、Oracleはユーザーを認証するためにオペレーティング・システムのセキュリティ・メカニズムを使用します。

Windowsクライアントがオペレーティング・システム認証のために正しく設定されると、OraOLEDBクライアントは、次のいずれかを設定することでこの機能を使用できるようになります。

- DBPROP\_AUTH\_USERNAMEを/に設定

- DBPROP\_INIT\_PROVIDERSTRINGをOSAuthent=1;に設定
- レジストリのOSAuthentを1に設定

#### 関連項目:

この件とWindowsクライアントでの設定方法の詳細は、[『Oracle Databaseプラットフォーム・ガイドfor Microsoft Windows』](#)を参照してください。

## パスワードの期限

Oracleには、パスワードの有効期限機能があり、データベース管理者はこれを使用して、ユーザーがパスワードを定期的に変更するように強制できます。PwdChgDlg属性は、パスワードの期限切れが原因でログオンできないときに、パスワード変更ダイアログ・ボックスの表示を有効化または無効化します。有効にした場合、プロバイダは、パスワードを変更するためのダイアログ・ボックスを表示します。無効にした場合は、ログオンが失敗し、エラー・メッセージが表示されます。有効値は0 (無効)および1 (有効)です。デフォルトは1(有効)です。

#### 関連項目:

パスワード失効機能の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。

例: ADOによるOracle Databaseへの接続

次の例で、OraOLEDBおよびADOによるOracle Databaseへの接続方法を示します。

#### 注意:



Data Source、User ID および Password が Open メソッドで指定されている場合、ADO はそれらのConnectionString 属性を無視します。

ConnectionStringを使用して接続

```
Dim con As New ADODB.Connection
con.ConnectionString = "Provider=OraOLEDB.Oracle;Data Source=MyOraDb;" & _
    "User ID=scott;Password=tiger;"
con.Open
```

ConnectionStringを使用しないで接続

```
Dim con As New ADODB.Connection
con.Provider = "OraOLEDB.Oracle"
con.Open "MyOraDb", "scott", "tiger"
```

接続してプロバイダ固有の属性を設定

```
Dim con As New ADODB.Connection
con.Provider = "OraOLEDB.Oracle"
con.ConnectionString = "FetchSize=200;CacheType=Memory;" & _
```

```
"OSAuthent=0;PLSQLRSet=1;Data Source=MyOraDb;" & _  
"User ID=scott;Password=tiger;"
```

```
con. Open
```

ユーザーIDに「/」を設定するオペレーティング・システム認証接続

```
Dim con As New ADODB.Connection  
con.Provider = "OraOLEDB.Oracle"  
con.Open "MyOraDb", "/", ""
```

OSAuthentを使用するオペレーティング・システム認証接続

```
Dim con As New ADODB.Connection  
con.Provider = "OraOLEDB.Oracle"  
con.ConnectionString = "Data Source=MyOraDb:OSAuthent=1;"  
con.Open
```

## VCharNull

VCharNull属性は、ストアド・プロシージャで、VARCHAR2 OUTパラメータの文字列での終了を有効または無効にします。有効値は0(使用不可)と1(使用可能)です。デフォルトは1で、VARCHAR2 OUTパラメータが文字列で終了することを示します。値0は、VARCHAR2 OUTパラメータに空白が埋め込まれることを示します。

この属性のデフォルト値は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下にあり、この`HOMENAME`はOracleホームです。この属性が接続時に指定されない場合、デフォルトのレジストリ値が使用されます。

この接続属性を有効にした場合、アプリケーションでストアド・プロシージャのINパラメータとIN OUT CHARパラメータをWHERE句で使用するときは、パラメータに明示的に空白を埋め込む必要があります。

## SPPrmDefVal

SPPrmDefVal属性は、アプリケーションでストアド・プロシージャのパラメータ値が指定されていない場合に、デフォルト値とNULL値のどちらを使用するかを指定します。有効値は0(FALSE)と1(TRUE)です。デフォルトはFALSEで、OraOLEDBでNULL値を渡すことができます。値がTRUEであれば、OraOLEDBでデフォルト値が使用されます。

この属性のデフォルト値は、`¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOMENAME¥OLEDB`レジストリ・キーの下にあります。この属性が接続時に指定されない場合、デフォルトのレジストリ値が使用されます。

## OraOLEDBセッション

OraOLEDBセッション・オブジェクトは、Oracle Databaseへの単一の接続を表します。セッション・オブジェクトは、データにアクセスして操作できるようにするインタフェースを公開します。

初期化データソースに作成される最初のセッションは、`IDBInitialize::Initialize()`で確立された初期接続を継承します。作成される後続のセッションは、データソース・プロパティで指定された特定のOracle Databaseへの、それぞれ個別の接続を確立します。

各セッション・オブジェクトもまた、データソースのトランザクション領域を定義します。特定のセッション・オブジェクトから作成されるすべてのコマンド・オブジェクトおよび行セット・オブジェクトは、そのセッションのトランザクションの一部になります。

そのセッション・オブジェクトへのすべての参照が解放されると、セッション・オブジェクトはメモリーから削除され、接続は切断されます。

## トランザクション

OraOLEDBは、トランザクション操作の明示的なコミットと異常終了を行う、ローカル・トランザクションおよび分散トランザクシ

ンをサポートします。

OraOLEDBでは、ネストしたトランザクションはサポートしません。さらに、現在分散トランザクションに登録されているセッションでは、ローカル・トランザクションを起動できません。逆も同様で、現在ローカル・トランザクションに登録されているセッションでは、分散トランザクションを起動できません。

## ローカル・トランザクション

OraOLEDBは、明示的トランザクションのためのITransactionLocalインタフェースをサポートします。デフォルトでは、OraOLEDBは自動コミット・モードで、データベース上で終了した各処理は自動的にまたは暗黙的にコミットされます。ITransactionLocalインタフェースを使用すると、コンシューマは明示的に特定のセッションに対するトランザクションを起動できるようになり、コンシューマが明示的に各処理をコミットまたは異常終了できるようになります。

OraOLEDBは、コミット読み取り(カーソル・スタビリティ)分離レベルをサポートします。このレベルでは、別のトランザクションによる変更は、そのトランザクションがコミットされるまで参照できなくなります。

## 分散トランザクション

OraOLEDBコンシューマを、Microsoft Transaction Server(またはCOM+)トランザクションに登録する、またはMicrosoft分散トランザクション・コーディネータ(MS DTC)によって調整された分散トランザクションに登録するには、Oracle Services for Microsoft Transaction Server(MTS)リリース10.2以上をインストールする必要があります。

ITransactionJoin::JoinTransaction()がコールされると、OraOLEDBはIsoLevel、IsoFlagsおよびpOtherOptionsの各パラメータを無視します。コンシューマがITransactionDispenser::BeginTransaction()メソッドをコールしてMS DTCからトランザクション・オブジェクトを取得する場合は、これらのオプションが必要です。

ただし、IsoFlagsが0でない場合は、XACT\_E\_NOISORETAINが返されます。

## 関連項目:

Oracle Services for MTSでの設定および構成情報の詳細は、[『Oracle Services for Microsoft Transaction Server開発者ガイドfor Microsoft Windows』](#)を参照してください。

## コマンド

OraOLEDBは、Oracle DatabaseによってサポートされるANSI SQLと、ODBC SQL構文をサポートします。

## ストアド・プロシージャ

コマンドを使用してOracle **PL/SQL** [ストアド・プロシージャ](#)を実行する場合、コマンドで、次のようにOracle固有の構文またはODBCプロシージャ・コールのエスケープ・シーケンスを使用します。

- Oracle固有の構文: BEGIN credit\_account(123, 40); END;
- ODBC構文: {CALL credit\_account(123, 40)}

## コマンドの準備

OraOLEDBは、SQLのSELECT文の場合にのみ、メタデータの妥当性をチェックして取り出します。

## コマンド・パラメータ

Oracle ANSI SQLを使用する場合は、コマンド内のパラメータの前にコロンを付けます。ODBC SQLの場合は、パラメータに

疑問符(?)を付けます。

OraOLEDBは、PL/SQLのストアード・プロシージャおよびストアード・ファンクションの入力、出力および入出力パラメータをサポートします。OraOLEDBは、SQL文の入力パラメータをサポートします。

### 注意:



OraOLEDB は、位置バインドのみをサポートします。

## コマンドのためのOraOLEDBカスタム・プロパティ

コマンドのためのOraOLEDBカスタム・プロパティは、カスタム・プロパティ・セットORAPROPSET\_COMMANDSの下にグループ化されません。次のプロパティを提供します。

表2-1 コマンドのためのカスタム・プロパティ

Visual Basicユーザーの場合	C++ユーザーの場合
PLSQLRSet	ORAPROP_PLSQLRSet
NDatatype	ORAPROP_NDatatype
SPPrmsLOB	ORAPROP_SPPrmsLOB
AddToStmtCache	ORAPROP_AddToStmtCache

### PLSQLRSet

このプロパティは、PLSQLRSet接続文字列属性と同様です。

このプロパティは、OraOLEDBがPL/SQLストアード・プロシージャから行セットを返す必要があるかどうかを指定します。コンシューマから提供されたストアード・プロシージャが行セットを返す場合は、PLSQLRSetをTRUE(使用可能)に設定する必要があります。このプロパティは、コマンドが実行された後はFALSEに設定する必要があります。デフォルトでは、このプロパティはFALSE(使用不可)に設定されます。

プロパティは、セッションではなくコマンド・オブジェクトで設定できるので、コンシューマは、属性よりプロパティを優先的に使用する必要があります。コマンド・オブジェクトで設定することにより、コンシューマは、行セットを返すストアード・プロシージャを実行するコマンド・オブジェクトに対してのみプロパティを設定することができます。属性の場合、コンシューマは、ADOアプリケーションによって実行される多くのストアード・プロシージャの1つのみが行セットを返した場合でも、属性を設定する必要があります。このプロパティを使用すると、以前に属性を使用していたアプリケーションのパフォーマンスを向上させることができます。

### 例: カスタム・プロパティPLSQLRSetの設定

```
Dim objRes As NEW ADODB.Recordset
Dim objCon As NEW ADODB.Connection
Dim objCmd As NEW ADODB.Command
....
objCmd.ActiveConnection = objCon
objCmd.CommandType = adCmdText

' Enabling the PLSQLRSet property indicates to the provider
```



```

' that the command returns one or more rowsets
objCmd.Properties("PLSQLRSet") = TRUE

' Assume Employees.GetEmpRecords() has a REF CURSOR as
' one of the arguments
objCmd.CommandText = "{ CALL Employees.GetEmpRecords(?, ?) }"

' Execute the SQL
set objRes = objCmd.Execute

' It is a good idea to disable the property after execute as the
' same command object may be used for a different SQL statement
objCmd.Properties("PLSQLRSet") = FALSE

```

## NDatatype

このプロパティを使用すると、コンシューマは、コマンドにバインドされているパラメータのいずれかがOracleのNデータ型(NCHAR、NVARCHAR2またはNLOB)かどうかを指定できます。この情報は、OraOLEDBがパラメータを検出およびバインドするために必要です。このプロパティは、SELECT文を実行するコマンドには設定しないでください。ただし、このプロパティはINSERT、UPDATE、DELETEなど、その他すべてのSQL文には設定する必要があります。

このプロパティを設定すると、データベースへの少なくとも1回のラウンドトリップという処理オーバーヘッドが課されるので、このプロパティの使用はNデータ型のパラメータを含むSQL文に限定する必要があります。デフォルトでは、このプロパティはFALSEに設定されています。

### 注意:



OraOLEDB は、SQL 文の WHERE 句内では N データ型のパラメータをサポートしません。

### 注意:



コンシューマは、ODBC プロシージャ・コールのエスケープ・シーケンスを使用して、N データ型パラメータを持つ**ストアド・プロシージャ**または**ファンクション**をコールする必要があります。

## 例: カスタム・プロパティNDatatypeの設定

```

Dim objCon As NEW ADODB.Connection
Dim objCmd As NEW ADODB.Command
Dim prEmpno As NEW ADODB.Parameter
Dim prEname As NEW ADODB.Parameter
...
objCmd.ActiveConnection = objCon
objCmd.CommandType = adCmdText

' Create and append the parameters to the command object
Set prEmpno = objCmd.CreateParameter("prEmpno", adSmallInt, adParamInput, , 8521)
' prEname is bound to a NVARCHAR2 column in the EMP table
Set prEname = objCmd.CreateParameter("prEname", adBSTR, adParamInput, , "Joe")
objCmd.Parameters.Append prEmpno
objCmd.Parameters.Append prEname

```

```

' Enabling the NDatatype property indicates to the provider
' that one or more of the bound parameters is of N datatype
objCmd.Properties("NDatatype") = TRUE

' Assume column ENAME in table EMP is of NVARCHAR2 type
objCmd.CommandText = "INSERT INTO EMP (EMPNO, ENAME) VALUES (?, ?)"

' Execute the SQL
objCmd.Execute

' It is a good idea to disable the property after execute as the same command
' object may be used for a different SQL statement
objCmd.Properties("NDatatype") = FALSE

```

## SPPrmsLOB

このプロパティを使用すると、コンシューマは、ストアド・プロシージャにバインドされているパラメータの1つ以上がOracleのLOBデータ型(CLOB、BLOBまたはNCLOB)かどうかを指定できます。OraOLEDBでは、ストアド・プロシージャを実行する前にそのパラメータ・リストをフェッチするために、このプロパティをTRUEに設定する必要があります。このプロパティを使用すると、1つ以上のLOBデータ型パラメータを持つストアド・プロシージャの処理オーバーヘッドが制限されます。このプロパティは、コマンドが実行された後はFALSEに設定する必要があります。デフォルトでは、このプロパティはFALSEに設定されています。

### 注意:



コンシューマは、ODBC プロシージャ・コールのエスケープ・シーケンスを使用して、LOB データ型パラメータを持つストアド・プロシージャまたはストアド・ファンクションをコールする必要があります。

### 例: カスタム・プロパティSPPrmsLOBの設定

```

Dim objCon As NEW ADODB.Connection
Dim objCmd As NEW ADODB.Command
Dim prCLOB As NEW ADODB.Parameter
...
objCmd.ActiveConnection = objCon
objCmd.CommandType = adCmdText

' Create and append the parameters to the command object
Set prCLOB = objCmd.CreateParameter("prCLOB", adLongVarchar, adParamOutput, _
10000)
objCmd.Parameters.Append prCLOB

' Enabling the SPPrmsLOB property indicates to the provider
' that one or more of the bound parameters is of LOB data type
objCmd.Properties("SPPrmsLOB") = TRUE

' Assume the Stored Procedure requires a CLOB parameter
objCmd.CommandText = "{ call storedproc(?) }"

' Execute the SQL
objCmd.Execute

' It is a good idea to disable the property after execute as the
' same command object may be used for a different SQL statement
objCmd.Properties("SPPrmsLOB") = FALSE

```

## AddToStmtCache

このプロパティを使用すると、プロパティがTRUEに設定され、文のキャッシュが有効になっているときに、実行された文をコンシューマがキャッシュできます。文のキャッシュが無効になっているか、このプロパティがFALSEに設定されていると、実行された文はキャッシュされません。

文のキャッシュが無効の場合、このプロパティは無視されます。文のキャッシュを有効にするには、StmtCacheSize接続文字列属性を0よりも大きい値に設定します。このプロパティを使用すると、文のキャッシュが有効な場合に、文を選択してキャッシュに追加することができます。デフォルトでは、このプロパティはTRUEに設定されています。

例: カスタム・プロパティAddToStmtCacheの設定

```
Dim objCon As NEW ADODB.Connection
Dim objCmd As NEW ADODB.Command
...

' Statement caching is enabled by setting the 'StmtCacheSize'
' connection string attribute to a value greater than zero
objCon.ConnectionString = "StmtCacheSize=10;Data Source=MyOraDb;" & _                               "User
ID=scott;Password=tiger;"
objCon.Open
objCmd.ActiveConnection = objCon
objCmd.CommandType = adCmdText
objCmd.CommandText = "SELECT * FROM EMP"

' "SELECT * FROM EMP" statement would be added to the statement cache because
' StmtCacheSize connection string attribute value is greater than 0 and
' AddToStmtCache property value is TRUE by default.
objCmd.Execute

' Do not add "SELECT * FROM DEPT" to the statement cache
objCmd.CommandText = "SELECT * FROM DEPT"
objCmd.Properties("AddToStmtCache") = FALSE

' "SELECT * FROM DEPT" statement would not be added to the statement cache
objCmd.Execute
```

## 行セットを返すストアード・プロシージャおよびストアード・ファンクション

Oracle Provider for OLE DBでは、コンシューマは、REF CURSOR型の引数を持つPL/SQLストアード・プロシージャまたはREF CURSOR値を返すストアード・ファンクションを実行できます。

OraOLEDBはREF CURSORバインド変数に対して行セットを返します。OLE DBの仕様にはREF CURSORのデータ型が定義されていないため、コンシューマはこのパラメータをバインドできません。

PL/SQLストアード・プロシージャにREF CURSOR型の引数が1つ以上ある場合、OraOLEDBはそれらの引数をバインドし、REF CURSOR型の各引数に対して行セットを返します。

PL/SQLストアード・ファンクションがREF CURSORを返す場合、またはREF CURSOR型の引数を持つ場合、OraOLEDBはそれらの引数をバインドし、各REF CURSOR型に対して行セットを返します。

この機能を使用するには、ODBCプロシージャ・コールのエスケープ・シーケンスでストアード・プロシージャまたはストアード・ファンクションをコールする必要があります。

コールされるストアード・プロシージャまたはストアード・ファンクションは、スタンドアロンまたはパッケージのどちらでも構いません。ただし、返されるREF CURSORは、データベース内のパッケージで明示的に定義されている必要があります。

## 複数の行セット

OraOLEDBは、ストアド・プロシージャから複数の行セットを返す機能をサポートしています。コンシューマは、この機能を使用して、ストアド・プロシージャから返されたすべてのREF CURSORにアクセスできます。

例: 複数の行セットを返すストアド・プロシージャ

PL/SQLパッケージ

```
CREATE OR REPLACE PACKAGE Employees AS
  TYPE empcur IS REF CURSOR;

  PROCEDURE GetEmpRecords (p_cursor OUT empcur,
                           q_cursor OUT empcur,
                           indeptno IN NUMBER,
                           p_errorcode OUT NUMBER);

  FUNCTION GetDept (inempno IN NUMBER,
                   p_errorcode OUT NUMBER)
    RETURN empcur;
END Employees;

CREATE OR REPLACE PACKAGE BODY Employees AS

  PROCEDURE GetEmpRecords (p_cursor OUT empcur,
                           q_cursor OUT empcur,
                           indeptno IN NUMBER,
                           p_errorcode OUT NUMBER) IS

  BEGIN
    p_errorcode := 0;
    OPEN p_cursor FOR
      SELECT *
      FROM emp
      WHERE deptno = indeptno
      ORDER BY empno;

  OPEN q_cursor FOR
    SELECT empno
    FROM emp
    WHERE deptno = indeptno
    ORDER BY empno;

  EXCEPTION
    WHEN OTHERS THEN
      p_errorcode:= SQLCODE;

  END GetEmpRecords;

  FUNCTION GetDept (inempno IN NUMBER,
                   p_errorcode OUT NUMBER)
    RETURN empcur IS
    p_cursor empcur;
  BEGIN
    p_errorcode := 0;
    OPEN p_cursor FOR
      SELECT deptno
      FROM emp
      WHERE empno = inempno;
    RETURN (p_cursor);

  EXCEPTION
```

```
WHEN OTHERS THEN
    p_errorcode:= SQLCODE;
```

```
END GetDept;
```

```
END Employees;
```

## ADOプログラム

```
Dim Con As New ADODB.Connection
Dim Rst1 As New ADODB.Recordset
Dim Rst2 As New ADODB.Recordset
Dim Rst3 As New ADODB.Recordset
Dim Cmd As New ADODB.Command
Dim Prm1 As New ADODB.Parameter
Dim Prm2 As New ADODB.Parameter

Con.Provider = "OraOLEDB.Oracle"
Con.ConnectionString = "Data Source=MyOraDb;" & _
    "User ID=scott:Password=tiger;"

Con.Open
Cmd.ActiveConnection = Con

' Although Employees.GetEmpRecords() takes four parameters, only
' two need to be bound because Ref cursor parameters are automatically
' bound by the provider.

Set Prm1 = Cmd.CreateParameter("Prm1", adSmallInt, adParamInput, , 30)
Cmd.Parameters.Append Prm1
Set Prm2 = Cmd.CreateParameter("Prm2", adSmallInt, adParamOutput)
Cmd.Parameters.Append Prm2

' Enable PLSQLRSet property
Cmd.Properties("PLSQLRSet") = TRUE

' Stored Procedures returning resultsets must be called using the
' ODBC escape sequence for calling stored procedures.
Cmd.CommandText = "{CALL Employees.GetEmpRecords(?, ?)}"

' Get the first recordset
Set Rst1 = Cmd.Execute

' Disable PLSQLRSet property
Cmd.Properties("PLSQLRSet") = FALSE

' Get the second recordset
Set Rst2 = Rst1.NextRecordset

' Just as in a stored procedure, the REF CURSOR return value must
' not be bound in a stored function.
Prm1.Value = 7839
Prm2.Value = 0

' Enable PLSQLRSet property
Cmd.Properties("PLSQLRSet") = TRUE

' Stored Functions returning resultsets must be called using the
' ODBC escape sequence for calling stored functions.
Cmd.CommandText = "{CALL Employees.GetDept(?, ?)}"
```

```
' Get the rowset
Set Rst3 = Cmd.Execute

' Disable PLSQLRSet
Cmd.Properties ("PLSQLRSet") = FALSE

' Clean up
Rst1.Close
Rst2.Close
Rst3.Close
```

## 文キャッシング

文のキャッシュにより、SQLまたはPL/SQL文の最初の実行時に作成されたサーバー・カーソルがキャッシュされるため、実行前に各文を解析する必要がなくなります。同じ文を後で実行する場合、カーソルから解析された情報を再利用して、文を解析せずに実行することで、パフォーマンスが向上します。

文のキャッシュによるパフォーマンスの向上を確認するには、繰り返して実行される文のみをキャッシュすることをお勧めします。さらに、SQL文またはPL/SQL文では、リテラル値ではなくパラメータを使用してください。このようにすると、文のキャッシュを最大限に活用できます。パラメータを含む文の解析情報は、後続の実行でパラメータ値が変わっても再利用できるためです。ただし、文のリテラル値が異なる場合、解析された情報は、後続の文にも同じリテラル値がないかぎり、再利用できません。

### StmtCacheSize接続文字列属性

この属性は、OraOLEDB文のキャッシングを有効化または無効化します。デフォルトでは、この属性は0（無効）に設定されています。0より大きい値に設定した場合、OraOLEDB文のキャッシングが有効になり、この値は接続用にキャッシュできる文の最大数を指定します。

1つの接続について指定の最大キャッシュ・サイズまでキャッシュされると、最近使用されていなかったカーソルが解放されて、新たに作成されたカーソルをキャッシュするための領域が空けられます。この値は、init.oraデータベース・コンフィギュレーション・ファイルにあるOPEN\_CURSORSパラメータ・セットの値以下にする必要があります。

### AddToStmtCacheコマンドのプロパティ

このプロパティは、文のキャッシュが有効な場合にのみ関係があります。文のキャッシュが有効な場合に、このプロパティがtrue（デフォルト）に設定されると、実行時に文がキャッシュに追加されます。文のキャッシュが無効になっているか、このプロパティがfalseに設定されていると、実行された文はキャッシュされません。

### レジストリを使用した文のキャッシュの有効化

システムで実行するすべてのOraOLEDBアプリケーションについて、アプリケーションを変更せずに、文のキャッシュをデフォルトで有効にするには、`%%HKEY_LOCAL_MACHINE%SOFTWARE%ORACLE%KEY_HOMENAME%OLEDB%StmtCacheSize`のレジストリ・キーを0よりも大きい値に設定します。このHOMENAMEは適切なOracleホームです。この値は、サーバーでキャッシュされるカーソルの数を指定します。デフォルトでは、これは10に設定されています。

### 接続と文キャッシング

文キャッシングは、接続ごとに個別に管理されます。このため、同じ文を別の接続で実行するときは、接続ごとに文を1回解析して個別のカーソルをキャッシュする必要があります。

## メタデータのキャッシュ

この機能は、SELECT文の最初の実行時にメタデータをキャッシュすることで、文のメタデータの取得を最小限に抑えます。その後、実行された同じ文は、キャッシュされたメタデータ情報を再利用して、パフォーマンスを向上させることができます。メタデータのキャッシュによるパフォーマンスの向上を確認するには、繰り返して実行される文のみをキャッシュすることをお勧めします。

## 注意:



メタデータのキャッシュは、各接続で個別に管理されます。そのため、同じ文を異なる接続で実行するには、メタデータを接続ごとに1回キャッシュする必要があります。

### 接続文字列属性を使用したメタデータのキャッシュの有効化

MetaDataCacheSize属性は、OraOLEDBメタデータ・キャッシングを有効化または無効化します。0より大きい値に設定した場合、OraOLEDBメタデータ・キャッシングが有効になり、この値はメタデータが接続用にキャッシュできる文の最大数を指定します。デフォルトでは、この属性は10に設定されています。

### レジストリを使用したメタデータのキャッシュの有効化

システムで実行中のすべてのOraOLEDBアプリケーションについて、アプリケーションを変更することなく、メタデータのキャッシュをデフォルトで有効にするには、次のレジストリ・キーを0よりも大きい値に設定します。デフォルトでは、これは10に設定されています。

```
¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOME¥OLEDB¥MetaDataCacheSize
```

### コマンド・タイムアウトおよびCancelメソッド

Cancelメソッドは、現在実行中のOraOLEDBコマンドを取り消します。このメソッドは、ネットワーク通信量やサーバーの使用負荷が高い時間帯に、アプリケーションで実行時間の長いコマンドを取り消す必要がある場合に有用です。

あるいは、開発者はCommandTimeoutプロパティを使用して、OraOLEDBでコマンドの取消しが試行されるまでの、コマンドの実行時間の制限を設定できます。OraOLEDBでCommandTimeoutを有効にするには、EnableCmdTimeoutレジストリ値を1に設定する必要があります。

OLE DBを使用する場合、デフォルトのDPBROP\_COMMANDTIMEOUTは0秒です。ADOを使用する場合、デフォルトのCommandTimeoutプロパティは30秒です。

### レジストリを使用したCommandTimeoutの有効化

インストール時に、デフォルト値が0に設定されたEnableCmdTimeoutというレジストリ値が追加されます。これを0に設定するとコマンド・タイムアウトが無効になり、1に設定すると有効になります。CommandTimeoutプロパティ値の設定は、EnableCmdTimeoutが1に設定されている場合にのみ有効になります。

レジストリ値は次のようになります。

```
¥¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥KEY_HOME¥OLEDB¥EnableCmdTimeout
```

## 行セット

ここでは、OraOLEDBで行セットを使用する方法を説明します。

### 行セットの作成

OraOLEDBは、行セットを作成するIOpenRowset::OpenRowsetおよびICommand::Executeをサポートします。

IOpenRowset::OpenRowsetによる行セットの作成

IOpenRowset::OpenRowsetを使用する場合は、次のガイドラインに従ってください。

- pTableIDパラメータには、実表またはビューを指定するDBID構造体が含まれている必要があります。

- DBID構造体のeKindメンバーには、DBKIND\_GUID\_NAME、DBKIND\_NAMEまたはDBKIND\_PGUID\_NAMEを設定する必要があります。
- DBID構造体のuNameメンバーには、実表またはビューの名前をUnicode文字列で指定する必要があります。これはNULLにはできません。
- OpenRowsetのpIndexIDパラメータにはNULLを設定する必要があります。

ICommand::Executeによる行セットの作成

OraOLEDBは、行セットを返すSQLのSELECT文をサポートします。また、OraOLEDBは、PL/SQLのストアド・プロシージャおよびストアド・ファンクションから行セットを返す機能をサポートします。

デフォルトでは、ADOはコマンド・オブジェクトから更新不可の行セットを作成します。更新可能な行セットは、コマンド・オブジェクトでUpdatabilityおよびIRowsetChangeの2つのプロパティを設定することで、作成できます。Updatabilityプロパティには、次の値を設定できます。

表2-2 Updatabilityプロパティに設定可能な値

値	説明
1	更新
2	削除
3	更新および削除
4	挿入
5	挿入および更新
6	挿入および削除
7	挿入、削除および更新

次のADOコードのサンプルでは、コマンド・オブジェクトのUpdatabilityプロパティに、行セット・オブジェクトに対する挿入、削除および更新の各操作ができるよう値を設定します。

```
Dim Cmd As New ADODB.Command
Dim Rst As New ADODB.Recordset
Dim Con As New ADODB.Connection
...
Cmd.ActiveConnection = Con
Cmd.CommandText = "SELECT * FROM emp"
Cmd.CommandType = adCmdText
cmd.Properties("IRowsetChange") = TRUE
Cmd.Properties("Updatability") = 7
' creates an updatable rowset
Set Rst = cmd.Execute
```



## 更新可能性

OraOLEDBは、即時と遅延の両方の更新モードをサポートします。ただし、その操作によってLONG型、BLOB型またはCLOB型などの非スカラー列が変更されるような挿入および更新は、遅延モードでは実行できません。非スカラー列の値が遅延更新モードで変更された場合は、それが即時更新モードで操作されたかのように、その行全体がデータベースに転送されます。また、これらの操作はUndoメソッド(ADO)あるいはIRowsetUpdate::Undo()を使用しても取り消すことができません。ただし、トランザクション内の操作の場合は、RollbackTransメソッド(ADO)またはITransactionLocal::Abort()によってロールバックできます。

結合を含む問合せを使用して作成された行セットは、Client Cursor Engineが使用可能になっているOraOLEDBでのみ更新できます。C/C++ OLE DBコンシューマは、これらの行セットを更新可能にするにはこのサービスを使用可能にする必要があります。ADOコンシューマは、これらの行セットを更新可能にするにはadUseClientとしてCursorLocationを指定する必要があります。

次に例を示します。

```
Dim objCon As New ADODB.Connection
Dim objRst As New ADODB.Recordset

objCon.Provider = "OraOLEDB.Oracle"
objCon.Open "MyOraDb", "scott", "tiger"
objRst.CursorLocation = adUseClient      ' ADO Client Cursor
objRst.Open "select ename, dname " & _
           "from emp, dept " & _
           "where emp.deptno = dept.deptno", _
           objCon, adOpenStatic, adLockOptimistic, adCmdText

' Recordset created is updatable. Please note that CursorLocation
' needs to be explicitly set to adUseClient for this join recordset
' to be updatable.
```

## Server Data on Insertプロパティ

OraOLEDBを使用してDBPROP\_SERVERDATAONINSERT(Server Data on Insert)にTRUEが設定されていると、挿入および更新が行セットを通じて操作された場合、コンシューマは新たに挿入および更新された行からデフォルト値、順序値、およびトリガーされた列の値を取得できます。

DBPROP\_SERVERDATAONINSERTがTRUEに設定されていると、OraOLEDBは新しく挿入または更新された行の行データをデータベースからフェッチするため、行セットを使用した挿入および更新の両方の操作でパフォーマンスが低下する可能性があります。ただし、DBPROP\_SERVERDATAONINSERTがデフォルト値のFALSEに設定されている場合は、それらの行の列値がリクエストされたときに、挿入および更新操作で明示的に提供された値のみが返されます。

行セットを作成する元になっている実表には、デフォルト値、順序値およびトリガーによる値は含まれません。

DBPROP\_SERVERDATAONINSERTはデフォルト値のFALSEのままにしておくことを強くお勧めします。

DBPROP\_SERVERDATAONINSERTプロパティは、コマンド・オブジェクトを使用した挿入および更新操作のパフォーマンスには影響しません。

## IRowsetFind::FindNextによる行の検索

OraOLEDBはCHAR型、DATE型、FLOAT型、NUMBER型、RAW型およびVARCHAR2型の列に対する検索のみをサポートします。それ以外の場合は、DB\_E\_NOTSUPPORTEDが返されます。

NULL値による検索を実行する場合は、DBCMPAREOPS\_EQ比較とDBCMPAREOPS\_NE比較の各操作のみがサポートされます。それ以外の場合は、DB\_E\_NOTSUPPORTEDが返されます。

## 行セットのためのOraOLEDB固有の接続文字列属性

行セットのパフォーマンスに影響を与えるOraOLEDB固有の接続文字列属性は、次のとおりです。

- CacheType - プロバイダが行セット・データの格納に使用する、キャッシュのタイプを指定します。OraOLEDBには、次の2つのキャッシュ・メカニズムがあります。
  - Memory - プロバイダはすべての行セット・データをインメモリーに格納します。このキャッシュ・メカニズムによりパフォーマンスは向上しますが、メモリーの使用量はかなり多くなります。デフォルトはMemoryです。
  - File - プロバイダはすべての行セット・データをディスク上に格納します。このキャッシュ・メカニズムによりメモリーの消費量は制限されますが、パフォーマンスは低下します。
- ChunkSize - この属性は、プロバイダのキャッシュでフェッチおよび格納される、LONG型およびLONG RAW型の列データのサイズをバイト数で指定します。この属性に大きな値を指定するとパフォーマンスが改善されますが、行セットにデータを格納するためにより多くのメモリーを必要とします。有効値は1から65535です。デフォルト値は100です。
- FetchSize: プロバイダが一度にフェッチする行数(フェッチ配列)を指定します。データ・サイズおよびネットワークの応答時間に基づいて設定する必要があります。この値の設定が高すぎる場合、問合せの実行時に待機時間が長くなることがあります。値の設定が低すぎる場合は、データベースへのラウンド・トリップが増える場合があります。有効値は、1から429、496および296です。デフォルトは100です。
- DeferUpdChk - DeferUpdChk属性は、更新可能性チェックを遅らせるかどうかを指定します。これにより、ADOの読取り専用の非接続行セットの更新がサポートされます。有効値は0(FALSE)と1(TRUE)です。デフォルトはFALSEで、OraOLEDBがチェックを遅らせないことを意味します。この属性が接続時に指定されない場合、デフォルトのレジストリ値が使用されます。

レジストリ内に、デフォルトの属性値が設定されています。次のADOコードは、デフォルトの属性値を上書きするサンプルです。

```
Dim con As ADODB.Connection
Set con = NEW ADODB.Connection
con.ConnectionString = "Provider=OraOLEDB.Oracle;User ID=scott;" & _
    "Password=tiger;Data Source=MyOraDB;" & _
    "FetchSize=200;CacheType=File;"
con.Open
```

### 関連項目:

デフォルト属性値の詳細は、[「デフォルト属性値」](#)を参照してください。

## ADOプログラマのためのヒント

ADO RowsetプロパティLockTypeのadLockPessimisticへの設定は、Oracle Provider for OLE DBではサポートされません。LockTypeがadLockPessimisticに設定されている場合、OraOLEDBは、adLockOptimisticに設定されているときと同様の動作をします。この動作は、OraOLEDBが変更される行に対して明示的なロックを実行しないために発生します。ただし、新しいデータがデータベースに送信されると、別のユーザーによって行セット・データがすでに更新されていない場合にかぎり、データベースは更新を実行しますが、これは内容を保証しない書き込みが許可されないことを意味します。LockType値のadLockReadOnly、adLockBatchOptimisticおよびadLockOptimisticがOraOLEDBでサポートされています。

ADO RowsetプロパティCursorTypeにadOpenKeysetまたはadOpenDynamicを設定しても、Oracle Provider for OLE DBではサポートされません。Oracleは文レベルの読取り一貫性をサポートすることで、問合せから返されるデータには、その問合せが実行された時点でコミット済のデータのみが含まれることが保証されるため、OraOLEDBは2つのどちらもサポートしません。

CursorTypeの値としては、adOpenStaticおよびadOpenForwardOnlyがOraOLEDBによってサポートされています。

## スキーマ行セット

Oracle Provider for OLE DBで使用可能なスキーマ行セットは、次のとおりです。

- DBSCHEMA\_COLUMNS
- DBSCHEMA\_INDEXES
- DBSCHEMA\_SCHEMATA
- DBSCHEMA\_VIEWS
- DBSCHEMA\_TABLES
- DBSCHEMA\_PROVIDER\_TYPES(前方スクロールのみ)
- DBSCHEMA\_FOREIGN\_KEYS
- DBSCHEMA\_PRIMARY\_KEYS
- DBSCHEMA\_PROCEDURES
- DBSCHEMA\_PROCEDURE\_PARAMETERS

## 日付書式

Oracleセッションの日付書式は、ALTER SESSION SET NLS\_DATE\_FORMATコマンドを使用しても設定できません。Visual Basicでは、日付書式はWindowsのコントロール パネルの「地域」設定で制御されます。Visual Basicの日付書式の詳細は、Visual Basicのドキュメントを参照してください。

Oracle Provider for OLE DBでは、ConnectionプロパティのUseSessionFormatがデフォルト値のFALSEに設定されている場合、セッションのNLS\_DATE\_FORMATが、プロバイダによって'YYYY-MM-DD HH24:MI:SS'に固定されています。日付を文字列としてOracle Databaseに渡す場合、'YYYY-MM-DD HH24:MI:SS'の形式にする必要があります。UseSessionFormatがTRUEに設定されている場合、NLS\_DATE\_FORMATはOracle Provider for OLE DBによって固定されず、デフォルト・セッションのNLS\_DATE\_FORMATが使用されます。次に例を示します。

```
SELECT * FROM EMP WHERE HIREDATE > '1981-06-15 17:32:12'
```

別の書式を使用するには、SQLファンクションTO\_DATE()を使用して、文字列として渡される日付の書式を指定する必要があります。次に例を示します。

```
SELECT * FROM EMP WHERE HIREDATE > TO_DATE('15-JUN-81', 'DD-MON-YY')
```

ただし、パラメータとして渡される日付については、ADOが日付書式を制御するため、Windowsのコントロール パネルの「地域」設定によって制御されます。この場合は、TO\_DATE()を使用しないでください。次に例を示します。

```
Private Sub Command1_Click()  
    Dim objCon As New ADODB.Connection  
    Dim objCmd As New ADODB.Command  
    Dim objRst As New ADODB.Recordset  
    Dim pDate As New ADODB.Parameter  
  
    objCon.Provider = "OraOLEDB.Oracle"  
    objCon.Open "MyOraDb", "scott", "tiger"  
    Set pDate = objCmd.CreateParameter("pDate", adDate, adParamInput)  
    objCmd.Parameters.Append pDate  
    objCmd.CommandText = _  
        "SELECT * FROM EMP WHERE HIREDATE > ?"  
    objCmd.ActiveConnection = objCon  
    objCmd.CommandType = adCmdText  
    pDate.Value = "06/15/1981"
```

```
Set objRst = objCmd.Execute
```

```
...  
End Sub
```

## オブジェクト名の大/小文字

Oracle Databaseのオブジェクト(表、列、ビューなど)の名前にはすべて、大文字と小文字の区別があります。このため、EMPとempという2つのオブジェクトを、データベースの同じ名前空間に置くことができます。

問合せSELECT *ename* FROM *emp*は、表名がデータベース内でEMP(すべて大文字)であっても正しく実行されます。一方、大文字と小文字が混在したオブジェクト名を指定する場合は、二重引用符で名前を囲むことによって指定できます。次に例を示します。

```
SELECT ename FROM "Emp"
```

この例は、データベース内の表名がEmpの場合に正常に実行されます。二重引用符で囲むことにより、Oracle Databaseでオブジェクト名の大/小文字が区別されます。

## SQL Server Integration Servicesによる高速読み込みの実行

OraOLEDBは行セットを開くとき、IRowsetFastLoadインタフェースを実装します。SQL Server Integration Services (SSIS)とともに使用すると、Oracle Provider for OLE DBを使用してOLE DB宛先への高速読み込みをアプリケーションで実行できます。これによって、読み込み時間は従来型の読み込みに比べて大幅に短縮されます。高速読み込みを実行するには、OLE DB宛先のAccessModeプロパティを"OpenRowset Using FastLoad"に設定します。

## データ型

OraOLEDBでサポートされるデータ型とUnicodeおよび非Unicodeのマッピングを[表A-1](#)に示します。

### 関連項目:

前述のデータ型とその他のデータ型、およびタイムゾーンの詳細は、『[Oracle Database SQL言語リファレンス](#)』を参照

## バイナリ・データ型

BINARY\_FLOATは、OLE DBのDBTYPE\_R4にマップされる単精度浮動小数点データ型(4バイト)です。

BINARY\_DOUBLEは、OLE DBのDBTYPE\_R8にマップされる倍精度浮動小数点データ型(8バイト)です。

## タイムスタンプ・データ型

ここでは、タイムスタンプ・データ型について説明し、次の例を示します。

- 各タイムスタンプ・データ型を使用した挿入操作および取得操作を示すサンプル・データ。
- タイムスタンプ・データ型を使用したVisual Basicのコード例。

タイムスタンプ・データ型は、OLE DBのDBTYPE\_DBTIMESTAMPにマップされます。OLE DBのDBTYPE\_DBTIMESTAMPデータ型には、TIME\_ZONE情報はありません。

タイムスタンプ・データ型には次のものがあります。

- TIMESTAMP

- TIME WITH TIME ZONE
- TIMESTAMP WITH LOCAL TIME ZONE

#### データの挿入

TIMESTAMP WITH TIME ZONEまたはTIMESTAMP WITH LOCAL TIME ZONE列へのデータの挿入には、クライアントのタイムゾーン設定が使用されます。

OLE DBのタイムスタンプ・データ型ではタイムゾーン情報が提供されません。挿入操作の場合、クライアント・セッションからのデフォルト・タイムゾーンがTIMESTAMP WITH TIME ZONE列のデータに追加されます。

#### データの取出し

データ取得では、TIMESTAMP WITH TIME ZONE列にはTIME ZONEが使用されませんが、TIMESTAMP WITH LOCAL TIME ZONE列にはTIME ZONEが使用されます。

OLE DBのタイムスタンプ・データ型にはタイムゾーン情報を格納できません。

#### 秒の端数

TIMESTAMPデータ型とCommandオブジェクトのバインドでは、秒の端数がサポートされていません。

タイムゾーン情報を変更するためにALTER SESSIONを使用すると新規または既存のRecordsetのタイムゾーン情報は変更されません(Recordsetはオペレーティング・システムの地域オプションのクライアント・タイムゾーン設定を使用します)。TIMESTAMPのfractional\_seconds\_precisionの最大値は9で、デフォルトの精度は6です。

#### ADOコンシューマ

タイムスタンプ・データ型の場合、ADOコンシューマは、CursorLocationの値をadUseServerとして指定し、DML操作にRecordsetを使用する必要があります。

#### タイムスタンプの挿入および取出し例

次の例では、デフォルトの精度6を使用するものと想定しています。

#### TIMESTAMP列

データ挿入: 4/16/2003 11:19:19 AM(タイムゾーンなし)

DB内のデータ: 4/16/2003 11.19.19.000000 AM

データ取出し: 4/16/2003 11:19:19 AM

#### TIMESTAMP WITH TIME ZONE列

データ挿入: 4/16/2003 11:19:19 AM(クライアント・セッションのタイムゾーンを使用)

DB内のデータ: 4/16/2003 11.19.19.000000 AM -07:00

データ取出し: 4/16/2003 11:19:19 AM(タイムゾーンをドロップ)

#### 関連項目:

- [「データの挿入」](#)
- [「データの取出し」](#)

#### TIMESTAMP WITH LOCAL TIME ZONE列

次の例では、クライアント・セッションのタイムゾーンが-04:00で、現在US EDT(米国東部夏時間)にあるものと仮定しています。挿入操作では、TIMESTAMP WITH LOCAL TIME ZONE列のデータにはタイムゾーンのずれは含まれませんが、そのTIMESTAMPデータは、US PDT (太平洋夏時間)と同じであるデータベースのタイムゾーン-07:00に標準化されます。

問合せに対して、データはクライアント・セッションのタイムゾーンで返されます。タイムゾーンによる時差とは、ローカル時間と協定世界時(UTC)との差異(時分単位)を指します。

データ挿入: 4/16/2003 4:30:23 PM(クライアント・タイムゾーンは-04:00)

DB内のデータ: 4/16/2003 01.30.23.000000 PM(データベースのタイムゾーンは-07:00)

データ取出し: 4/16/2003 4:30:23 PM(クライアント・タイムゾーンは-04:00)

データ取出し: 4/16/2003 3:30:23 PM(クライアント・タイムゾーンは-05:00)

データ取出し: 4/16/2003 2:30:23 PM(クライアント・タイムゾーンは-06:00)

データ取出し: 4/16/2003 1:30:23 PM(クライアント・タイムゾーンは-07:00)

#### Visual Basicの例

```
...
Dim DT As Date
DT = Now()
con.ConnectionString = "Provider=OraOLEDB.Oracle.1;User ID=user_name;" & _
    "Password=pwd;Data Source=Oracle;"
con.Open
' Must use adUseServer
rec.CursorLocation = adUseServer
rec.ActiveConnection = con
rec.Open "select timestamp_column from test_table", con, adOpenDynamic, _
    adLockOptimistic
rec.AddNew Array("timestamp_column"), Array(DT)

update data
rec.Update Array("timestamp_column"), Array("07/07/07 07:17:17 AM")
...
```

## 期間データ型

期間データ型は、OLE DBのDBTYPE\_STRにマップされます。期間データ型には次のものがあります。

- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND

INTERVAL YEAR TO MONTH列では、year\_precisionの最大は9で、デフォルトは2です。INTERVAL DAY TO SECOND列では、day\_precisionの最大は9、デフォルトは2で、fractional\_seconds\_precisionの最大は9、デフォルトは6です。

### 注意:



符号が指定されていない場合、デフォルトは+です。

INTERVALYEARTOMONTH

使用方法: (符号)years-months

例:

- 2-3  
2年3か月
- +2-3  
2年3か月
- -2-3  
-2年3か月

INTERVALDAYTOSECOND

使用方法: (符号)days hours:minutes:seconds. second\_fraction

例:

- 7 10:20:30.123456  
7日と10時間20分30.123456秒
- +7 10:20:30.123456  
7日と10時間20分30.123456秒
- -7 10:20:30.123456  
-7日と10時間20分30.123456秒

Visual Basicの例

```
...
con.ConnectionString = "Provider=OraOLEDB.Oracle.1;User ID=user_name;" & _
    "Password=pwd;Data Source=Oracle;"
con.Open
'no restriction on using adUseServer or adUseClient
rec.CursorLocation = adUseServer
rec.ActiveConnection = con
rec.Open "select * from test_table2", con, adOpenDynamic, adLockOptimistic
rec.AddNew Array("year_to_month_column", "day_to_second_column"), _
    Array("8-1", "3 20:30:10.12")

'update data
rec.Update Array("year_to_month_column", "day_to_second_column"), _
    Array("2-3", "7 10:20:30.123456")
...
```

## LOBサポート

ISequentialStreamインタフェースは、すべてのLONG、LONG RAWおよびLOB (BLOB、CLOB、NCLOBおよびBFILE)列でサポートされます。コンシューマはこのインタフェースを使用して、読み取り専用のBFILEを除くすべてのLOB列に対して読み取りおよび書き込みを行えます。これらの列に対して読み取りおよび書き込みアクセスを行うには、行セットを作成するために使用するSELECT SQL文に結合を含めないでください。



### 注意:

Oracle Database の LOB 列のほとんどが、2GB を超えるデータ記憶域の LOB をサポートしますが、ADO

の列サイズの最大値は 2GB です。

BFILEデータ型を持つ列は、Rowsetインタフェースでは更新できません。ただし、更新が、BFILE列で示される外部ファイルのディレクトリと名前の変更に限定される場合は、コマンド・インタフェースを使用してこれらの列を更新できます。次に例を示します。

```
INSERT INTO topomaps (areanum, topomap)
VALUES (158, BFILENAME('mapdir', 'topo158.tps'))
```

LOBの詳細は、[Oracle Database SecureFilesおよびラージ・オブジェクト開発者ガイド](#)を参照してください。

## Unicodeサポート

OraOLEDBはUnicodeキャラクタ・セットをサポートします。コンシューマはこの機能を使用して、同じクライアント・コンピュータにある複数言語のデータにアクセスするためにOraOLEDBを使用できます。これは、Unicode標準で必要とされる数の言語をサポートするグローバル・インターネット・アプリケーションの作成に特に役立ちます。たとえば、Oracle Databaseにアクセスする単一のActive Server Page (ASP)アプリケーションを作成して、日本語、アラビア語、英語、タイ語などのコンテンツを動的に生成できます。

### Unicodeエンコーディングのタイプ

Oracle Databaseは、UnicodeのASCII互換のマルチバイト・エンコーディングであるUTF-8コード体系でUnicodeデータを格納します。Microsoft Windowsは、2バイトの固定幅コード体系であるUTF-16エンコーディングを使用します。

OraOLEDBは、2つのコード体系間でデータを透過的に変換し、コンシューマがUTF-16のみに対処できるようにします。

#### 注意:



Unicode サポートは、ADO コンシューマに対して透過的です。C または C++ を使用している OLE DB コンシューマは、Unicode データが含まれている場合には、データ型バインドで DBTYPE\_WSTR を明示的に指定する必要があります。

## Oracle Unicodeサポートの機能

OraOLEDBは、Unicodeモードと非Unicodeモードの2つのモードで動作します。クライアントのキャラクタ・セットがサーバーのキャラクタ・セットのスーパーセットでない場合、またはデータベースのキャラクタ・セットがマルチバイト・キャラクタ・セットである場合、OraOLEDBは自動的にUnicodeモードを有効にします。このモードでは、OraOLEDBはUTF-16コード体系でキャッシュ内にデータを格納します。ユーザーは、データの損失を防ぐために、データベースのキャラクタ・セットがAL32UTF8であることを確認する必要があります。

クライアントのキャラクタ・セットがサーバーのキャラクタ・セットのスーパーセットである場合、プロバイダは非Unicodeモードで動作します。このモードでは、UTF-16エンコーディングで必要となる大規模な文字バッファを扱う必要がないため、パフォーマンスが多少向上します。

クライアントとサーバーのキャラクタ・セットの検出は、ログオン時に実行されます。



#### 注意:



OraOLEDB では、Unicode モードを有効にするためにクライアントのキャラクタ・セットを UTF-8 に設定する必要はなくなりました。プロバイダは、現在もこのような設定をサポートしていますが、必須ではなくなりました。

詳細は、[行セットおよびパラメータのデータ型マッピング](#)を参照してください。

## Unicodeサポートの設定

データの損失を防ぐために、データベースのキャラクタ・セットをAL32UTF8にする必要があります。これ以外に、Unicodeサポートで必須の設定はありません。

### データベースの設定

Oracle DatabaseがAL32UTF8キャラクタ・セットでデータを格納するように構成されていることを確認する必要があります。キャラクタ・セット構成は、通常はデータベースの作成時に指定します。データベースのキャラクタ・セット設定を確認するには、SQL\*Plusで次の問合せを実行します。

```
SQL> SELECT parameter, value FROM nls_database_parameters
       WHERE parameter = 'NLS_CHARACTERSET';
```

データベースのキャラクタ・セットがAL32UTF8でない場合は、AL32UTF8キャラクタ・セットで新規のデータベースを作成し、そこにデータをインポートする必要があります。

### 関連項目:

- [Oracle Database管理者ガイド](#)
- [Oracle Databaseグローバル化・サポート・ガイド](#)

## エラー

OLEオブジェクトおよびCOMオブジェクトは、オブジェクト・メンバー・ファンクションのHRESULTリターン・コードによってエラーを報告します。OLE/COMのHRESULTリターン・コードはビット圧縮された構造体です。OLEには構造体メンバーの参照を解除するマクロがあります。OraOLEDBは、エラーに関する情報を取り出すために、IErrorLookupを公開します。

すべてのオブジェクトは拡張エラー情報をサポートします。このため、コンシューマは、OLE DB拡張エラー・オブジェクトをインスタンス化し、続いてGetErrorDescription() メソッドをコールして、エラー・テキストを取得する必要があります。

```
// Instantiate OraOLEDBErrorLookup and obtain a pointer to its
// IErrorLookup interface
CoCreateInstance(CLSID_OraOLEDBErrorLookup, NULL, CLSCTX_INPROC_SERVER,
                 IID_IErrorLookup, (void **)&pIErrorLookup)
//Call the method GetErrorDescription() to get the full error text
pIErrorLookup->GetErrorDescription()
```

OraOLEDBプロバイダは、エラー全体を1つのテキスト・ブロックにスタックして返します。

ADOユーザーは、次のサンプルを参考にしてください。

```
Dim oerr As ADODB.Error
For Each oerr in con.Errors
    MsgBox "Error: " & oerr.Description & vbCrLf _
        & "Source: " & oerr.Source
Next
```

## OLEDB.NETデータ・プロバイダの互換性

OLE DB .NETデータ・プロバイダは、Oracle DatabaseにアクセスするためのOLE DBプロバイダとしてOraOLEDBを利用できます。

OraOLEDBとOLE DB .NETデータ・プロバイダとの間に互換性を持たせるには、接続文字列属性OLEDB.NETをTrueに設定します。

OLEDB.NET属性をFalseに設定すると、.NETとの互換性が無効になります。

### 注意:



OLEDB.NET 接続文字列属性は、ADO アプリケーションでは使用しないでください。

## 接続文字列でのOLEDB.NET属性の使用

OLE DB .NETデータ・プロバイダでOraOLEDBを使用する際に、OLEDB.NET接続属性は、次の例に示すようにTrueに設定する必要があります。

```
// in VB.NET
Dim con As New OleDbConnection()
con.ConnectionString = "Provider=OraOLEDB.Oracle;User Id=scott;" & _
    "Password=tiger;Data Source=Oracle;OLEDB.NET=True;"
con.Open

// in C#
...
OleDbConnection con = new OleDbConnection();
con.ConnectionString = "Provider=OraOLEDB.Oracle;User Id=scott;" +
    "Password=tiger;Data Source=Oracle;OLEDB.NET=true;"
con.Open();
...
```

## OraOLEDBカスタム・プロパティの使用

ADOでは、OraOLEDBプロバイダ固有のプロパティをオブジェクト・レベルで設定できます。OraOLEDB固有のプロパティSPPrmsLOBおよびNDatatypeは、接続文字列属性としても、コマンド・オブジェクト・レベルでも設定できます。StmtCacheSizeプロパティは接続文字列属性として設定でき、AddToStmtCacheプロパティはコマンド・オブジェクト・レベルで設定できます。次の例では、コマンド・レベルでのプロパティの設定を示します。

```
// in VB
Dim cmd as new ADO.Command
...
cmd.Properties("SPPrmsLOB") = True
cmd.Properties("NDatatype") = True
cmd.Properties("AddToStmtCache") = True
...
```

一方、OLEDB.NETデータ・プロバイダは、OLE DBプロバイダ固有のプロパティをオブジェクト・レベルで公開できません。したがって、OraOLEDBがOLE DB .NETデータ・プロバイダで使用される場合、SPPrmsLOBプロパティおよびNDatatypeプロパティは接続文字列属性としてのみ設定でき、AddToStmtCacheプロパティはサポートされません。

```
// in VB.NET
```

```
Dim con As New OleDbConnection()  
con.ConnectionString = "Provider=OraOLEDB.Oracle;User Id=scott;" & _  
    "Password=tiger;Data Source=Oracle;OLEDB.NET=True;" & _  
    "SPPrmsLOB=False;NDatatype=False;"  
con.Open()
```

SPPrmsLOBおよびNDatatype接続文字列属性が指定されていない場合は、デフォルトでFalseに設定されます。

これらの接続文字列属性をTrueに設定すると、パラメータ付きのコマンドを実行する際に、処理オーバーヘッドが増加します。

#### 関連項目:

いずれかの属性をTrueに設定する前に「[コマンドのためのOraOLEDBカスタム・プロパティ](#)」を参照してください

## DataTable変更でのOracleの更新

OleDbDataAdapter.Update() メソッドがDataTableで行われた変更でOracle Databaseを適切に更新するには、DataTableにデータベース表の主キーが含まれている必要があります。データベース表に主キーが含まれていない場合は、DataTableに移入する際にROWIDを明示的に選択して、データベース内の行の更新時にROWIDを使用して行を一意に識別できるようにする必要があります。

主キーを含むデータベース表でROWIDを選択しないでください。ROWIDが主キーとともに選択されている場合、ROWIDは主キーとしてマークされる唯一の列になります。

#### 関連項目:

OLE DB .NETデータ・プロバイダの使用の詳細は、次の資料を参照してください。

- Microsoft .NETドキュメント
- Microsoft .NET Frameworkクラス・ライブラリ

## Visual BasicでのOraOLEDBの使用

次の簡単なサンプルで、Visual Basic 6.0でADOとOracle Provider for OLE DBを使用してOracle Databaseに接続し、PL/SQLのストアド・プロシージャとストアド・ファンクションを実行する方法を説明します。

### Oracle Databaseの設定

この例では、Oracle Databaseに、デモ用の表EMPとユーザー・アカウントscottが存在することを前提にしています。scottアカウントは、Oracleの初期データベースに含まれています。使用しているデータベースにこのアカウントがない場合は、サンプル・プログラムを実行する前にこのアカウントを作成してください。使用しているデータベースにemp表がない場合は、demo1d.sqlスクリプトを使用してこのデモ用の表を作成できます。

また、このサンプルでは、Oracle Databaseに接続するときに、データベースのネットワーク別名として *exampledb* を使用します。使用しているシステムに合わせて、このネットワーク別名を変更する必要があります。

#### ステップ1: サンプル表の作成

1. SQL\*Plusを起動します。

2. ユーザー名scottとパスワードtigerを使用して接続します。

3. demobl.d.sqlスクリプトを実行します。

```
SQL> @ORACLE_BASE\ORACLE_HOME\sqlplus\demo\demobl.d.sql;
```

scottアカウントでemp表を作成した後で、Visual Basicのサンプルで実行されるストアード・プロシージャとストアード・ファンクションを含むPL/SQLパッケージを作成する必要があります。

ステップ2: PL/SQLパッケージの作成

1. ユーザー名scottとパスワードtigerを使用して接続します。
2. [PL/SQLのパッケージ](#)で示した、PL/SQLパッケージを作成します。

### 注意:

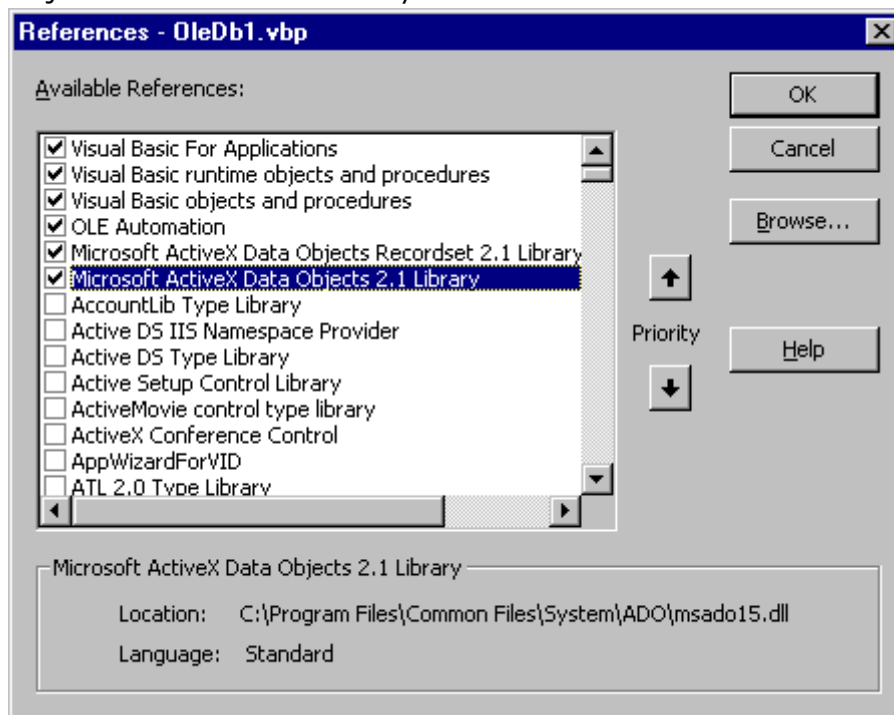


PL/SQL パッケージを作成するときには、/文字が終了記号として使用されます。各 CREATE PACKAGE...END ブロックに続く別の行に、この文字を追加する必要があります。

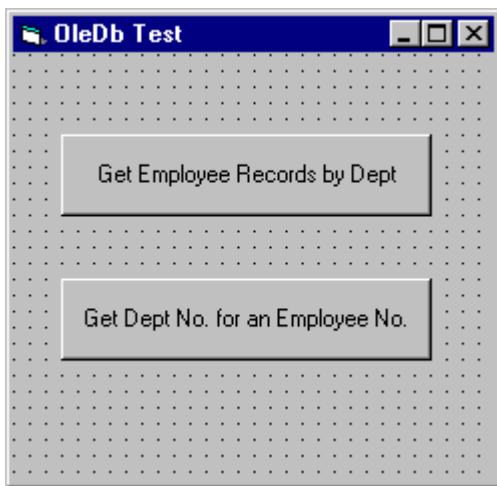
## Visual Basicプロジェクトの設定

Oracle Databaseの設定が完了すると、Visual Basic 6.0プロジェクトを作成できます。

1. Visual Basic 6.0を起動して新しいプロジェクトを作成します。
2. プロジェクト参照として、Microsoft ActiveX Data Objects 2.1 LibraryおよびMicrosoft ActiveX Data Objects Recordset 2.1 Libraryが含まれていることを確認します。



3. フォームにコマンド・ボタンを2つ追加します。その1つは、PL/SQLプロシージャGetEmpRecordsを実行するコードを実行するためのボタンです。もう1つは、PL/SQLファンクションGetDeptを実行するコードを実行するためのボタンです。



4. PL/SQLプロシージャGetEmpRecordsを実行するコードを実行するためのボタンのClickサブルーチンに、次のコードを追加します。

```

Dim Oracon As ADODB.Connection
Dim recset As New ADODB.Recordset
Dim cmd As New ADODB.Command
Dim param1 As New ADODB.Parameter
Dim param2 As New ADODB.Parameter
Dim objErr As ADODB.Error
Dim Message, Title, Default, EmpNoValue

Message = "Enter an employee number (5000 - 9000)"
Title = "Choose an Employee"
Default = "7654"

On Error GoTo err_test

EmpNoValue = InputBox(Message, Title, Default)
If EmpNoValue = "" Then Exit Sub
If EmpNoValue < 5000 Or EmpNoValue > 9000 Then EmpNoValue = 7654

Set Oracon = CreateObject("ADODB.Connection")
Oracon.ConnectionString = "Provider=OraOLEDB.Oracle;" & _
    "Data Source=exampledb;" & _
    "User ID=scott;" & _
    "Password=tiger;"

Oracon.Open
Set cmd.ActiveConnection = Oracon
Set param1 = cmd.CreateParameter("param1", adSmallInt, adParamInput, ,
    EmpNoValue)
cmd.Parameters.Append param1
Set param2 = cmd.CreateParameter("param2", adSmallInt, adParamOutput)
cmd.Parameters.Append param2

' Enable PLSQLRSet property
Cmd.Properties ("PLSQLRSet") = TRUE

cmd.CommandText = "{CALL Employees.GetDept(?, ?)}"
Set recset = cmd.Execute

' Disable PLSQLRSet property
Cmd.Properties ("PLSQLRSet") = FALSE

MsgBox "Number: " & EmpNoValue & " Dept: " & recset.Fields("deptno").Value

```

```
Exit Sub
```

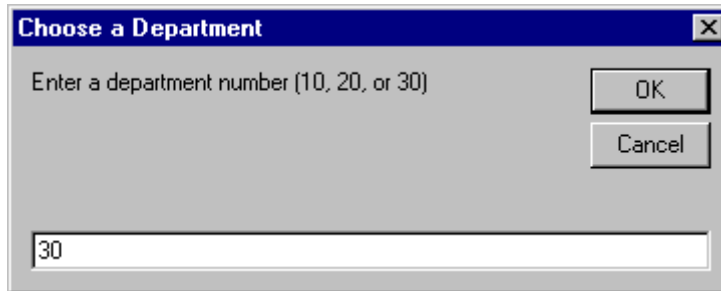
```
err_test:  
    MsgBox Error$  
    For Each objErr In Oracon.Errors  
        MsgBox objErr.Description  
    Next  
    Oracon.Errors.Clear  
    Resume Next
```

5. PL/SQLファンクションGetDeptを実行するコードを実行するためのボタンのClickサブルーチンに、次のコードを追加します。

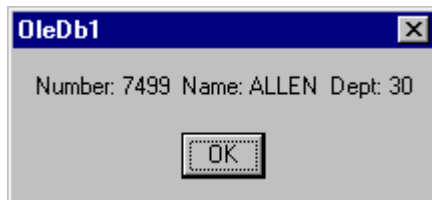
```
Dim Oracon As ADODB.Connection  
Dim recset As New ADODB.Recordset  
Dim cmd As New ADODB.Command  
Dim param1 As New ADODB.Parameter  
Dim param2 As New ADODB.Parameter  
Dim objErr As ADODB.Error  
  
Dim Message, Title, Default, DeptValue  
Message = "Enter a department number (10, 20, or 30)"  
Title = "Choose a Department"  
Default = "30"  
  
On Error GoTo err_test  
DeptValue = InputBox(Message, Title, Default)  
If DeptValue = "" Then Exit Sub  
If DeptValue < 10 Or DeptValue > 30 Then DeptValue = 30  
  
Set Oracon = CreateObject("ADODB.Connection")  
Oracon.ConnectionString = "Provider=OraOLEDB.Oracle;" & _  
    "Data Source=exampledb;" & _  
    "User ID=scott;" & _  
    "Password=tiger;"  
  
Oracon.Open  
Set cmd = New ADODB.Command  
Set cmd.ActiveConnection = Oracon  
Set param1 = cmd.CreateParameter("param1", adSmallInt, adParamInput, ,  
    DeptValue)  
cmd.Parameters.Append param1  
Set param2 = cmd.CreateParameter("param2", adSmallInt, adParamOutput)  
cmd.Parameters.Append param2  
  
' Enable PLSQLRSet property  
Cmd.Properties ("PLSQLRSet") = TRUE  
  
cmd.CommandText = "{CALL Employees.GetEmpRecords(?, ?)}"  
Set recset = cmd.Execute  
  
' Disable PLSQLRSet property  
Cmd.Properties ("PLSQLRSet") = FALSE  
  
Do While Not recset.EOF  
    MsgBox "Number: " & recset.Fields("empno").Value & " Name: " &  
        recset.Fields("ename").Value & " Dept: " & recset.Fields("deptno").Value  
    recset.MoveNext  
Loop  
  
Exit Sub
```

```
err_test:
  MsgBox Error$
  For Each objErr In Oracon.Errors
    MsgBox objErr.Description
  Next
  Oracon.Errors.Clear
  Resume Next
```

6. プロジェクトを実行して結果を確認します。たとえば、「Get Employee Records by Dept」ボタンを選択すると、部門番号の入力を要求するダイアログ・ボックスが表示されます。



部門番号を入力して「OK」をクリックすると、別のダイアログ・ボックスに、その部門の従業員名と従業員番号が表示されます。



# A プロバイダ固有の情報

次の各項では、Oracle Provider for OLE DBに固有のOLE DB情報について説明します。すべてのOLE DBプロパティおよびインタフェースの詳細なリストを含むOLE DBの一般的な情報については、Microsoft社の*OLE DB Programmer's Reference Guide*を参照してください。

- [行セットおよびパラメータのデータ型マッピング](#)
- [サポートされるプロパティ](#)
- [サポートされるインタフェース](#)
- [サポートされるメタデータ列](#)
- [OraOLEDBトレース](#)

## 行セットおよびパラメータのデータ型マッピング

ここでは、Oracleデータ型とOLE DBで定義される型間のデータ型マッピングのリストを示します。Oracle Provider for OLE DBは、パラメータと同様に行セットでもOLE DBで定義される特定のデータ型を使用して、Oracleデータ型を表します。OLE DBで定義される型は、表を作成するときにもOracleデータ型にマッピングされます。

それぞれのOracleデータ型が、次の表に示すように特定のOLE DBデータ型にマッピングされます。この対応関係は、データ型の情報がOracle Databaseから取り出されるときに使用されます。

表A-1 データ型マッピング

Oracleデータ型	OLE DBデータ型 - 通常(非Unicode)モード	OLE DBデータ型 - Unicodeモード
BFILE	DBTYPE_BYTES	DBTYPE_BYTES
BINARY_FLOAT	DBTYPE_R4	DBTYPE_R4
BINARY_DOUBLE	DBTYPE_R8	DBTYPE_R8
BLOB	DBTYPE_BYTES	DBTYPE_BYTES
CHAR	DBTYPE_STR	DBTYPE_WSTR
CLOB	DBTYPE_STR	DBTYPE_WSTR
DATE	DBTYPE_DBTIMESTAMP	DBTYPE_DBTIMESTAMP
FLOAT	DBTYPE_R8	DBTYPE_R8
INTERVAL DAY TO SECOND	DBTYPE_STR	DBTYPE_WSTR
INTERVAL YEAR TO MONTH	DBTYPE_STR	DBTYPE_WSTR



Oracleデータ型	OLE DBデータ型 - 通常(非Unicode)モード	OLE DBデータ型 - Unicodeモード
LONG	DBTYPE_STR	DBTYPE_WSTR
LONG RAW	DBTYPE_BYTES	DBTYPE_BYTES
NCHAR	DBTYPE_STR	DBTYPE_WSTR
NCLOB	DBTYPE_STR	DBTYPE_WSTR
NUMBER	DBTYPE_VARNUMERIC	DBTYPE_VARNUMERIC
NUMBER (p, s)	DBTYPE_NUMERIC	DBTYPE_NUMERIC
NVARCHAR2	DBTYPE_STR	DBTYPE_WSTR
RAW	DBTYPE_BYTES	DBTYPE_BYTES
ROWID	DBTYPE_STR	DBTYPE_STR
TIMESTAMP	DBTYPE_DBTIMESTAMP	DBTYPE_DBTIMESTAMP
TIME WITH TIME ZONE	DBTYPE_DBTIMESTAMP	DBTYPE_DBTIMESTAMP
TIMESTAMP WITH LOCAL TIME ZONE	DBTYPE_DBTIMESTAMP	DBTYPE_DBTIMESTAMP
VARCHAR	DBTYPE_STR	DBTYPE_WSTR

## サポートされるプロパティ

ここでは、Oracle Provider for OLE DBでサポートされるプロパティのリストを示します。読み取り/書き込みの状態と初期値も示します。

- [データソース・プロパティ](#)
- [DataSourceInfoのプロパティ](#)
- [初期化のプロパティと認可のプロパティ](#)
- [セッション・プロパティ](#)
- [行セットのプロパティ](#)

## データソース・プロパティ

[表A-2](#)は、データソースのプロパティの一覧です。

表A-2 DBPROPSET\_DATASOURCEプロパティ

プロパティ	状態	初期値
DBPROP_CURRENTCATALOG	読取り専用	NULL

## DataSourceInfoプロパティ

[表A-3](#)は、DataSourceInfoのプロパティの一覧です。

表A-3 DBPROPSET\_DATASOURCEINFOプロパティ

プロパティ	状態	初期値
DBPROP_ACTIVESESSIONS	読取り専用	0、無制限セッション
DBPROP_ASYNCNXNABORT	読取り専用	VARIANT_FALSE
DBPROP_ASYNCXNCOMMIT	読取り専用	VARIANT_FALSE
DBPROP_BYREFACCESSORS	読取り専用	VARIANT_TRUE
DBPROP_CATALOGLOCATION	読取り専用	DBPROPVAL_CL_END
DBPROP_CATALOGTERM	読取り専用	"Database link"
DBPROP_CATALOGUSAGE	読取り専用	DBPROPVAL_CU_DML_STATEMENTS
DBPROP_COLUMNDEFINITION	読取り専用	DBPROPVAL_CD_NOTNULL
DBPROP_CONCATNULLBEHAVIOR	読取り専用	DBPROPVAL_CB_NON_NULL
DBPROP_CONNECTIONSTATUS	読取り専用	DBPROPVAL_CS_INITIALIZED
DBPROP_DATASOURCENAME	読取り専用	" ", 実行時に設定
DBPROP_DATASOURCEREADONLY	読取り専用	VARIANT_FALSE
DBPROP_DBMSNAME	読取り専用	" ", 実行時に設定
DBPROP_DBMSVER	読取り専用	実行時に設定
DBPROP_DSOTHREADMODEL	読取り/書込み	DBPROPVAL_RT_FREETHREAD
DBPROP_GROUPBY	読取り専用	DBPROPVAL_GB_CONTAINS_SELECT

プロパティ	状態	初期値
DBPROP_HETEROGENEOUSTABLES	読取り専用	DBPROPVAL_HT_DIFFERENT_CATALOGS
DBPROP_IDENTIFIERCASE	読取り専用	DBPROPVAL_IC_UPPER
DBPROP_MAXINDEXSIZE	読取り専用	0、制限はブロックサイズに依存
DBPROP_MAXOPENCHAPTERS	読取り専用	0、未サポート
DBPROP_MAXORSINFILTER	読取り専用	0、未サポート
DBPROP_MAXROWSIZE	読取り専用	0、制限なし
DBPROP_MAXROWSIZEINCLUDESBLOB	読取り専用	VARIANT_FALSE
DBPROP_MAXSORTCOLUMNS	読取り専用	0、未サポート
DBPROP_MAXTABLESINSELECT	読取り専用	0、制限なし
DBPROP_MULTIPLEPARAMSETS	読取り専用	VARIANT_TRUE
DBPROP_MULTIPLERESULTS	読取り専用	DBPROP_MR_SUPPORTED   DBPROPVAL__MR_CONCURRENT
DBPROP_MULTIPLESTORAGEOBJECTS	読取り専用	VARIANT_FALSE
DBPROP_MULTITABLEUPDATE	読取り専用	VARIANT_FALSE
DBPROP_NULLCOLLATION	読取り専用	DBPROPVAL_NC_HIGH
DBPROP_OLEOBJECTS	読取り専用	DBPROPVAL_OO_BLOB
DBPROP_ORDERBYCOLUMNSINSELECT	読取り専用	VARIANT_FALSE
DBPROP_OUTPUTPARAMETERAVAILABILITY	読取り専用	DBPROPVAL_OA_ATEXECUTE
DBPROP_PERSISTENTIDTYPE	読取り専用	DBPROPVAL_PT_NAME
DBPROP_PREPAREABORTBEHAVIOR	読取り専用	DBPROPVAL_CB_PRESERVE

プロパティ	状態	初期値
DBPROP_PREPARECOMMITBEHAVIOR	読取り専用	DBPROPVAL_CB_PRESERVE
DBPROP_PROCEDURETERM	読取り専用	"PL/SQL Stored Procedure"
DBPROP_PROVIDERFRIENDLYNAME	読取り専用	"Oracle Provider for OLE DB"
DBPROP_PROVIDERNAME	読取り専用	OraOLEDBver.dll
DBPROP_PROVIDEROLEDBVER	読取り専用	"02.01"
DBPROP_PROVIDERSERVER	読取り専用	現在の OraOLEDB バージョンに設定
DBPROP_QUOTEDIDENTIFIERCASE	読取り専用	DBPROPVAL_IC_SENSITIVE
DBPROP_ROWSETCONVERSIONSONCOMMAND	読取り専用	VARIANT_TRUE
DBPROP_SCHEMATERM	読取り専用	"Owner"
DBPROP_SCHEMAUSAGE	読取り専用	DBPROPVAL_SU_DML_STATEMENTS   DBPROPVAL_SU_TABLE_DEFINITION   DBPROPVAL_SU_INDEX_DEFINITION   DBPROPVAL_SU_PRIVILEGE_DEFINITION
DBPROP_SERVERNAME	読取り専用	" "、実行時に設定
DBPROP_SORTONINDEX	読取り専用	VARIANT_FALSE
DBPROP_SQLSUPPORT	読取り専用	DBPROPVAL_SQL_ODBC_MINIMUM   DBPROPVAL_SQL_ANSI92_ENTRY   DBPROPVAL_SQL_ESCAPECLAUSES
DBPROP_STRUCTUREDSTORAGE	読取り専用	DBPROPVAL_SS_ISEQUENTIAL_STREAM
DBPROP_SUBQUERIES	読取り専用	DBPROPVAL_SQ_CORRELATEDSUBQUERIES
DBPROP_SUPPORTEDTXNDDL	読取り専用	DBPROPVAL_TC_DDL_COMMIT
DBPROP_SUPPORTEDTXNISOLEVELS	読取り専用	DBPROPVAL_TI_CURSORSTABILITY   DBPROPVAL_TI_READCOMMITTED
DBPROP_SUPPORTEDTXNISORETAIN	読取り専用	DBPROPVAL_TR_DONTCARE

プロパティ	状態	初期値
DBPROP_TABLETERM	読取り専用	"Table"
DBPROP_USERNAME	読取り専用	" ", 実行時に設定

## 初期化のプロパティと認可のプロパティ

[表A-4](#)は、初期化のプロパティと認可のプロパティの一覧です。

表A-4 DBPROPSET\_DBINITプロパティ

プロパティ	状態	初期値
DBPROP_AUTH_PERSIST_SENSITIVE_AUTHINFO	読取り専用	VARIANT_FALSE
DBPROP_AUTH_USERID	読取り/書込み	ユーザーID
DBPROP_INIT_DATASOURCE	読取り/書込み	接続文字列
DBPROP_INIT_HWND	読取り/書込み	プロンプト用のウィンドウ・ハンドル
DBPROP_INIT_LCID	読取り/書込み	システムの LCID
DBPROP_INIT_OLEDBSERVICES	読取り/書込み	DBPROPVAL_OS_ENABLEALL
DBPROP_INIT_PROMPT	読取り/書込み	DBPROMPT_NOPROMPT

## セッションのプロパティ

[表A-5](#)は、セッションのプロパティの一覧です。

表A-5 DBPROPSET\_SESSIONプロパティ

プロパティ	状態	初期値
DBPROP_SESS_AUTOCOMMITISOLEVELS	読取り専用	DBPROPVAL_TI_CURSORSTABILITY   DBPROPVAL_TI_READCOMMITTED

## 行セットのプロパティ

[表A-6](#)は、行セットのプロパティの一覧です。

表A-6 DBPROPSET\_ROWSETプロパティ

プロパティ	状態	初期値
-------	----	-----

プロパティ	状態	初期値
DBPROP_ABORTPRESERVE	読取り/書込み	VARIANT_TRUE
DBPROP_ACCESSORORDER	読取り専用	DBPROP_AO_RANDOM
DBPROP_APPENDONLY	読取り専用	VARIANT_FALSE
DBPROP_BLOCKINGSTORAGEOBJECTS	読取り専用	VARIANT_FALSE
DBPROP_BOOKMARKINFO	読取り専用	0
DBPROP_BOOKMARKS	読取り/書込み	VARIANT_FALSE
DBPROP_BOOKMARKSKIPPED	読取り/書込み	VARIANT_TRUE
DBPROP_BOOKMARKTYPE	読取り専用	DBPROP_BMK_NUMERIC
DBPROP_CACHEDDEFERRED	読取り専用	VARIANT_FALSE
DBPROP_CANFETCHBACKWARDS	読取り/書込み	VARIANT_FALSE
DBPROP_CANHOLDROWS	読取り/書込み	VARIANT_FALSE
DBPROP_CANSROLLBACKWARDS	読取り/書込み	VARIANT_FALSE
DBPROP_CHANGEINSERTEDROWS	読取り専用	VARIANT_TRUE
DBPROP_CLIENTCURSOR	読取り/書込み	VARIANT_TRUE
DBPROP_COLUMNRESTRICT	読取り専用	VARIANT_TRUE
DBPROP_COMMANDTIMEOUT	読取り/書込み	0
DBPROP_COMMITPRESERVE	読取り/書込み	VARIANT_TRUE
DBPROP_DEFERRED	読取り専用	VARIANT_TRUE
DBPROP_DELAYSTORAGEOBJECTS	読取り専用	VARIANT_TRUE、遅延更新なし
DBPROP_FINDCOMPAREOPS	読取り専用	DBPROPVAL_CO_EQUALITY   DBPROPVAL_CO_STRING

プロパティ	状態	初期値
		DBPROPVAL_CO_CASESENSITIVE   DBPROPVAL_CO_CASEINSENSITIVE   DBPROPVAL_CO_CONTAINS   DBPROPVAL_CO_BEGINSWITH
DBPROP_HIDDENCOLUMNS	読取り専用	0
DBPROP_IACCESSOR	読取り専用	VARIANT_TRUE
DBPROP_ICOLUMNSINFO	読取り専用	VARIANT_TRUE
DBPROP_ICOLUMNSROWSET	読取り/書込み	VARIANT_TRUE
DBPROP_ICONNECTIONPOINTCONTAINER	読取り専用	VARIANT_TRUE
DBPROP_ICONVERTTYPE	読取り専用	VARIANT_TRUE
DBPROP_IMMOBILEROWS	読取り専用	VARIANT_TRUE
DBPROP_IMULTIPLERESULTS	読取り/書込み	VARIANT_TRUE
DBPROP_IROWSET	読取り専用	VARIANT_TRUE
DBPROP_IROWSETCHANGE	読取り/書込み	VARIANT_FALSE
DBPROP_IROWSETFIND	読取り/書込み	VARIANT_FALSE
DBPROP_IROWSETIDENTITY	読取り専用	VARIANT_TRUE
DBPROP_IROWSETINFO	読取り専用	VARIANT_TRUE
DBPROP_IROWSETLOCATE	読取り/書込み	VARIANT_FALSE
DBPROP_IROWSETREFRESH	読取り/書込み	VARIANT_FALSE
DBPROP_IROWSETSCROLL	読取り/書込み	VARIANT_FALSE
DBPROP_IROWSETUPDATE	読取り/書込み	VARIANT_FALSE
DBPROP_ISEQUENTIALSTREAM	読取り/書込み	VARIANT_TRUE

プロパティ	状態	初期値
DBPROP_ISUPPORTERRORINFO	読取り/書込み	VARIANT_TRUE
DBPROP_LITERALBOOKMARKS	読取り専用	VARIANT_FALSE
DBPROP_LITERALIDENTITY	読取り専用	VARIANT_FALSE
DBPROP_LOCKMODE	読取り専用	DBPROPVAL_LM_NONE
DBPROP_MAXOPENROWS	読取り/書込み	0, 無制限
DBPROP_MAXPENDINGROWS	読取り専用	0, 無制限
DBPROP_MAXROWS	読取り/書込み	0
DBPROP_MAXROWSIZE	読取り専用	0
DBPROP_MAXROWSIZEINCLUDESBLOB	読取り専用	VARIANT_FALSE
DBPROP_NOTIFICATIONGRANULARITY	読取り/書込み	DBPROPVAL_NT_MULTIPLEROWS
DBPROP_NOTIFICATIONPHASES	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODO   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODO   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYCOLUMNSET	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODO   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODO   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWDELETE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODO   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODO   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWFIRSTCHANGE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODO
DBPROP_NOTIFYROWINSERT	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODO   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODO



プロパティ	状態	初期値
		DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWRESYNCH	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETRELEASE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWUNDOCHANGE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODD   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDODELETE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODD   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDOINSERT	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODD   DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDOUPDATE	読取り/書込み	DBPROPVAL_NP_OKTODO   DBPROPVAL_NP_ABOUTTODD   DBPROPVAL_NP_SYNCHAFTER   DBPROPVAL_NP_FAILEDTODD   DBPROPVAL_NP_DIDEVENT
DBPROP_ORDEREDBOOKMARKS	読取り専用	VARIANT_TRUE
DBPROP_OTHERINSERT	読取り専用	VARIANT_FALSE
DBPROP_OTHERUPDATEDELETE	読取り専用	VARIANT_FALSE
DBPROP_OWNINGINSERT	読取り専用	VARIANT_TRUE

プロパティ	状態	初期値
DBPROP_OWNUPDATEDELETE	読取り専用	VARIANT_TRUE
DBPROP_QUICKRESTART	読取り/書込み	VARIANT_FALSE
DBPROP_REENTRANTEVENTS	読取り専用	VARIANT_FALSE
DBPROP_REMOVEDELETED	読取り専用	VARIANT_TRUE
DBPROP_REPORTMULTIPLECHANGES	読取り専用	VARIANT_FALSE
DBPROP_RETURNPENDINGINSERTS	読取り/書込み	VARIANT_TRUE
DBPROP_ROWRESTRICT	読取り/書込み	VARIANT_FALSE
DBPROP_ROWTHREADMODEL	読取り専用	DBPROPVAL_RT_FREETHREAD
DBPROP_SERVERCURSOR	読取り/書込み	VARIANT_FALSE
DBPROP_SERVERDATAONINSERT	読取り/書込み	VARIANT_TRUE
DBPROP_STRONGIDENTITY	読取り/書込み	VARIANT_TRUE
DBPROP_TRANSACTEDOBJECT	読取り専用	VARIANT_TRUE
DBPROP_UNIQUEROWS	読取り/書込み	VARIANT_FALSE
DBPROP_UPDATABILITY	読取り/書込み	DBPROPVAL_UP_CHANGE   DBPROPVAL_UP_DELETE   DBPROPVAL_UP_INSET

## 行セットのプロパティの含意

特定のプロパティがVARIANT\_TRUEに設定されている場合、Oracle Provider for OLE DBは、その他の必要なプロパティを設定します。

- DBPROP\_IROWSETLOCATEにVARIANT\_TRUEが設定されている場合、次のプロパティにもVARIANT\_TRUEが設定されません。
  - DBPROP\_IROWSETIDENTITY
  - DBPROP\_CANHOLDROWS
  - DBPROP\_BOOKMARKS
  - DBPROP\_CANFETCHBACKWARDS
  - DBPROP\_CANSROLLBACKWARDS

- DBPROP\_IROWSETSCROLLにVARIANT\_TRUEが設定されている場合、次のプロパティにもVARIANT\_TRUEが設定されま  
す。
  - DBPROP\_IROWSETIDENTITY
  - DBPROP\_IROWSETLOCATE
  - DBPROP\_CANHOLDROWS
  - DBPROP\_BOOKMARKS
  - DBPROP\_CANFETCHBACKWARDS
  - DBPROP\_CANSROLLBACKWARDS
- DBPROP\_IROWSETUPDATEにVARIANT\_TRUEが設定されている場合、DBPROP\_IROWSETCHANGEプロパティにも  
VARIANT\_TRUEが設定されます。

## サポートされるインタフェース

ここでは、Oracle Provider for OLE DBでサポートされるOLE DBインタフェースを示します。

- [データ・ソース](#)
- [セッション](#)
- [コマンド](#)
- [行セット](#)
- [複数結果](#)
- [トランザクション・オプション](#)
- [カスタム・エラー・オブジェクト](#)

### データ・ソース

```
CoType TDataSource {
    interface IDBCreateSession;
    interface IDBInitialize;
    interface IDBProperties;
    interface IPersist;
    interface IDBInfo;
    interface ISupportErrorInfo;
}
```

### セッション

```
CoType TSession {
    interface IGetDataSource;
    interface IOpenRowset;
    interface ISessionProperties;
    interface IDBCreateCommand;
    interface IDBSchemaRowset;
    interface ISupportErrorInfo;
    interface ITransactionJoin;
    interface ITransactionLocal;
    interface ITransaction;
}
```

## コマンド

```
CoType TCommand {
    interface IAccessor;
    interface IColumnsInfo;
    interface ICommand;
    interface ICommandProperties;
    interface ICommandText;
    interface IConvertType;
    interface IColumnsRowset;
    interface ICommandPrepare;
    interface ICommandWithParameters;
    interface ISupportErrorInfo;
}
```

## 行セット

```
CoType TRowset {
    interface IAccessor;
    interface IColumnsInfo;
    interface IConvertType;
    interface IRowset;
    interface IRowsetInfo;
    interface IColumnsRowset;
    interface IConnectionPointContainer;
    interface IRowsetChange;
    interface IRowsetFind;
    interface IRowsetIdentity;
    interface IRowsetLocate;
    interface IRowsetRefresh;
    interface IRowsetFastLoad;
    interface IRowsetScroll;
    interface IRowsetUpdate;
    interface ISupportErrorInfo;
}
```

## 複数結果

```
CoType TMultipleResults {
    interface IMultipleResults;
    interface ISupportErrorInfo;
}
```

## トランザクションのオプション

```
CoType TTransactionOptions {
    interface ITransactionOptions;
    interface ISupportErrorInfo;
}
```

## カスタム・エラー・オブジェクト

```
CoType TCustomErrorObject {
    interface IErrorLookup;
}
```

## サポートされるメタデータ列

DBTYPE\_BASECOLUMNNAME、DBTYPE\_BASETABLENAMEおよびDBTYPE\_BASESCHEMANAMEの各メタデータ列には、読み取り専用レコードセット用のデータは移入されません。OraOLEDBは、DISTINCTまたはUNIQUEキーワードを使用したSQL問合せのためのサーバー・カーソル用読み取り専用レコードセットを作成します。OraOLEDBは、JOIN問合せのためのサーバー・カーソル用読み取り専用レコードセットも作成します。

次のメタデータ列は、OraOLEDBの列の行セットでサポートされます。

- DBCOLUMN\_IDNAME
- DBCOLUMN\_PROPID
- DBCOLUMN\_NAME
- DBCOLUMN\_NUMBER
- DBCOLUMN\_TYPE
- DBCOLUMN\_TYPEINFO
- DBCOLUMN\_COLUMNSIZE
- DBCOLUMN\_PRECISION
- DBCOLUMN\_SCALE
- DBCOLUMN\_FLAGS
- DBCOLUMN\_BASECATALOGNAME
- DBCOLUMN\_BASECOLUMNNAME
- DBCOLUMN\_BASESCHEMANAME
- DBCOLUMN\_BASETABLENAME
- DBCOLUMN\_COMPUTEMODE
- DBCOLUMN\_ISAUTOINCREMENT
- DBCOLUMN\_ISCASESENSITIVE
- DBCOLUMN\_ISSEARCHABLE
- DBCOLUMN\_OCTETLENGTH
- DBCOLUMN\_KEYCOLUMN

## OraOLEDBトレース

OraOLEDBには、デバッグを目的とする、インタフェース・コールのトレース機能があります。この機能は、顧客の問題をデバッグする際に、Oracleサポート・サービスの役に立つよう提供されています。

プロバイダは、次の情報を記録するように構成できます。

- OLE DBインタフェース・メソッドの入力および出力
  - 提供されているパラメータ値(入力)
  - 戻り値、HRESULT(出力)
  - スレッドID(開始および終了)
- 分散トランザクションの登録および登録解除
  - セッション・オブジェクト情報
  - トランザクションID

## 注意:



グローバル・トランザクションの登録および登録解除情報を記録するには、TraceLevel 値にセッション・オブジェクトを設定する必要があります。[コールをトレースするためのレジストリ設定](#)の TraceOption を参照してください。

## コールをトレースするためのレジストリ設定

ODAC 12c リリース4では、OraOLEDBがデフォルトでトレース・ファイルを書き込むために次の新しいディレクトリを使用するようになりました: `<Windows user temporary folder>%oledb%trace`。

Windowsユーザーの一時フォルダは、Windows TMPまたはTEMP環境変数など、ローカルのWindows設定で定義されています。通常、`C:%temp`または`C:%Users%<user name>%AppData%Local%Temp`です。

トレース・ファイルの場所を変更するには、TraceFileLocation属性を使用して新しい場所を設定します。

TraceFileLocationは、デフォルトではWindowsレジストリに作成されません。OraOLEDBは、トレース・ファイルを作成するたびにトレースが作成されたWindowsイベント・ログにエントリを作成します。

インタフェース・コールをトレースするには、`HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_<HOMENAME%\OLEDB%`に対して次のレジストリ値を構成する必要があります。

- TraceFileLocation

有効な値: 任意の有効なディレクトリ・パス

TraceFileLocationは、トレース・ファイルの保存先ディレクトリを指定します。デフォルトのトレースの場所は `<Windows user temporary folder>%oledb%trace`です。

- TraceCategory

有効な値:

- 0 = なし
- 1 = OLEDBインタフェース・メソッドの入力
- 2 = OLEDBインタフェース・メソッドの出力
- 4 = 分散トランザクションの登録および登録解除

TraceCategoryは、トレースする情報を指定します。有効値を加算することによって、異なるトレース・カテゴリの組合せを作成できます。たとえば、TraceCategoryに3を設定すると、すべてのOLE DBインタフェース・メソッドの入力と出力がトレースされます。

- TraceLevel

有効な値:

- 0 = なし
- 1 = データソース・オブジェクト
- 2 = セッション・オブジェクト
- 4 = コマンド・オブジェクト

- 8 = 行セット・オブジェクト
- 16 = エラー・オブジェクト
- 64 = 複数結果オブジェクト

TraceLevelは、トレースするOLE DBオブジェクトを指定します。すべてのOLE DBオブジェクトに対するすべての入出力のコールをトレースする必要はない場合、TraceLevelによって、トレース対象を1つまたは複数のOLE DBオブジェクトに限定できます。複数のオブジェクトに対するトレースを取得するには、有効値を加算します。たとえば、TraceLevelが12に、TraceCategoryが3に設定されている場合は、トレース・ファイルにはコマンド・オブジェクトと行セット・オブジェクトに対するメソッドの入出力のみが含まれます。

グローバル・トランザクションの登録および登録解除情報をトレースするには、TraceLevel値にセッション・オブジェクト(2)を設定する必要があります。

- TraceOption

有効な値:

- 0 = 単一のトレース・ファイル
- 1 = 複数のトレース・ファイル

TraceOptionは、各スレッドIDに対してトレース情報を単一のファイルと複数のファイルのどちらに記録するかを指定します。複数のトレース・ファイルが要求されている場合は、各スレッドのトレース・ファイルを作成するために指定されたファイル名にスレッドIDが追加されます。

# 用語集

## Component Object Model(COM)

オブジェクト間通信を、個々のオブジェクトの記述に使用されるプログラム言語の種類に関係なく実現するためのバイナリ標準。

## コンシューマ

コンシューマとは、データソースをコールするアプリケーションまたはツール、あるいはデータにアクセスするためのプロバイダのインタフェースのことです。「**プロバイダ**」を参照してください。

## Oracle Net Services

任意のネットワーク・プロトコルおよびオペレーティング・システムを介してOracle Toolsまたはデータベースに対して透過的操作を提供するOracleのクライアント/サーバー通信ソフトウェア。

## PL/SQL

Oracleによって提供される、SQLを拡張した手続き型言語。

## プロバイダ

プロバイダとは、コンシューマにデータを提供するインタフェースまたは一連のコンポーネントのことです。Oracle Provider for OLE DBでは、データ・プロバイダとは、要求時にデータソースからコンシューマにデータを転送し、表形式でデータを提供する一連のCOMコンポーネントをいいます。「**コンシューマ**」を参照してください。

## ストアド・プロシージャ

ストアド・プロシージャとは、Oracle Databaseに格納されるPL/SQLブロックのことで、アプリケーションから名前でも呼ぶことが可能です。



# 索引

記号 [A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#)

---

## 記号

- [.NET 1](#)
- 

## A

- [ADO 1, 2, 3](#)
  - [ADO.NET 1](#)
  - [ADOアプリケーションでのOLE DBサービス 1](#)
  - 属性
    - [接続文字列 1](#)
- 

## C

- C#
  - [接続文字列 1](#)
  - [例 1](#)
- [C++/COM 1](#)
- [C++/COMアプリケーションでのOLE DBサービス 1](#)
- [キャッシュ 1](#)
- [Cancelメソッド 1](#)
- [オブジェクト名の大/小文字 1](#)
- クラスID
  - [CLSID\\_OraOLEDB 1](#)
- [CLSCTX\\_INPROC\\_SERVERマクロ 1](#)
- [CLSID\\_MSDAINITIALIZEクラス 1](#)
- [CoCreateInstance](#)
  - [データソース・オブジェクトのインスタンスの作成 1](#)
- 列
  - [メタデータ 1](#)
- [コマンド 1](#)
  - [パラメータ 1](#)
  - [準備 1](#)
- [CommandTimeoutプロパティ 1](#)
- [OLE DBサービスとの互換性 1](#)
- コンポーネントの認証
  - [My Oracle Support 1](#)
- 接続
  - [サポートされるOracle Database 1](#)

- Oracle Database [1](#)
  - ADOによるOracle Database [1](#)
  - 特定のデータベース [1](#)
  - 接続文字列属性 [1](#)
    - デフォルト値 [1](#)
    - レジストリ [1](#)
    - 行セット [1](#)
  - コンシューマ
    - OLE DB [1](#)
  - 作成
    - データソース・オブジェクトのインスタンス [1](#)
    - 行セット [1](#)
  - カーソル・スタビリティ [1](#)
  - カスタム・エラー・オブジェクト
    - サポートされるインタフェース [1](#)
- 

## D

- データ・ソース
  - インスタンスの作成 [1](#)
  - 分散トランザクション [1](#)
  - オブジェクト [1](#)
  - プロパティ [1](#)
- データ・ソース情報
  - プロパティ [1](#)
- DataTable [1](#)
- データ型
  - Oracleデータ型とOLE DBの型のマッピング [1](#)
  - 行セットおよびパラメータでのマッピング [1](#)
  - OLE DB [1](#)
  - Oracle [1](#)
- 日付書式
  - NLS\_DATE\_FORMAT [1](#)
  - 設定 [1](#)
- DBNotificationPort [1](#)
- DBNotifications [1](#), [2](#)
- DBPROP\_INIT\_OLEDBSERVICESプロパティ [1](#)
- DBPROP\_SERVERDATAONINSERT property [1](#)
- デバッグ [1](#)
- DeferUpdChk
  - 更新可能性を遅らせるかどうかを示す接続文字列属性 [1](#)
- 設計
  - OLE DB [1](#)
- 分散トランザクション [1](#)

- [分散トランザクション 1](#)
- 

## E

- [EnableCmdTimeoutレジストリ値 1](#)
  - [拡張フェイルオーバー機能 1](#)
  - [登録 1](#)
    - [分散トランザクション 1](#)
  - エラー
    - [HRESULT 1](#)
    - [OLEおよびCOM 1](#)
  - [例 1](#)
    - [ADOによるOracle Databaseへの接続 1](#)
    - [複数の行セットを返すストアド・プロシージャ 1](#)
    - [Visual BasicでのOraOLEDBの使用 1](#)
- 

## F

- 機能
    - [Oracle Provider for OLE DB 1](#)
  - ファイル
    - [Oracle Provider for OLE DBのシステムへのインストール 1](#)
    - [Oracle Provider for OLE DB 1](#)
- 

## G

- [グローバル・トランザクション 1](#)
- 

## H

- HRESULT
    - [エラー・リターン・コード 1](#)
- 

## I

- 初期化と認可
  - [プロパティ 1](#)
- [インストール 1](#)
  - [Oracle Provider for OLE DBのファイル 1](#)
- [インタフェース・コールのトレース 1](#)
- インタフェース
  - [カスタム・エラー・オブジェクト 1](#)
  - [行セット 1](#)

- [セッション 1](#)
  - [Oracle Provider for OLE DBでサポート 1](#)
  - [トランザクションのオプション 1](#)
- 

## L

- [LOBサポート 1](#)
    - [ISequentialStreamインタフェース 1](#)
- 

## M

- [MDAC 1](#)
  - [メタデータ 1](#)
  - [MetaDataCacheSize 1](#)
  - [メタデータのキャッシュ 1](#)
  - [メタデータ列](#)
    - [Oracle Provider for OLE DBでサポート 1](#)
  - [Microsoft Data Access Components 1](#)
  - [Microsoft分散トランザクション・コーディネータ 1](#)
  - [Microsoft Transaction Server 1](#)
  - [MTS, 「Microsoft Transaction Server」を参照 1](#)
  - [My Oracle Support 1](#)
- 

## N

- [NDatatype 1](#)
- 

## O

- [オブジェクト名](#)
  - [大/小文字 1](#)
- [OLE DB](#)
  - [コンシューマ 1](#)
  - [データ型 1](#)
  - [設計 1](#)
  - [MicrosoftのWebサイト 1](#)
  - [プロバイダ 1](#)
- [OLEDB.NET 1, 2](#)
- [OLE DB .NETデータ・プロバイダ 1](#)
  - [互換性 1](#)
- [OLE DBサービス](#)
  - [ADOアプリケーションでの使用 1](#)
  - [C++/COMアプリケーションでの使用 1](#)

- [互換性 1](#)
  - [オペレーティング・システム認証 1](#)
    - [DBPROP\\_INIT\\_PROVIDERSTRING 1](#)
  - Oracle
    - [データ型 1](#)
  - [OracleMetaLink 1](#)
  - Oracle Provider for OLE DB
    - [クラスID 1](#)
    - [機能 1](#)
    - [プロバイダ固有の情報 1](#)
    - [システム要件 1](#)
  - [Oracle Services for Microsoft Transaction Server 1, 2](#)
  - [OraOLEDBセッション 1](#)
- 

## P

- [パラレルDML 1](#)
  - [パスワードの有効期限](#)
    - [接続文字列属性 1](#)
    - [PwdChgDlg 1](#)
  - [パフォーマンス 1](#)
  - [PLSQLRSet 1](#)
  - [プロパティ](#)
    - [データ・ソース 1](#)
    - [データ・ソース情報 1](#)
    - [初期化と認可 1](#)
    - [行セット 1](#)
    - [行セットの含意 1](#)
    - [セッション 1](#)
    - [Oracle Provider for OLE DBでサポート 1](#)
  - [プロバイダ](#)
    - [OLE DB 1](#)
  - [PwdChgDlg](#)
    - [データ・ソースのための接続文字列属性 1](#)
- 

## R

- [レジストリ](#)
  - [デフォルトの属性値 1](#)
- [行セットを返す](#)
  - [ストアド・プロシージャとストアド・ファンクション 1](#)
- [行セット 1](#)
  - [作成 1](#)
  - [ICommandによる作成 1](#)

- IOpenRowsetによる作成 [1](#)
  - 日付書式 [1](#)
  - サポートされるインタフェース [1](#)
  - プロパティ [1](#)
  - プロパティの含意 [1](#)
  - プロシージャとファンクションによって返される [1](#)
  - スキーマ [1](#)
  - IRowsetFindによる検索 [1](#)
  - updatability [1](#)
- 

## S

- スキーマ行セット [1](#)
  - Server Data on Insertプロパティ [1](#)
  - セッション
    - サポートされるインタフェース [1](#)
    - オブジェクト [1](#)
    - プロパティ [1](#)
  - SPPrmDefVal [1](#)
  - SPPrmsLOB [1](#)
  - ストアド・プロシージャとストアド・ファンクション
    - 実行 [1](#)
    - 行セットを返す [1](#)
  - システム要件
    - Oracle Provider for OLE DB [1](#)
- 

## T

- ヒント
    - ADOプログラマ用 [1](#)
  - トレース [1](#)
  - トランザクションのオプション
    - サポートされるインタフェース [1](#)
  - トランザクション
    - 分散 [1](#)
    - グローバル [1](#)
    - 分離レベル [1](#)
    - ローカル [1](#)
    - サポートされているタイプ [1](#)
- 

## U

- UCS-2キャラクタ・セット [1](#)

- Unicode [1](#)
- 

## V

- VB.NET
  - 接続文字列 [1](#)
  - 例 [1](#)