

Oracle Provider for OLE DB

開発者ガイド

リリース 9.2

2002 年 7 月

部品番号 : J06332-01

ORACLE®

Oracle Provider for OLE DB 開発者ガイド, リリース 9.2

部品番号 : J06332-01

原本名 : Oracle Provider for OLE DB Developer's Guide, Release 9.2

原本部品番号 : A95498-01

原本協力者 : Janis Greenberg, Eric Belden, Riaz Ahmed, Kiminari Akiyama, Neeraj Gupta, Sinclair Hsu, Gopal Kirsur, Sunil Mushran, Rajendra Pingte, Helen Slattery

Copyright © 1999, 2002, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、**Oracle Corporation**（米国オラクル）または**日本オラクル株式会社**（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションに用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である **Oracle Corporation**（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『**Restricted Rights**』と共に提供してください。この場合次の **Notice** が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	v
対象読者	vi
このマニュアルの構成	vi
関連資料	vii
表記規則	viii
OraOLEDB の新機能	xiii
Oracle9i リリース 2 (9.2) の Oracle Provider for OLE DB における新機能	xiv
Oracle9i リリース 1 (9.0.1) の Oracle Provider for OLE DB における新機能	xiv
Oracle8i リリース 8.1.7 の Oracle Provider for OLE DB における新機能	xiv
1 Oracle Provider for OLE DB の概要	
OLE DB の概要	1-2
OLE DB の設計	1-2
OraOLEDB の概要	1-3
システム要件	1-4
OraOLEDB のインストール	1-5

2 OraOLEDB の機能

OraOLEDB プロバイダ固有の機能	2-2
データ・ソース	2-2
Oracle データベースへの接続	2-3
OraOLEDB 固有の接続文字列属性	2-3
デフォルトの属性値	2-4
分散トランザクション	2-4
OS 認証	2-4
パスワードの期限	2-5
OraOLEDB セッション	2-6
トランザクション	2-6
コマンド	2-7
ストアド・プロシージャ	2-7
コマンドの準備	2-7
コマンド・パラメータ	2-8
コマンドのための OraOLEDB カスタム・プロパティ	2-8
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	2-11
複数の行セット	2-12
行セット	2-15
行セットの作成	2-15
更新可能性	2-16
Server Data on Insert プロパティ	2-17
IRowsetFind::FindNext による行の検索	2-17
行セットのための OraOLEDB 固有の接続文字列属性	2-17
ADO プログラマのためのヒント	2-18
スキーマ行セット	2-19
日付書式	2-19
オブジェクト名の大 / 小文字	2-20
LOB サポート	2-20
Unicode サポート	2-21
Unicode エンコーディングのタイプ	2-21
Oracle Unicode サポートの機能	2-21
Unicode サポートの設定	2-22
エラー	2-22

OLE DB.NET データ・プロバイダとの互換性	2-23
接続文字列での OLEDB.NET 属性の使用	2-23
OraOLEDB カスタム・プロパティの使用	2-24
DataTable 変更での Oracle の更新	2-24
Visual Basic での OraOLEDB の使用	2-25
Oracle データベースの設定	2-25
Visual Basic プロジェクトの設定	2-26

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	A-2
サポートされるプロパティ	A-3
データ・ソースのプロパティ	A-3
データ・ソース情報のプロパティ	A-3
初期化のプロパティと認可のプロパティ	A-6
セッションのプロパティ	A-6
行セットのプロパティ	A-6
行セットのプロパティの含意	A-10
サポートされるインタフェース	A-11
データ・ソース	A-11
セッション	A-11
コマンド	A-12
行セット	A-12
複数結果	A-13
トランザクションのオプション	A-13
カスタム・エラー・オブジェクト	A-13
サポートされるメタデータ列	A-13
OraOLEDB トレース	A-14
コールをトレースするためのレジストリ設定	A-15

用語集

索引

はじめに

Oracle Provider for OLE DB (OraOLEDB) は、オープンで標準的な方法で Oracle データベースにアクセスします。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。

このマニュアルで説明するのは、Windows NT、Windows 2000、Windows XP および Windows 98 オペレーティング・システムに適用される Oracle9i for Windows ソフトウェアの機能のみです。

次の項目について説明します。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連資料](#)
- [表記規則](#)

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 データベース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 Structured Query Language (SQL) の使用方法についても理解する必要があります。

このマニュアルの構成

このマニュアルは、次のように構成されています。

第 1 章「Oracle Provider for OLE DB の概要」

OLE DB、Oracle Provider for OLE DB (OraOLEDB)、要件およびインストールについて説明します。

第 2 章「OraOLEDB の機能」

OraOLEDB コンポーネント、およびコンシューマ・アプリケーションを開発するための OraOLEDB の使用方法について説明します。

付録 A「プロバイダ固有の情報」

Oracle Provider for OLE DB 固有の OLE DB の情報について説明します。

用語集

関連資料

詳細は、次の Oracle マニュアルを参照してください。

- 『Oracle9i Database for Windows インストレーション・ガイド』
- 『Oracle9i Database for Windows リリース・ノート』
- 『Oracle9i Database for Windows 管理者ガイド』
- 『Oracle Services for Microsoft Transaction Server 開発者ガイド』
- 『Oracle Enterprise Manager 管理者ガイド』
- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle9i データベース新機能』
- 『Oracle9i データベース概要』
- 『Oracle9i データベース・リファレンス』
- 『Oracle9i データベース・エラー・メッセージ』
- 『Oracle9i Database グローバリゼーション・サポート・ガイド』

リリース・ノート、インストール・ドキュメント、ホワイト・ペーパー、またはその他の関連資料を無償でダウンロードするには、OTN-J (Oracle Technology Network Japan) にアクセスしてください。OTN-J を利用する前に、オンライン登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

OTN-J のユーザー名およびパスワードをすでにお持ちの場合は、次の OTN-J の Web サイトのドキュメント・セクションに直接アクセスできます。

<http://otn.oracle.co.jp/document/>

追加情報は、次の Web サイトを参照してください。

<http://www.microsoft.com/data/oledb/>

表記規則

ここでは、このマニュアルの本文およびサンプル・コードで使用される表記規則について説明します。表記規則は次の 3 種類です。

- 本文の表記規則
- サンプル・コードの表記規則
- Windows オペレーティング・システムの表記規則

本文の表記規則

本文中では、特定の用語をより簡単に識別できるように、様々な表記規則を使用しています。次の表は、本文中で使用される表記規則とその使用例を説明したものです。

規則	意味	例
太字	太字は、本文中で定義されている用語、または用語集で説明されている用語、あるいはその両方を示します。	この句を指定する場合、 索引構成表 を作成します。
大文字（固定幅） フォント	大文字固定幅フォントは、システムによって指定される要素を示します。これらの要素には、パラメータ、権限、データ型、Recovery Manager のキーワード、SQL のキーワード、SQL*Plus またはユーティリティのコマンド、パッケージ、メソッドの他に、システムで表示される列名、データベースのオブジェクトおよび構造、ユーザー名およびロールがあります。	この句は NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用して、データベースをバックアップできます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
小文字（固定幅） フォント	小文字固定幅フォントは、実行可能ファイル、ファイル名、ディレクトリ名、およびサンプルのユーザー指定要素を示します。これらの要素には、コンピュータ名およびデータベース名、ネット・サービス名、および接続識別子の他に、ユーザー指定のデータベースのオブジェクトおよび構造、列名、パッケージおよびクラス、ユーザー名およびロール、プログラム・ユニット、およびパラメータ値があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。これらの要素は、記載されているとおりに入力してください。	sqlplus を入力して、SQL*Plus を開きます。 パスワードは、orapwd ファイルで指定されます。 ¥disk1¥oracle¥dbs ディレクトリのデータ・ファイルと制御ファイルをバックアップします。 department_id、department_name および location_id 列は、hr.departments 表にあります。 QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。 oe ユーザーとして接続します。 JRepUtil クラスは、これらのメソッドを実装します。

規則	意味	例
小文字イタリック (固定幅) フォント	小文字イタリック固定幅フォントは、プ レースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。 <i>Uold_release</i> .SQL を実行します。 <i>old_release</i> は、アップグレード前にインス トールしたリリースを表します。

サンプル・コードの表記規則

サンプル・コードは、SQL、PL/SQL、SQL*Plus またはその他のコマンドライン文を示しま
す。これらは固定幅フォントで示され、次の例のように、通常の本文とは区別されていま
す。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表は、サンプル・コードで使用する表記規則とそれらの使用例を説明したものです。

規則	意味	例
[]	大カッコは、1 つ以上のオプション項目を囲 みます。大カッコは入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコは複数の項目を囲み、そのうちの 1 つが必要であることを示します。中カッコ は入力しないでください。	{ENABLE DISABLE}
	縦線は、大カッコまたは中カッコ内にある 複数のオプションの選択肢を区切るために 使用します。オプションの 1 つを入力しま す。縦線は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示しま す。 <ul style="list-style-type: none">■ 例に直接関係のないコードの一部を省 略■ コードの一部の繰り返しが可能	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	垂直の省略記号は、例に直接関係のない コードの数行を省略したことを示します。	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.

規則	意味	例
その他の表記規則	大カッコ、中カッコ、縦線および省略記号以外の記号は、示されているとおりに入力してください。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システムによって指定される要素を示します。ユーザーが定義する語句と区別するために、大文字で示しています。語句が大カッコ内に表示されている場合を除き、記載されているとおりの順序とスペルで入力します。ただし、これらの語句には大文字と小文字の区別がないため、小文字で入力できます。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
小文字	小文字は、ユーザーが指定するプログラム要素を示します。たとえば、小文字は表、列またはファイルの名前を示します。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。これらの要素は、記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mJones IDENTIFIED BY ty3MU9;

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則とその使用例を説明したものです。

規則	意味	例
「スタート」→を選択	プログラムの起動方法。たとえば、Database Configuration Assistant を起動するには、タスクバーの「スタート」ボタンをクリックし、「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。	「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。
ファイル名およびディレクトリ名	ファイルおよびディレクトリ名には、大文字と小文字の区別がありません。＜、＞、：、"、/、 、および-の特殊文字は使用できません。特殊文字¥は、引用符に囲まれている場合でも、要素の区切り文字として扱われます。ファイル名が¥¥で始まる場合、Windows では汎用命名規則を使用しているものと認識されます。	c:¥winnt"¥"system32 は、C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行のハード・ディスク・ドライブの Windows コマンド・プロンプトを示します。コマンド・プロンプトのエスケープ文字は、カレット (^) です。プロンプトは、現在作業中のサブディレクトリを示しています。このマニュアルでは、コマンド・プロンプトと呼びます。	C:¥oracle¥oradata>
特殊文字	特殊文字の円記号 (¥) は、Windows コマンド・プロンプトで特殊文字の二重引用符 (") のエスケープ文字として必要な場合があります。カッコおよび特殊文字の一重引用符 (') は、エスケープ文字を必要としません。エスケープ文字および特殊文字の詳細は、Windows オペレーティング・システムのドキュメントを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を示します。 ホーム名は、英数字 16 文字までです。ホーム名で利用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

規則	意味	例
ORACLE_HOME および ORACLE_BASE	<p>Oracle8 リリース 8.0 以下のリリースでは、Oracle コンポーネントをインストールすると、サブディレクトリはすべて、最上位の ORACLE_HOME ディレクトリ（デフォルトでは次のとおり）の下に置かれました。</p> <ul style="list-style-type: none">■ Windows NT の場合は C:¥orant■ Windows 98 の場合は C:¥orawin98 <p>あるいは、Oracle ホームと呼ばれるディレクトリの下に置かれました。</p> <p>今回のリリースは、Optimal Flexible Architecture (OFA) に準拠しています。すべてのサブディレクトリが最上位の ORACLE_HOME ディレクトリの下にあるわけではありません。ORACLE_BASE という最上位ディレクトリがあり、デフォルトは C:¥oracle です。コンピュータに最新の Oracle リリースをインストールし、他の Oracle ソフトウェアをインストールしない場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は、C:¥oracle¥orann です。nn は、最新のリリース番号です。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下に置かれます。</p> <p>このマニュアルでは、ディレクトリ・パスの例は、すべて OFA 表記規則に準拠しています。</p>	%ORACLE_HOME%¥rdbms¥admin ディレクトリに移動します。

OraOLEDB の新機能

次の各項では、OraOLEDB の新機能について説明します。

- [Oracle9i リリース 2 \(9.2\) の Oracle Provider for OLE DB における新機能](#)
- [Oracle9i リリース 1 \(9.0.1\) の Oracle Provider for OLE DB における新機能](#)
- [Oracle8i リリース 8.1.7 の Oracle Provider for OLE DB における新機能](#)

Oracle9i リリース 2 (9.2) の Oracle Provider for OLE DB における新機能

次の項目について説明します。

- OLE DB .NET データ・プロバイダでの OraOLEDB の使用のサポート

ADO .NET アプリケーション開発者は、OLE DB .NET データ・プロバイダを介して OraOLEDB を使用できます。接続属性 OLEDB .NET は、接続時に OraOLEDB が OLE DB .NET データ・プロバイダと互換性を持つよう設定できます。2-23 ページの「[OLE DB .NET データ・プロバイダとの互換性](#)」を参照してください。

Oracle9i リリース 1 (9.0.1) の Oracle Provider for OLE DB における新機能

次の項目について説明します。

- Windows 2000 での Oracle9i の使用

Windows 2000 と Windows NT 4.0 とでは、Oracle9i の使用方法が多少異なります。

関連資料：『Oracle9i Database for Windows スタート・ガイド』

Oracle8i リリース 8.1.7 の Oracle Provider for OLE DB における新機能

Oracle8i リリース 8.1.7 では、次の機能が追加されました。

- 複数の行セットを返す機能のサポート

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。2-12 ページの「[複数の行セット](#)」を参照してください。

- Unicode キャラクタ・セットのサポート

コンシューマは、この機能を使用して、OraOLEDB で同一クライアント・コンピュータ上の複数の言語のデータにアクセスできます。この機能は、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、Oracle8i データベースにアクセスする Active Server Pages (ASP) を 1 つ作成するだけで、日本語、アラビア語、英語およびタイ語のコンテンツを動的に生成することが可能です。2-21 ページの「[Unicode サポート](#)」および A-2 ページの「[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。

Oracle Provider for OLE DB の概要

この章では、Oracle Provider for OLE DB（OraOLEDB）の概要を示します。

この章の項目は次のとおりです。

- [OLE DB の概要](#)
- [OraOLEDB の概要](#)
- [システム要件](#)
- [OraOLEDB のインストール](#)

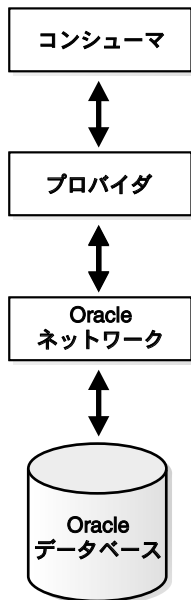
OLE DB の概要

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 「OLE DB の流れ」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 응답してデータを表形式で用意します。プロバイダは単純にも複雑にもできます。**プロバイダ** は表を返すことができます。また、コンシューマに表の形式を決定させることや、データを操作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 インタフェー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ます。

Oracle Provider for OLE DB のサンプル・コード、最新のパッチおよびその他の技術情報は、次の Web サイトを参照してください。

<http://otn.oracle.co.jp/>

.NET フレームワークの出現により、OraOLEDB での OLE DB .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、OLE DB .NET データ・プロバイダで OraOLEDB を利用して Oracle にアクセスできます。

関連項目： OLE DB .NET データ・プロバイダのサポートの詳細は、2-23 ページの「[OLE DB .NET データ・プロバイダとの互換性](#)」を参照してください。

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 98、Windows NT 4.0、Windows 2000 または Windows XP
- Oracle サーバー（リリース 8.0 以上）へのアクセス
- **Oracle Net Services**
- Microsoft Data Access Components (MDAC) 2.1 以上で提供される再配布可能ファイル。プロバイダ用に必要です。ファイルは次の Microsoft の Web サイトから入手できます。

<http://www.microsoft.com/data/oledb/>

- Oracle Services for Microsoft Transaction Server（リリース 9.0 およびリリース 9.2）。Microsoft Transaction Server (MTS) または COM+ を使用するコンシューマに必要です。

注意： Oracle Services for Microsoft Transaction Server がインストールされている場合、OraOLEDB では Oracle9i リリース 1 (9.0.1)、Oracle8i (リリース 8.1.5 以上) および Oracle8 (リリース 8.0.6 以上) のデータベース・リリースで、MTS がサポートされます。

OraOLEDB のインストール

Oracle Provider for OLE DB は、Oracle のインストール時に一緒にインストールできます。Oracle Provider for OLE DB には、機能、およびこの製品を使用した実際の問題の解決方法を示すデモが含まれています。

関連資料： インストール手順は、『Oracle9i Database for Windows インストール・ガイド』を参照してください。

インストール・プロセス中に、表 1-1 にリストされているファイルがシステムにインストールされます。

表 1-1 Oracle Provider for OLE DB ファイル

ファイル	説明	場所
OraOLEDB.dll	Oracle Provider for OLE DB	%ORACLE_HOME%\bin
OraOLEDBrfc.dll	Oracle 行セット・ファイル・キャッシュ・マネージャ	%ORACLE_HOME%\bin
OraOLEDBrmc.dll	Oracle 行セット・メモリー・キャッシュ・マネージャ	%ORACLE_HOME%\bin
OraOLEDBrst.dll	Oracle 行セット	%ORACLE_HOME%\bin
OraOLEDBgmr.dll	Oracle ODBC SQL パーサー	%ORACLE_HOME%\bin
OraOLEDBlang.dll (lang は対応する言語)	言語固有のリソース DLL	%ORACLE_HOME%\bin
OraOLEDBpus.dll	プロパティの説明	%ORACLE_HOME%\bin
OraOLEDButl.dll	OraOLEDB ユーティリティ DLL	%ORACLE_HOME%\bin
OraOLEDB.tlb	OraOLEDB タイプ・ライブラリ	%ORACLE_HOME%\bin
OraOLEDB.h	OraOLEDB ヘッダー・ファイル	%ORACLE_HOME%\oledb\include
OraOLEDB.lib	OraOLEDB ライブラリ・ファイル	%ORACLE_HOME%\oledb\lib
OraOLEDBlang.msb (lang は対応する言語)	言語固有のメッセージ・ファイル	%ORACLE_HOME%\oledb\mesg
README および ドキュメント・ファイル	リリース・ノートおよびオンライン・マニュアル	%ORACLE_HOME%\oledb\doc
サンプル・ファイル	サンプル・コード	%ORACLE_HOME%\oledb\samples

OraOLEDB の機能

この章では、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。

この章の項目は次のとおりです。

- 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](#)
- [LOB サポート](#)
- [Unicode サポート](#)
- [Visual Basic での OraOLEDB の使用](#)
- [OLE DB .NET データ・プロバイダとの互換性](#)

その他のプロバイダ固有の機能の詳細は、[付録 A「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 データベース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、CLSID_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);
```

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、リリース 7.3.4 以上の Oracle データベ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_COMPLETE_REQUIRED 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。

Oracle データベースへの接続

OraOLEDB を使用して Oracle データベースに接続するには、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。詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

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。それらは、プロバイダおよびユーザー ID の設定方法と同様の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。2-17 ページの「[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。2-17 ページの「[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。2-4 ページの「[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。2-17 ページの「[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。

- **OSAuthent:** Oracle データベースに接続するときに、OS 認証が使用されるかどうかを指定します。2-4 ページの「[OS 認証](#)」を参照してください。
- **PLSQLRSet:** [PL/SQL](#) ストアド・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。2-8 ページの「[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。2-5 ページの「[パスワードの期限](#)」を参照してください。
- **OLEDB.NET:** OLE DB .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。2-23 ページの「[OLE DB .NET データ・プロバイダとの互換性](#)」を参照してください。

デフォルトの属性値

これらの属性のデフォルト値は、¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥OLEDB レジストリ・キーの下にあります。

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;"
```

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 Transaction Server 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。

OS 認証

OSAuthent 属性は、Oracle データベースに接続するときに、OS 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、OS 認証が使用されないことを示します。

OS 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。この詳細と Windows NT クライアントの設定方法は、『Oracle9i for Windows セキュリティおよびネットワーク統合ガイド』で、Windows NT でのデータベース・ユーザーの認証に関する情報を参照してください。

Windows NT クライアントが OS 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 に "/" を設定
- DBPROP_INIT_PROVIDERSTRING に "OSAuthent=1;" を設定
- レジストリの OSAuthent に "1" を設定

パスワードの期限

Oracle9i にはパスワードに期限を設定する機能があります。データベース管理者は、この機能を使用して、パスワードを定期的に変更するようユーザーに要求できます。PwdChgDlg 属性は、パスワードが期限切れになったためにログオンできないときに、パスワード変更ダイアログ・ボックスを表示可能または不可にします。使用可能の場合、プロバイダはパスワードを変更するダイアログ・ボックスを表示します。使用不可の場合、ログオンは失敗してエラー・メッセージが表示されます。有効値は 0（使用不可）と 1（使用可能）です。デフォルトは 1（使用可能）です。パスワードに期限を設定する機能の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

例： ADO による Oracle データベースへの接続

次の例で、OraOLEDB および ADO による Oracle データベース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;PLSQLSet=1;Data Source=MyOraDb;" & _
    "User ID=scott;Password=tiger;"
con.Open
```

ユーザー ID に "/" を設定する OS 認証接続

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

OSAuthent を使用する OS 認証接続

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

OraOLEDB セッション

OraOLEDB セッション・オブジェクトは、Oracle データベース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 サーバーへの、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、明示的にコミットおよび異常終了する、ローカル・トランザクションおよび分散トランザクショ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、コミット読み込み（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 9.0.1 以上をインストールする必要があります。Oracle Services for MTS の設定および構成の詳細は、『Oracle Services for Microsoft Transaction Server 開発者ガイド』を参照してください。

`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。

コマンド

OraOLEDB は、Oracle 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 のストアド・プロシージャおよびストアド・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す。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

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いずれかのパラメータが Oracle の N データ型 (NCHAR、NVARCHAR または NCLOB) かどうかを指定できます。この情報は、OraOLEDB がパラメータを適切に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 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 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 **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 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 NVARCHAR 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 NVARCHAR 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たパラメータの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 datatype
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

```

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

Oracle Provider for OLE DB によって、コンシューマは、REF CURSOR 型の引数を含む PL/SQL ストアド・プロシージャ、または REF CURSOR を返すストアド・ファンクションを実行できるようになります。

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 はそれらの引数を適切にバインドし、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 はそれらの引数を適切にバインドし、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。

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、ストアード・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、ストアード・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ード・プロシージャ

PL/SQL のパッケージ

```
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
```

行セット

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。 `NULL` は使用できません。
- `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、すべての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または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 に TRUE が設定されていると、行セットを使用する挿入と更新の両方のパフォーマンスが低下する可能性があります。これは、OraOLEDB がデータベースから、新しく挿入および更新された行の行データを取り出すためです。ただし、DBPROP_SERVERDATAONINSERT にデフォルト値である FALSE が設定されている場合は、それらの行の列の値が要求されたときに、挿入および更新のために明示的に指定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て挿入および更新を実行するとき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、DBCOMPAREOPS_EQ 比較と DBCOMPAREOPS_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・データをメモリー内に格納します。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を必要とします。有効値は 1 ～ 65535 です。デフォルトは 100 です。
- **FetchSize:** プロバイダが 1 度に取り出す行数（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従って、適切な値に設定する必要があります。この値が大きすぎると、問合せを実行したときの待機時間が長くなることがあります。この値が小さすぎると、データベースへの通信回数が増えることがあります。有効値は 1 ～ 429,496,296 です。デフォルトは 100 です。

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。詳細は、2-4 ページの「[デフォルトの属性値](#)」を参照してください。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
```

ADO プログラマのためのヒント

ADO Rowset プロパティ `LockType` に `adLockPessimistic` を設定しても、Oracle Provider for OLE DB ではサポートされません。`LockType` に `adLockPessimistic` が設定されている場合でも、OraOLEDB は `adLockOptimistic` が設定されているときと同様に機能します。これは、OraOLEDB が変更されている行に対して明示的なロックを実行しないためです。ただし、データベースに新しくデータを更新するとき、その行セット・データがまだ別のユーザーによって更新されていない場合にかぎり更新が実行されます。内容を保証しない書き込みは許可されません。`LockType` の値としては、`adLockReadOnly`、`adLockBatchOptimistic` および `adLockOptimistic` が OraOLEDB によって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 が文レベルの読み取り一貫性をサポートするため、OraOLEDB は 2 つのどちらもサポートしません。この文レベルの読み取り一貫性によって、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みのデータのみが含まれることが保証されます。`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、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 に渡す場合、YYYY-MM-DD HH24:MI:SS の形式にする必要があります。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 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 でオブジェクト名の大 / 小文字が区別されます。

LOB サポート

すべての LONG 型、LONG RAW 型および LOB 型（BLOB 型、CLOB 型、NCLOB 型および BFILE 型）列のために、`ISequentialStream` インタフェースがサポートされます。コンシューマは、すべての LOB 型列の読取りおよび書込みにこのインタフェースを使用できます。ただし、BFILE 型は読取り専用です。これらの列に読取りおよび書込みアクセスするには、行セットの作成に使用する SQL の SELECT 文の中に結合を含めないでください。

注意： Oracle データベースの LOB 型列のほとんどが最大 4GB のデータ記憶域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 型の詳細は、『Oracle9i アプリケーション開発者ガイド - ラージ・オブジェクト』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、OraOLEDB で同クライアント・コンピュータ上の複数の言語のデータにアクセスできます。この機能は、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、Oracle9i データベースにアクセスする ASP を 1 つ作成するだけで、日本語、アラビア語、英語およびタイ語のコンテンツを動的に生成することが可能です。

Unicode エンコーディングのタイプ

Oracle データベースは、Unicode データを UTF-8 コード体系で格納します。このコード体系は、Unicode の ASCII 互換のマルチバイト・エンコーディングです。Microsoft Windows 2000 および Windows NT 4.0 は、2 バイトの固定幅コード体系である UCS2 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 UCS2 のみ扱えるようにします。

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/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は、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 UCS2 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 UTF-8 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、UCS2 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、A-2 ページの「[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 UTF-8 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 データベースが UTF-8 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 UTF-8 でない場合は、UTF-8 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。詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

関連資料： 全般的な情報は、『Oracle9i Database グローバリゼーション・サポート・ガイド』を参照してください。

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 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、続いてメソッド GetErrorDescription() 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
```

OLE DB .NET データ・プロバイダとの互換性

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

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、次の例に示すように、ADO コマンド・オブジェクト・レベルでのみ設定できます。

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

一方、OLE DB .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、SPPrmsLOB プロパティおよび NDatatype プロパティは、OraOLEDB が OLE DB .NET データ・プロバイダで使用される場合に接続文字列属性としてのみ設定できます。

```
// in VB.NET
Dim con As New OleDbConnection()
con.ConnectionString = "Provider=OraOLEDB.Oracle;User Id=scott;" & _
    "Password=tiger;Data Source=Oracle;OLEDB.NET=True;" & _
    "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。このため、このどちらかの属性を True に設定する前に、2-8 ページの「[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 を正しく更新するには、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。

関連資料： 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 データベースに接続し、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 データベースの設定

この例では、Oracle データベースに、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、demobld.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 データベースに接続するときに、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。

手順 1 サンプル表を作成します。

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. demobld.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\sqlplus\demo\demobld.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ド・プロシージャとストアド・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。

手順 2 PL/SQL パッケージを作成します。

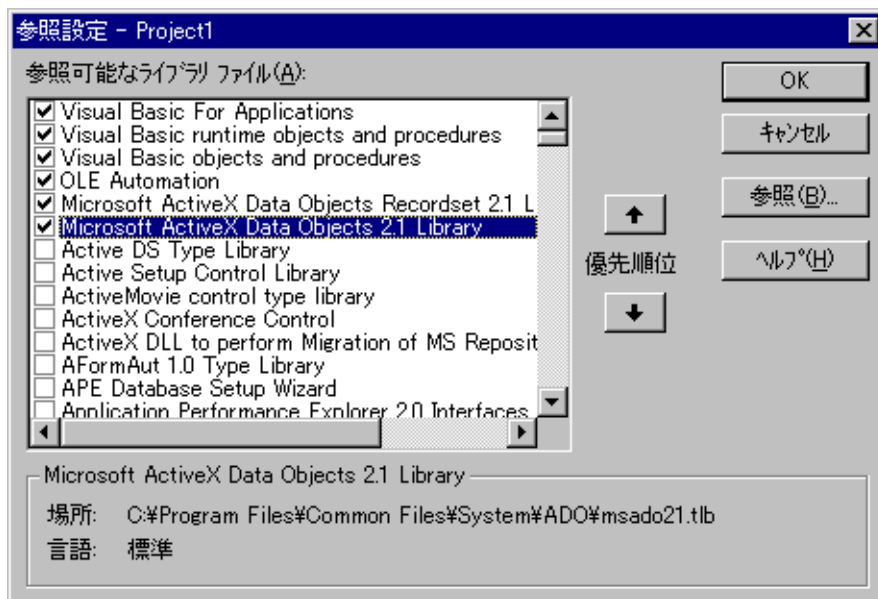
1. SQL*Plus を起動します。
2. ユーザー名 scott とパスワード tiger を使用して接続します。
3. 2-12 ページの「PL/SQL のパッケージ」で示した、PL/SQL パッケージを作成します。

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。

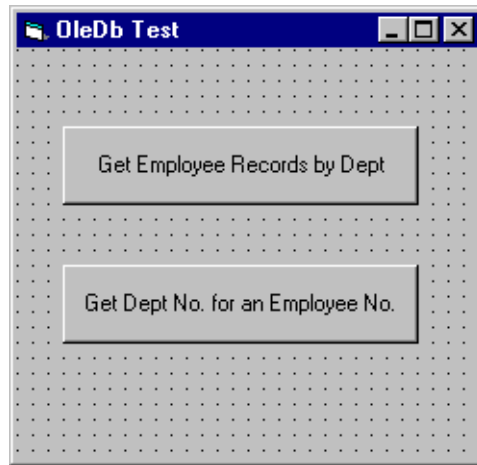
Visual Basic プロジェクトの設定

Oracle データベース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, 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
```

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, 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

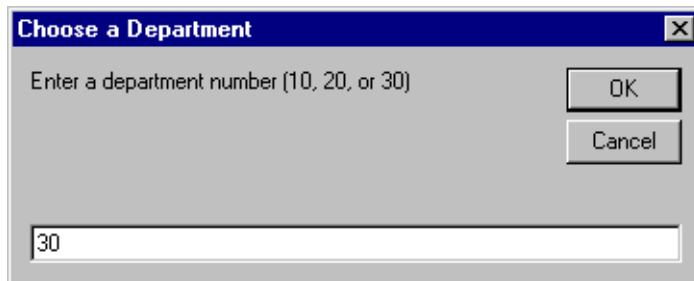
' Disable PLSQLRSet property
Cmd.Properties ("PLSQLRSet") = FALSE

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

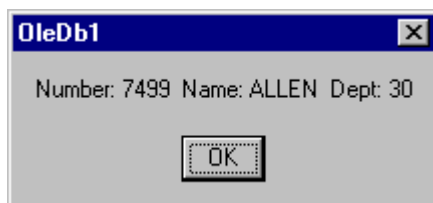
Exit Sub

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

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。



プロバイダ固有の情報

この付録では、Oracle Provider for OLE DB 固有の OLE DB の情報について説明します。すべての OLE DB の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 全般に関しては、『Microsoft OLE DB Programmer's Reference and Data Access SDK』を参照してください。

この付録の項目は次のとおりです。

- [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 で定義される型の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 データ型は、表 A-1 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 データベース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
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
LONG	DBTYPE_STR	DBTYPE_WSTR
LONG RAW	BTYPE_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
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。
読取り / 書込みの状態と初期値も示します。

- データ・ソースのプロパティ
- データ・ソース情報のプロパティ
- 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

データ・ソースのプロパティ

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

データ・ソース情報のプロパティ

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_ASYNCCTXNABORT	読取り専用	VARIANT_FALSE
DBPROP_ASYNCCTXNCOMMIT	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

表 A-3 DBPROPSET_DATASOURCEINFO プロパティ (続き)

プロパティ	状態	初期値
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
DBPROP_HETEROGENEOUSTABLES	読取り専用	DBPROPVAL_HT_DIFFERENT_CATALOGS
DBPROP_IDENTIFIER_CASE	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、サポートなし
DBPROP_MAXORSINFILTER	読取り専用	0、サポートなし
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、サポートなし
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
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	OraOLEDB.dll
DBPROP_PROVIDEROLEDBVER	読取り専用	"02.01"

表 A-3 DBPROPSET_DATASOURCEINFO プロパティ (続き)

プロパティ	状態	初期値
DBPROP_PROVIDERVER	読取り専用	現在の OraOLEDB パージョンに設定
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_SUPPORTEDTXNDL	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
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 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 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 DBPROPSET_ROWSET プロパティ

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

表 A-6 DBPROPSET_ROWSET プロパティ (続き)

プロパティ	状態	初期値
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、現在は操作不可能
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 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

表 A-6 DBPROPSET_ROWSET プロパティ (続き)

プロパティ	状態	初期値
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
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、制限なし
DBPROP_MAXPENDINGROWS	読取り専用	0、制限なし
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 DBPROPVAL_NP_DIDEVENT

表 A-6 DBPROPSET_ROWSET プロパティ (続き)

プロパティ	状態	初期値
DBPROP_NOTIFYROWRESYNCH	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETRELEASE	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWUNDOCHANGE	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODO DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDODELETE	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODO DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDOINSERT	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODO DBPROPVAL_NP_DIDEVENT
DBPROP_NOTIFYROWUNDOUPDATE	読取り / 書込み	DBPROPVAL_NP_OKTODO DBPROPVAL_NP_ABOUTTODO DBPROPVAL_NP_SYNCHAFTER DBPROPVAL_NP_FAILEDTODO 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
DBPROP_OWNINGUPDATEDELETE	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

表 A-6 DBPROPSET_ROWSET プロパティ (続き)

プロパティ	状態	初期値
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

特定のプロパティに VARIANT_TRUE が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にも VARIANT_TRUE 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 が設定されます。
 - 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、次のプロパティにも VARIANT_TRUE が設定されます。
 - DBPROP_IROWSETCHANGE

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](#)
- [トランザクションのオプション](#)
- [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 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

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_GUID
- 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、オラクル社カスタマ・サポート・センター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（入力と出力）
- 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。A-15 ページの「[TraceLevel](#)」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

インタフェース・コールをトレースするには、
¥HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥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。

■ TraceFileName

有効値： 任意の有効パスおよびファイル名

TraceFileName は、トレース情報のログに使用するファイル名を指定します。
TraceOption が 0 に設定されている場合は、名前がそのまま使用されます。一方、
TraceOption が 1 の場合は、指定されたファイル名にスレッド ID が追加されます。
詳細は、「[TraceOption](#)」を参照してください。

■ TraceCategory

有効値：

- 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。有効値を単純に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、単純に有効値を加

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。単一のトレース・ファイルが指定されている場合は、TraceFileName で指定されたファイル名が使用されます。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。

Oracle Net Services

Oracle クライアント / サーバー通信ソフトウェア。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 ツールまたはデータベースへの透過的操作を提供する。

PL/SQL

SQL を拡張した、オラクル社の手続き型言語。

コンシューマ (consumer)

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。「[プロバイダ](#)」を参照。

ストアド・プロシージャ (stored procedure)

ストアド・プロシージャとは、Oracle がデータベースに格納する PL/SQL ブロックのことで、アプリケーションから名前でもコールすることが可能。

プロバイダ (provider)

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。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う。「[コンシューマ](#)」を参照。

A

ActiveX Data Objects, vi
ADO, 2-5, 2-13

C

C#

接続文字列, 2-23
例, 2-23

CacheType

行セットのための接続文字列属性, 2-3, 2-17

ChunkSize

行セットのための接続文字列属性, 2-3, 2-18

CLSCTX_INPROC_SERVER マクロ, 2-2

CoCreateInstance

データ・ソース・オブジェクトのインスタンスの作成, 2-2

COM

Component Object Model, vi

Component Object Model (COM), vi

CursorType

ADO プログラマのためのヒント, 2-18

D

DataTable, 2-24

DBPROP_AUTH_PASSWORD プロパティ
設定, 2-3

DBPROP_AUTH_USERNAME プロパティ
OS 認証を使用可能に設定, 2-5
設定, 2-3

DBPROP_INIT_DATASOURCE プロパティ
設定, 2-3

DBPROP_INIT_PROMPT プロパティ
設定, 2-3

DBPROP_INIT_PROVIDERSTRING プロパティ
OS 認証を使用可能に設定, 2-5
設定, 2-4

DBPROP_IROWSETUPDATE プロパティ
その他のプロパティの設定, A-11

DBPROP_SERVERDATAONINSERT プロパティ, 2-17

DBPROPSET_DBINIT プロパティ・セット
プロパティの設定, 2-3, 2-4

demobld.sql, 2-25

DistribTX

コマンドのための接続文字列属性, 2-3

F

FetchSize

行セットのための接続文字列属性, 2-3, 2-18

H

HRESULT

エラー・リターン・コード, 2-22

L

LOB サポート, 2-20

ISequentialStream インタフェース, 2-20

LockType

ADO プログラマのためのヒント, 2-18

M

MDAC, 1-4
Microsoft Data Access Components, 1-4
Microsoft Transaction Server, 2-7
Microsoft 分散トランザクション・コーディネータ,
2-7
MTS, 「Microsoft Transaction Server」を参照

N

NDatatype, 2-9
.NET, 2-23

O

OLE DB
 Microsoft の Web サイト, 1-4
 コンシューマ, 1-3
 設計, 1-2
 データ型, A-2
 プロバイダ, 1-3
OLE DB .NET データ・プロバイダ, 2-4
 互換性, 2-23
OleDbDataAdapter.Update(), 2-24
OLEDB.NET, 2-4, 2-23
Oracle
 データ型, A-2
Oracle Provider for OLE DB
 機能, 2-2
 クラス ID, 2-2
 システム要件, 1-4
 対象読者, vi
 プロバイダ固有の情報, A-1
Oracle Services for Microsoft Transaction Server, 1-4,
2-7
OraOLEDB
 「Oracle Provider for OLE DB」を参照
 インストール, 1-5
OraOLEDB セッション, 2-6
OSAuthent
 OS 認証を使用可能に設定, 2-5
 データ・ソースのための接続文字列属性, 2-4
OS 認証, 2-4
 DBPROP_INIT_PROVIDERSTRING, 2-4

P

PLSQLRSet, 2-8
 コマンドのための接続文字列属性, 2-4
PwdChgDlg
 コマンドのための接続文字列属性, 2-4
 データ・ソースのための接続文字列属性, 2-5

R

ROWID, 2-24

S

Server Data on Insert プロパティ, 2-17
SPPrmsLOB, 2-10

T

TraceCategory, A-15
TraceFileName, A-15
TraceLevel, A-15

U

UCS2 キャラクタ・セット, 2-21
Unicode, 2-21

V

VB.NET
 接続文字列, 2-23
 例, 2-23

い

インストール, 1-5
 Oracle Provider for OLE DB のファイル, 1-5
 OraOLEDB, 1-5
インタフェース
 Oracle Provider for OLE DB でサポート, A-11
 カスタム・エラー・オブジェクト, A-13
 行セット, A-12
 セッション, A-11
 トランザクションのオプション, A-13
インタフェース・コールのトレース, A-14

え

エラー

- HRESULT, 2-22
- OLE と COM, 2-22

お

オブジェクト名

- 大 / 小文字, 2-20
- オブジェクト名の大 / 小文字, 2-20

か

- カーソル・スタビリティ, 2-7
- 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, A-13

き

機能

- Oracle Provider for OLE DB, 2-2
- 機能, 新しい, xiii
- 行セット, 2-15
 - ICommand による作成, 2-15
 - IOpenRowset による作成, 2-15
 - IRowsetFind による検索, 2-17
 - 更新可能性, 2-16
 - 作成, 2-15
 - サポートされるインタフェース, A-12
 - スキーマ, 2-19
 - 日付書式, 2-19
 - プロシージャとファンクションによって返される, 2-11
 - プロパティ, A-6
 - プロパティの含意, A-10
- 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, 2-11

く

クラス ID

- CLSID_OraOLEDB, 2-2

こ

コマンド

- 準備, 2-7
- パラメータ, 2-8
- コンシューマ
- OLE DB, 1-3

さ

作成

- 行セット, 2-15
- データ・ソース・オブジェクトのインスタンス, 2-2
- サンプル表
- 作成, 2-25

し

システム要件

- Oracle Provider for OLE DB, 1-4
- 初期化と認可
- プロパティ, A-6
- 新機能, xiii

す

- スキーマ行セット, 2-19
- ストアド・プロシージャとストアド・ファンクション
- 行セットを返す, 2-11
- 実行, 2-7

せ

設計

- OLE DB, 1-2
- セッション
- オブジェクト, 2-6
- サポートされるインタフェース, A-11
- プロパティ, A-6
- 接続
- ADO による Oracle データベースへの, 2-5
- Oracle データベース, 2-3
- サポートされる Oracle データベース, 2-3
- 特定のデータベース, 2-3
- 接続文字列属性, 2-3
- 行セット, 2-17

デフォルト, 2-4
レジストリ, 2-4

そ

属性
接続文字列, 2-3

て

データ型
OLE DB, A-2
Oracle, A-2
Oracle データ型と OLE DB の型のマッピング, A-2
行セットおよびパラメータでのマッピング, A-2
データ・ソース
インスタンスの作成, 2-2
オブジェクト, 2-2
プロパティ, A-3
分散トランザクションの登録, 2-4
データ・ソース情報
プロパティ, A-3
デバッグ, A-14

と

登録
分散トランザクション, 2-4
トランザクション
サポートされるタイプ, 2-6
分散, 2-7
分離レベル, 2-7
ローカル, 2-7
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, A-13
トレース, A-14

は

パスワードの期限
PwdChgDlg, 2-5
接続文字列属性, 2-5

ひ

日付書式
NLS_DATE_FORMAT, 2-19
設定, 2-19
ヒント
ADO プログラマのための, 2-18

ふ

ファイル
Oracle Provider for OLE DB, 1-5
Oracle Provider for OLE DB 用にシステムにインス
トールされる, 1-5
プロバイダ
OLE DB, 1-3
プロパティ
Oracle Provider for OLE DB でサポート, A-3
行セット, A-6
行セットの含意, A-10
初期化と認可, A-6
セッション, A-6
データ・ソース, A-3
データ・ソース情報, A-3
分散トランザクション, 2-4, 2-7

め

メタデータ列
Oracle Provider for OLE DB でサポート, A-13

れ

例, 2-23
ADO による Oracle データベースへの接続, 2-5
Visual Basic での OraOLEDB の使用, 2-25
複数の行セットを返すストアード・プロシージャ,
2-12
レジストリ
デフォルトの属性値, 2-4
列
メタデータ, A-13