

Oracle® Database

2日で.NET開発者ガイド

11gリリース2 (11.2)

部品番号: B56266-01

2009年11月

Oracle Database 2日で.NET開発者ガイド, 11gリリース2 (11.2)

部品番号: B56266-01

Oracle Database 2 Day + .NET Developer's Guide, 11g Release 2 (11.2)

原本部品番号: E10767-01

原本著者: Janis Greenberg, Roza Leyderman

原本協力者: John Paul Cook, Mark Williams, Alex Keh, Christian Shay

Copyright © 2006, 2009, Oracle and/or its affiliates. All rights reserved.

制限付権利の説明

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が、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される場合は、次のNoticeが適用されます。

U. S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフト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。 このソフトウェアを危険が伴うアプリケーションで使用する際、このソフトウェアを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。

このソフトウェアおよびドキュメントは、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。 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。 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

[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](#)

[1 概要](#)

- [このマニュアルについて](#)
- [Microsoft .NET Frameworkの概要](#)
- [Oracle Data Provider for .NETの概要](#)
- [Oracle Developer Tools for Visual Studioの概要](#)
- [.NETストアド・プロシージャの概要](#)
- [Oracle Providers for ASP.NETの概要](#)

[2 .NET製品のインストール](#)

- [必要事項](#)
 - [Oracle Database](#)
 - [サンプル・データ](#)
 - [Oracle Data Access Components](#)
 - [Oracle Database Extensions for .NET](#)
 - [Visual Studioのバージョン](#)
- [.NET製品のインストール](#)
- [NET接続の別名の構成](#)
- [Oracle Providers for ASP.NETのセットアップ](#)
 - [Oracle Providers for ASP.NETデータベース・ユーザーのセットアップ](#)
 - [ユーザーの作成および権限の付与](#)
 - [すべてのOracle Providers for ASP.NETの構成](#)
 - [個別のOracle Providers for ASP.NETの構成](#)
 - [Oracle Providers for ASP.NET用スキーマのアンインストール](#)
 - [接続文字列の設定](#)
 - [様々なセットアップに対応するためのOracle Providers for ASP.NETのカスタマイズ](#)

[3 ODP.NETによる単純な.NETアプリケーションの作成](#)

- [新しいプロジェクトの作成](#)
- [参照の追加](#)
- [名前空間ディレクティブの追加](#)
- [ユーザー・インタフェースの設計](#)
- [接続コードの記述](#)
- [アプリケーションのコンパイルと実行](#)
- [エラー処理](#)
 - [Try-Catch-Finallyブロック構造の使用](#)
 - [一般的なエラーの処理](#)
 - [一般的なOracleエラーの処理](#)

[4 Oracle Data Provider for .NETでの取得と更新](#)

- [コマンド・オブジェクトの使用](#)
- [データの取得: 単純な問合せ](#)
- [データの取得: バインド変数](#)
- [データの取得: 複数の値](#)
- [Oracle Data Provider for .NETでのDataSetクラスの使用](#)
- [データベースの更新の有効化](#)
- [データの挿入、削除および更新](#)

[5 Oracle Developer Tools for Visual Studioの使用](#)

- [Oracle Developer Toolsの使用](#)
- [Oracle Databaseへの接続](#)
- [表および表の列の作成](#)
- [表の索引の作成](#)
- [表の制約の追加](#)
- [表へのデータの追加](#)
- [データを表示および更新するためのコードの自動生成](#)

[6 PL/SQLストアド・プロシージャおよびREF CURSORの使用](#)

- [PL/SQLストアド・プロシージャの概要](#)
- [PL/SQLパッケージとパッケージ本体の概要](#)
- [REF CURSORの概要](#)
- [REF CURSORを使用するPL/SQLストアド・プロシージャの作成](#)
- [ストアド・プロシージャを実行するためのODP.NETアプリケーションの変更](#)
- [ODP.NETアプリケーションによるPL/SQLストアド・プロシージャの実行](#)

[7 Oracle DatabaseでのASP.NETの使用](#)

- [概要: Oracle Developer Toolsを使用したASP.NETアプリケーションの構築](#)
- [このチュートリアルを始める前に](#)
- [Webサイトの作成およびデータベースへの接続](#)
 - [ASP.NET Webサイトの作成](#)
 - [データソースの作成](#)
- [認証用のWebサイトの有効化](#)
- [Oracle Providers for ASP.NETの有効化および軽量Webユーザーの作成](#)
- [Webサイト認証のテスト](#)

[8 .NETストアド・プロシージャの開発とデプロイ](#)

- [.NETストアド・プロシージャの概要](#)
- [共通言語ランタイム・サービスの開始](#)
- [SYSDBAとしての接続の作成](#)
- [Oracleプロジェクトの作成](#)
- [.NETストアド・ファンクションおよびプロシージャの作成](#)
- [.NETストアド・ファンクションおよびプロシージャのデプロイ](#)
- [.NETストアド・ファンクションおよびプロシージャの実行](#)
- [問合せウィンドウでの.NETストアド・プロシージャの実行](#)

[9 グローバリゼーション・サポートの組み込み](#)

- [グローバル・アプリケーションの概要](#)
- [.NET Frameworkを使用したグローバル・アプリケーションの開発](#)
- [ユーザーの正しいローカル規則でのデータ表示](#)

- [SQL*Plusへの接続](#)
- [Oracleの日付書式の使用](#)
- [Oracleの数値書式の使用](#)
- [Oracleの言語ソートの使用](#)
- [Oracleのエラー・メッセージ](#)
- [.NETとOracle Databaseのロケール環境の同期化](#)
- [Oracle Data Provider for .NETでのクライアント・グローバリゼーションのサポート](#)
 - [クライアント・グローバリゼーション設定](#)
 - [セッション・グローバリゼーション設定の使用](#)
 - [スレッドベースのグローバリゼーション設定](#)

[A Oracle Databaseインスタンスの起動および停止](#)

[B フォームのコピー](#)

[索引](#)

はじめに

このマニュアルでは、Microsoft .NET Frameworkに対応するOracleテクノロジーを使用した、Oracle Databaseでのアプリケーション開発について説明します。

対象読者

このマニュアルは、『Oracle Database 2日でデータベース管理者』および『Oracle Database 2日で開発者ガイド』をすでに読んでいるユーザーで、SQLおよびPL/SQLの基礎知識があり、Microsoft Visual Studioの使用方法を理解しているユーザーを対象としています。

ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社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/accessibility/>を参照してください。

ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかしJAWSは括弧だけの行を読まない場合があります。

外部Webサイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しないWebサイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらのWebサイトのアクセシビリティに関しての評価や言及は行っておりません。

聴覚に障害があるお客様のOracleサポート・サービスへのアクセス

Oracleサポート・サービスに連絡するには、テレコミュニケーション・リレー・サービス (TRS) を使用してOracleサポート (+1-800-223-1711) までお電話ください。Oracleサポート・サービスの技術者が、Oracleサービス・リクエストのプロセスに従って、技術的な問題を処理し、お客様へのサポートを提供します。TRSの詳細は、<http://www.fcc.gov/cgb/consumerfacts/trs.html>を参照してください。電話番号の一覧は、<http://www.fcc.gov/cgb/dro/trsphonebk.html>を参照してください。

関連ドキュメント

詳細は、次のOracle Databaseドキュメントを参照してください。

- 『Oracle Data Provider for .NET開発者ガイド』
- 『Oracle Database Extensions for .NET開発者ガイド』
- 『Oracle Database 2日でデータベース管理者』
- 『Oracle Database 2日で開発者ガイド』
- Oracle Developer Tools for Visual Studioのダイナミック・ヘルプ
- 『Oracle Database Net Services管理者ガイド』
- 『Oracle Database Express Edition Installation Guide for Microsoft Windows』

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。

サポートおよびサービス

次の各項に、各サービスに接続するためのURLを記載します。

Oracleサポート・サービス

オラクル製品サポートの購入方法、およびOracleサポート・サービスへの連絡方法の詳細は、次のURLを参照してください。

<http://www.oracle.com/lang/jp/support/index.html>

製品マニュアル

製品のマニュアルは、次のURLにあります。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

研修およびトレーニング

研修に関する情報とスケジュールは、次のURLで入手できます。

http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3

その他の情報

オラクル製品やサービスに関するその他の情報については、次のURLから参照してください。

<http://www.oracle.com/lang/jp/index.html>

<http://www.oracle.com/technology/global/jp/index.html>

注意:

ドキュメント内に記載されているURLや参照ドキュメントには、Oracle Corporationが提供する英語の情報も含まれています。日本語版の情報については、前述のURLを参照してください。

1 概要

この章の内容は次のとおりです。

- [このマニュアルについて](#)
- [Microsoft .NET Frameworkの概要](#)
- [Oracle Data Provider for .NETの概要](#)
- [Oracle Developer Tools for Visual Studioの概要](#)
- [.NETストアド・プロシージャの概要](#)
- [Oracle Providers for ASP.NETの概要](#)

このマニュアルについて

このマニュアルは、クイック・スタート・ガイドであり、Oracle Data Provider for .NETおよびOracle Developer Tools for Visual Studioの主な機能をはじめとする、Microsoft .NET Frameworkに対応したOracleテクノロジーについて説明しています。 インストールと構成、Oracle .NET製品を使用した基本的なアプリケーションの作成方法、PL/SQLおよび.NETストアド・プロシージャの両方の作成方法と使用方法についても説明します。

注意:

このマニュアルはMicrosoft Visual Studio 2008を使用して作成されました。Microsoft Visual Studio 2005をご使用の場合は、スクリーンショット、ショートカット、メニュー・オプションおよび生成されるコードが若干異なることはありますが、通常、問題が発生することはありません。

このマニュアルの内容を理解した後は、Oracle Databaseドキュメント・ライブラリに含まれる様々な情報の理解に進むことができます。

参照:

- Visual Studioのダイナミック・ヘルプ
- 『Oracle Data Provider for .NET開発者ガイド』
- 『Oracle Database Extensions for .NET開発者ガイド』
- 『Oracle Database 2日でデータベース管理者』
- 『Oracle Database 2日で開発者ガイド』

Microsoft .NET Frameworkの概要

Microsoft .NET Frameworkは、アプリケーションやXML Webサービスを作成、デプロイおよび実行するための多言語環境です。主なコンポーネントは次のとおりです。

共通言語ランタイム

共通言語ランタイム (CLR) は、実行中のアプリケーションの管理に役立つサービスを提供する、言語に依存しない開発環境およびランタイム環境です。

Frameworkクラス・ライブラリ

Frameworkクラス・ライブラリ (FCL) には、事前にパッケージ化された機能における一貫性のあるオブジェクト指向ライブラリが含まれています。

Oracle Data Provider for .NETの概要

Oracle Data Provider for .NET (ODP.NET) は、.NETクライアント・アプリケーションからOracle Databaseへの高速かつ効率的なADO.NETデータ・アクセス、およびその他のOracle Database機能へのアクセスを提供します。

開発者は、ODP.NETを使用して、Real Application ClustersやXML DBなどのOracle Databaseの拡張機能、および高度なセキュリティを利用できます。

Oracle Developer Tools for Visual Studioの概要

Oracle Developer Tools for Visual Studio (ODT) は、Visual Studio環境と統合された一連のアプリケーション・ツールです。これらのツールでは、Oracle機能に対し、Graphical User Interfaceによってアクセスを行います。また、ユーザーは様々なアプリケーション開発タスクを実行できる他、開発の生産性や使いやすさが向上します。Oracle Developer Toolsでは、Visual Basic、C#、およびその他の.NET言語による.NETストアド・プロシージャのプログラミングおよび実装をサポートしています。

次に、Oracle Developer Toolsの機能の一部を示します。

- Oracleスキーマを参照するためのServer Explorerとの統合
- スキーマ・オブジェクトを作成および変更するためのデザイナーおよびウィザード
- コードを自動生成するためにスキーマ・オブジェクトを.NETフォーム上にドラッグ・アンド・ドロップする機能
- 状況依存のダイナミック・ヘルプが統合されたPL/SQLエディタおよびデバッガ
- データの挿入および更新、Visual Studio環境でのストアド・プロシージャのテストなど、データベースの日常的なタスクを実行するためのOracle Data Window
- SQL文またはPL/SQLスクリプトを実行するためのOracle Query Window
- Oracle Deployment Wizard for .NET ([「Microsoft Visual Studioとの統合」](#)を参照)

.NETストアド・プロシージャの概要

Oracle Database Extensions for .NETは、WindowsでのOracle Databaseのデータベース・オプションです。このオプションによって、Microsoft Windows用のOracle Databaseを使用する.NETストアド・プロシージャまたはファンクションを、Visual Basic .NETまたはVisual C#を使用して作成および実行できるようになります。

参照:

『Oracle Database Extensions for .NET開発者ガイド』

Microsoft Visual Studioとの統合

.NETアセンブリに.NETプロシージャおよびファンクションを作成した後は、Oracle Developer Tools for Visual StudioのコンポーネントであるOracle Deployment Wizard for .NETを使用してそれらをOracle Databaseにデプロイできます。

Oracle Providers for ASP.NETの概要

ASP.NETの開発者は、Oracle Providers for ASP.NETを使用すると、Webアプリケーション（Webユーザー情報、ショッピング・カートなど）に共通のアプリケーション状態をOracle Databaseに簡単に格納できるようになります。これらのプロバイダは、既存のMicrosoft ASP.NETプロバイダをモデルにしており、類似スキーマおよびプログラミング・インタフェースを共有することで、.NET開発者に使い慣れたインタフェースを提供します。

Oracleでは、次のプロバイダが用意されています。

- メンバーシップ・プロバイダ
- ロール・プロバイダ
- サイト・マップ・プロバイダ
- セッション状態プロバイダ
- プロファイル・プロバイダ
- Webイベント・プロバイダ
- Webパーツ・パーソナライズ・プロバイダ
- キャッシュ依存性プロバイダ

それぞれのASP.NETプロバイダは、個別に使用したり、ご使用のWebサイトのその他のOracle ASP.NETプロバイダと組み合わせて使用することができます。各プロバイダは、特定のセットのWebサイト情報を格納されます。

Oracle Providers for ASP.NETのクラス、その使用方法、インストールおよび要件については、『Oracle Providers for ASP.NET開発者ガイド』を参照してください。この内容は、ダイナミック・ヘルプとしても提供されています。

参照:

- [第7章「Oracle DatabaseでのASP.NETの使用」](#)
- 『Oracle Providers for ASP.NET開発者ガイド』

2 .NET製品のインストール

この章の内容は次のとおりです。

- [必要事項](#)
- [.NET製品のインストール](#)
- [NET接続の別名の構成](#)
- [Oracle Providers for ASP.NETのセットアップ](#)

必要事項

この項では、このマニュアルで紹介する例を実行するために必要な製品とデータベース・スキーマを示します。

Oracle Database

ローカルまたはリモート・コンピュータのいずれかに、Oracle Databaseをインストールしておく必要があります。

注意:

このマニュアルで使用するすべての例で、Oracle Database 11gクライアントが必要です。ただし、このクライアントではOracle Database 9iリリース2以上がサポートされているため、このうちのいずれかのリリースを使用することもできます。

Oracle Database Extensions for .NETの使用を予定している場合、クライアントはOracle Database 11gへも接続する必要があります。

データベースは、ユーザー・インタフェースであるEnterprise Managerを使用して管理できます。Enterprise Managerでは、スクリプトや問合せを実行したりその他の操作を実行できます。

参照:

Oracle Databaseをインストールおよび構成していない場合は、『*Oracle Database Express Edition Installation Guide for Microsoft Windows*』を参照してください。

サンプル・データ

このマニュアルで使用するサンプル・データは、Oracleサンプル・スキーマの1つであるHRスキーマに含まれています。このサンプル・スキーマは、Oracle Databaseインストールの一部として含まれています。

参照:

HRのデータ・モデルおよび表については、『*Oracle Databaseサンプル・スキーマ*』を参照してください。

Oracle Data Access Components

Oracle Data Access Components (ODAC) は、次のツールのコレクションです。

- Oracle Developer Tools for Visual Studio
- Oracle Data Provider for .NET
- Oracle Providers for ASP.NET
- Oracle Provider for OLE DB
- Oracle Objects for OLE
- Oracle ODBC Driver
- Oracle Services for Microsoft Transaction Server
- Oracle SQL*Plus
- Oracle Instant Client

Oracle Database Extensions for .NET

Oracle Database Extensions for .NETは、Oracle Database 11gをWindowsにインストールするときに、その一部としてインストールされます。Oracle Database Extensions for .NETのインストール後、ODACインストールによってOracle Database Extensions for .NETへのアップグレードが提供されます。このアップグレードは、「Oracle Data Access Components for Oracle Server」オプションの一部として含まれています。これは、[「.NET製品のインストール」](#)の手順4のスクリーンショットに示されています。このアップグレードを実行し、Oracle Database Extensions for .NETをインストールする必要があるのは、このマニュアルの[第8章](#)に示す内容を実行する予定がある場合のみです。

Visual Studioのバージョン

Visual Studio 2008を使用する場合は、このマニュアルに示す手順を進める前にインストールする必要があります。Microsoft Visual Studio 2005を使用した場合、スクリーンショット、ショートカット、メニュー・オプションおよび生成されるコードが若干異なることはありますが、通常、問題が発生することはありません。

.NET製品のインストール

次の手順は、Visual Studioをインストールした後で行う、Oracle Developer Tools for Visual Studio (ODT)、Oracle Data Provider for .NETおよびその他のODAC製品のインストール方法を示しています。

注意:

新しいバージョンのOracle .NET製品のリリースにより、このマニュアルに示しているインストール手順が若干変更される場合があります。スクリーンショットは、Oracle Data Access Components (ODAC) バージョン11.1.0.6.21に基づいています。

インストールを行うには、次の手順を実行します。

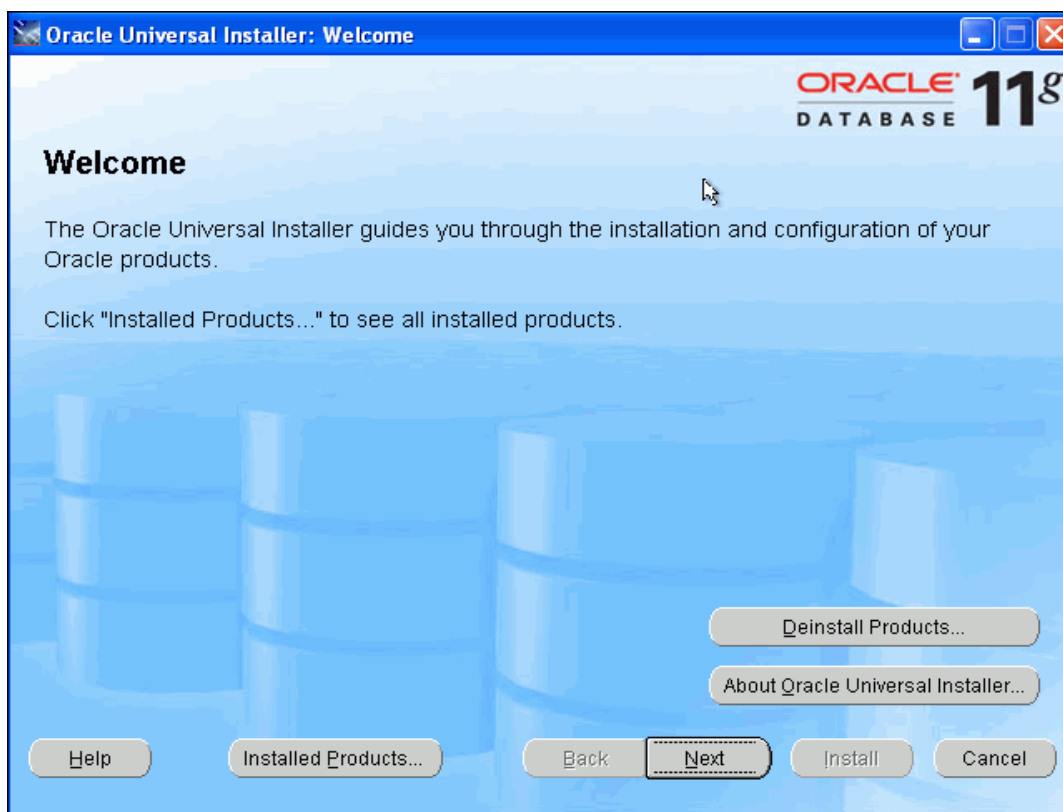
1. インターネット・ブラウザで、次の場所に移動してODACとOracle Developer Tools for Visual Studioをダウンロードします。

<http://www.oracle.com/technology/software/tech/windows/odpnet/index.html>

2. すべてのファイルをzipファイルからファイル・システムのフォルダに抽出します。
3. **Setup.exe**をダブルクリックします。

Oracle Installerが起動します。必要な依存オブジェクトを検出する画面が短い間表示された後、Oracle Universal Installer

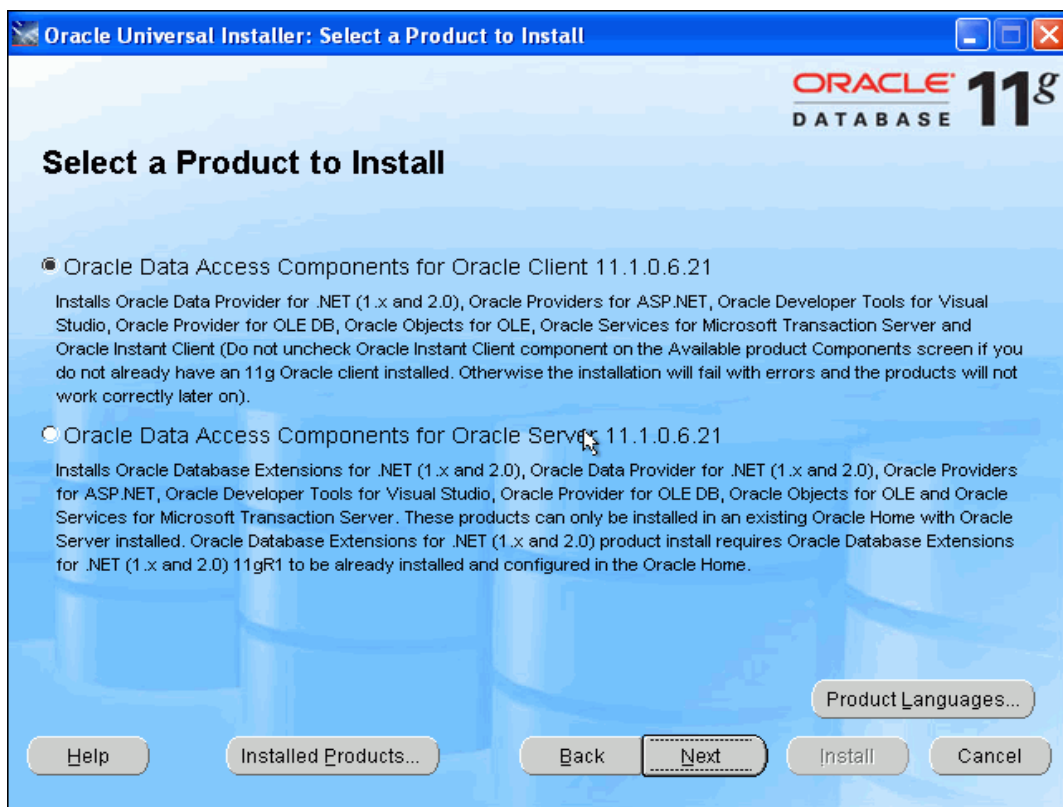
(OUI) の「ようこそ」画面が表示されます。



[install01.gifの説明](#)

4. 「次へ」をクリックします。

「インストールする製品の選択」画面が表示されます。



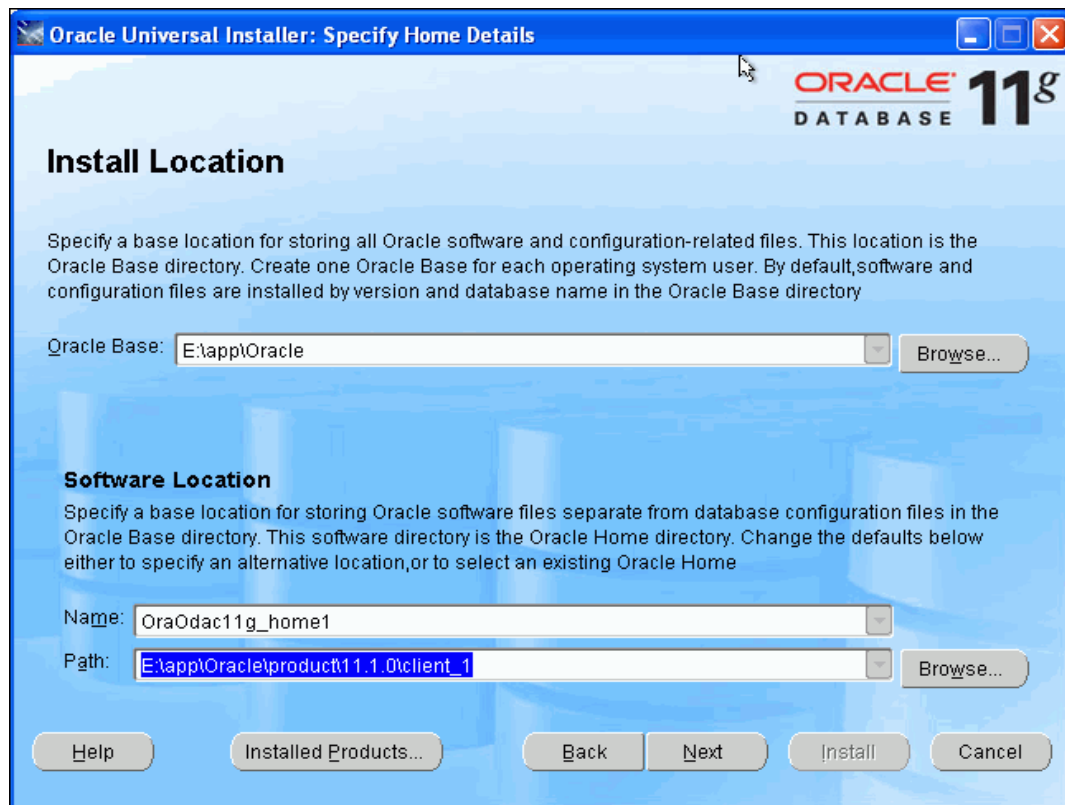
[install02.gifの説明](#)

5. 最初のオプションを選択します。

このオプション（ODAC for Oracle Client）を選択すると、クライアントのOracleホームで使用する製品のみがインストールされます。2番目のオプション（ODAC for Oracle Server）を選択すると、Oracle Databaseが含まれるOracleホームに直接インストールできます。

6. 「次へ」をクリックします。

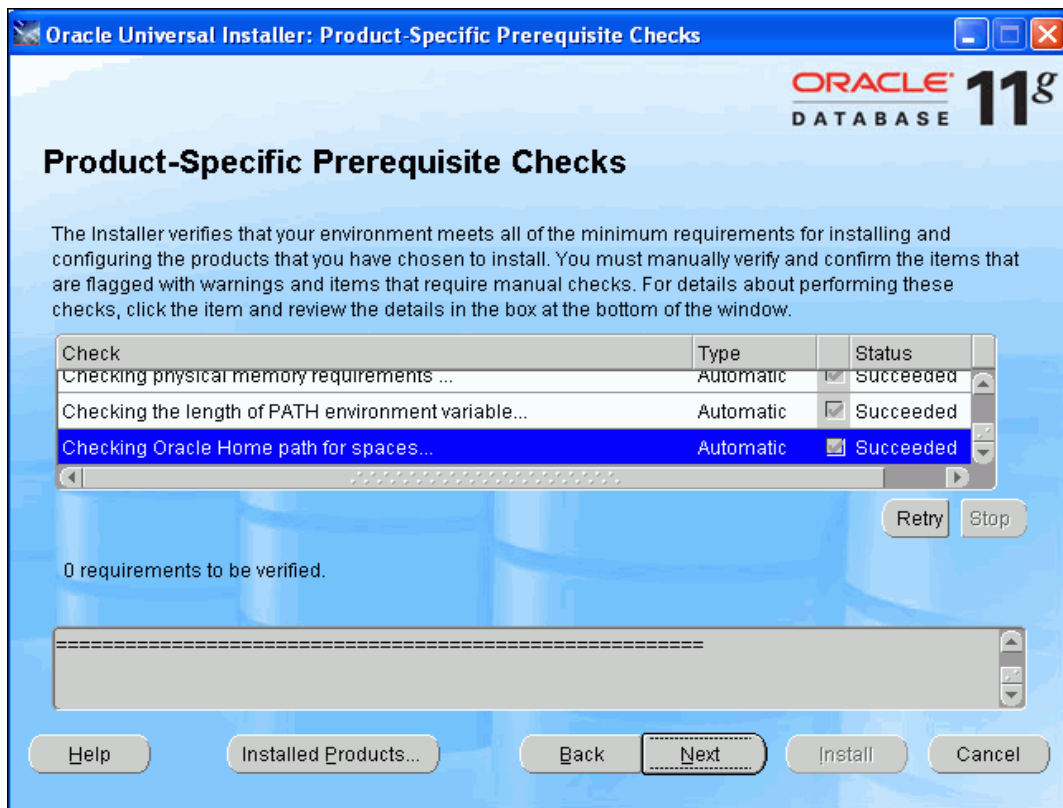
「インストールの場所」ウィンドウが表示されます。このウィンドウで、インストールする場所を選択できます。デフォルトでは、クライアントのOracleホームが新しく作成されます。このマニュアルでは、デフォルトを受け入れて新しいOracleホームを作成します。



[install03.gifの説明](#)

7. 「次へ」をクリックします。

インストーラにより前提条件のチェックが実行されます。各チェックのステータスは「成功しました」になる必要があります。



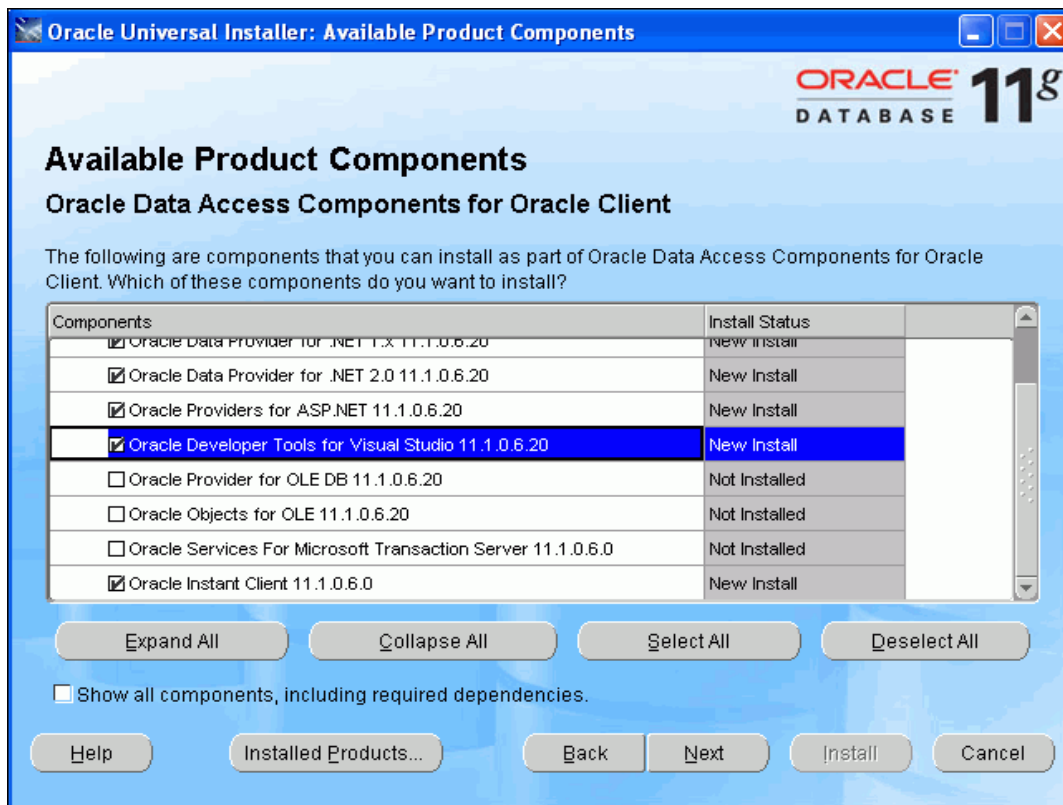
[install04.gifの説明](#)

8. 「次へ」をクリックします。

「使用可能な製品コンポーネント」画面が表示されます。

次の製品が選択されていることを確認してください。

- Oracle Data Provider for .NET 2.0
- Oracle Providers for ASP.NET
- Oracle Developer Tools for Visual Studio
- Oracle Instant Client



[install05.gifの説明](#)

9. 「次へ」をクリックします。

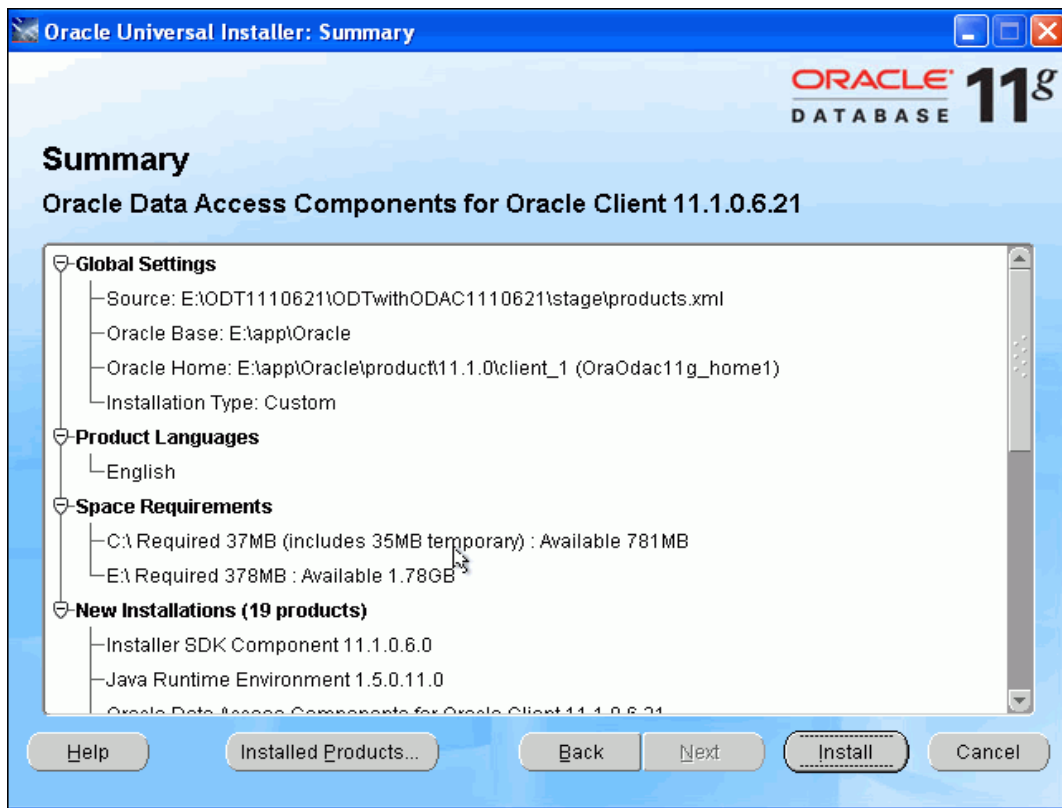
Oracle Providers for ASP.NETを使用する場合は、`ORACLE_BASE¥¥ORACLE_HOME¥client_1¥ASP.NET¥SQL`にあるSQLスクリプトを実行する必要があることを示す画面が表示されます。

注意:

`ORACLE_BASE¥¥ORACLE_HOME`は、使用しているOracleホームを表すディレクトリです。

10. 「次へ」をクリックします。

「サマリー」ウィンドウが表示されます。



[install06.gifの説明](#)

11. 「インストール」をクリックしてインストールを完了します。

インストールの最後の画面が表示されます。再度、ASP.NETスクリプトをインストールするように指示されます。Oracle Providers for ASP.NETを使用する予定がある場合は、スクリプトをインストールします。

12. 「終了」をクリックします。

NET接続の別名の構成

Oracleクライアントが短縮名を使用してデータベースに接続できるように、tnsnames.oraファイルにデータベース・サーバーのアドレスを定義します。DBAから事前構成済のtnsnames.oraファイルがすでに提供されている場合もあります。

それ以外の場合は、ORACLE_BASE¥ORACLE_HOME¥network¥admin¥sampleディレクトリに移動して、このディレクトリにあるtnsnames.oraファイルとsqlnet.oraファイルをORACLE_BASE¥ORACLE_HOME¥network¥adminディレクトリにコピーする必要があります。

tnsnames.oraファイルに含まれる次の接続記述子を使用し、イタリック体で示されている値を使用環境に合わせて変更します。

例2-1 tnsnames.ora接続記述子

```
address name =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (Host = hostname) (Port = port))
  )
(CONNECT_DATA =
  (SERVICE_NAME = sid)
)
)
```

値の説明:

sid: データベース・サービス名

hostname: データベース・コンピュータ名

port: データベースとの通信に使用するポート

address name: 接続記述子として使用するユーザー定義の短縮名。この短縮名を、.NETアプリケーションの接続文字列で使用します。

[例2-2](#)に、tnsnames.oraファイルの例を示します。

例2-2 tnsnames.oraファイルの例

```
ORCL =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = ORCL)
  )
)
```

参照:
『Oracle Database Net Services管理者ガイド』

Oracle Providers for ASP.NETのセットアップ

Oracle Providers for ASP.NETは、Oracle Database内のデータベース・ユーザーのスキーマのコンテキストにWebアプリケーションの状態を格納します。管理者は、アプリケーションの状態を格納するための新しいデータベース・ユーザーを作成できます。

このデータベース・ユーザーは単一の物理ユーザーにはマップされません。すべてのWebサイト・ユーザーのASP.NET情報を格納するためのリポジトリとして機能します。このため、この新しいデータベース・ユーザーのスキーマ内には、単一または複数のWebユーザーのアプリケーションの状態を格納できます。

注意:
このチュートリアルでは、Oracle Providers for ASP.NETデータベース・ユーザーであることを示すために、このデータベース・ユーザーをASPNET_DB_USERと呼びます。

実行時に、ASP.NETアプリケーションは、接続文字列内でデータベース・ユーザーの資格証明を使用してデータベースに接続します。

Oracle Databaseを設定するには、データベース管理者がOracle Providers for ASP.NETデータベース・ユーザーのスキーマに特定のデータベース権限を付与する必要があります。これらの権限によって、データベース・ユーザーはOracle Providers for ASP.NETで必要な表、ビュー、ストアド・プロシージャなどのデータベース・オブジェクトを作成できます。

権限が付与されると、データベース・ユーザーはOracle Provider for ASP.NET構成スクリプトを実行できます。

参照:
詳細は、『Oracle Providers for ASP.NET開発者ガイド』を参照してください。

この項では、データベースにOracle Providers for ASP.NETをセットアップする手順について説明します。このチュートリアルのASP.NETプロバイダの部分（[第7章](#)の後半）の内容を実行する予定がない場合は、この項を完了する必要はありません。Oracle Providers for ASP.NETのセットアップでは、Oracle Developer Tools for Visual Studioを使用します。これは、セットアップの開始前にインストールする必要があります。

この項の内容は次のとおりです。

- [Oracle Providers for ASP.NETデータベース・ユーザーのセットアップ](#)
- [接続文字列の設定](#)
- [様々なセットアップに対応するためのOracle Providers for ASP.NETのカスタマイズ](#)

Oracle Providers for ASP.NETデータベース・ユーザーのセットアップ

このチュートリアルでは、ASP.NETプロバイダのデータを格納するための新しいデータベース・ユーザー・スキーマASPNET_DB_USERを作成します。ユーザーASPNET_DB_USERに特定のデータベース権限を付与し、ASP.NETプロバイダのデータベース・スクリプトを実行してスキーマをセットアップします。このスキーマには、表とストアド・プロシージャ以外に、Oracle Providers for ASP.NETに必要なその他のデータベース・オブジェクトが含まれます。

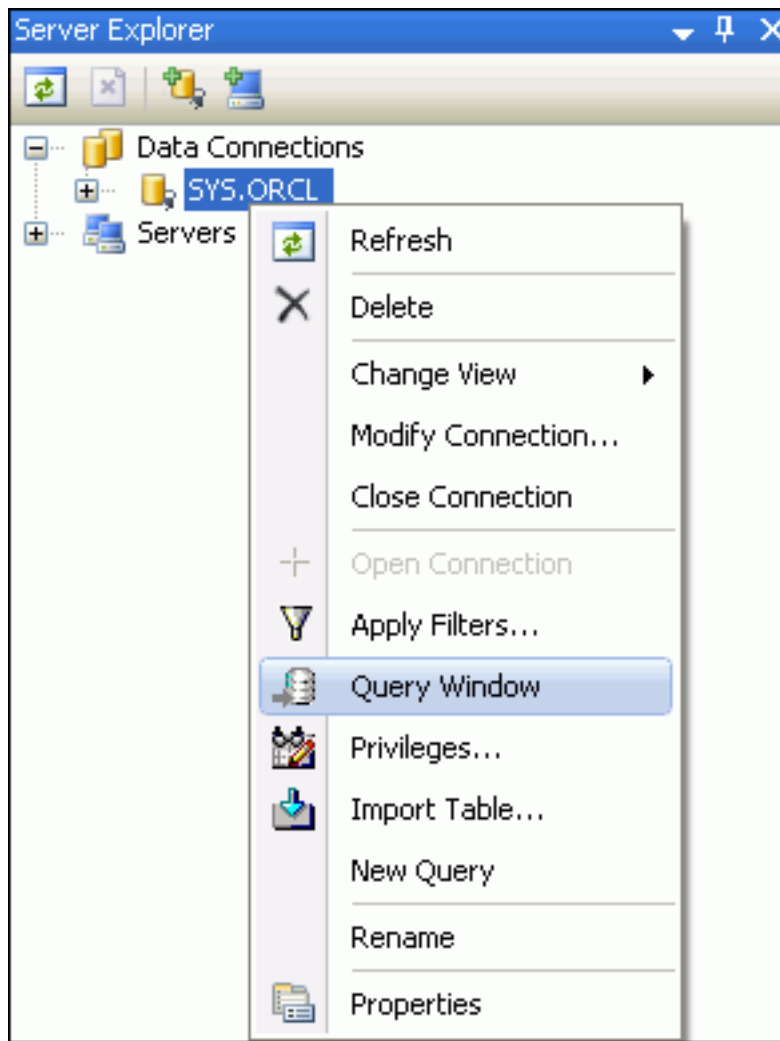
この項の内容は次のとおりです。

- [ユーザーの作成および権限の付与](#)
- [すべてのOracle Providers for ASP.NETの構成](#)
- [個別のOracle Providers for ASP.NETの構成](#)
- [Oracle Providers for ASP.NET用スキーマのアンインストール](#)

ユーザーの作成および権限の付与

新しいユーザーを追加し、必要な権限を付与するには、次の手順を実行します。

1. SYSまたは別のデータベース管理ユーザーとしてログインします。詳細は、[「SYSDBAとしての接続の作成」](#)を参照してください。
2. Server Explorerの問合せウィンドウで、次のように新しいASPNET_DB_USERユーザーを作成します。
 1. Server Explorerで、「SYS.ORCL」を右クリックして「Query Window」を右クリックします。

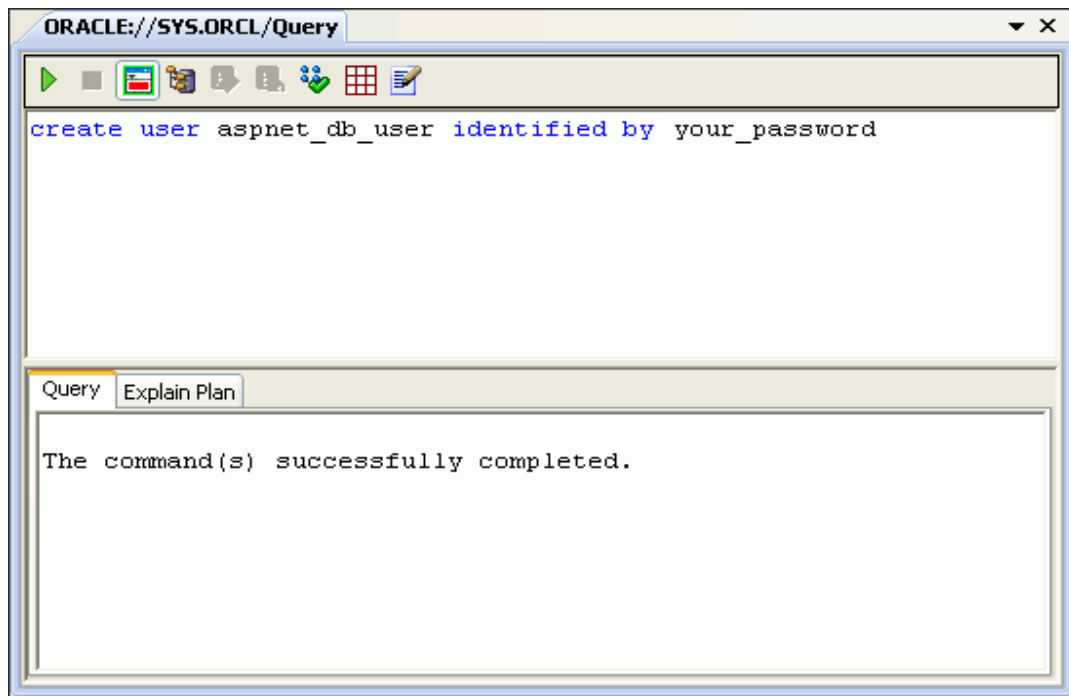


[aspnt_install01.gifの説明](#)

2. 問合せウィンドウで、次のコマンドを入力します。

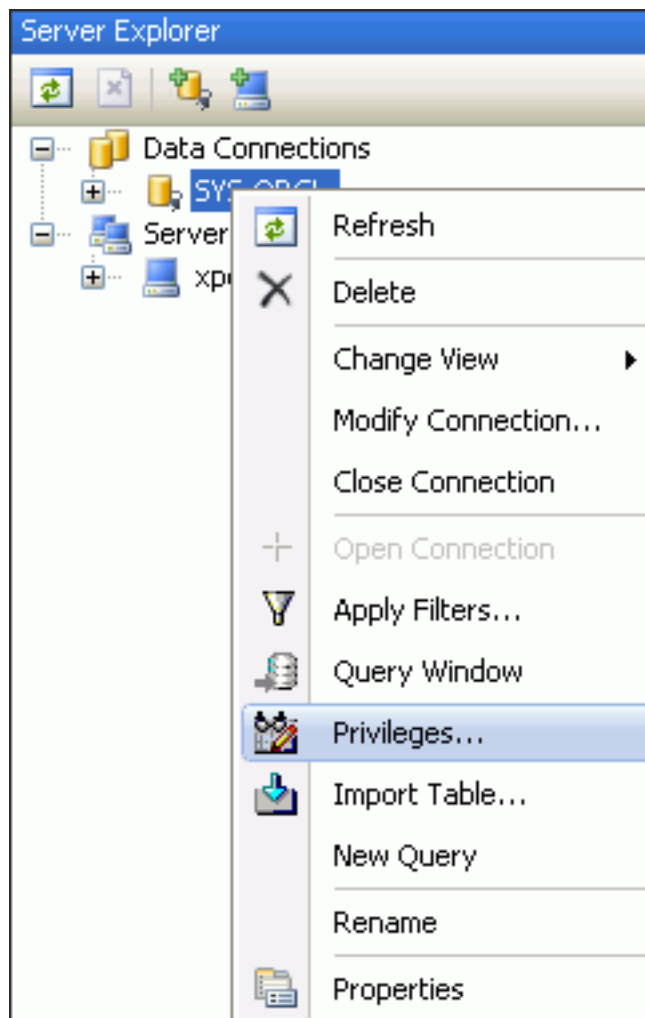
```
create user ASPNET_DB_USER identified by your_password
```

この手順によって、入力したパスワードを持つユーザーASPNET_DB_USERがデータベースに作成されます。



[aspnt_install02.gifの説明](#)

3. 「Execute Query」（左上の緑色の矢印）をクリックしてこのコマンドを実行します。下部のウィンドウに、コマンドが正常に完了したことが示されます。
-
3. Server Explorerに戻り、「SYS.ORCL」を再度選択し、右クリックして「Privileges...」を選択します。



[aspnt_install03.gifの説明](#)

ODTの「Grant/Revoke Privileges」ウィザードが表示されます。

4. ASP.NETプロバイダ用にスキーマを作成し、Webサイトの状態を格納できるように、新しいデータベース・ユーザーに権限を付与します。
 - 「Object type」をUSERに設定し、「User」をASPNET_DB_USERに設定します。

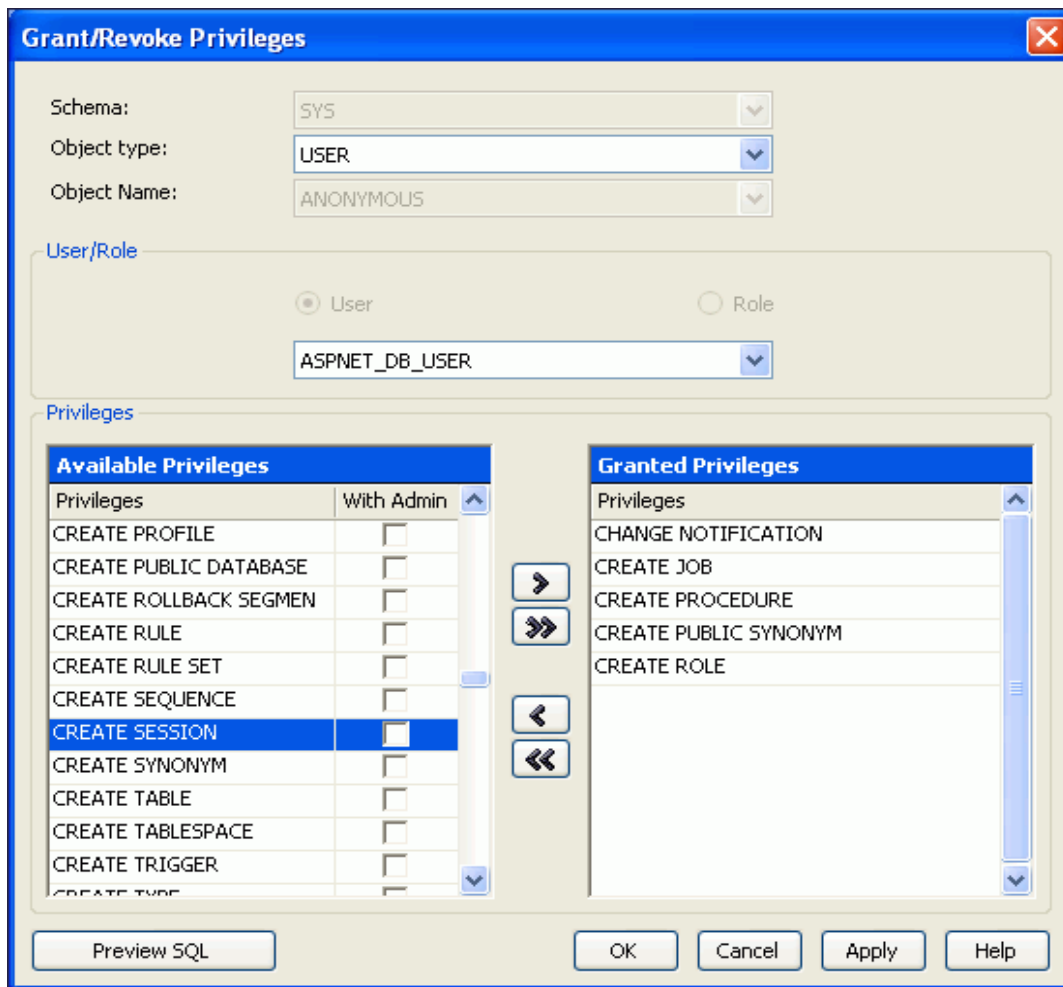
注意:

ASPNET_DB_USERを表示するには、リフレッシュが必要な場合があります。

5. 中央の右矢印 (➤) を使用して、権限を「Available Privileges」リストから「Granted Privileges」リストに移動します。

通常、次の権限が必要となります。

- 変更の通知
- ジョブの作成
- プロシージャの作成
- パブリック・シノニムの作成
- ロールの作成
- セッションの作成
- 表の作成
- ビューの作成
- パブリック・シノニムの削除
- 無制限の表領域: この例では、ASPNET_DB_USERに無制限の表領域を付与します。ただし、ほとんどの場合、管理者はデータベース・ユーザーに特定の表領域の割当て制限を割り当てます。



[aspnt_install04.gifの説明](#)

「Apply」をクリックすると、出力ウィンドウに正常に実行されたことが示されます。「OK」をクリックします。

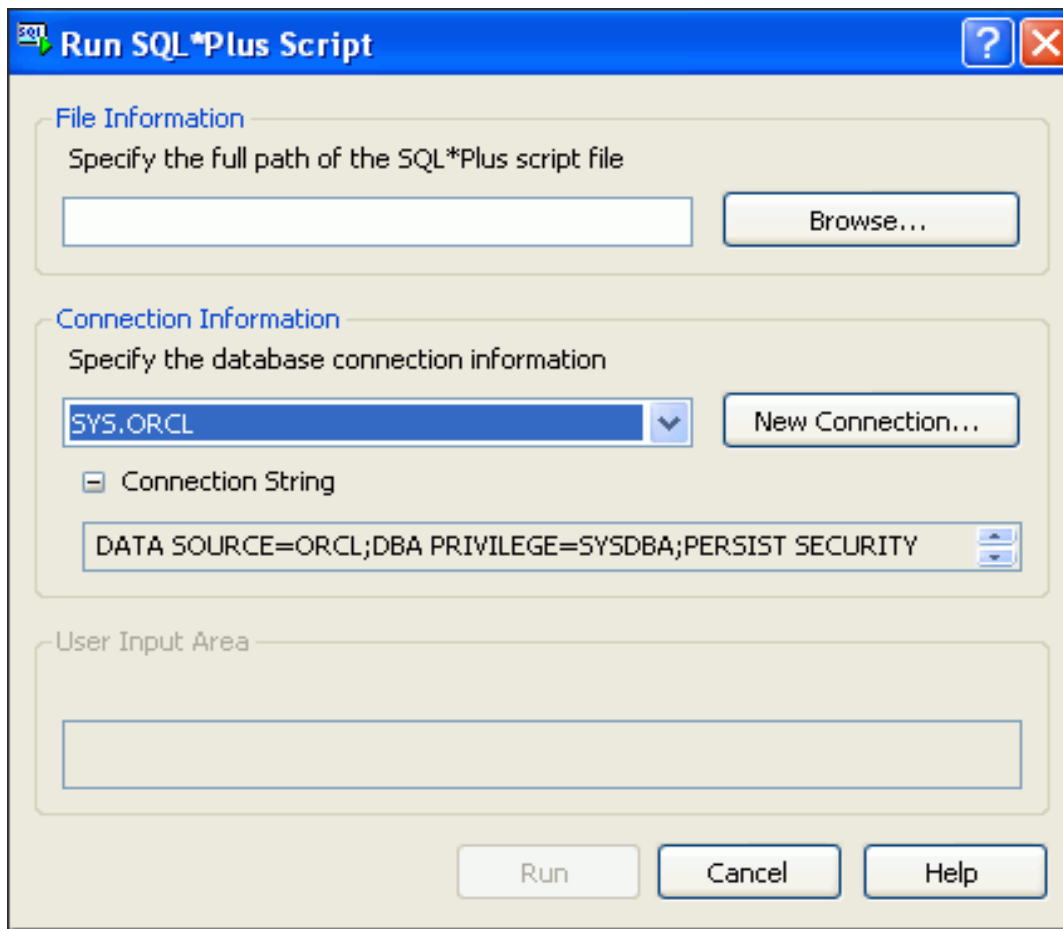
Oracle Providers for ASP.NETユーザーに必要な権限が付与されていない場合、設定スクリプトの実行時にエラーが発生する可能性があります。

すべてのOracle Providers for ASP.NETの構成

データベース内のすべてのプロバイダを一度に構成するには、InstallAllOracleASPNETProviders.sqlを実行します。

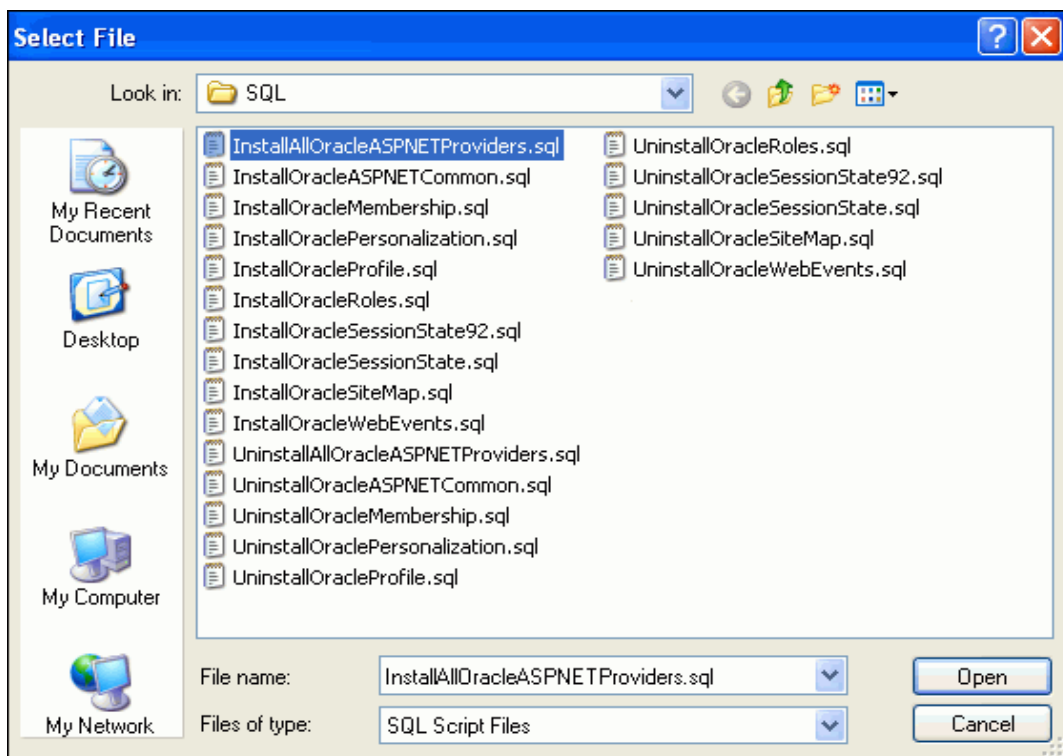
このスクリプトをOracle Developer Toolsで実行するには、次の手順を実行します。

1. Visual Studioで、「Tools」を選択し、「Run SQL*Plus Script」を選択します。画面が表示されたら、「Browse」を選択します。



[aspnt_install105a.gifの説明](#)

2. `ORACLE_BASE\ORACLE_HOME\ASP.NET\sql`ディレクトリ（`ORACLE_BASE\ORACLE_HOME`はOracleホームを表す）を参照し、`InstallAllOracleASPNETProviders.sql`を選択して「Open」をクリックします。



[aspnt_install105.gifの説明](#)

3. 「Run SQL*Plus Script」画面が表示されたら、「New Connection」を選択します。

「Connection Properties」画面が表示されたら、データソースが「Oracle Database (Oracle ODP.NET)」で、データソース名が

「ORCL」であることを確認します。その後、ユーザー名ASPNET_DB_USERとパスワードを入力し、「Role」に「Default」を指定します。「OK」をクリックします。

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Oracle Database (Oracle ODP.NET) Change...

Connection Details Apply Filters

Data source name: ORCL

☐ Use Windows integrated authentication
☒ Use a specific user name and password

User name: ASPNET_DB_USER
Password: *****

☐ Save password

Role: Default

Connection name: ASPNET_DB_USER.ORCL

Advanced...

Test Connection OK Cancel

[aspnt_install106.gifの説明](#)

場合によっては、「Oracle Server Login」ダイアログ・ボックスが表示されることもあります。このような場合は、パスワードを保存するかどうかを選択できます。

4. 「Run SQL*Plus Script」画面が再度表示されたら、「Run」を選択します。

Run SQL*Plus Script

File Information
Specify the full path of the SQL*Plus script file

Connection Information
Specify the database connection information

☒ Connection String

User Input Area

[aspnt_install07.gifの説明](#)

このSQLファイルが実行され、バックグラウンドで、このスクリプトが正常に実行されたことが出力ウィンドウで確認されます。

5. スクリプトの実行が終了したら、「Cancel」を選択します。

個別のOracle Providers for ASP.NETの構成

アプリケーションで、すべてのOracle Providers for ASP.NETが必要であるとはかぎりません。 プロバイダは個別に設定できます。通常、他のインストール・スクリプトを実行する前にInstallOracleASPNETCommon.sqlインストール・スクリプトを実行する必要があります。このスクリプトによって、ASP.NETプロバイダの共通インフラストラクチャがセットアップされます。その後、必要なOracle Provider for ASP.NETに対して、それぞれ特定のSQLスクリプトを（任意の順序で）実行します。

これらのインストール・スクリプトは、`ORACLE_BASE\ORACLE_HOME\ASP.NET\sql`ディレクトリにあります。

表2-1 Oracle Providers for ASP.NETの個別インストール・スクリプト

プロバイダ	必要なインストール・スクリプト
Oracleメンバーシップ・プロバイダ	InstallOracleMembership.sql
Oracleパーソナライズ・プロバイダ	InstallOraclePersonalization.sql
Oracleプロファイル・プロバイダ	InstallOracleProfile.sql

Oracleロール・プロバイダ	InstallOracleRoles.sql
Oracleセッション状態プロバイダ	InstallOracleSessionState.sql (Oracle Database 10g リリース1以上の場合) InstallOracleSessionState92.sql (Oracle Database 9i リリース2の場合) これらのインストール・スクリプトには、対応する名前が付けられたアンインストール・スクリプトがあります。 注意: このプロバイダでは、InstallOracleASPNETCommon.sqlの実行は不要です。プロバイダ固有の.sqlインストール・スクリプトのみを実行する必要があります。
Oracleサイト・マップ・プロバイダ	InstallOracleSiteMap.sql
Oracle Webイベント・プロバイダ	InstallOracleWebEvents.sql
Oracleキャッシュ依存性プロバイダ	スクリプトの実行は不要です。

Oracle Providers for ASP.NET用スキーマのアンインストール

インストール・スクリプトで作成したデータベース・オブジェクトを削除するには、対応するアンインストール・スクリプトを使用します。これらのスクリプトには、Uninstallという接頭辞が付いています。

接続文字列の設定

Oracle Providers for ASP.NET情報を格納および取得するように構成されたデータベースでは、中間層またはクライアントからASPNET_DB_USERユーザーに接続する必要があります。

コンピュータの接続情報を構成するには、次の手順を実行します。

1. drive:¥WINDOWS¥Microsoft.NET¥Framework¥v2.0.50727¥CONFIGにあるmachine.configファイルに移動します。
2. テキスト・エディタで、<connectionStrings>を検索し、<add name="OraAspNetConString"..で始まる行を変更して、ユーザーID、パスワード、データ・ソース・エントリ、データ・ソースおよびプロバイダ名を次のように追加します。

```
<connectionStrings>
<add name="OraAspNetConString" connectionString="User
  Id=aspnet_db_user;Password=your_password;Data Source=orcl;"
  providerName="Oracle.DataAccess.Client" />
</connectionStrings>
```

注意:
パスワードは、作成済のパスワードに変更してください。 また、改行は必ず削除してください。接続文字列にコピーしてしまった場合があります。

様々なセットアップに対応するためのOracle Providers for ASP.NETのカスタマイズ

開発者は、`machine.config`ファイルの<`system.web`>セクションで、各ASP.NETプロバイダのプロパティをカスタマイズできます。

`machine.config`ファイルはOracle Universal Installerによって自動的に構成されますが、開発者は`web.config`ファイルを使用してOracle Providers for ASP.NETに対してよりファイングレインなアプリケーション・レベルの制御を適用できます。このファイルは、`machine.config`ファイルのエントリより優先されますが、関連付けられている特定のWebアプリケーションに対してのみ適用されます。開発者は、`machine.config`ファイルと同じXML構文で`web.config`ファイルを設定できます。

3 ODP.NETによる単純な.NETアプリケーションの作成

この章の内容は次のとおりです。

- [新しいプロジェクトの作成](#)
- [参照の追加](#)
- [名前空間ディレクティブの追加](#)
- [ユーザー・インタフェースの設計](#)
- [接続コードの記述](#)
- [アプリケーションのコンパイルと実行](#)
- [エラー処理](#)

新しいプロジェクトの作成

Visual Studioでは、作成するすべての開発コードがグループ化され、プロジェクトと呼ばれるコンテナに入れられます。単純なプロジェクトのほとんどは、1つのファイルのみで構成されます。この項では、新しい開発プロジェクトの作成方法を説明します。

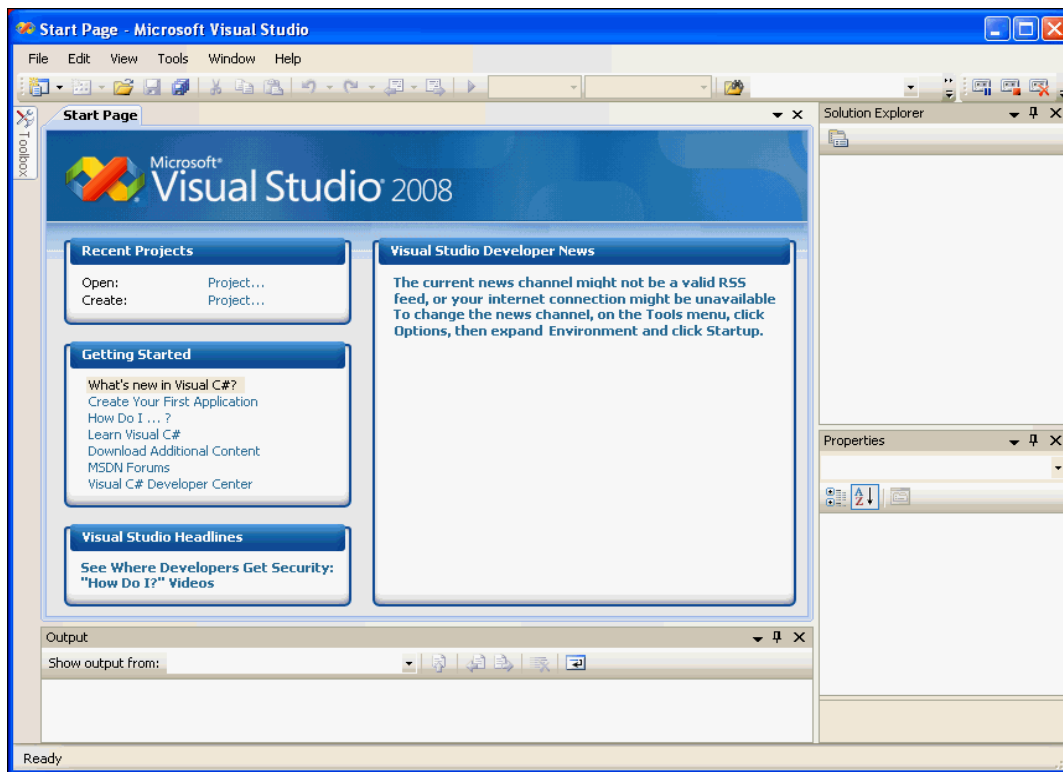
この章で作成するアプリケーションは、後続の章で行う操作の開始点になるため、このマニュアルに示す順に操作してください。

注意: 必要に応じて、**Visual C#**または**Visual Basic**が指定されています。

新しいプロジェクトを開始するには、次の手順を実行します。

1. Visual Studioを起動します。

「スタート」メニューを開き、「すべてのプログラム」から「Microsoft Visual Studio 2008」を選択します。



[connect00.gifの説明](#)

Microsoft Visual Studio IDE環境が表示されます。

2. 「Start Page」の「Recent Projects」ヘッダーの下にある「**Create: Project**」をクリックします。

または、「File」メニューから「New」、「Project」の順に選択します。

「New Project」ダイアログ・ボックスが表示されます。

3. 「Project Types」で、作成するプロジェクトの種類を選択します。

Visual C#:

Visual C#: Windows

Visual Basic:

Other Languages: Visual Basic: Windows

4. 「Templates」で「**Windows Forms Application**」を選択します。
5. 「Name」フィールドに適切な名前を入力します。

Visual C#:

HR_Connect_CS

Visual Basic:

HR_Connect_VB

略語のCSはC#プロジェクトを、VBはVisual Basicプロジェクトを示します。

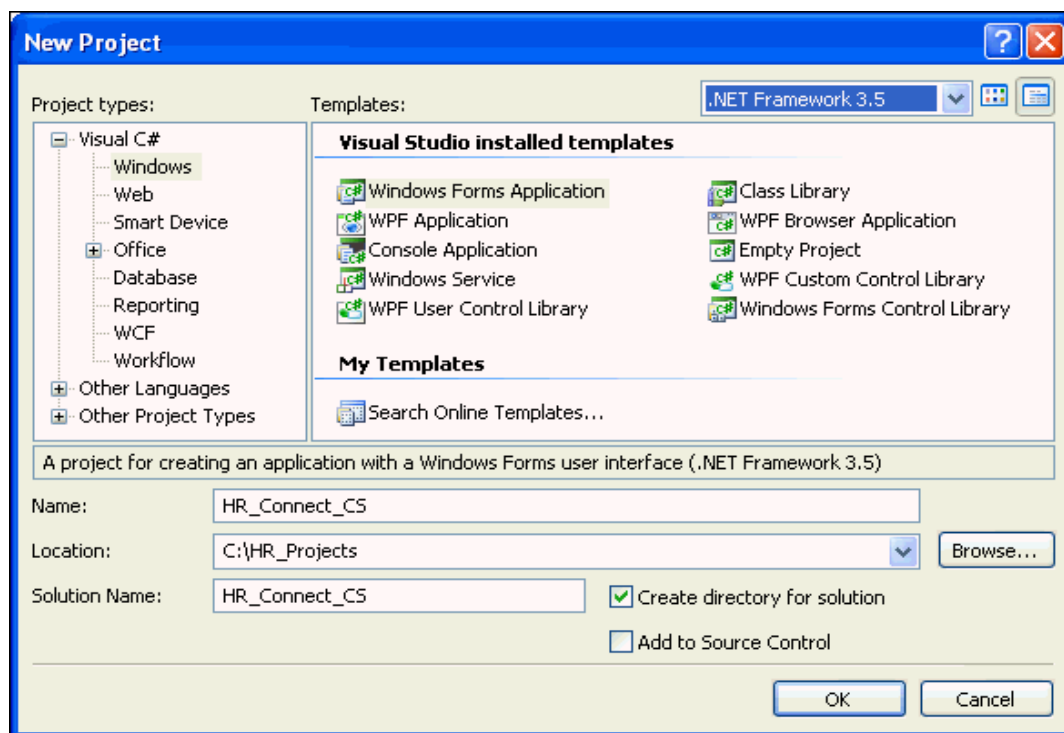
6. 「Location」にファイルを保存するディレクトリを入力します。

このマニュアルに合わせて、このディレクトリをC:\HR_Projectsと入力します。

7. 「Solution Name」に適切な名前（HR_Connect_CSまたはHR_Connect_VB）が表示されます。

1つのソリューションに複数のプロジェクトを格納できます。ソリューションに含まれるプロジェクトが1つのみの場合は、両方に同じ名前を使用できます。

8. 「Create directory for solution」を選択します。
9. 「OK」をクリックします。



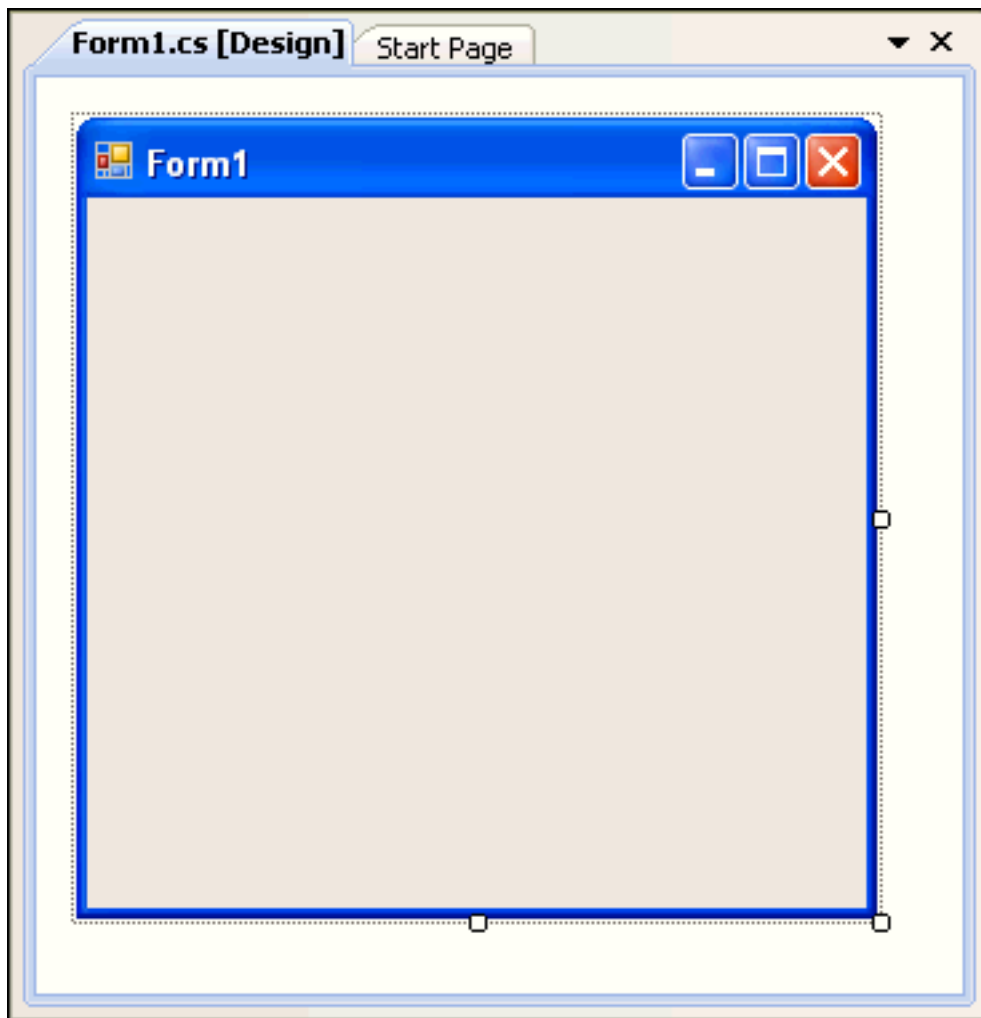
10. [connect01.gifの説明](#)

プロジェクトが作成されます。

メイン・ウィンドウには、言語に応じて「HR_Connect_CS - Microsoft Visual Studio」または「HR_Connect_VB - Microsoft Visual Studio」のいずれかが新しいタイトルとして表示され、次に示すForm1が作成されます。

多くのプロジェクトの最初のフォームには、Form1という名前が自動的に付けられることに注意してください。これはフォーム・コントロールの名前です。この名前と、コード・ファイルに付けられる実際の名前を混同しないでください。通常、コード・ファイルの名前はForm1.csまたはForm1.vbです。

Form1とForm1.xxは、両方とも名前を変更できます。このマニュアルでは、Form1.xxの名前を何回か変更します。



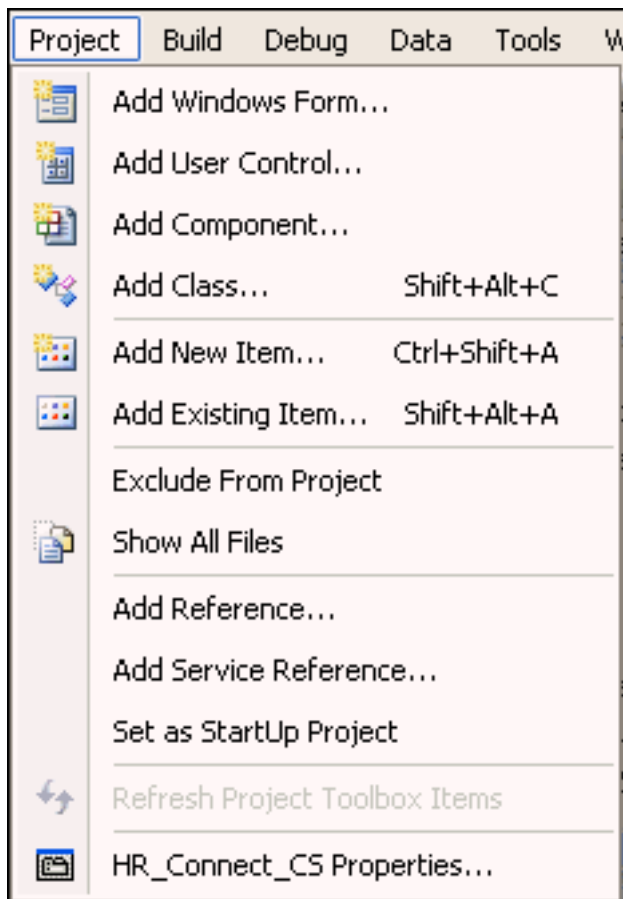
[connect2a.gifの説明](#)

参照の追加

この項では、データ・プロバイダであるOracle Data Provider for .NETが含まれるOracle.DataAccess.dllファイルへの参照を追加する方法を説明します。

参照を追加するには、次の手順を実行します。

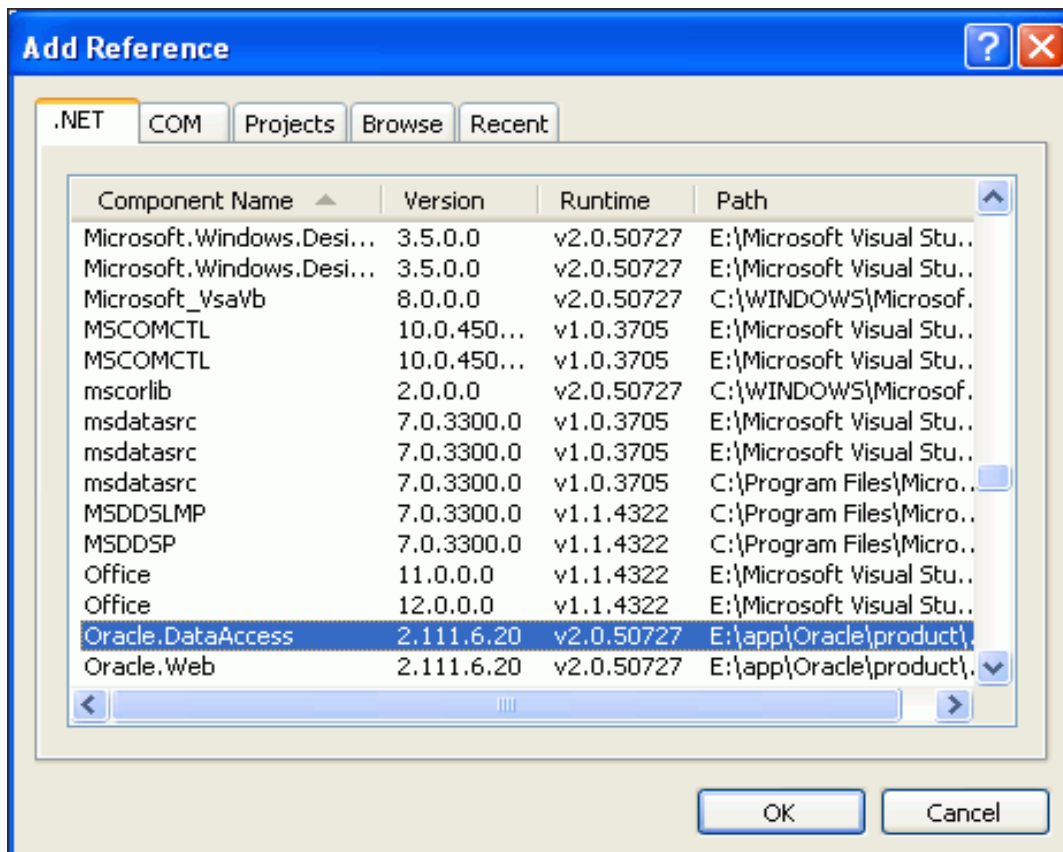
1. 「Project」メニューから「Add Reference」を選択します。



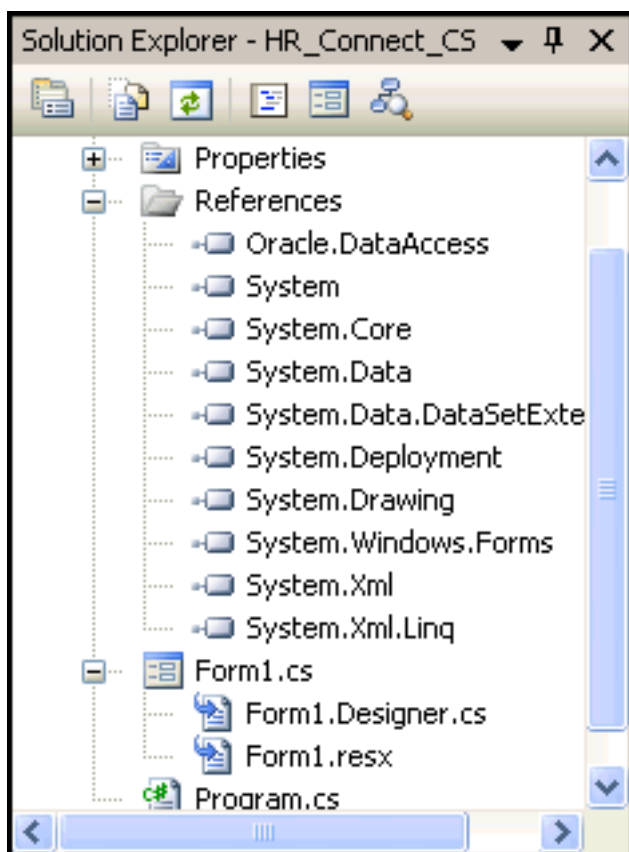
[reference01.gifの説明](#)

「Add Reference」ウィンドウが表示されます。

2. 「Add Reference」ウィンドウの「.NET」タブで**Oracle.DataAccess**を選択します。「OK」をクリックします。



「Solution Explorer」に新しい参照が表示されていることを確認します。



[reference03.gifの説明](#)

名前空間ディレクティブの追加

モジュール内でアセンブリの名前空間を指定できるように、Oracle名前空間ディレクティブを追加できます。これを行うには、C#のusing文またはVisual BasicのImports文をコード・ファイルの先頭または先頭付近に追加します。

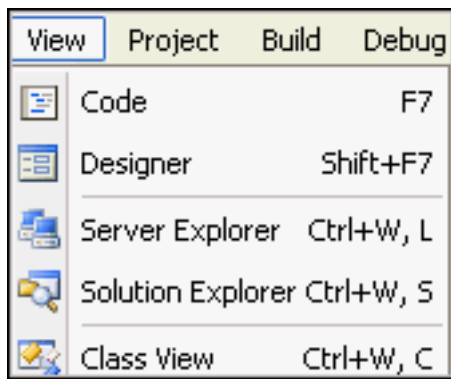
注意:

参照を追加すると、アプリケーション内で名前空間を使用できるようになります。名前空間ディレクティブをアプリケーション・コードに追加することで、名前空間が明確になり、スコープの追加が可能になります。

Oracle名前空間ディレクティブを追加するには、次の手順を実行します。

1. Form 1 をアクティブにして、「View」メニューから「Code」を選択します。

または、[F7]キーボード・ショートカットを使用することもできます。



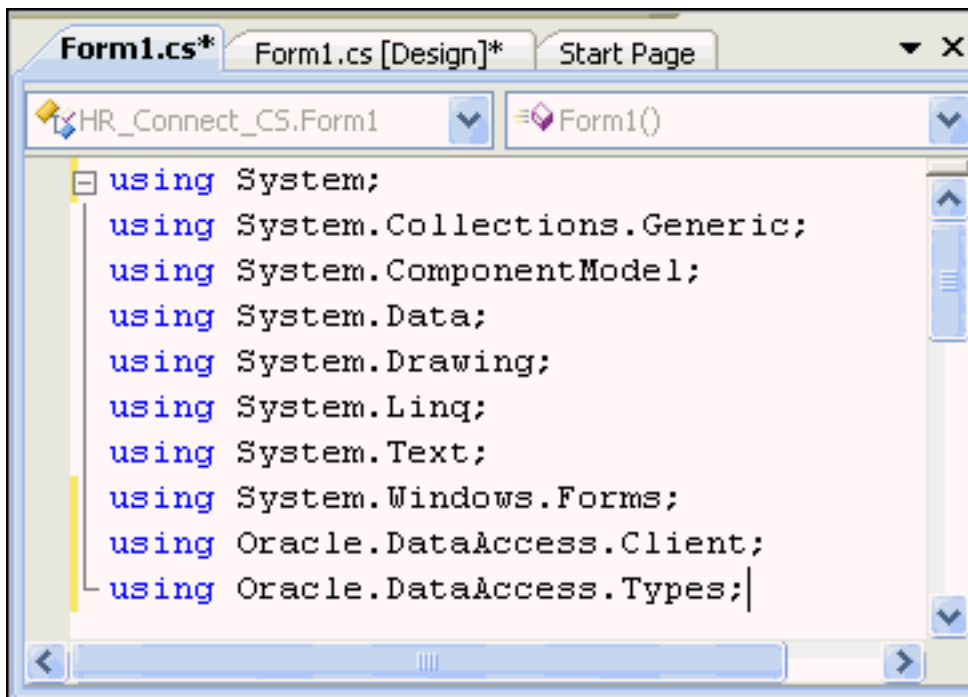
[connect04.gifの説明](#)

2. 使用している言語に応じて、次の文を宣言のリストに追加します。

○ Visual C#:

他のusing文とともに、名前空間の前に追加します。

```
using Oracle.DataAccess.Client;
using Oracle.DataAccess.Types;
```

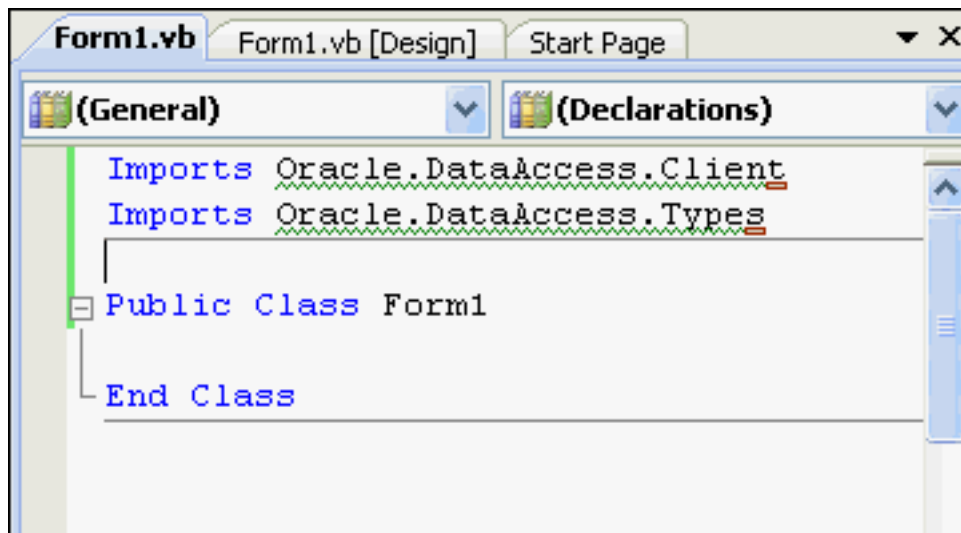


[connect05.gifの説明](#)

○ Visual Basic:

ファイルの先頭の宣言部に追加します。

```
Imports Oracle.DataAccess.Client
Imports Oracle.DataAccess.Types
```



[connect06.gifの説明](#)

3. 「File」メニューから「Save」を選択するか、[Ctrl] + [S]キーボード・ショートカットを使用して、変更を保存します。

ユーザー・インタフェースの設計

ユーザー・インタフェースは、デザイン・フォームにツールボックス・コントロールを追加して作成できます。このインタフェースは、ユーザーからの接続情報を受け入れます。

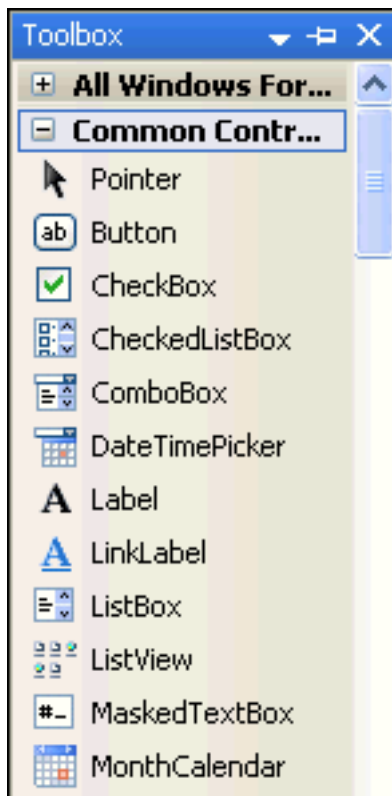
ツールボックス・コントロールを追加するには、次の手順を実行します。

1. 「View」メニューから「Designer」を選択します。

Form1をまだ開いていない場合は、この操作でForm1が設計ビューに表示されます。

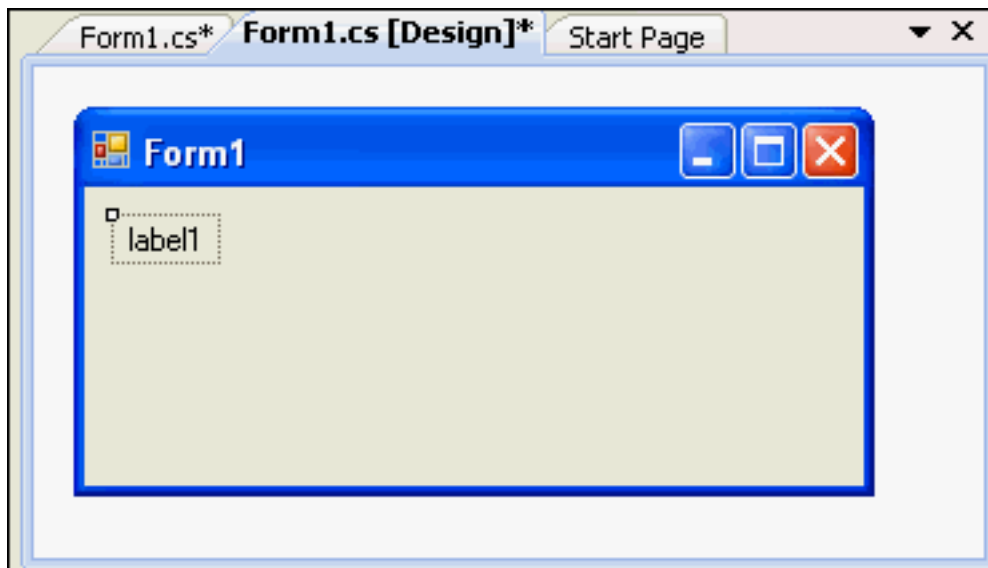
後で、「Code」と「Designer」を何度も切り替えます。 キーボード・ショートカットはそれぞれ[F7]と、[Shift] + [F7]です。

2. 「View」メニューから「Toolbox」を選択します。
3. 「Toolbox」で「Common Controls」を開きます。



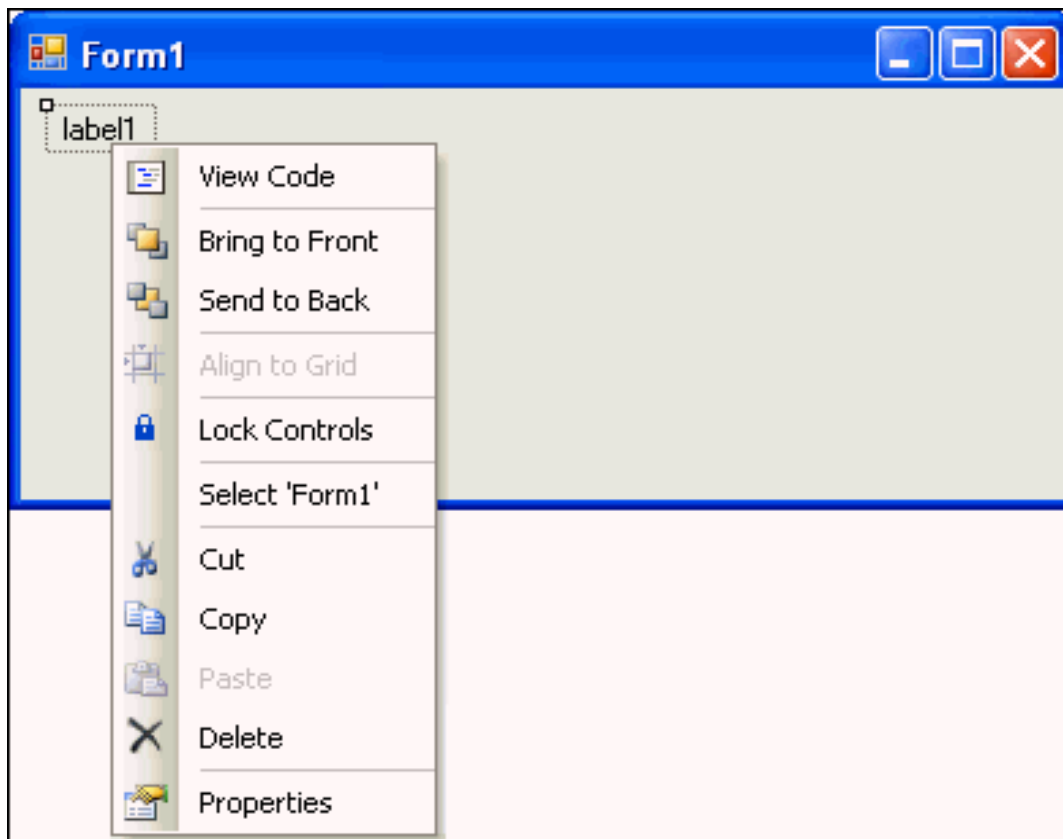
[control01.gifの説明](#)

4. 「Toolbox」で「Label」を選択し、Form1にドラッグします。



[control02.gifの説明](#)

5. Form1で「label1」を右クリックします。

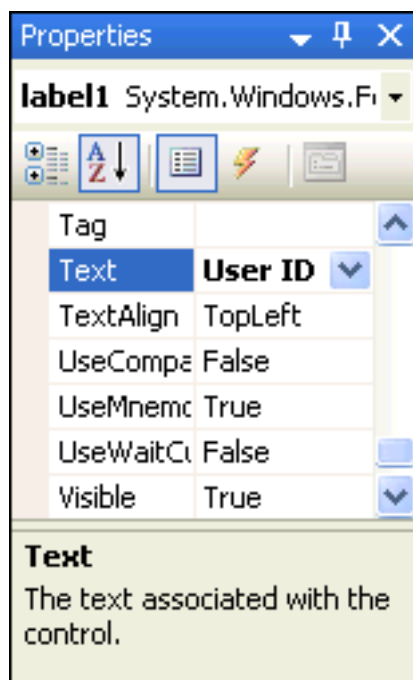


[control03.gifの説明](#)

6. 「Properties」 ウィンドウが表示されていない場合は、メニューから「**Properties**」を選択します。

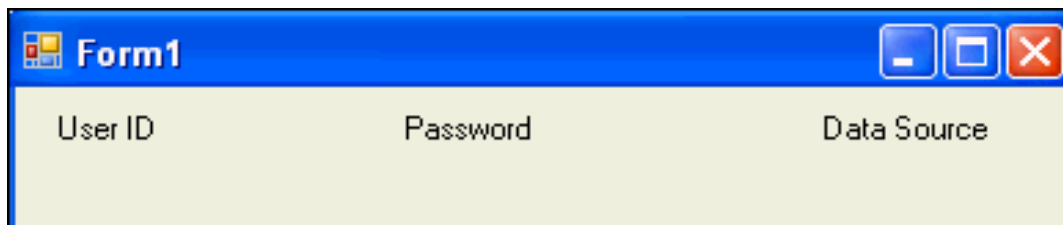
「Properties」 ウィンドウが表示されます。

7. 「Properties」 ウィンドウでTextプロパティを**label1**から**User ID**に変更します。



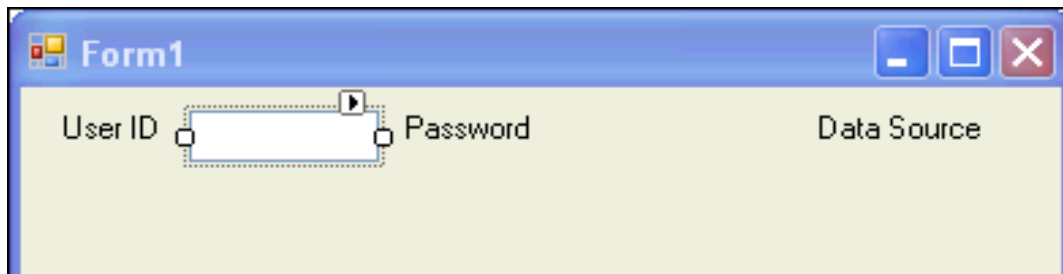
[control04.gifの説明](#)

8. 手順4から7を2回繰り返して、さらに2つのラベルをForm1に置きます。それらのTextプロパティを**Password**と**Data Source**に変更します。



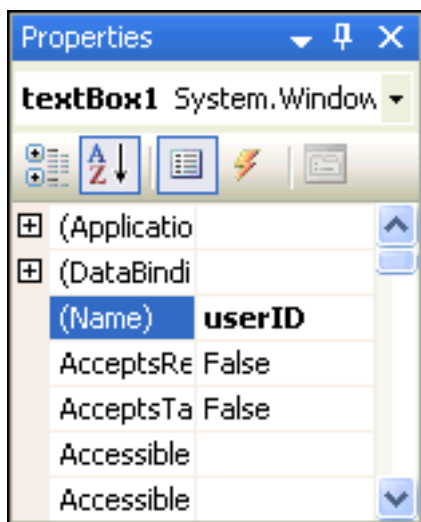
[control05.gifの説明](#)

9. 「Toolbox」で「TextBox」を選択し、Form1の「User ID」ラベルの隣にドラッグします。



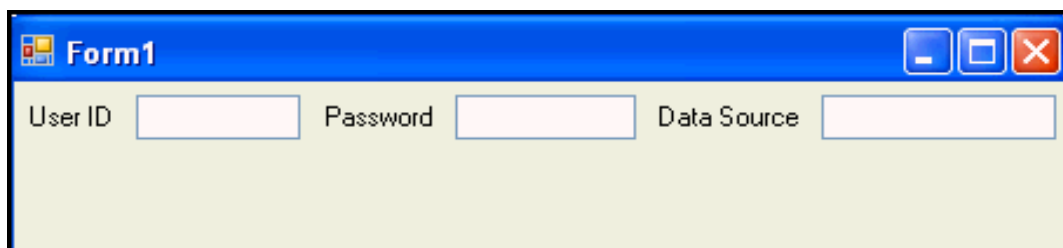
[control06.gifの説明](#)

10. 「Properties」ウィンドウで、Nameプロパティを**userID**に変更します。



[control07.gifの説明](#)

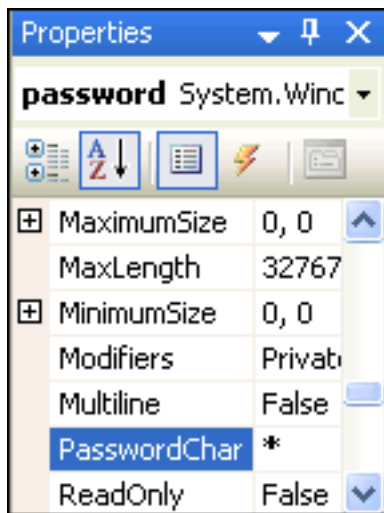
11. 手順9から10を2回繰り返し、さらに2つテキスト・ボックスをForm1の既存のラベルの隣に配置します。それらのNameプロパティを**password**と**dataSource**に変更します。



[control08.gifの説明](#)

12. 「Password」ラベルの隣のテキストボックスを選択します。「Properties」ウィンドウで、PasswordCharプロパティまでスクロールして、このプロパティをアスタリスク (*) に設定します。

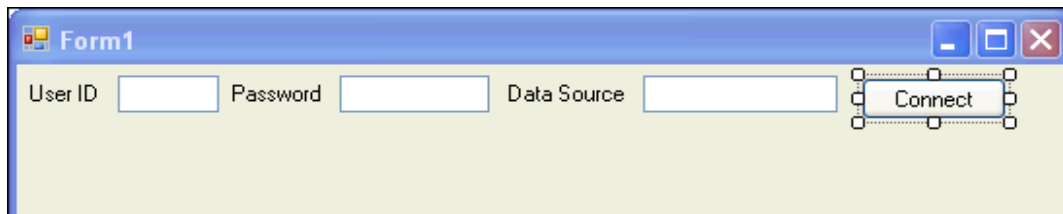
これにより、入力中のパスワードがマスクされます。



[control09.gifの説明](#)

- 「Toolbox」で「Button」を選択し、Form1にドラッグします。

「Properties」ウィンドウで、このボタンのTextプロパティを「button1」から「Connect」に変更し、Nameプロパティをconnectに変更します。



[control10.gifの説明](#)

- 保存します。
- ツールボックスを閉じます。

接続コードの記述

ここでは、ユーザー・インタフェースに入力された情報を使用してデータベースに接続するコードを記述します。

データベースに接続するには、接続オブジェクトを作成する必要があります。

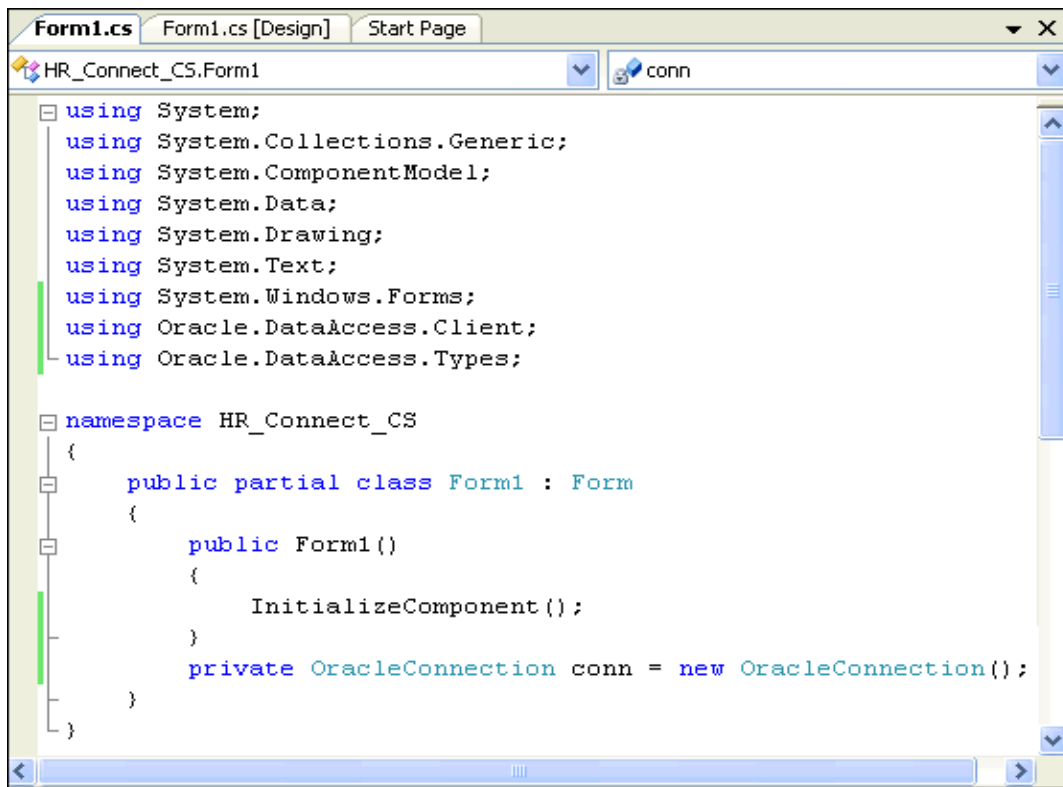
データベースに接続するコードを記述するには、次の手順を実行します。

この手順を実行すると、ユーザーがForm1のコントロールに入力したデータに基づいて、アプリケーションからデータベースに接続できるようになります。 [「アプリケーションのコンパイルと実行」](#)を参照してください。

- 「View」メニューから「Code」を選択します。
- データベースの接続文字列をインスタンス化するために、次のコードを追加します。

Visual C#: 次のコードを使用して、Form1クラスのpublic Form1()ブロックの直後にクラス変数connを追加します。

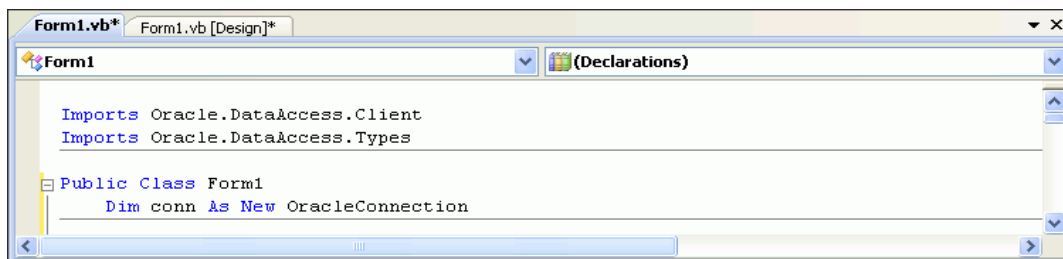
```
private OracleConnection conn = new OracleConnection();
```

[connect1.gifの説明](#)

Visual Basic: 次のコードを使用して、Form1クラス宣言にconnクラス変数を追加します。

```
Public Class Form1
    Dim conn As New OracleConnection
```



[connect2.gifの説明](#)

3. 変更を保存します。
4. 「View」メニューをクリックして「Designer」を選択し、「Designer」ビューに変更します。
5. Form1の「Connect」ボタンをダブルクリックし、connect_Click()メソッドのコード・ウィンドウを開きます。

次に示すコードをconnect_Click()メソッドに挿入します。

Visual C#:

```
conn.ConnectionString = "User Id=" + userID.Text +
    ";Password=" + password.Text +
    ";Data Source=" + dataSource.Text + ";";
conn.Open();
```

Visual Basic:

```
conn.ConnectionString = "User Id=" + userID.Text & _
    ";Password=" + password.Text & _
    ";Data Source=" + dataSource.Text
conn.Open()
```

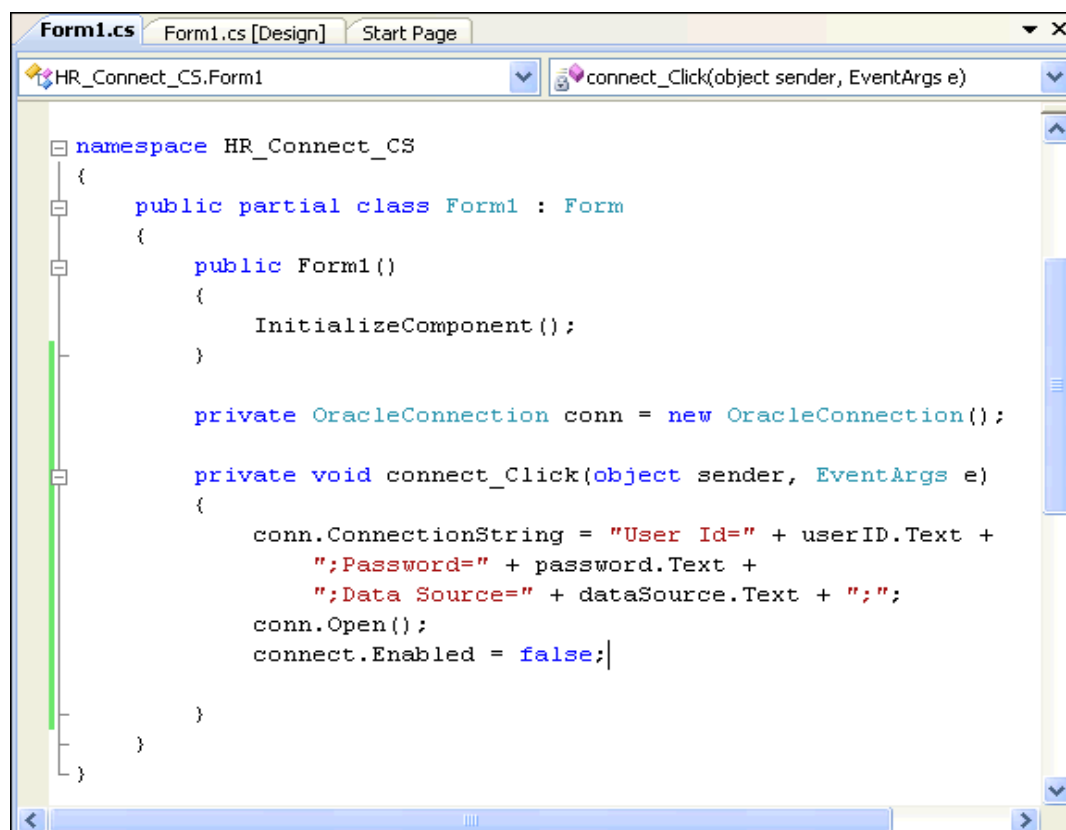
注意: 接続をオープンするには、ユーザーが入力したUser ID、PasswordおよびData Sourceの値から接続を作成する必要があります。Open()メソッドにより、実際の接続が作成されます。

- connect_Click()メソッドの最後に次のコードを挿入し、このボタンのEnabled属性をfalseに設定します。

これにより、「Connect」ボタンは無効になります。正しく接続できた後は、このようにすることをお勧めします。

Visual C#:

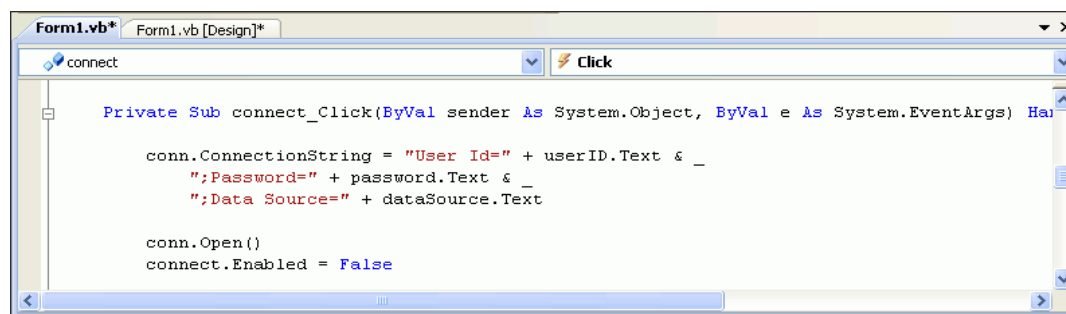
```
connect.Enabled = false;
```



[connect_obj1.gifの説明](#)

Visual Basic:

```
connect.Enabled = false
```



[connect_obj2.gifの説明](#)

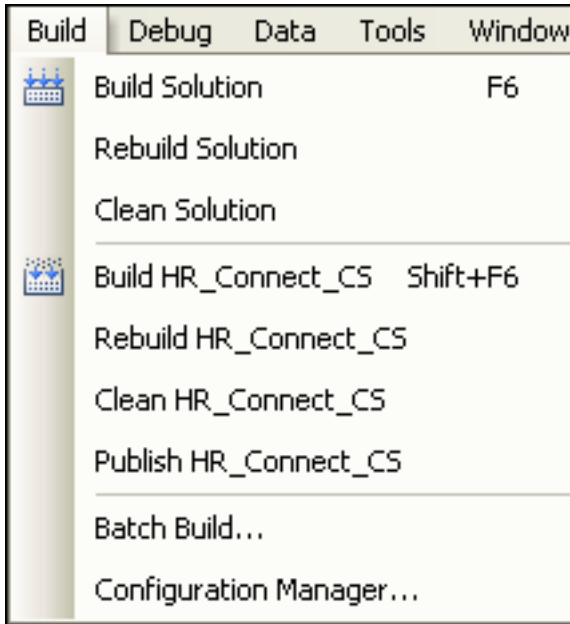
これで、Oracle Databaseに接続できるアプリケーションを記述できました。この後の項では、このアプリケーションの使用方法を説明します。

アプリケーションのコンパイルと実行

この項では、前の項で作成したアプリケーションをコンパイルして実行する方法を説明します。

アプリケーションをコンパイルして実行するには、次の手順を実行します。

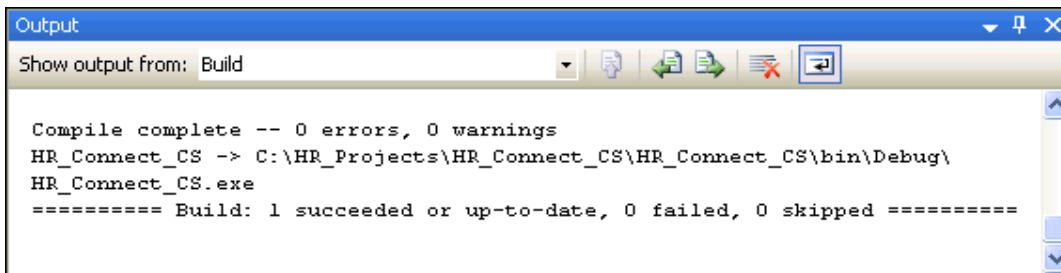
1. 「Build」メニューから「Build Solution」を選択します。



[build1.gifの説明](#)

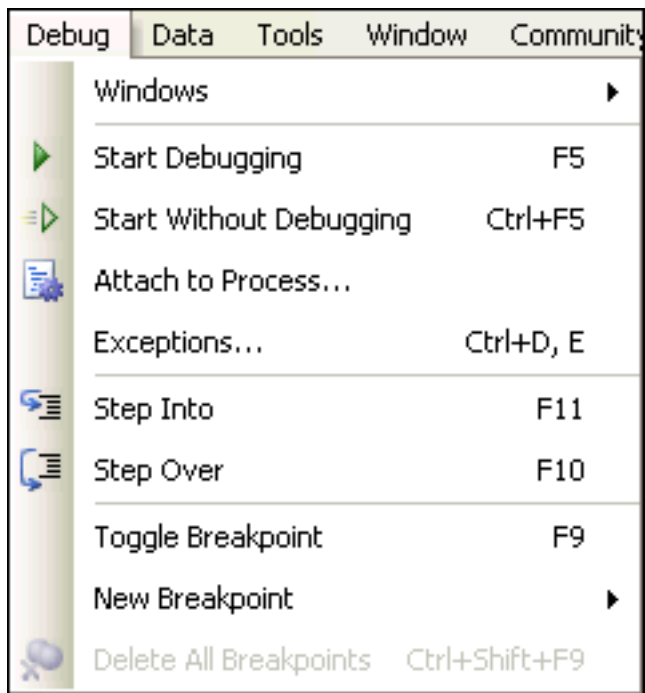
2. 「Output」ウィンドウにエラーが表示されておらず、「View」メニューから使用できることを確認します。

標準的な出力結果は次のとおりです。



[build2.gifの説明](#)

3. エラーがある場合は、「View」メニューから「Error List」を選択し、エラーを修正します。
4. 「Debug」メニューから「Start Without Debugging」を選択して、アプリケーションを実行します。



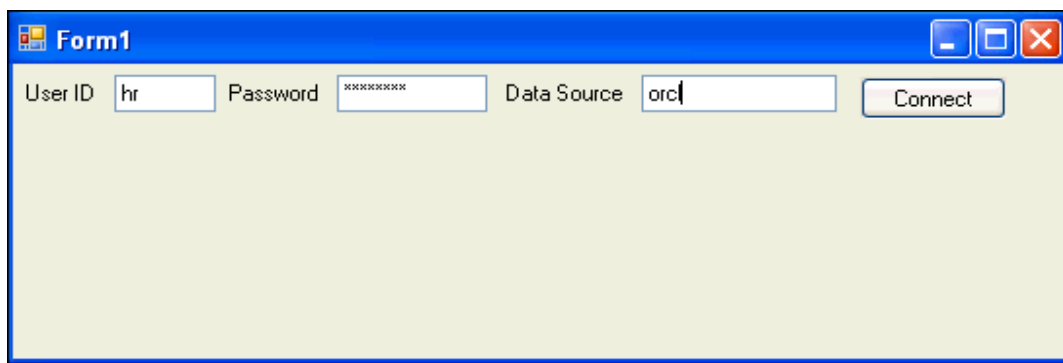
[build3.gifの説明](#)

- Form1アプリケーションの「User ID」、「Password」および「Data Source」を入力します。

「Connect」をクリックします。

これにより、アプリケーションでtnsnames.oraファイルが使用されます。 [「NET接続の別名の構成」](#)を参照してください。

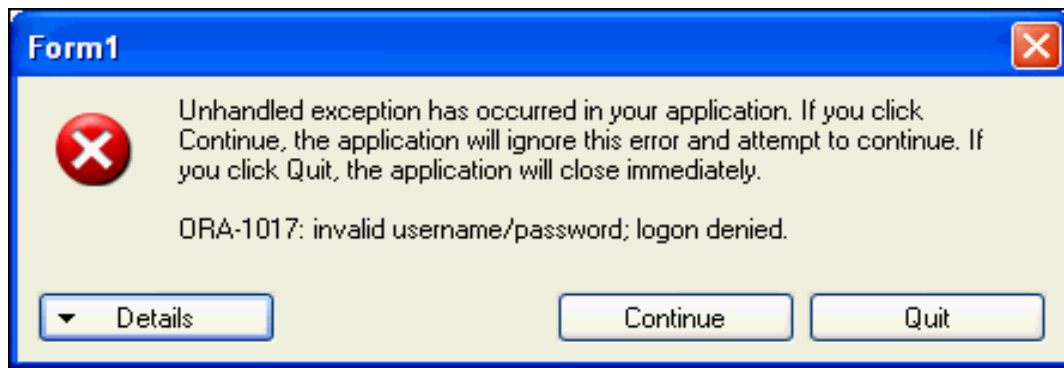
接続がオープンされたら、「Connect」ボタンは無効になります。これで、Oracle Databaseインスタンスへの接続が正しく実装されました。



[build4.gifの説明](#)

エラー処理

アプリケーションでは、実行時エラーを正常に処理する必要があります。たとえば、間違ったパスワードを使用してログインしようとする、ここまでに開発したアプリケーションではデータベース接続を確立できず、「ORA-1017: ユーザー名/パスワードが無効です。ログオンは拒否されました。」という未処理例外エラーで終了します。



[error1.gifの説明](#)

「Start Without Debugging」を再度選択して、別のパスワードでログインを試行する必要があります。

エラー処理では、プログラム実行の標準フローとは異なる状況が発生した場合に、その対処方法を決定します。Oracle Data Provider for .NETには、エラーを処理およびサポートする次の3つのクラスが含まれています。

- OracleErrorクラスは、Oracleにより報告される警告またはエラーを表します。
- OracleErrorCollectionクラスは、Oracle Data Provider for .NETによりスローされるすべてのエラーの集合を表します。これは、OracleErrorのリストを保持する単純なArrayListです。
- OracleExceptionクラスは、Oracle Data Provider for .NETでエラーが発生したときにスローされる例外を表します。各OracleExceptionオブジェクトの「Error」プロパティには、エラーまたは警告を示す1つ以上のOracleErrorオブジェクトが含まれます。

Try-Catch-Finallyブロック構造の使用

.NET言語では、エラー処理にTry-Catch-Finallyブロック構造を使用します。この構造では、Tryコードがメイン・コードとなり、アプリケーションで実現する目標になります。次の2つの項で説明するように、Catchコードで様々な種類のエラーが取得されます。最後にあるFinallyブロックは常に実行されます。

Finallyブロックには、接続をクローズして破棄するDisposeメソッドが含まれることがよくあります。FinallyブロックにDisposeメソッドを記述すると、Try-Catch-Finallyブロックが完了した後、データベース接続は常にクローズされます。アプリケーションでのデータベース・アクセスが不要になった後にデータベース接続をクローズすることは、特にデータ・セキュリティなど、多くの理由で重要です。

クローズされているデータベース接続をクローズしようとしてもエラーは発生しません。この試行は適切でないように思われます。しかし、Dispose()をFinallyコード・ブロックに置くことで、接続が確実にクローズされるようになります。

次の項では、一般的なエラーでのTry-Catch-Finallyブロック構造の使用方法を説明します。その後の項では、Oracleエラーでの使用方を説明します。

一般的なエラーの処理

この項では、Try-Catch-Finallyブロックを使用して一般的なエラーを処理する方法を説明します。

一般的なエラーを処理するには、次の手順を実行します。

1. Try-Catch-Finally構文の実装を追加して、Form1のconnect_Click()メソッドのコードを変更します。

新しいコードは太字で示しています。

Visual C#:

```
private void connect_Click(object sender, EventArgs e)
```

```

{
    conn.ConnectionString = "Data Source=ORCL;User Id="
        + userID.Text + ";Password=" + password.Text + ";";
    try
    {
        conn.Open();
        connect.Enabled = false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
    finally
    {
        conn.Dispose();
    }
}

```

または、次のようにusingキーワードを使用して、有効でなくなった接続を破棄するC#構文を使用することもできます。

```

using (OracleConnection conn = new OracleConnection())
{
    conn.Open();
    // application code
    ...
}

```

Visual Basic:

```

Try
    conn.Open()
    connect.Enabled = false

Catch ex As Exception
    MessageBox.Show(ex.Message.ToString())

Finally
    conn.Dispose()
End Try

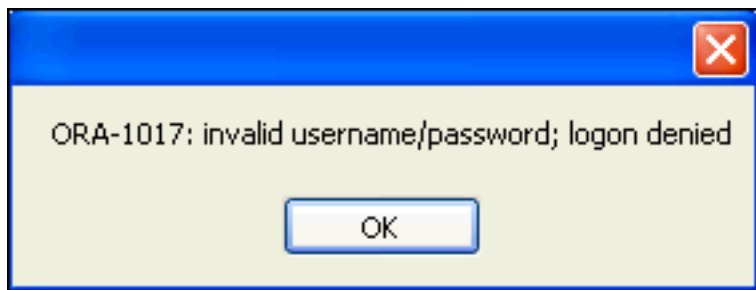
```

2. 「Build」メニューから「Rebuild Solution」を選択します。

エラーがないことを確認します。

3. 「Debug」メニューから「Start Without Debugging」を選択します。

4. [「アプリケーションのコンパイルと実行」](#)の手順に従ってアプリケーションを再度実行し、間違ったパスワードを使用して接続を試行します。今度はアプリケーションでエラーが取得され、ポップアップ・ウィンドウに「ORA-1017: ユーザー名/パスワードが無効です。ログオンは拒否されました。」と表示されます。



[error2.gifの説明](#)

一般的なOracleエラーの処理

次に示す完成したTry-Catch-Finallyブロック・コードでは、OracleExceptionがない場合は、最初のCatch文のブランチがスキップされます。2番目のCatch文のブランチで、その他のすべてのExceptionが取得されます。

最初のCatch文にはCase文が含まれており、これにより一般的なデータベース・エラーがトラップされ、ユーザーにわかりやすい方法で表示されます。

2番目のCase文では、データベースにアクセスできない場合に、特定のOracleExceptionが取得されます。

特定のエラーを処理するには、次の手順を実行します。

1. データベース・インスタンスを停止します。 [付録A「Oracle Databaseインスタンスの起動および停止」](#)を参照してください。
2. 次の太字で示されているCatch OracleExceptionブロックを、connect_Click()メソッドに以前に追加したCatch Exceptionブロックの前に追加します。

Visual C#:

```
try
{
    conn.Open();
    connect.Enabled = false;
}
catch (OracleException ex)
{
    switch (ex.Number)
    {
        case 1:
            MessageBox.Show("Error attempting to insert duplicate data.");
            break;
        case 12560:
            MessageBox.Show("The database is unavailable.");
            break;
        default:
            MessageBox.Show("Database error: " + ex.Message.ToString());
            break;
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString());
}
finally
{
    conn.Dispose();
}
}
```

Visual Basic:

```
Try
    conn.Open()
    connect.Enabled = false

Catch ex As OracleException ' catches only Oracle errors
    Select Case ex.Number
        Case 1
            MessageBox.Show("Error attempting to insert duplicate data.")
        Case 12560
            MessageBox.Show("The database is unavailable.")
        Case Else
            MessageBox.Show("Database error: " + ex.Message.ToString())
    End Select

Catch ex As Exception
    MessageBox.Show(ex.Message.ToString())

Finally
    conn.Dispose()
End Try
```

3. [「アプリケーションのコンパイルと実行」](#)の手順に従ってアプリケーションを再度コンパイルして実行します。

ORA-12560エラーが、エラー番号なしで「The database is unavailable」としてポップアップ・ウィンドウに表示されることを確認します。



[error04.gifの説明](#)

4. データベース・インスタンスを再起動します。 [付録A「Oracle Databaseインスタンスの起動および停止」](#)を参照してください。

4 Oracle Data Provider for .NETでの取得と更新

この章の内容は次のとおりです。

- [コマンド・オブジェクトの使用](#)
- [データの取得：単純な問合せ](#)
- [データの取得：バインド変数](#)
- [データの取得：複数の値](#)
- [Oracle Data Provider for .NETでのDataSetクラスの使用](#)
- [データベースの更新の有効化](#)
- [データの挿入、削除および更新](#)

コマンド・オブジェクトの使用

データベース内のデータを表示、編集、挿入または削除するには、SQLコマンド、ストアド・プロシージャまたは表名を指定して、OracleCommandオブジェクトにリクエストをカプセル化する必要があります。OracleCommandオブジェクトは、リクエストを作成してデータベースに送信し、結果を戻します。

コマンド・オブジェクトを使用するには、次の手順を実行します。

1. [第3章「ODP.NETによる単純な.NETアプリケーションの作成」](#)で作成したアプリケーションHR_Connect_xxから、Form1.xxのコピーを2つ作成します。コピーを作成する方法については、[付録B「フォームのコピー」](#)を参照してください。

コピーにForm2.csまたはForm2.vb、およびForm3.csまたはForm3.vbと名前を付けます。最初のコピーは、この章の前半で使用します。2つ目のコピーは、この章の後半で使用します。

2. Form2.csまたはForm2.vbを開きます。

コード・ファイルの名前は変更しましたが、プロジェクト内の実際のフォーム・コントロールの名前は変更していないため、デザイナーのフォームはまだForm1となっていることに注意してください。

3. SQL問合せを表す文字列を作成し、try文の本体に追加します。

新しいコードは太字で示しています。

Visual C#:

```
try
{
    conn.Open();
    connect.Enabled = false;

    // SQL Statement
    string sql = "select department_name from departments"
        + " where department_id = 10";
}
```

Visual Basic:

```
Try
    conn.Open()
    connect.Enabled = False

    Dim sql As String = "select department_name from departments" & _
        "where department_id = 10"
```

4. 新しいsql変数を使用してOracleCommandオブジェクトを作成し、テキスト・コマンドが実行されるようにCommandTypeプロパティを設定します。

Visual C#:

```
try
{
    conn.Open();
    connect.Enabled = false;

    // SQL Statement
    string sql = "select department_name from departments"
        + " where department_id = 10 ";

    OracleCommand cmd = new OracleCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
}
```

Visual Basic:

```
Try
    conn.Open()
    connect.Enabled = False

    Dim sql As String = "select department_name from departments" & _
        "where department_id = 10"

    Dim cmd As New OracleCommand(sql, conn)
    cmd.CommandType = CommandType.Text
```

5. 実行した内容を保存します。

データの取得：単純な問合せ

この項では、データベースからデータを取得する方法を説明します。

OracleCommandオブジェクトのExecuteReader()メソッドによりOracleDataReaderオブジェクトが戻されます。これにアクセスすることでフォームに結果を表示できます。このアプリケーションでは、ListBoxを使用して結果を表示します。

データを取得するには、次の手順を実行します。

1. 次に示すコードをconnect_Click()メソッドのTryブロックの最後に追加して、OracleDataReaderオブジェクトを作成します。

これにより、問合せ結果を読み取ることができます。

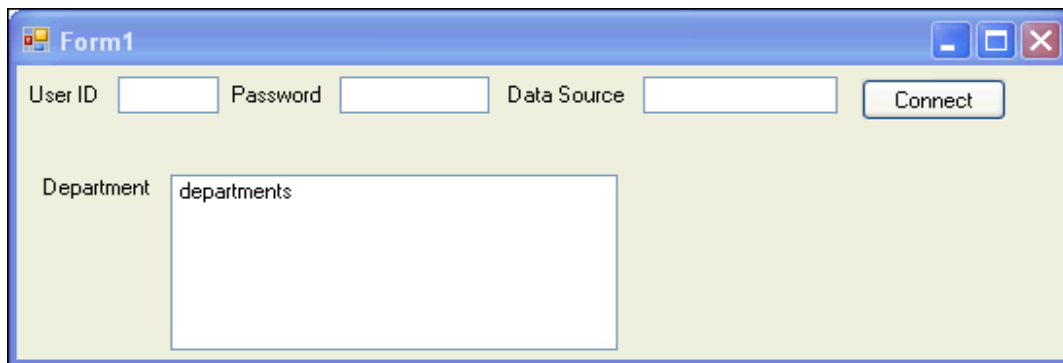
Visual C#:

```
OracleDataReader dr = cmd.ExecuteReader();
dr.Read();
```

Visual Basic:

```
Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()
```

2. 設計ビューでForm1を開きます。「View」メニューから「Designer」を選択します。
3. 「View」メニューから「Toolbox」を選択します。
4. 「Toolbox」から「Label」を選択し、Form1にドラッグします。
5. 「View」メニューから「Properties Window」を選択します。
6. 「Properties」ウィンドウで、ラベルの「Text」をDepartmentに変更します。
7. 「Toolbox」の「Window」フォームから、「ListBox」を選択し、Form1にドラッグします。
8. 「Properties」ウィンドウの「Design」で、「Name」をdepartmentsに変更します。



[simple1.gifの説明](#)

9. 問合せ結果からデータを取得するためのアクセッサ型メソッドを追加します。

「Connect」ボタンをダブルクリックしてconnect_click()メソッドを編集し、次に示すコードをTryブロックの最後に追加します。

Visual C#:

```
departments.Items.Add(dr.GetString(0));
```

Visual Basic:

```
departments.Items.Add(dr.GetString(0))
```

GetStringなどの型指定されたアクセッサは、ネイティブの.NETデータ型およびネイティブのOracleデータ型を戻します。結果セットのどの列を戻すかは、アクセッサに渡されたゼロベースの序数で指定します。

10. アプリケーションをビルドして保存します。
11. アプリケーションを実行します。ログインおよびデータソースを入力します。

接続すると、「Department」リスト・ボックスにAdministrationと表示されます。これはHRスキーマに含まれる部門番号10の正しい名前です、SELECT文でリクエストしたものです。

[simple2.gifの説明](#)

データの取得：バインド変数

バインド変数はSQL文内のプレースホルダです。データベースではSQL文を受信すると、その文がすでに実行されたことがありメモリーに格納されているかどうかを確認されます。その文がメモリーに存在する場合、Oracle Databaseではその文を再利用でき、文の解析と最適化のタスクがスキップされます。バインド変数を使用すると、異なる入力値でも文の再利用が可能です。また、バインド変数を使用すると、データベースの問合せパフォーマンスが向上するだけでなく、入力に含まれるリテラル引用符の特別な処理が不要になり、SQLインジェクション攻撃から保護することができます。

次のコードは、バインド変数を使用せずに、文のWHERE句に値10を指定する標準的なSELECT文です。

```
SELECT department_name FROM departments WHERE department_id = 10
```

次のコードは、数値をバインド変数:department_idで置き換えたものです。バインド変数識別子は、常に1つのコロン (:) で始まります。

```
SELECT department_name FROM departments WHERE department_id = :department_id
```

バインド変数はUPDATE、INSERTおよびDELETE文でも使用でき、ストアド・プロシージャでも使用できます。次のコードは、UPDATE文でバインド変数を使用する方法を示しています。

```
UPDATE departments SET department_name = :department_name  
WHERE department_id = :department_id
```

詳細は、[「データの挿入、削除および更新」](#)を参照してください。

.NETコードで各バインド変数を表すには、OracleParameterクラスを使用できます。OracleParameterCollectionクラスには、各文のOracleCommandオブジェクトと関連付けられたOracleParameterオブジェクトが含まれています。OracleCommandクラスは、SQL文をデータベースに渡し、結果をアプリケーションに戻します。

変数は、OracleCommandプロパティBindByNameの設定（デフォルトはfalse）によって、位置または名前でバインドできます。

- 位置指定によるバインド

Add() メソッドを使用して、SQL文またはストアド・プロシージャで記述されるのと同じ順序でパラメータをOracleParameterCollectionに追加する必要があります。

- 名前指定によるバインド

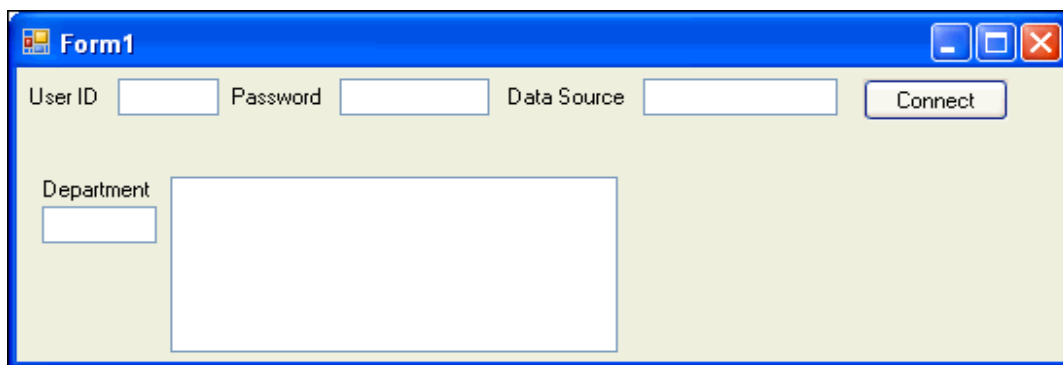
任意の順序でパラメータをコレクションに追加できます。ただし、パラメータ・オブジェクトのParameterNameプロパティを、ストアド・プロシージャで宣言されているバインド変数識別子と同じ名前に設定する必要があります。

バインド・モード（位置指定または名前指定）の他に、.NET開発者はDirection、OracleDbType、SizeおよびValueの各プロパティを、各パラメータ・オブジェクトに設定します。

- **Direction**バインド変数は、出力、入力または入出力パラメータとして使用できます。Directionプロパティは、各パラメータの向きを示します。Directionプロパティのデフォルト値はInputです。
- **OracleDbType**プロパティは、パラメータがnumber、date、VARCHAR2などのいずれであるかを示します。
- **Size**は、VARCHAR2などの可変長データ型のパラメータで保持できる最大データ・サイズを示します。
- **Value**には、文の実行前（入力パラメータ）、実行後（出力パラメータ）または実行前後の両方（入出力パラメータ）のパラメータ値が保持されます。

バインド変数を使用してデータを取得するには、次の手順を実行します。

1. Departmentsという名前のListBoxを右側に移動します。
2. 「View」メニューから「Toolbox」を選択します。
3. 「Toolbox」から「TextBox」を選択し、Form1のDepartmentという名前のラベルの下にドラッグします。
4. 「View」メニューから「Properties Window」を選択します。
5. 「Properties」ウィンドウで、「Name」をdepartmentIDに変更します。



[bind0.gifの説明](#)

6. 次を示すコードをconnect_Click()メソッドのTryブロックに追加して、バインド変数が使用されるようにSELECT文を変更します。

変更されたコードまたは新しいコードは太字で示しています。

Visual C#:

```
string sql = "select department_name from departments where department_id = " +
    "":department_id"";
OracleCommand cmd = new OracleCommand(sql, conn);
cmd.CommandType = CommandType.Text;
OracleParameter p_department_id = new OracleParameter();
p_department_id.OracleDbType = OracleDbType.Decimal;
p_department_id.Value = departmentID.Text;
cmd.Parameters.Add(p_department_id);

OracleDataReader dr = cmd.ExecuteReader();
dr.Read();

departments.Items.Add(dr.GetString(0));
```

Visual Basic:

```
Dim sql As String = "select department_name from departments where" & _
    "department_id= ":department_id""
```

```
Dim cmd As OracleCommand = New OracleCommand(sql, conn)
cmd.CommandType = CommandType.Text
Dim p_department_id as OracleParameter = new OracleParameter()
p_department_id.OracleDbType = OracleDbType.Decimal
p_department_id.Value = departmentID.Text
cmd.Parameters.Add(p_department_id)

Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()

departments.Items.Add(dr.GetString(0))
```

このコードでは、パラメータ・オブジェクトはOracleDbTypeプロパティを設定しますが、Directionプロパティにはデフォルト値Inputを使用するため、設定は不要です。オブジェクトは入力パラメータであり、データ・プロバイダは値からサイズを判断できるため、Sizeプロパティを設定する必要はありません。

7. アプリケーションを保存して実行します。
8. ログイン情報と、HRスキーマの代表的な部門番号（50など）を入力します。
9. 「Connect」をクリックします。

部門IDに対応する部門名がアプリケーションから戻されます。

[bind1.gifの説明](#)

データの取得：複数の値

データベースから複数の値を取得することが必要になる場合はよくあります。複数の列および複数の行の値を取得するには、DataReaderオブジェクトを使用できます。次の例で、複数の列や複数の行に対する問合せについて考えてみます。

```
SELECT department_id, department_name, manager_id, location_id
FROM departments
WHERE department_id < 100
```

DataReaderオブジェクトから複数の行を処理するには、ループ構造が必要です。また、複数の行を表示できるコントロールが役立ちます。OracleDataReaderオブジェクトは前進専用で読取り専用のカーソルであるため、Windows FormsのDataGridコントロールなどの、更新可能なコントロールまたは後方にスクロールできるコントロールにはバインドできません。ただし、OracleDataReaderオブジェクトはListBoxコントロールと互換性があります。

複数の値を取得するには、次の手順を実行します。

1. connect_Click()メソッドのTryブロックで、複数の行の結果セットが戻され、部門名を表示するreadメソッドを囲むwhileループを追加するようにSQL問合せを変更します。

Visual C#:

```
try
{
```

```

...
string sql = "select department_name from departments where department_id" +
"<:&department_id";
...
while (dr.Read())
{
    departments.Items.Add(dr.GetString(0));
}
}

```

Visual Basic:

```

Try
...
Dim sql As String = "select department_name from departments " & _
    "where department_id <:&department_id"
...
While (dr.Read())
    departments.Items.Add(dr.GetString(0))
End While

```

2. アプリケーションを保存して実行します。
3. ログイン情報を入力し、部門に50と入力します。
4. 「Connect」をクリックします。

問合せに対応する部門名が、アプリケーションから戻されます。

[bind2.gifの説明](#)

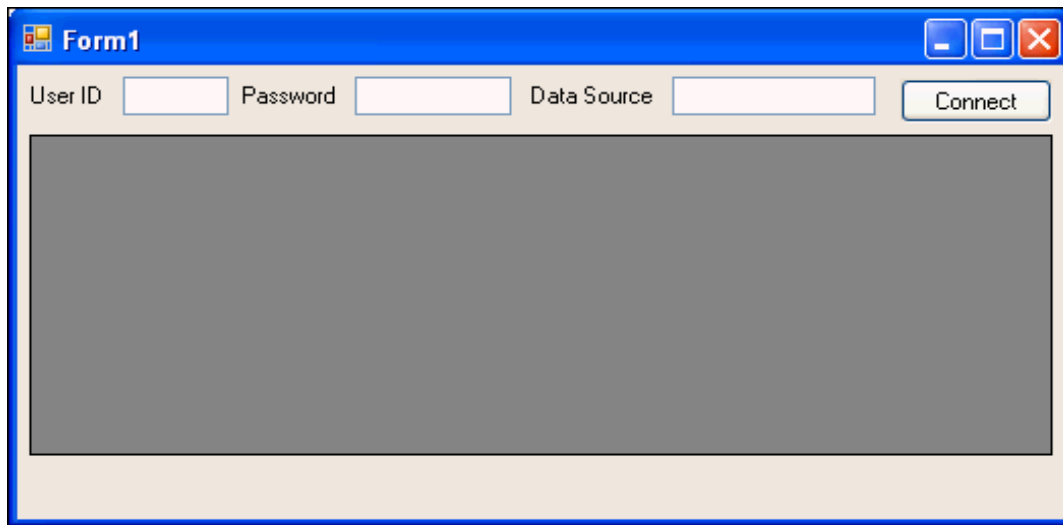
Oracle Data Provider for .NETでのDataSetクラスの使用

DataSetクラスでは、メモリーに常駐するデータベース・データのコピーが提供されます。これは、リレーショナル・データまたはXMLデータを格納する1つ以上の表で構成されます。OracleDataReaderオブジェクトとは異なり、DataSetは更新可能で後方にスクロールできます。

Datasetクラスを使用するには、次の手順を実行します。

1. 第3章で作成したForm1のコピーを作成していない場合はForm1をコピーし、[付録B「フォームのコピー」](#)で説明するとおり、Form3.vb または .cs という名前を付けます。Form1.xxがSolution Explorerに表示されない場合は、「Project」メニューから「Show All Files」を選択します。
2. 「View」メニューから「Designer」ビューを選択します。
3. 「View」メニューから「Toolbox」を選択します。
4. 「Toolbox」から「DataGridView」を選択し、Form1にドラッグします。

5. 「View」メニューから「Properties Window」を選択します。
6. 「Properties」ウィンドウで、データ・グリッド・ビューの「Name」をdepartmentsに変更します。



[dataset1.gifの説明](#)

7. 「View」メニューから「Code」を選択します。
8. 次に示すように、コードのconn宣言の直後に、クラス変数への変数宣言を追加します。

Visual C#:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private OracleConnection conn = new OracleConnection();
    private OracleCommand cmd;
    private OracleDataAdapter da;
    private OracleCommandBuilder cb;
    private DataSet ds;
    ...
}
```

Visual Basic:

```
Public Class Form1
    Dim conn As New OracleConnection
    Private cmd As OracleCommand
    Private da As OracleDataAdapter
    Private cb As OracleCommandBuilder
    Private ds As DataSet
End Class
```

9. connect_Click()メソッドのTryブロックに、次のコードを追加します。
 - データベースを問い合わせるコード
 - コマンドの問合せ結果をDataSetに埋め込むコード
 - DataSetをデータ・グリッド (departments) にバインドするコード

Visual C#:

```
conn.Open();
connect.Enabled = false;
```



```

string sql = "select * from departments where department_id < 60";
cmd = new OracleCommand(sql, conn);
cmd.CommandType = CommandType.Text;

da = new OracleDataAdapter(cmd);
cb = new OracleCommandBuilder(da);
ds = new DataSet();

da.Fill(ds);

departments.DataSource = ds.Tables[0];

```

Visual Basic:

```

conn.Open()
connect.Enabled = False

Dim sql As String = "select * from departments where department_id < 60"
cmd = New OracleCommand(sql, conn)
cmd.CommandType = CommandType.Text

da = New OracleDataAdapter(cmd)
cb = New OracleCommandBuilder(da)
ds = New DataSet()

da.Fill(ds)

departments.DataSource = ds.Tables(0)

```

10. アプリケーションをビルドして保存します。
11. アプリケーションを実行し、ログインおよびデータソースを入力します。

データベースに正しく接続されると、データ・グリッドに問合せ結果が移入されます。

	DEPARTMENT_ID	DEPARTMENT_N	MANAGER_ID	LOCATION_ID
▶	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500
*				

[dataset3.gifの説明](#)

データベースの更新の有効化

この時点で、DataSetにはデータベース・データのクライアント・コピーが保持されています。この項では、クライアント・データの変更をデータベースに保存できるようにするボタンを追加します。その後の項では、データの更新、挿入および削除のテスト方法を説明します。

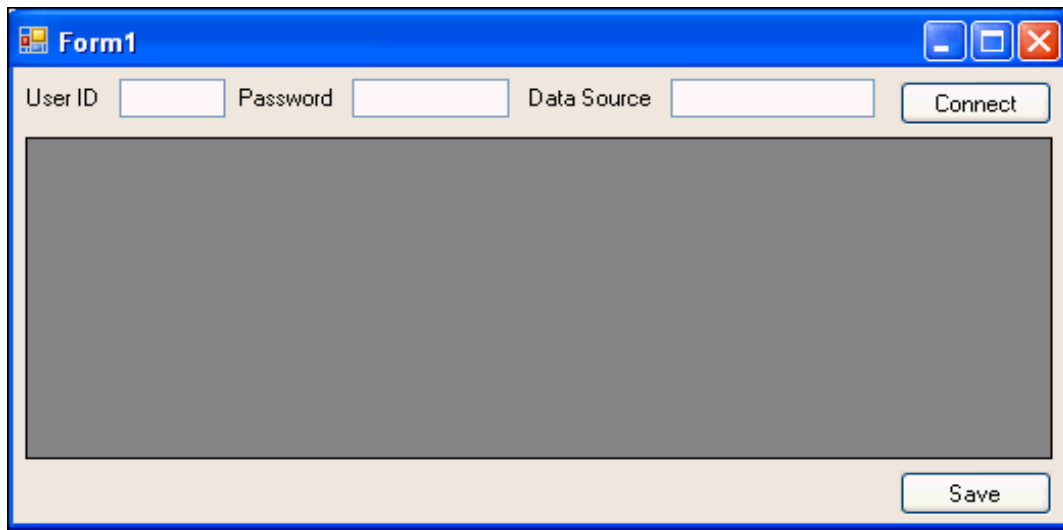
DataSetからデータベースにデータを保存できるようにするには、次の手順を実行します。

1. 「Toolbox」から「Button」をForm1にドラッグ・アンド・ドロップします。

2. 「Properties」 ウィンドウで、ボタンの「Name」をsaveに変更します。

TextプロパティをSaveに変更します。

3. 「Properties」 ウィンドウの上部で、「Events」 (稲妻のアイコン) をクリックします。イベントのリストでクリック・イベントを選択します。2番目の列にイベント名save_Clickを入力します。



[dataset2.gifの説明](#)

4. 「View」メニューから「Code」を選択します。
5. 次に示すように、データを更新するコードをsave_Click()メソッドの本体に追加します。

Visual C#:

```
da.Update(ds.Tables[0]);
```

Visual Basic:

```
da.Update(ds.Tables(0))
```

「Error List」にエラーが表示される場合があります。これらのエラーは、次の手順でコードを追加する表示されなくなります。

6. Form()メソッドまたはForm1_Loadメソッドに、次のコードを追加します。

Visual C#:

```
public Form1()
{
    InitializeComponent();
    save.Enabled = false;
}
```

Visual Basic:

```
Private Sub Form1_Load(ByVal sender As System.Object, &_
    ByVal e As System.EventArgs) Handles MyBase.Load
    save.Enabled = false
```

7. 次のように、connect_Click()メソッドのTryブロックに、「Save」ボタンを有効にするコードを追加します。

Visual C#:

```
conn.Open();  
...  
departments.DataSource = ds.Tables[0];  
  
save.Enabled = true;
```

Visual Basic:

```
conn.Open()  
...  
departments.DataSource = ds.Tables(0)  
  
save.Enabled = True
```

- conn.Dispose() コールをconnect_Click() メソッドのFinally ブロックから削除します。

注意: この例で前に使用したコードでは、接続を破棄またはクローズするためにこのメソッドが必要でした。しかし、コードを変更したため、問合せ結果が戻された後も接続をオープン状態のままにし、エンド・ユーザーによるデータ変更がデータベースに伝播されるようにする必要があります。一般的なオーバーライド・コールである components.Dispose() は、すでに Form1 の定義に含まれています。

- アプリケーションをビルドして保存します。
- アプリケーションを実行し、ログインおよびデータソースを入力します。

データベースに正しく接続されると、データ・グリッドに問合せ結果が移入されます。

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
▶	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500
*				

[dataset3a.gifの説明](#)

データの挿入、削除および更新

この項では、新しいアプリケーションを使用してデータベースのデータを直接操作する方法を説明します。

データを挿入、削除および更新するには、次の手順を実行します。

- 前の項で作成したアプリケーションを実行し、ログインおよびデータソースを入力して、データベースに接続します。
- データ・グリッドの一番下にある*プロンプトで、新しいレコードを入力します。

- 「DEPARTMENT_ID」に5と入力します。
- 「DEPARTMENT_NAME」にCommunity Outreachと入力します。
- 「MANAGER_ID」には値を入力せず、そのままにします。
- 「LOCATION_ID」に1700と入力します。

Form1

Userid: hr Password: ***** Data Source: orcl Connect

	DEPARTMENT_ID	DEPARTMENT_N	MANAGER_ID	LOCATION_ID
	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500
▶	5	Community Outre...		1700
*				

Save

[dataset4.gifの説明](#)

3. 「Save」をクリックします。
4. アプリケーションを閉じ、新しいレコードが保存されているかどうかを確認します。
5. アプリケーションを再度実行し、データベースに接続します。

DEPARTMENT_IDが番号順に表示され、新しい部門がDEPARTMENTS表の先頭にあることを確認します。

6. 部門名をCommunity Volunteersに変更して、「Save」ボタンをクリックします。

Form1

Userid: hr Password: ***** Data Source: orcl Connect

	DEPARTMENT_ID	DEPARTMENT_N	MANAGER_ID	LOCATION_ID
▶	5	Community Volun...		1700
	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500
*				

Save

[dataset10.gifの説明](#)

7. 手順4を繰り返します。アプリケーションを再度実行し、データベースに接続して、部門名が変更されていることを確認します。
8. 変更したばかりのレコード全体を選択し（一番左の列にあるカーソル・アイコンをクリック）、[Delete]キーを使用してこのレコード

を削除します。「Save」ボタンをクリックします。

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
5	Community Volun...		1700
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500

[dataset11.gifの説明](#)

9. 手順4を繰り返します。アプリケーションを再度実行し、データベースに接続して、新しいレコードがDEPARTMENTS表に含まれていないことを確認します。
10. アプリケーションを閉じます。

5 Oracle Developer Tools for Visual Studioの使用

この章の内容は次のとおりです。

- [Oracle Developer Toolsの使用](#)
- [Oracle Databaseへの接続](#)
- [表および表の列の作成](#)
- [表の索引の作成](#)
- [表の制約の追加](#)
- [表へのデータの追加](#)
- [データを表示および更新するためのコードの自動生成](#)

Oracle Developer Toolsの使用

Oracle Developer Tools for Visual Studio (ODT) は、Visual Studio用に緊密に統合されたアドインです。ODTによりServer Explorerに組み込まれる拡張機能を使用することで、表、索引、制約、データ接続などのデータベース・スキーマ・オブジェクトを自動作成できます。また、アプリケーション・コードも自動生成できます。

参照:

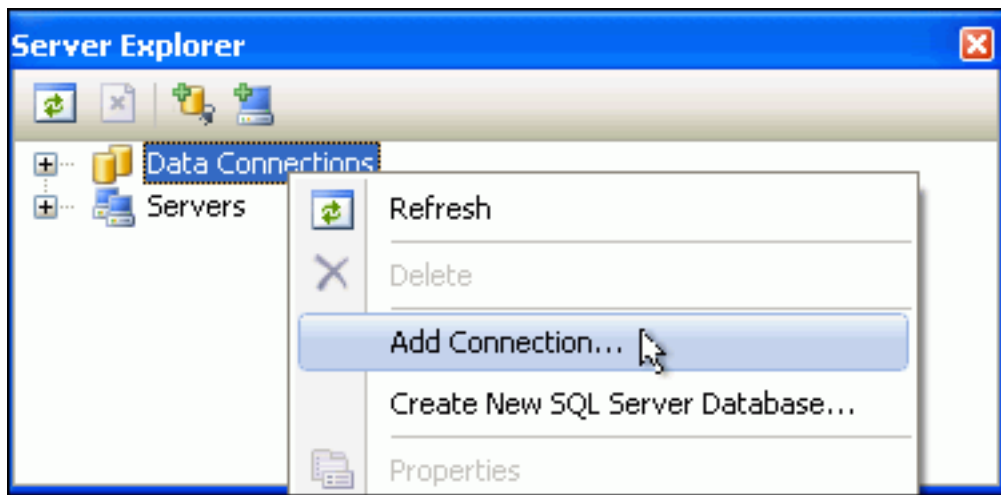
[「Oracle Developer Tools for Visual Studioの概要」](#)

Oracle Databaseへの接続

この項では、データベース・スキーマ・オブジェクトを自動的に作成または変更するために、Server Explorerを使用してOracle Databaseに接続する方法を説明します。

データベースに接続するには、次の手順を実行します。

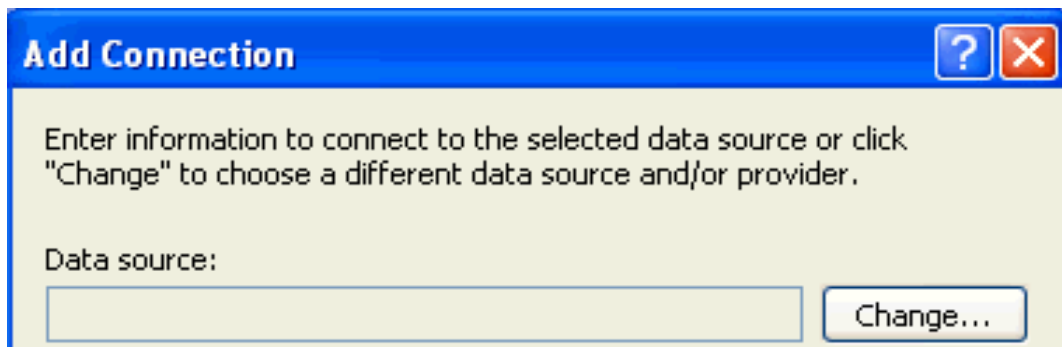
1. 「View」メニューから「Server Explorer」を選択します。
2. Server Explorerで、「Data Connections」を右クリックします。
3. 「Add Connection」を選択します。



[addconnection1.gifの説明](#)

4. 「Add Connection」ウィンドウが表示されたら、「Data source」に「Oracle Database (Oracle ODP.NET)」と表示されているかどうかを確認します。

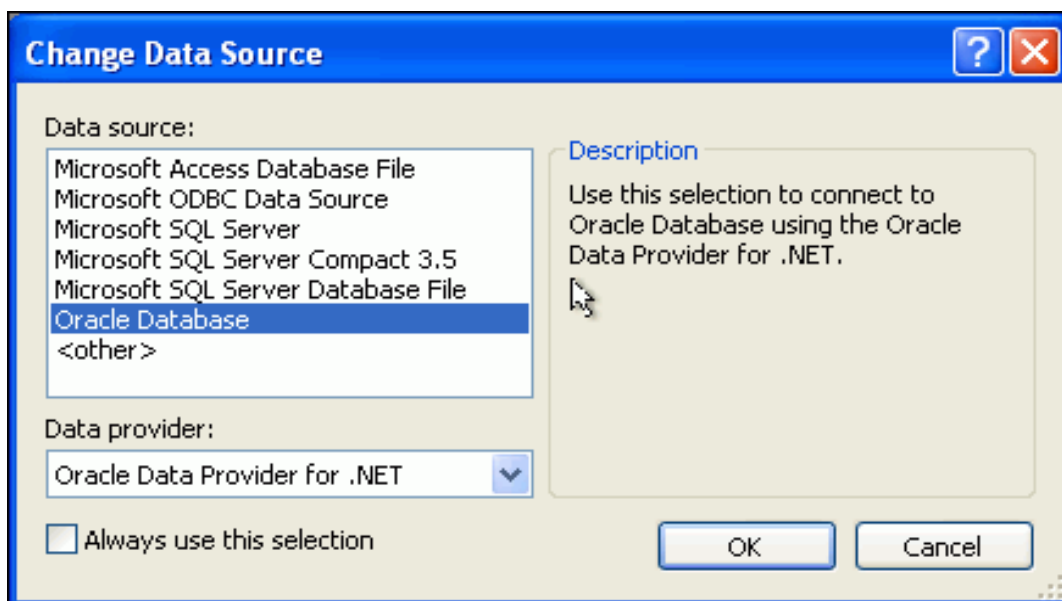
表示されている場合は、手順6に進みます。



[addconnection1a.gifの説明](#)

「Data source」に「Oracle Database (Oracle ODP.NET)」と表示されていない場合は、「Change」を選択します。

「Change Data Source」ウィンドウが表示されます。



[addconnection2a.gifの説明](#)

5. 「Oracle Database」を選択してから、「Oracle Data Provider for .NET」を選択します。
6. 「Add Connection」ウィンドウの「Connection Details」タブで、次の情報を入力します。

Data source name: リモート・データベース・インスタンスの別名orclなどを使用します。

同じコンピュータ上のデータベースに接続する場合は、Local Databaseを使用します。

「Use a specific user name and password」オプションを選択します。

「User name」にhrと入力します。

「Password」に、hrアカウントのロック解除と設定を行ったときに作成したパスワードを入力します。

後続のセッション用にパスワードを保存するには、「Save password」ボックスを選択します。

「Role」がDefaultに設定されていることを確認します。これにより、ユーザーhrに付与されているデフォルトのロールが参照されます。

「Connection name」は「Data source name」および「User name」の値から自動生成されます。この演習では、HR.orclになります。

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Oracle Database (Oracle ODP.NET) Change...

Connection Details Apply Filters

Data source name: orcl

☐ Use Windows integrated authentication

☒ Use a specific user name and password

User name: HR

Password: *****

☒ Save password

Role: Default

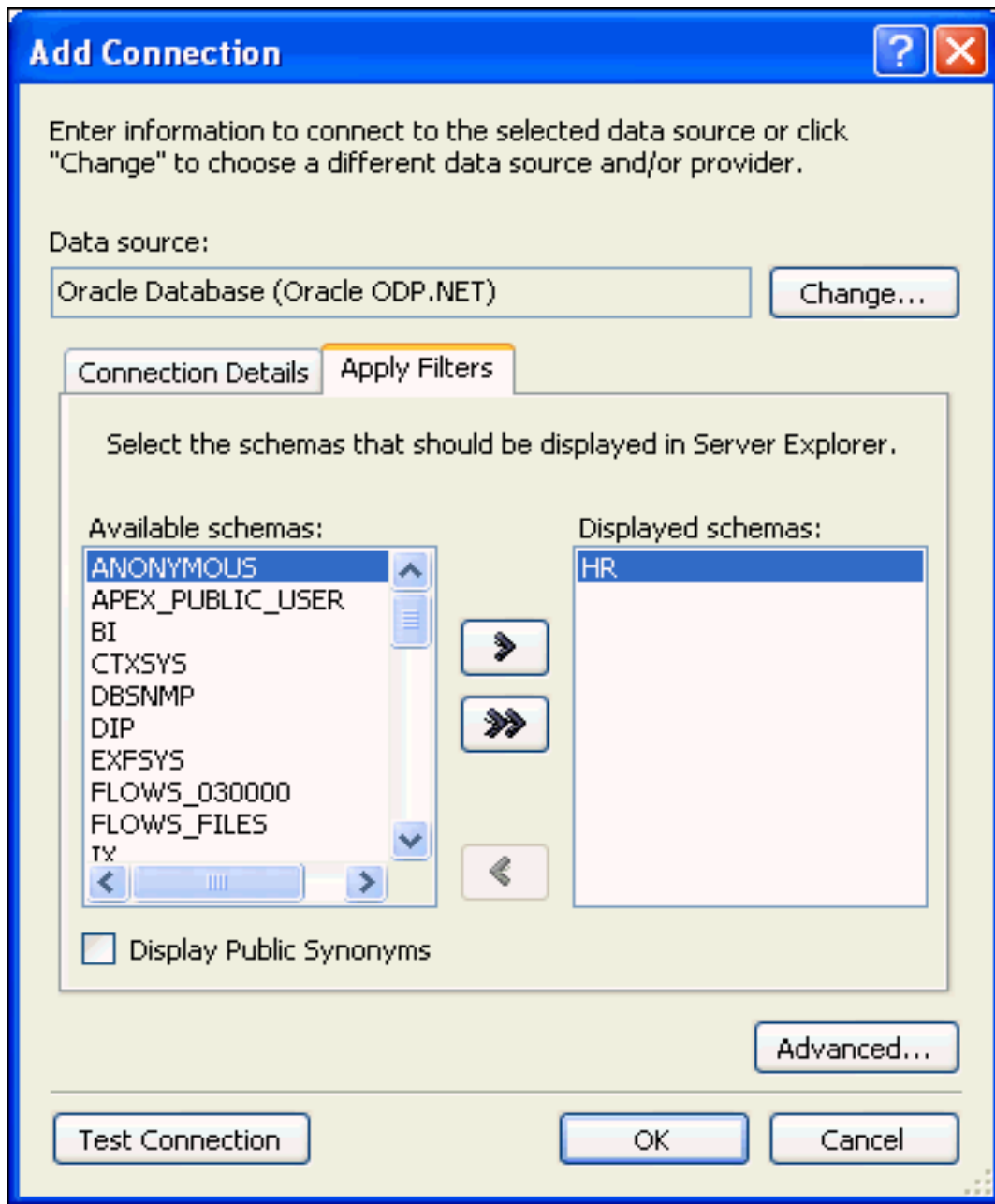
Connection name: HR.orcl

Advanced...

Test Connection OK Cancel

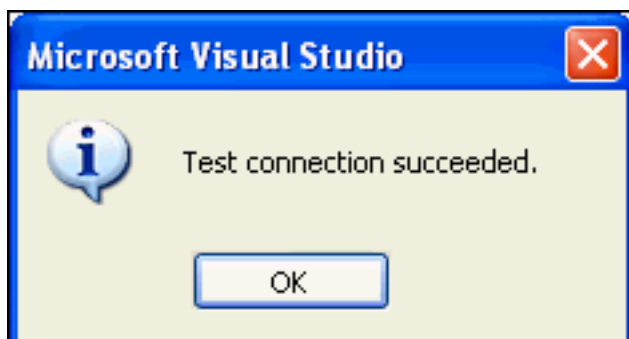
[addconnection2.gifの説明](#)

7. 「Apply Filters」タブをクリックし、HRスキーマが「Displayed schemas」列に含まれていることを確認します。データ接続のスキーマ・カテゴリ・ノードを開くと、「Apply Filters」タブで選択したスキーマ・オブジェクト（表、ビューなど）のみが表示されます。



[addconnection3.gifの説明](#)

8. 「Test connection」をクリックします。

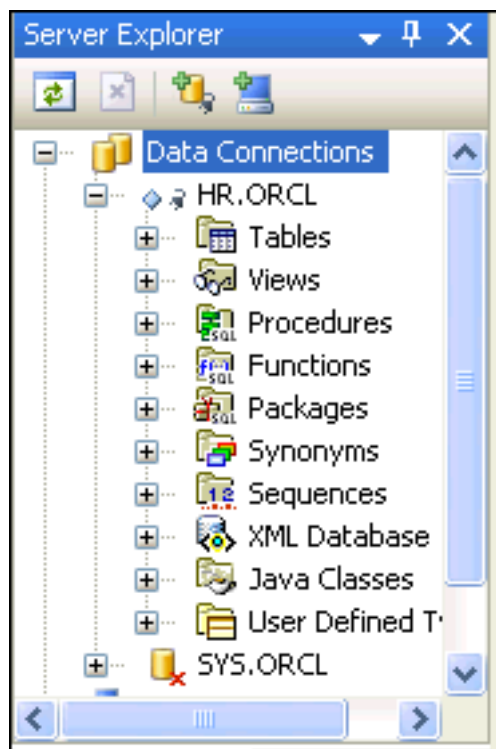


[addconnection4.gifの説明](#)

テストは成功するはずです。「OK」をクリックします。

テストが失敗した場合は、次のような問題が原因であると考えられます。後続の手順に進む前に、問題を解決する必要があります。

- データベースが起動していない。
 - データベース・リスナーが起動していない。
 - データベース接続が正しく構成されていない。
 - ユーザー名、パスワードまたはロールが正しく入力されていない。
9. 「Add Connection」ウィンドウで、「OK」をクリックします。
 10. Server ExplorerでHR.ORCL接続を開き、HRスキーマの内容を表示します。「Tables」、「Views」、「Procedures」、「Functions」、「Packages」、「Synonyms」、「Sequences」などが表示されます。



[addconnection5.gifの説明](#)

表および表の列の作成

Oracle Developer Toolsには、データベース・オブジェクトを作成するためのユーザー・インタフェースが含まれています。この項では、DEPENDENTSという名前の表を作成します。

表を作成するには、次の手順を実行します。

1. Server Explorerで、「Tables」を右クリックして「New Relational Table」を選択します。

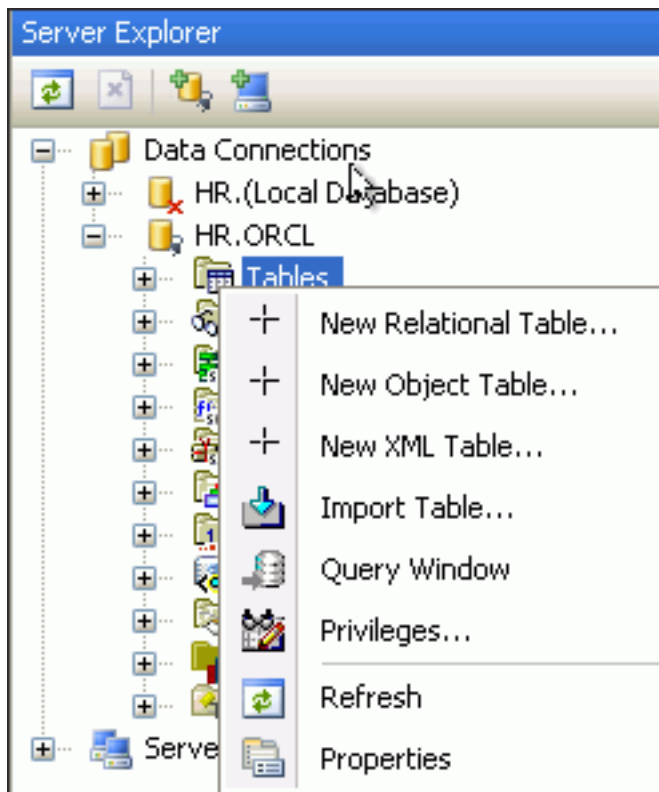


table1.gifの説明

表の設計ウィンドウが表示されます。

2. 設計ビューで、「Table name」にDEPENDENTSと入力します。

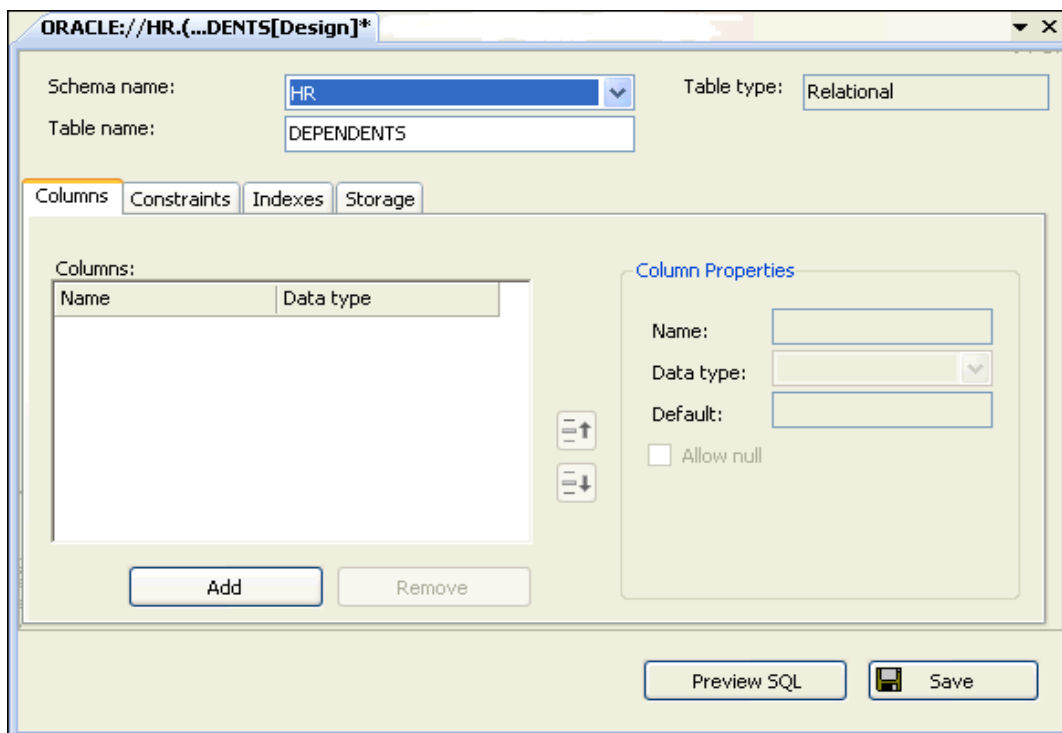


table2.gifの説明

3. 「Column Properties」タブで、次に示す6つの列を次の方法で追加します。

「Add」をクリックします。続いて、新しい列の情報を入力します。新しい列がすべて追加されるまで、繰り返し「Add」をクリックします。

フィールドはデータ型によって異なる場合があります。タブ全体にアクセスするには、Server ExplorerやSolution Explorerなどのウィンドウを閉じることが必要になる場合があります。

- 「Name」にLAST_NAME、「Data Type」にVARCHAR2、「Size」に30を指定します。他のプロパティは、すべてデフォルト値のままにします。
- 「Name」にFIRST_NAME、「Data Type」にVARCHAR2、「Size」に20を指定します。他のプロパティは、すべてデフォルト値のままにします。
- 「Name」にBIRTH_DATE、「Data Type」にDATEを指定します。他のプロパティは、すべてデフォルト値のままにします。
- 「Name」にRELATIONSHIP、「Data Type」にVARCHAR2、「Size」に20を指定します。他のプロパティは、すべてデフォルト値のままにします。
- 「Name」にEMPLOYEE_ID、「Data Type」にNUMBERを指定し、「Allow null」を選択解除します。「Precision」に6、「Scale」に0を入力します。
- 「Name」にDEPENDENT_ID、「Data Type」にNUMBERを指定し、「Allow null」チェック・ボックスを選択解除します。「Precision」に6、「Scale」に0を入力します。

ORACLE://HR.(...DENTS[Design]*

Schema name: HR Table type: Relational

Table name: DEPENDENTS

Columns Constraints Indexes Storage

Columns:

Name	Data type
LAST_NAME	VARCHAR2

Add Remove

Column Properties

Name: LAST_NAME

Data type: VARCHAR2

Default:

☒ Allow null

Size: 30

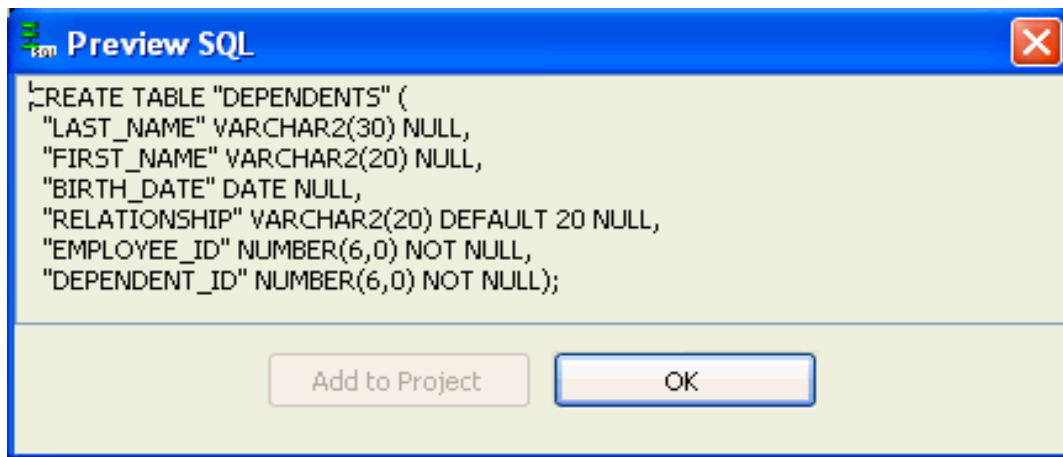
Length semantic: Default

Preview SQL Save

[table3.gifの説明](#)

4. 「Preview SQL」をクリックします。

表を作成するSQL文が「Preview SQL」ウィンドウに次のように表示されます。

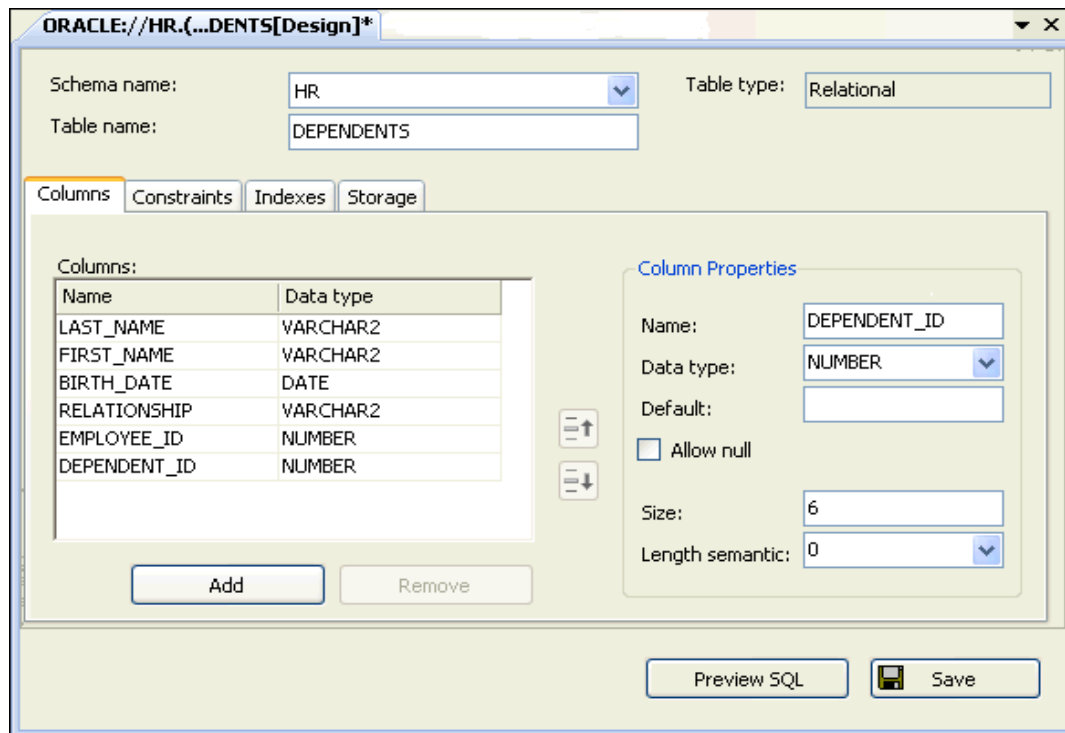


[table4.gifの説明](#)

「OK」をクリックして「Preview SQL」ウィンドウを閉じます。

5. 表の設計ビューで「Save」をクリックします。

この操作により、HRスキーマに新しい表DEPENDENTSが作成されます。新しい表がServer Explorerにリストされます。



[table5.gifの説明](#)

表の索引の作成

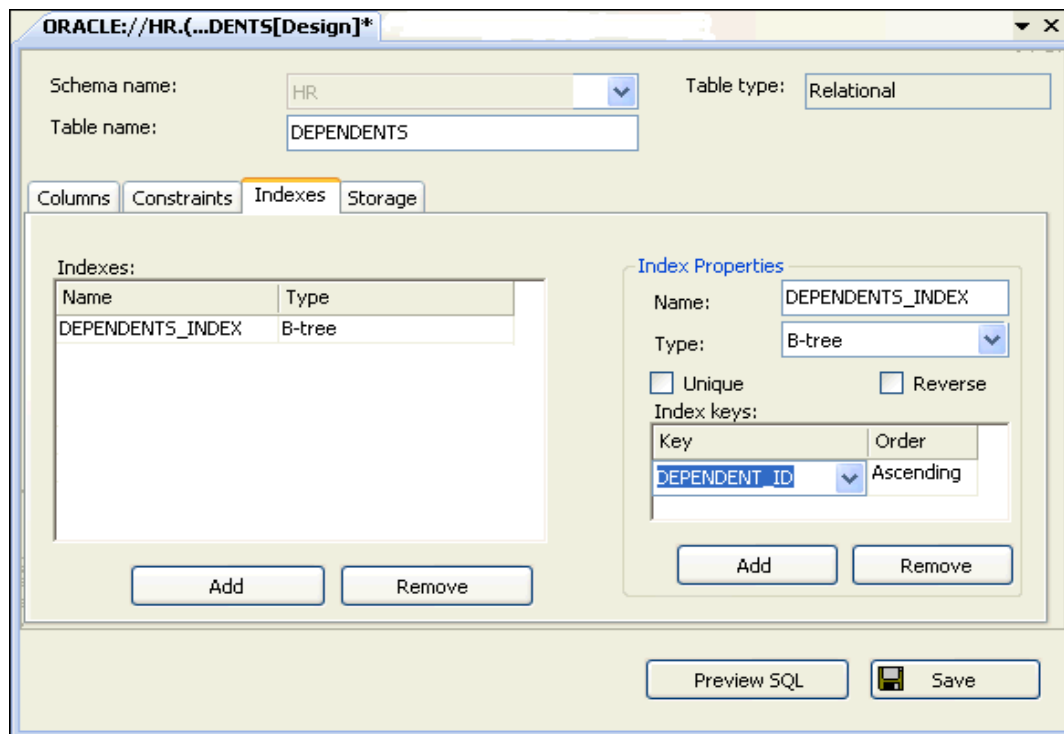
索引はオプションですが、非常に役立つリレーショナル・データベースの機能です。索引を使用すると表の行（またはレコード）にすばやくアクセスできます。この項では、DEPENDENTS表に索引を作成します。

索引を作成するには、次の手順を実行します。

1. DEPENDENTSの表の設計ビューで「Indexes」タブをクリックします。
2. 「Indexes」領域の下にある「Add」をクリックします。

「Index Properties」領域がアクティブになります。

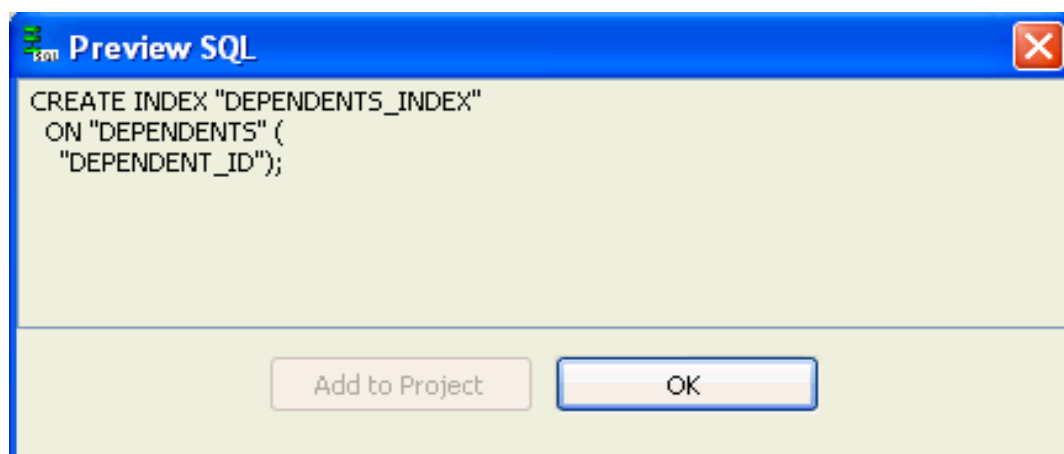
3. 「Index Properties」の下（右側）にある「Name」にDEPENDENTS_INDEXと入力し、他のプロパティはすべてデフォルトのままにします。
4. 「Index Properties」領域の下部で、「Add」をクリックします。
5. 「Index keys」の下で「Key」列の最初のセルをクリックし、リストから「DEPENDENT_ID」を選択します。



[index1.gifの説明](#)

6. 「Preview SQL」をクリックします。

「Preview SQL」ウィンドウが表示され、索引を作成するSQL文が表示されます。

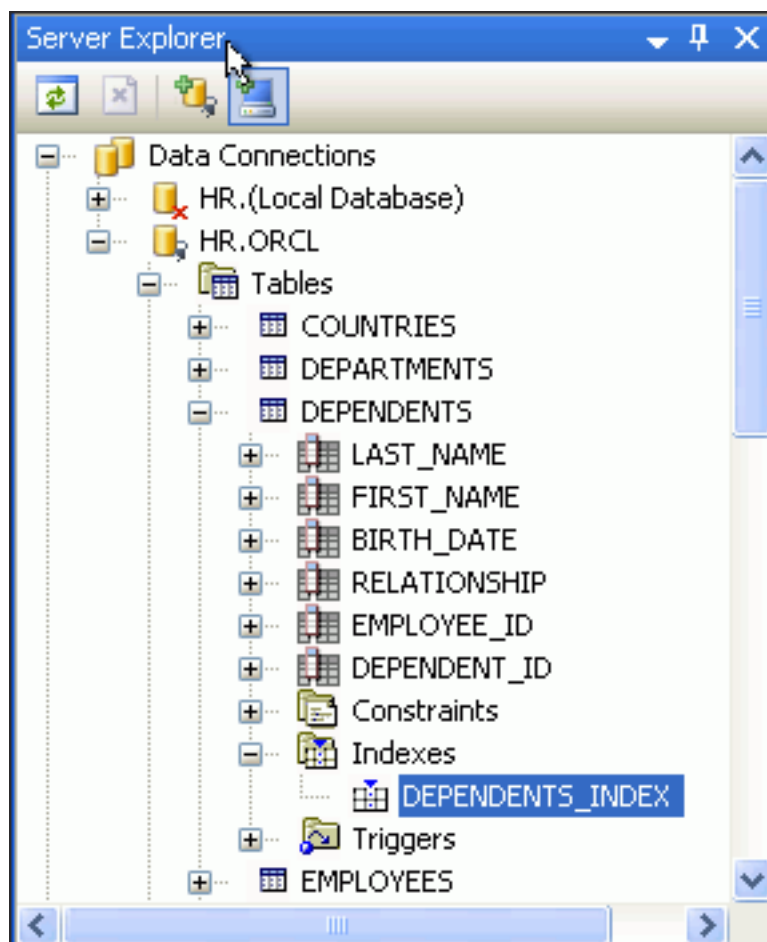


[index2.gifの説明](#)

「OK」をクリックして「Preview SQL」ウィンドウを閉じます。

7. 表の設計ビューで「Save」をクリックします。

これで、HRスキーマの表DEPENDENTSに新しい索引が作成されます。作成された索引をServer Explorerで確認するには、DEPENDENTS表および関連する索引を開きます。



[index3.gifの説明](#)

表の制約の追加

データベースで制約を使用すると、許容できるデータ値に対し、データ整合性の定義ルールが自動的に施行されます。また、この制約により、表の主キーや外部キーも実装されます。この項では、新しい表DEPENDENTSにこのような制約を追加します。

外部キーおよび主キーを追加するには、次の手順を実行します。

1. DEPENDENTS表の設計ビューで、「Constraints」タブをクリックします。

構成によっては、すでにデフォルトのチェック制約がリストに含まれていることがあります。

2. 「Constraints」領域の下で、次に示す制約を次の方法で追加します。

「Constraint Properties」の下で「Add」をクリックします。続いて、新しい制約の情報を入力します。新しい制約がすべて追加されるまで、繰り返し「Add」をクリックします。

- 「Name」にEMPLOYEES_FK、「Type」にForeignKey、「Table」にEMPLOYEES、「Constraint」にEMP_EMP_ID_PKを指定します。「Association」の下で、「Referenced Column:」にEMPLOYEE_ID、「Local Column:」にEMPLOYEE_IDを選択し、「On delete」値にCascadeを設定します。他のプロパティは、すべてデフォルト値のままにします。

Schema name: Table type:

Table name:

Columns Constraints Indexes Storage

Constraints:

Name	Type
SYS_C009997	Check
SYS_C009996	Check
EMPLOYEES_FK	Foreign Key

Add Remove

Constraint Properties

Name:

Type:

☐ Deferrable ☒ Enabled ☒ Validate

Execution:

Select the foreign key constraint

Table:

Constraint:

Association:

Referenced Column	Local Column
EMPLOYEE_ID	EMPLOYEE_ID

On delete:

Preview SQL Save

[constraint1.gifの説明](#)

- 「Name」 にDEPENDENTS_PK、 「Type」 にPrimary Keyを指定します。

「Primary key columns」 領域の下で、「Add」をクリックします（下にスクロールすることが必要な場合があります）。
「Primary Key Columns」 の下で、「Key:」 にDEPENDENT_IDを選択し、「Using index」 値にDEPENDENTS_INDEXを設定しま
す。他のプロパティは、すべてデフォルト値のままにします。

Schema name: Table type:

Table name:

Columns Constraints Indexes Storage

Constraints:

Name	Type
SYS_C009997	Check
SYS_C009996	Check
EMPLOYEES_FK	Foreign Key
DEPENDENTS_PK	Primary Key

Add Remove

Constraint Properties

Name:

Type:

☐ Deferrable ☒ Enabled ☒ Validate

Execution:

Primary key columns:

Key:

Add Remove

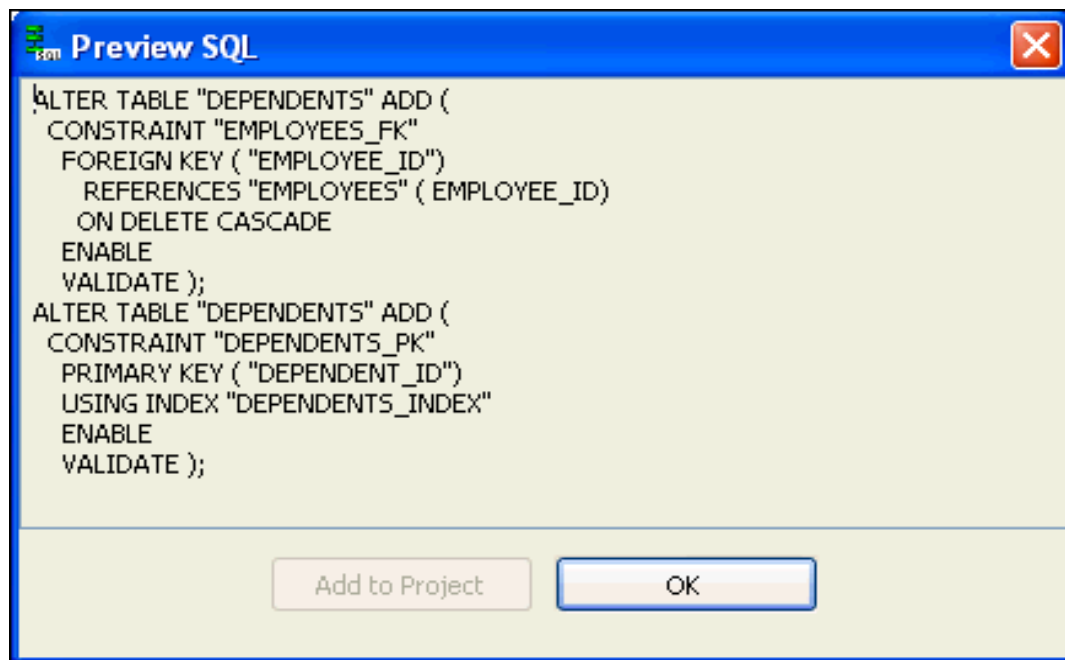
Using index:

Preview SQL Save

[constraint2.gifの説明](#)

3. 「Preview SQL」をクリックします。

「Preview SQL」ウィンドウに、表DEPENDENTSの制約用に生成されたコードが表示されます。制約により、表のDEPENDENT_ID列とEMPLOYEE_ID列の定義が変更されるため、制約の追加にはALTER TABLEコマンドが使用されます。

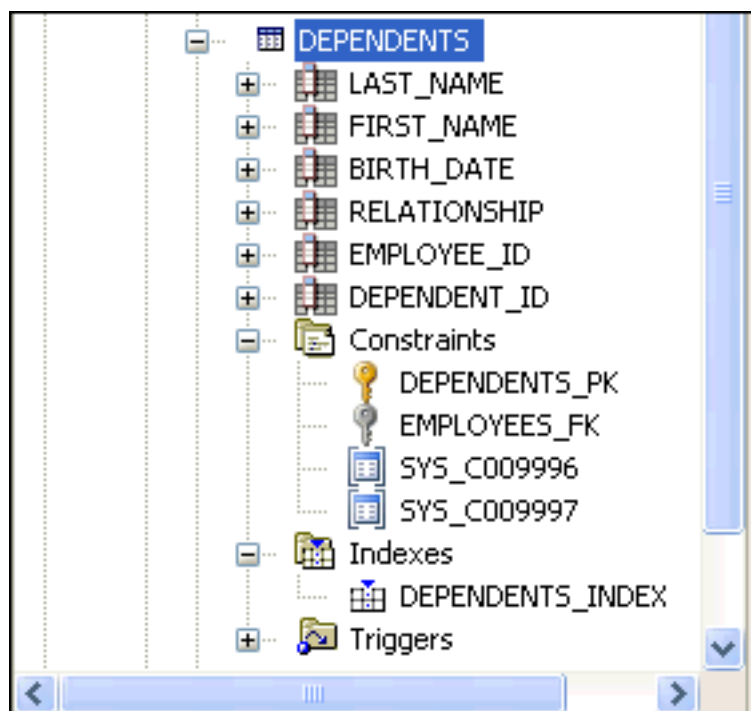


[constraint3.gifの説明](#)

「OK」をクリックして「Preview SQL」ウィンドウを閉じます。

4. 表の設計ビューで「Save」をクリックします。

この操作により、新しい2つの制約がHRスキーマのDEPENDENTS表に作成されます。Server Explorerを確認するには、表DEPENDENTSおよび制約の階層ツリーを開きます。



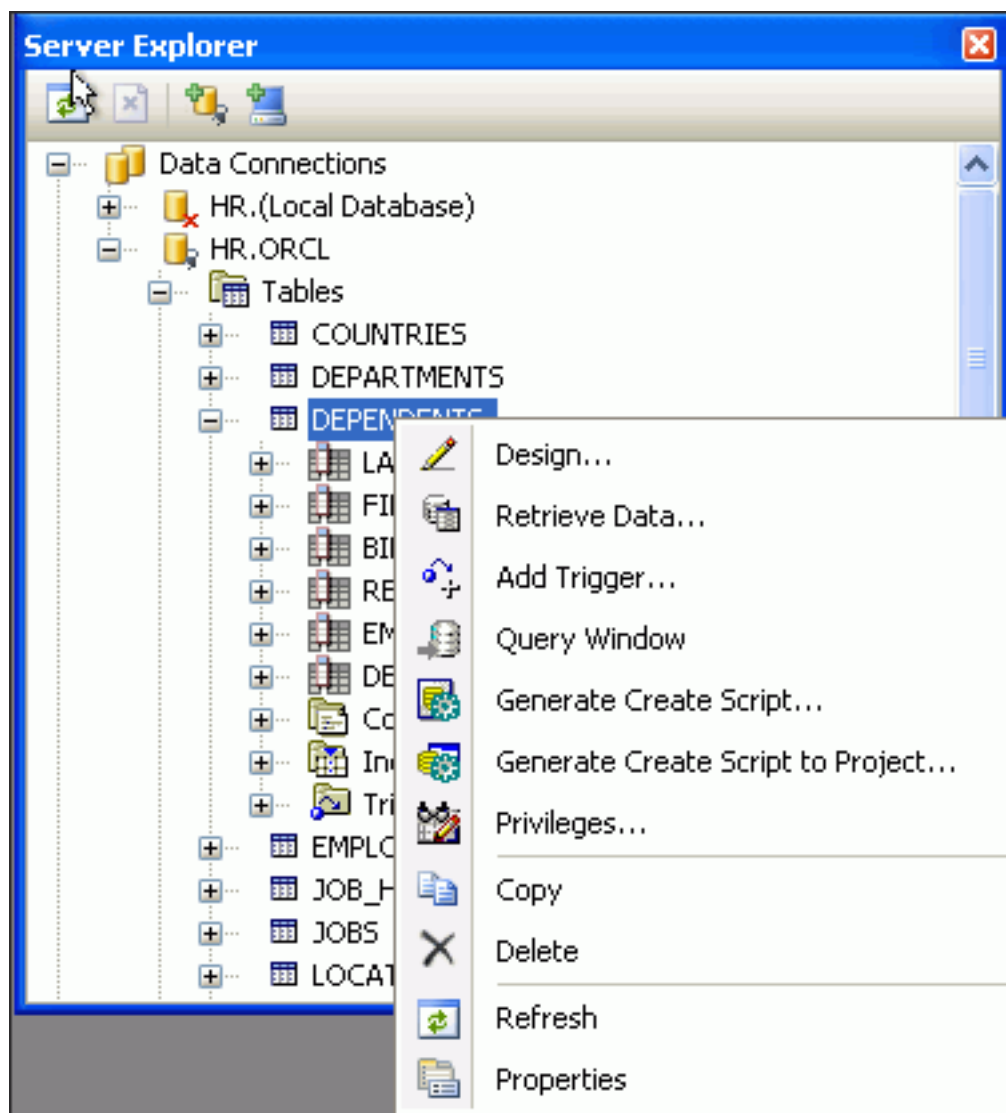
[constraint4.gifの説明](#)

表へのデータの追加

ここで、新しいDEPENDENTS表にデータを追加する必要があります。

表にデータを移入するには、次の手順を実行します。

1. Server Explorerで、DEPENDENTS表を右クリックして「Retrieve Data」を選択します。



[retrievel.gifの説明](#)

DEPENDENTSの表グリッドが設計ビューに表示されます。

2. [表5-1](#)に示す4つのレコードを、この表グリッドに追加します。

表5-1 DEPENDENTS表の新しいデータ

LAST_NAME	FIRST_NAME	BIRTH_DATE	RELATIONSHIP	EMPLOYEE_ID	DEPENDENT_ID
Ernst	Mary	06-MAY-2000	daughter	104	1041
Atkinson	Sue	12-JUL-1998	daughter	130	1301

Ernst	David	02-APR-2007	son	104	1042
Sciarra	Aaron	31-JAN-2008	son	111	1111

グリッドは次のようになります。

LAST_NAME	FIRST_NAME	BIRTH_DATE	RELATIONSHI	EMPLOYEE_I	DEPENDENT_
Ernst	Mary	06-MAY-00	daughter	104	1041
Atkinson	Sue	12-JUL-98	daughter	130	1301
Ernst	David	02-APR-07	son	104	1042
Sciarra	Aaron	31-JAN-08	son	111	1111
*					

[retrieve2の説明](#)

行を移動すると、データは自動的に保存されます。

データを表示および更新するためのコードの自動生成

DEPENDENTS表の内容を確認するために、表の単純な問合せを使用するフォームを作成します。この項では、Visual Studio統合開発環境（IDE）を使用して、操作に対応するコードを自動生成します。

新しいデータソースを作成するには、次の手順を実行します。

1. [「新しいプロジェクトの作成」](#)の説明に従って、新しいプロジェクトを開始します。新しいプロジェクトには、次に示す名前を付けます。

Visual C#:

HR_ODT_CS

Visual Basic:

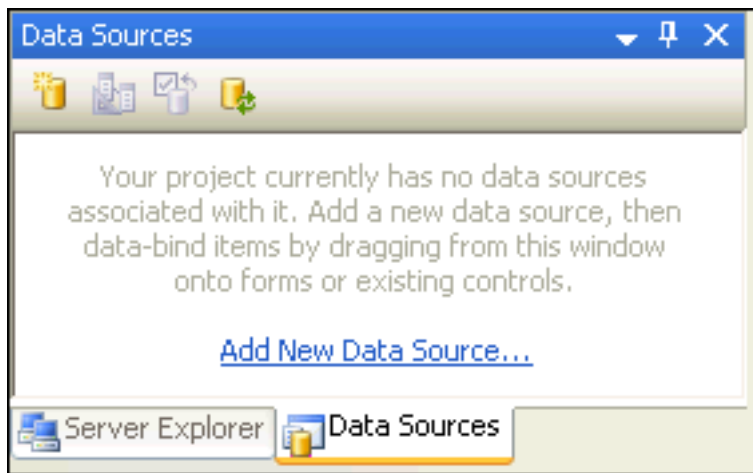
HR_ODT_VB

2. 「Create Directory for Solution」を選択します。「OK」をクリックします。
3. 設計ビューが表示されていない場合は、Form1の設計ビューに切り替えます。

注意: すべてのアプリケーションはForm1から開始しますが、これまでの章で作成したアプリケーションとの関係はありません。

4. 「Server Explorer」ウィンドウをクリックし、「Show Data Sources」ウィンドウを有効にします。
5. Visual Studioの「Data」メニューから「Show Data Sources」を選択します。

「Data Source」ウィンドウが表示されます。



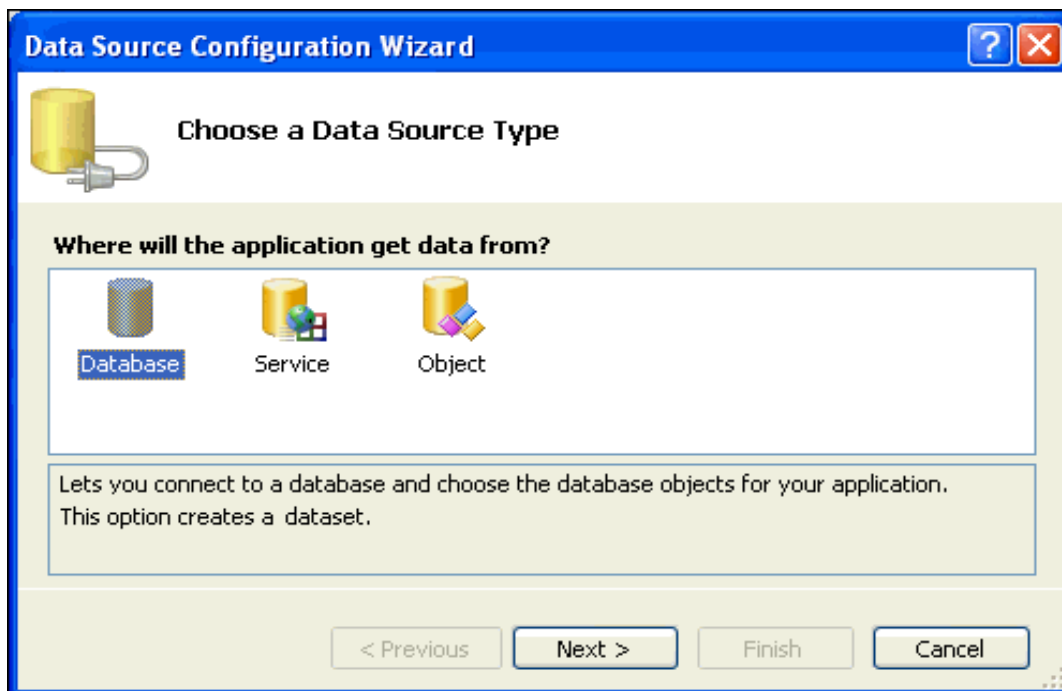
[datasource1.gifの説明](#)

6. 「Data Sources」 ウィンドウで、「Add New Data Source」をクリックします。

「Data Source Configuration Wizard」が開きます。

7. 「Data Source Configuration Wizard」の「Choose a Data Source Type」で「Database」を選択します。

「Next」をクリックします。

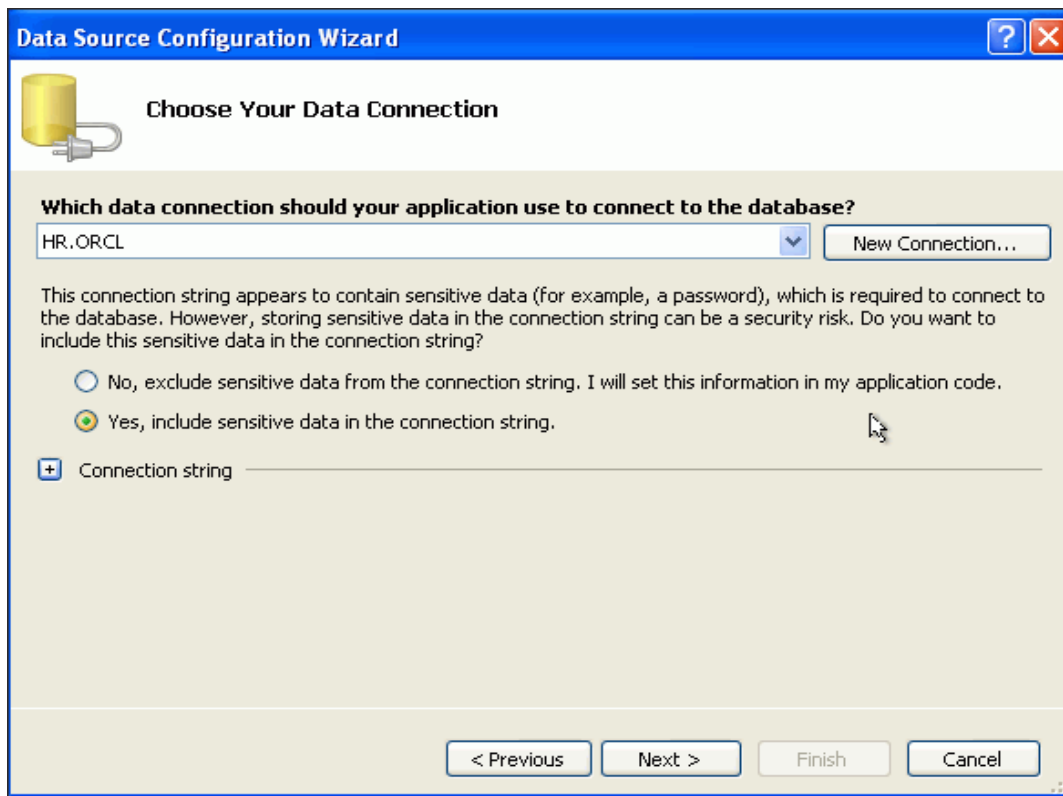


[datasource2.gifの説明](#)

8. 「Choose Your Data Connection」で、「HR.ORCL」または「HR.(Local Database)」を選択します。この例では、HR. ORCLを使用します。

「Yes, include sensitive data in the connection string」を選択します。

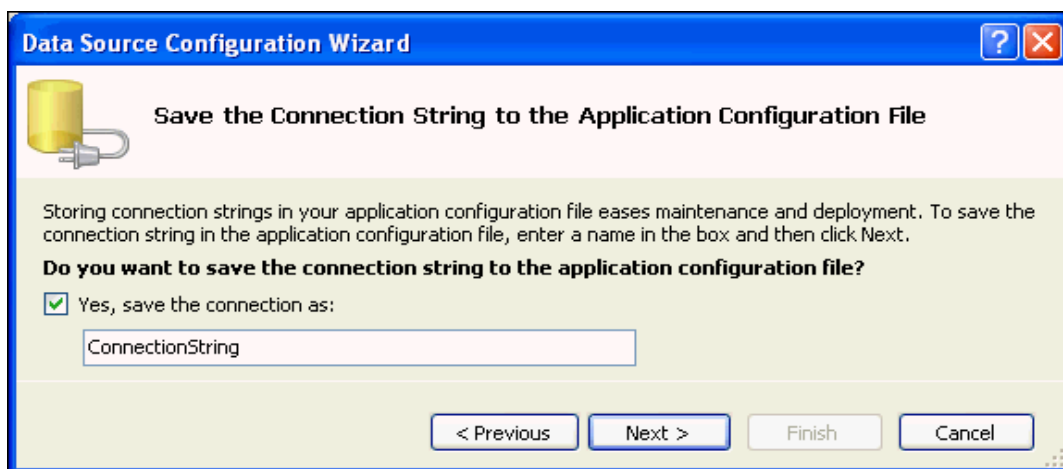
「Next」をクリックします。



[datasource3.gifの説明](#)

9. 「Save the Connection String to the Application Configuration File」で、「Yes, save the connection as:」にConnectionStringを選択します。

「Next」をクリックします。



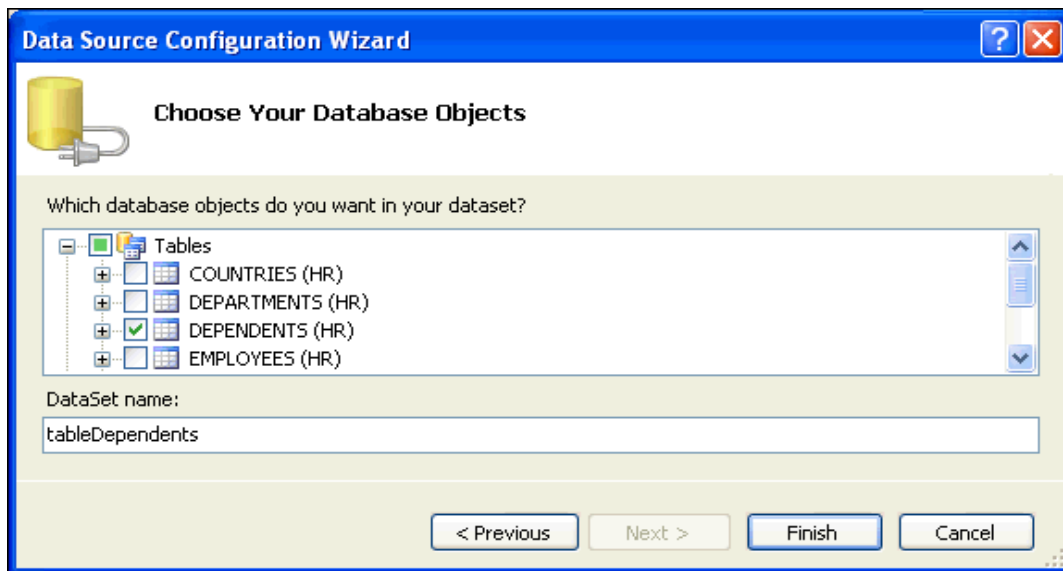
[datasource4.gifの説明](#)

10. 「Choose Your Database Objects」で、「Tables」を開きます。

「DEPENDENTS(HR)」表を選択します。

「DataSet name」をtableDependentsに変更します。

「Finish」をクリックします。



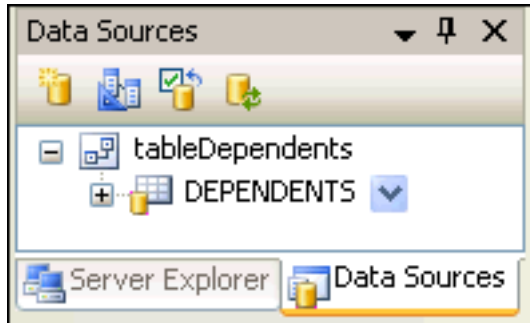
[datasource5.gifの説明](#)

参照:

DataSet クラスの詳細は、[「Oracle Data Provider for .NETでのDataSetクラスの使用」](#)を参照してください。

ドラッグ・アンド・ドロップでコードを自動生成するには、次の手順を実行します。

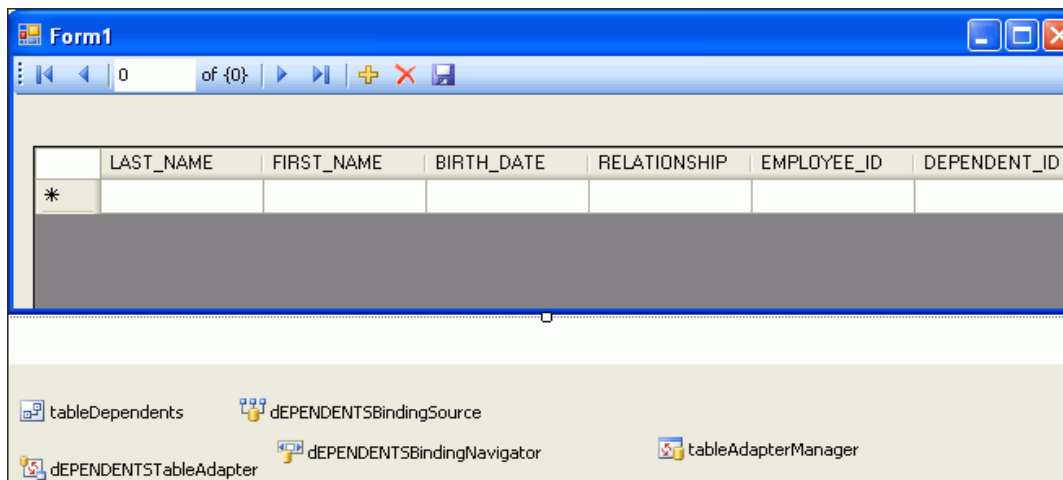
1. Form1の設計ビューに切り替えます。
2. 「Data Sources」ウィンドウで、「tableDependents」を開きます。



[datasource6.gifの説明](#)

3. **DEPENDENTS**表を選択し、Form1にドラッグします。

フォームと表グリッドの両方のサイズ調整が必要になる場合があります。



表グリッド（レコード・ナビゲーション要素を含む）の他に、次のコンポーネントがプロジェクトの設計ビューに追加されます。これらのオブジェクトは、Form1に自動生成されるコードを表しています。

Visual C#:

tableDependents、dDEPENDENTSBindingSource、dDEPENDENTSTableAdapter、tableAdapterManagerおよびdDEPENDENTSBindingNavigator

Visual Basic:

TableDependents、DEPENDENTSBindingSource、DEPENDENTSTableAdapter、TableAdapterManagerおよびDEPENDENTSBindingNavigator

- Form1の上部付近にある「Save」アイコン（フロッピー・ディスク）をダブルクリックします。

これにより、Form1の「Save」アイコンのコード・ウィンドウが開きます。

- プライベート・メソッドxxxSaveItem_Click()で、try...catchブロックに既存のコードをカプセル化します。この自動生成されたメソッドのVisual C#およびVisual Basicの完全な名前は、次に示すコードを参照してください。

また、MessageBox.show()コールをTryセクションおよびCatchセクションの両方に追加します。更新したメソッド・コードを次に示します。新しいコードまたは変更されたコードは太字で表しています。

Visual C#:

```
private void dDEPENDENTSBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    try
    {
        this.Validate();
        this.dDEPENDENTSBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.tableDependents);

        MessageBox.Show("Update successful");
    }
    catch (System.Exception ex)
    {
        MessageBox.Show("Update failed: " + ex.Message.ToString());
    }
}
```

Visual Basic:

```
Private Sub DEPENDENTSBindingNavigatorSaveItem_Click(
```



```
ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles DEPENDENTSBindingNavigatorSaveItem.Click
```

```
Try  
Me.Validate()  
Me.DEPENDENTSBindingSource.EndEdit()  
Me.TableAdapterManager.UpdateAll(Me.TableDependents)  
MessageBox.Show("Update successful")
```

```
Catch ex As Exception  
MessageBox.Show("Update failed: " + ex.Message.ToString())
```

```
End Try
```

```
End Sub
```

6. アプリケーションをコンパイルして実行するには、[「アプリケーションのコンパイルと実行」](#)の手順に従います。

次の方法で、新しいアプリケーションをテストできます。フロッピー・ディスク・アイコンは、「Save」コマンドを表しています。

アプリケーションをテストするには、次の手順を実行します。

1. Mary Ernstの「DEPENDENT_ID」の値を1110に変更し、「Save」アイコンをクリックします。メッセージ・ボックスUpdate successfulが表示されます。「OK」をクリックしてメッセージ・ボックスを閉じます。
2. David Ernstの「EMPLOYEE_ID」の値を99999に変更し、「Save」アイコンをクリックします。「更新に失敗しました。:ORA-02291:整合性制約(HR.EMPLOYEES_FK)に違反しました - 親キーがありません」というメッセージが表示されます。「OK」をクリックしてメッセージ・ボックスを閉じます。

6 PL/SQLストアド・プロシージャおよびREF CURSORの使用

この章の内容は次のとおりです。

- [PL/SQLストアド・プロシージャの概要](#)
- [PL/SQLパッケージとパッケージ本体の概要](#)
- [REF CURSORの概要](#)
- [REF CURSORを使用するPL/SQLストアド・プロシージャの作成](#)
- [ストアド・プロシージャを実行するためのODP.NETアプリケーションの変更](#)
- [ODP.NETアプリケーションによるPL/SQLストアド・プロシージャの実行](#)

PL/SQLストアド・プロシージャの概要

ストアド・プロシージャとは、ある操作を実行するために設計された一連のPL/SQL文を1つにまとめて名前を付けたものです。ストアド・プロシージャはデータベース内に格納されます。これは、クライアント・アプリケーションがデータベース・オブジェクトと直接対話できるようにするものではなく、データベースのプログラミング・インタフェースを定義するものです。一般的にストアド・プロシージャを使用するのは、データを検証する場合や、複数のSQL問合せを組み合わせた大規模で複雑な処理の指示をカプセル化する場合です。

ストアド・ファンクションは、戻り値パラメータを1つとります。ファンクションとは異なり、プロシージャは値を戻す場合と戻さない場合があります。

PL/SQLパッケージとパッケージ本体の概要

PL/SQLパッケージには、関連する項目が単一の論理エンティティとして格納されます。パッケージは、次の2つの部分で構成されます。

- **パッケージ仕様部**では、パッケージに含まれるものを定義します。これは、C++などの言語のヘッダー・ファイルに似ています。仕様部では、すべてのパブリック項目を定義します。仕様部はパッケージの公開インタフェースです。
- **パッケージ本体**には、仕様部で定義したプロシージャおよびファンクションのコードと、仕様部で宣言していないプライベート・プロシージャおよびファンクションのコードが含まれます。プライベート・コードはパッケージ本体内でのみ確認できます。

パッケージ仕様部とパッケージ本体は、データ・ディクショナリに別々のオブジェクトとして格納され、`user_source`ビューで確認できます。仕様部はPACKAGE型として格納され、本体はPACKAGE BODY型として格納されます。

一連のパブリック定数を宣言する場合と同様、本体のない仕様部を保持することはできますが、仕様部のない本体を保持することはできません。

REF CURSORの概要

REF CURSORを使用することは、Oracle Databaseからの問合せ結果をクライアント・アプリケーションに戻す最も強力かつ柔軟で、拡張性のある方法の1つです。

REF CURSORはPL/SQLデータ型であり、この値はデータベース上の問合せ作業領域のメモリー・アドレスです。つまり、REF CURSORは、データベース上にある結果セットへのポインタまたはハンドルとなります。REF CURSORは、OracleRefCursor ODP.NETクラスを使用して表現します。

REF CURSORには次の特性があります。

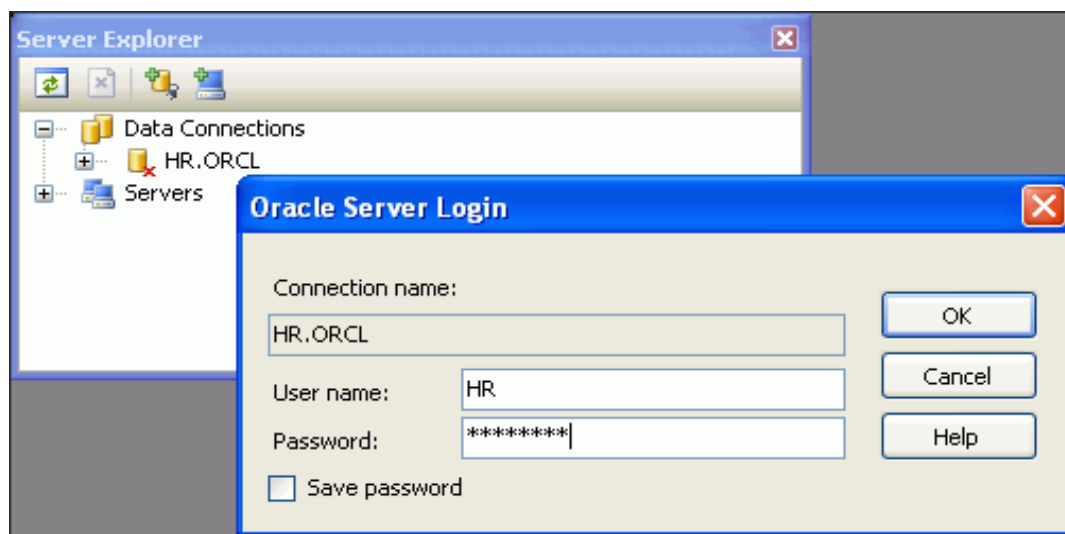
- REF CURSORは、データベース上のメモリー・アドレスを参照します。そのため、REF CURSORにアクセスするには、REF CURSORの存続期間中、クライアントがデータベースに接続されている必要があります。
- REF CURSORにより、追加のデータベース・ラウンドトリップが発生します。REF CURSORがクライアントに戻されても、クライアントがREF CURSORをオープンしてデータをリクエストするまで、実際のデータは戻されません。ユーザーがREF CURSORの読取りを試行するまで、データは取得されないことに注意してください。
- REF CURSORは更新できません。REF CURSORで表される結果セットは読取り専用です。REF CURSORを使用してデータベースを更新することはできません。
- REF CURSORは後方にスクロールできません。REF CURSORで表される結果セットは、前進専用で順次アクセスされます。結果セット内のレコードをランダムにポイントするためにREF CURSOR内にレコード・ポインタを配置することはできません。
- REF CURSORはPL/SQLデータ型です。PL/SQLコード・ブロック内でREF CURSORを作成して戻すことができます。

REF CURSORを使用するPL/SQLストアド・プロシージャの作成

この項では、PL/SQLストアド・プロシージャの作成方法を説明します。

ストアド・プロシージャを作成するには、次の手順を実行します。

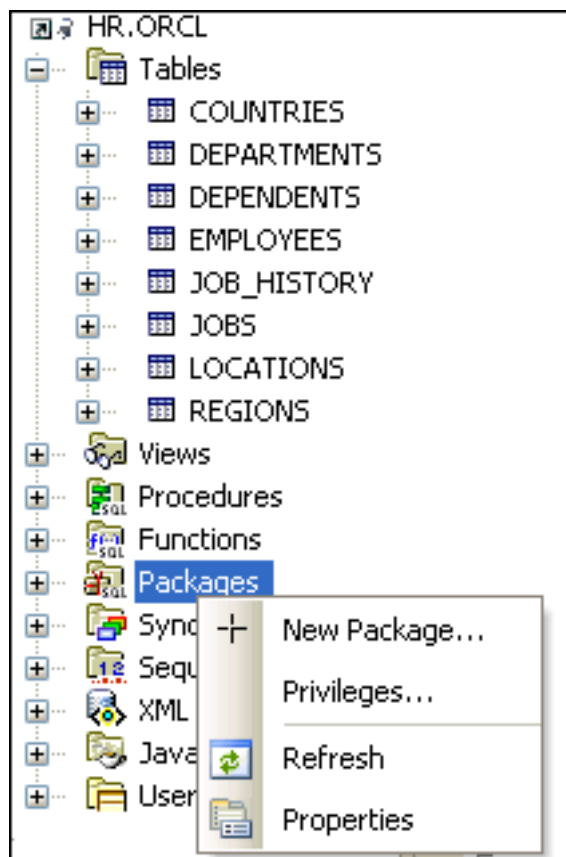
1. Server Explorerを開いてHRをダブルクリックし、[「Oracle Databaseへの接続」](#)で作成したHRスキーマへの接続をオープンします。



[connect_se.gifの説明](#)

以前にパスワードを保存しなかった場合は、「Oracle Server Login」が開き、パスワードを入力できます。パスワードを以前に保存している場合は、すぐに接続がオープンします。

2. Server Explorerで、「Packages」を右クリックして「New Package」を選択します。



[procedure1.gifの説明](#)

「New Package」ウィンドウが表示されます。

3. 「New Package」ウィンドウで、「**Package Name**」をHR_DATAに変更します。
4. 「Methods」領域の下「**Add**」をクリックします。

+ New Package

Schema name:

Package name:

Authentication identifier:

☒ Generate empty package body

Use the Add button to add functions and procedures to this package.

Methods:

Name	Method Type
------	-------------

[procedure2.gifの説明](#)

「Add Method」ウィンドウが表示されます。

5. 「Add Method」ウィンドウで、「**Method Name**」にGETCURSORSと入力し、「**Method Type**」をProcedureに変更します。
6. 「Parameters」の下「**Add**」をクリックします。

これにより、パラメータを追加するプロセスが開始されます。

右側の「Parameter Details」グループで、次の3つのパラメータを入力します。「Add」をクリックしてから、必要なパラメータを1つずつ追加します。

- 「**Name:**」にDEP_IDと入力し、「**Direction:**」は「IN」を選択します。「**Data Type:**」には「NUMBER」を選択します。
- 「**Name:**」にEMPLOYEES_Cと入力し、「**Direction:**」は「OUT」を選択します。「**Data Type:**」には「SYS_REFCURSOR」を選択します。
- 「**Name:**」にDEPENDENTS_Cと入力し、「**Direction:**」は「OUT」を選択します。「**Data Type:**」には「SYS_REFCURSOR」を選択します。

+ Add Method

Method name:

Method type:

Return type:

☐ Use pipelining to return rows

Parameters:

Name	Data Type
DEP_ID	NUMBER
EMPLOYEES_C	SYS_REFCURSOR
DEPENDENTS_C	SYS_REFCURSOR

Parameter Details

Name:

Direction:

Data type:

Default:

☐ No copy

☐ REF

[procedure3.gifの説明](#)

7. パラメータの追加が終了したら、「OK」をクリックします。

「New Package」ウィンドウが再度表示されます。

8. 「New Package」ウィンドウで「Preview SQL」をクリックし、作成されたSQLコードを確認します。

次のようなコードを含む「Preview SQL」ウィンドウが表示されます。このコードは、コメントの大部分を削除して短縮したものです。

```
CREATE PACKAGE "HR"."HR_DATA" IS -- Declare types, variables, constants, exceptions, cursors,
-- and subprograms that can be referenced from outside the package.
```

```
PROCEDURE "GETCURSORS" (
  "DEP_ID" IN NUMBER,
  "EMPLOYEES_C" OUT SYS_REFCURSOR,
  "DEPENDENTS_C" OUT SYS_REFCURSOR);
```

```
END "HR_DATA";
```

```
CREATE PACKAGE BODY "HR"."HR_DATA" IS
```

```
-- Implement subprograms, initialize variables declared in package
-- specification.
```

```
-- Make private declarations of types and items, that are not accessible
-- outside the package
```

```
PROCEDURE "GETCURSORS" (
  "DEP_ID" IN NUMBER,
  "EMPLOYEES_C" OUT SYS_REFCURSOR,
  "DEPENDENTS_C" OUT SYS_REFCURSOR) IS
```

```
-- Declare constants and variables in this section.
```

```
BEGIN -- executable part starts here
```

```
NULL;
```

```
-- EXCEPTION -- exception-handling part starts here
```

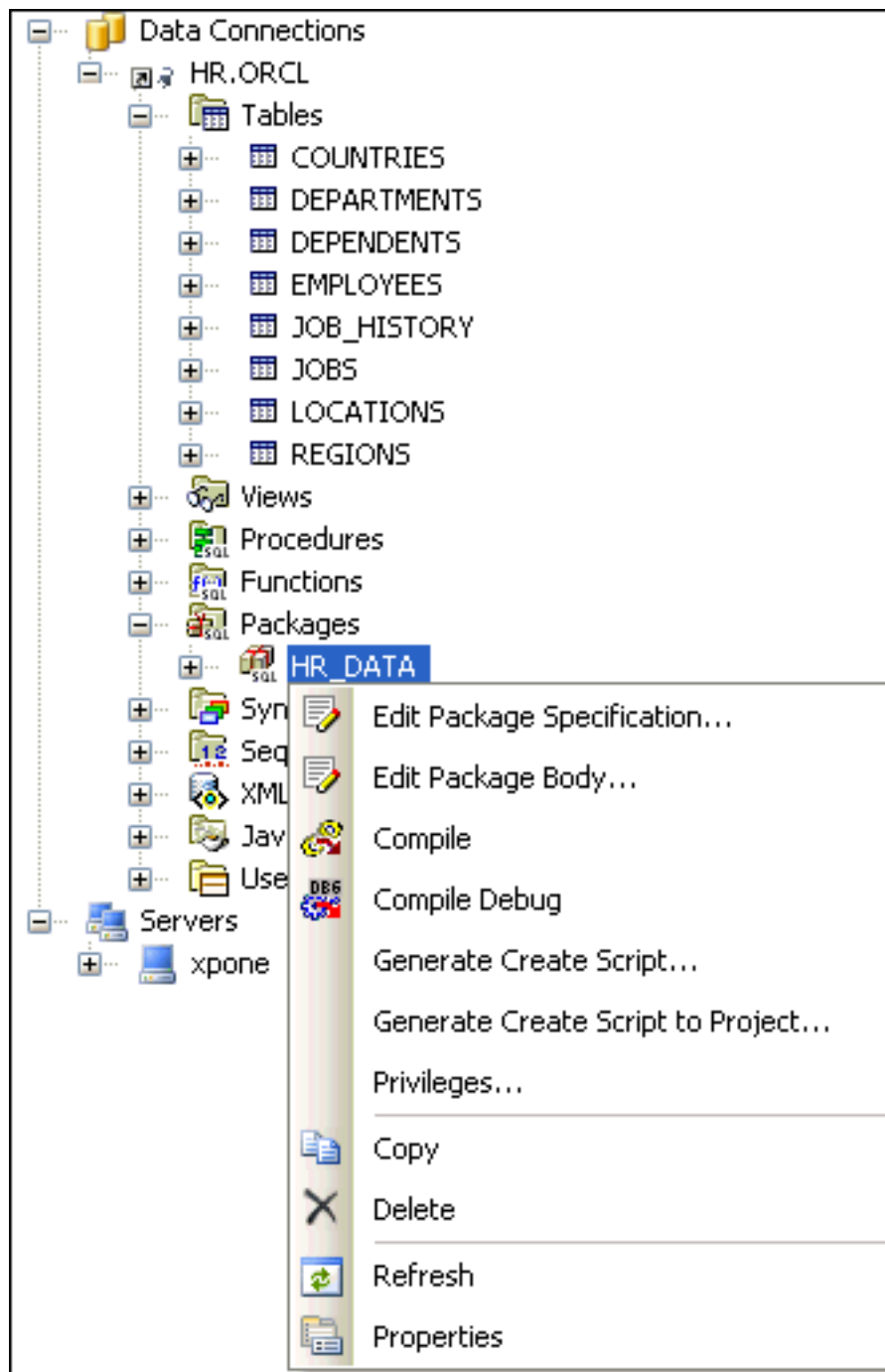
```
END "GETCURSORS";
```

```
END "HR_DATA";
```

9. 「OK」をクリックして「Preview SQL」ウィンドウを閉じます。
10. 「New Package」ウィンドウで「OK」をクリックし、新しいパッケージを保存します。

新しいパッケージHR_DATAがServer Explorerに表示されます。

11. Server Explorerで、パッケージHR_DATAを右クリックして「Edit Package Body」を選択します。



[procedure4.gifの説明](#)

パッケージのコードが表示されます。

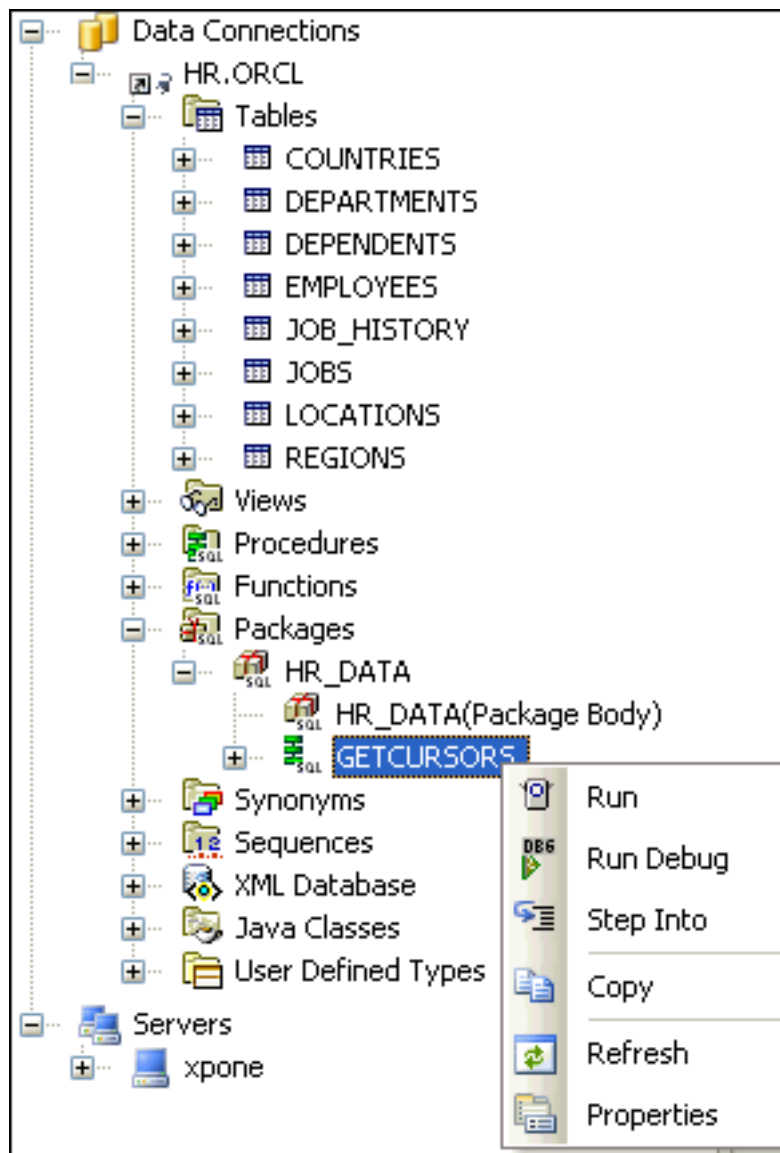
12. GETCURSORSプロシージャの本体までスクロールし、BEGINの後にある行NULL;を次のコードに置き換えます。

```
OPEN EMPLOYEES_C FOR SELECT * FROM EMPLOYEES
WHERE DEP_ID=DEPARTMENT_ID;
```

```
OPEN DEPENDENTS_C FOR SELECT * FROM DEPENDENTS;
```

13. パッケージへの変更を保存します。
14. ストアド・プロシージャを実行するには、Server ExplorerでHR_DATAパッケージを開きます。

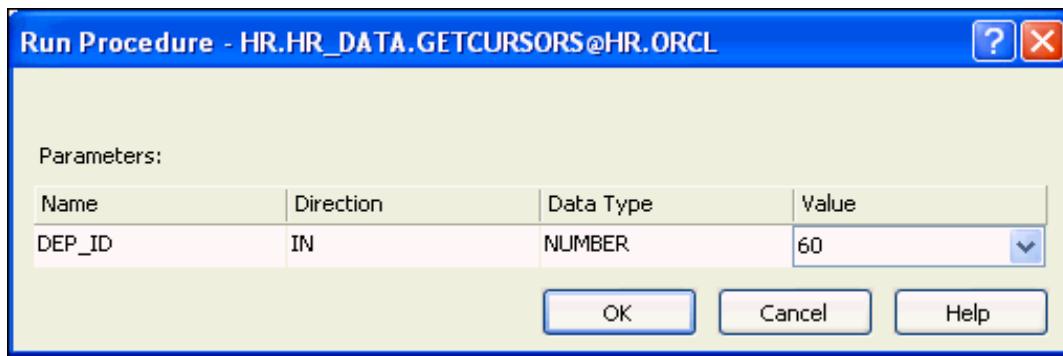
GETCURSORSメソッドを右クリックして、「Run」を選択します。



[procedure5.gifの説明](#)

「Run Procedure」ウィンドウが表示されます。

15. 「Run Procedure」ウィンドウで、DEP_IDの「Value」に60を入力します。



[procedure6.gifの説明](#)

- 「OK」をクリックします。

「Output」ウィンドウが表示され、正しく実行されたことが示されます。

結果ウィンドウに次のメッセージが表示されます。

Procedure <HR.HR_DATA.GETCursors@hr.database> was run successfully.

このメッセージの下で (DEP_IDとともに表示される) 2つの出力パラメータ (EMPLOYEES_CおよびDEPENDENTS_C) を確認します。

- EMPLOYEES_Cの「Value」列のエントリを選択します。

「Parameter Details」領域が表示され、部門60の従業員が表示されます。DEP_IDの値は60です。

Parameters:			
Name	Direction	Data Type	Value
DEP_ID	IN	NUMBER	60
EMPLOYEES_C	OUT	REF CURSOR	<Click here for details...>
DEPENDENTS_C	OUT	REF CURSOR	<Click here for details...>

Parameter Details - EMPLOYEES_C:						
EMPLOYEE_I	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUM	HIRE_DATE	JOB_ID
103	Alexander	Hunold	AHUNOLD	590.423.4567	1/3/1990	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	5/21/1991	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	6/25/1997	IT_PROG
106	Valli	Pataballa	VPATABAL	590.423.4560	2/5/1998	IT_PROG
107	Diana	Lorentz	DLORENTZ	590.423.5567	2/7/1999	IT_PROG

[procedure7.gifの説明](#)

- DEPENDENTS_Cの「Value」列のエントリを選択します。

「Parameter Details」領域が表示され、DEPENDENTS_Cの値が表示されます。

Parameters:			
Name	Direction	Data Type	Value
DEP_ID	IN	NUMBER	60
EMPLOYEES_C	OUT	REF CURSOR	<Click here for details...>
DEPENDENTS_C	OUT	REF CURSOR	<Click here for details...>

Parameter Details - DEPENDENTS_C:					
LAST_NAME	FIRST_NAME	BIRTH_DATE	RELATIONSHI	EMPLOYEE_ID	DEPENDENT_
Ernst	Mary	5/6/2000	daughter	104	1122
Atkinson	Sue	7/12/1998	daughter	130	1301
Ernst	David	4/2/2007	son	104	1042
Sciarra	Aaron	1/31/2008	son	111	1111

[procedure8.gifの説明](#)

ストアド・プロシージャを実行するためのODP.NETアプリケーションの変更

この項では、Oracle Data Provider for .NETアプリケーションを変更してPL/SQLストアド・プロシージャを実行できるようにする方法を、GETCURSORSストアド・プロシージャを例に説明します。

ストアド・プロシージャを実行できるようにアプリケーションを変更するには、次の手順を実行します。

1. アプリケーションHR_Connect_CSまたはHR_Connect_VBを開きます。
2. [付録B「フォームのコピー」](#)の手順に従って、[第4章](#)の最後で完成させたForm3.xxのコピーを作成し、Form4.xxという名前を付けます。
3. Form1を選択し、コード・ビューに切り替えます。
4. connect_Click()メソッドのTryブロックで、コマンドを割り当てる2つの行（cmd = New OracleCommand...で始まる行）を、次に示すコードと置き換えます。

Visual C#:

```
cmd = new OracleCommand("HR_DATA.GETCURSORS", conn);
cmd.CommandType = CommandType.StoredProcedure;
```

Visual Basic:

```
cmd = new OracleCommand("HR_DATA.GETCURSORS", conn)
cmd.CommandType = CommandType.StoredProcedure
```

5. 手順4で追加したコードの下に、GETCURSORSストアド・プロシージャの3つのパラメータの定義とバインドを、それぞれdep_id、employees_cおよびdependents_cという名前のOracleParameterオブジェクトとして追加します。

Visual C#:

```
OracleParameter dep_id = new OracleParameter();
dep_id.OracleDbType = OracleDbType.Decimal;
dep_id.Direction = ParameterDirection.Input;
dep_id.Value = 60;
cmd.Parameters.Add(dep_id);

OracleParameter employees_c = new OracleParameter();
employees_c.OracleDbType = OracleDbType.RefCursor;
employees_c.Direction = ParameterDirection.Output;
cmd.Parameters.Add(employees_c);
```

```
OracleParameter dependents_c = new OracleParameter();
dependents_c.OracleDbType = OracleDbType.RefCursor;
dependents_c.Direction = ParameterDirection.Output;
cmd.Parameters.Add(dependents_c);
```

Visual Basic:

```
Dim dep_id As OracleParameter = New OracleParameter
dep_id.OracleDbType = OracleDbType.Decimal
dep_id.Direction = ParameterDirection.Input
dep_id.Value = 60
cmd.Parameters.Add(dep_id)

Dim employees_c As OracleParameter = New OracleParameter
employees_c.OracleDbType = OracleDbType.RefCursor
employees_c.Direction = ParameterDirection.Output
cmd.Parameters.Add(employees_c)

Dim dependents_c As OracleParameter = New OracleParameter
dependents_c.OracleDbType = OracleDbType.RefCursor
dependents_c.Direction = ParameterDirection.Output
cmd.Parameters.Add(dependents_c)
```

6. アプリケーションをビルドします。

ODP.NETアプリケーションによるPL/SQLストアド・プロシージャの実行

この項では、GETCursorsストアド・プロシージャなどのPL/SQLストアド・プロシージャを、ODPアプリケーションから実行する方法を説明します。

ストアド・プロシージャを実行するには、次の手順を実行します。

1. アプリケーションを実行します。

「Form1」ウィンドウが表示されます。

2. 「Form1」ウィンドウで接続情報を入力し、「**Connect**」をクリックします。
3. DataGridViewオブジェクトで、水平方向にスクロールして最後の列DEPARTMENT_IDに含まれる値が60のみであることを確認します。

DataGridViewには、ストアド・プロシージャの最初の結果セットが含まれており、これはEMPLOYEES表の問合せと一致することに注意してください。

	SALARY	COMMISSION_PC	MANAGER_ID	DEPARTMENT_ID
▶	9000		102	60
	6000		103	60
	4800		103	60
	4800		103	60
	4200		103	60

[procedure9.gifの説明](#)

4. アプリケーションを閉じます。

7 Oracle DatabaseでのASP.NETの使用

この章の内容は次のとおりです。

- [概要: Oracle Developer Toolsを使用したASP.NETアプリケーションの構築](#)
- [このチュートリアルを始める前に](#)
- [Webサイトの作成およびデータベースへの接続](#)
- [認証用のWebサイトの有効化](#)
- [Oracle Providers for ASP.NETの有効化および軽量Webユーザーの作成](#)
- [Webサイト認証のテスト](#)

概要: Oracle Developer Toolsを使用したASP.NETアプリケーションの構築

Oracleは、様々な方法でMicrosoft ASP.NETと直接統合されます。

- Oracle Developer Tools for Visual Studioでは、データ駆動型のWebサイトを設計する簡単な方法が提供されます。
- ODP.NETによって、ASP.NETデータ・アクセスが可能になります。
- Oracle Providers for ASP.NETは、Microsoft ASP.NETのコントロールおよびサービスと直接統合され、Webサイトの状態管理機能が提供されます。

このチュートリアルでは、Oracle Developer Toolsを使用してデータ駆動型のWebアプリケーションを構築する方法や、Oracle Providers for ASP.NETを使用した簡単な方法でそのアプリケーションにセキュリティを追加する方法など、これらの機能のいくつかを紹介します。

まず、これらのツールを使用して、Oracle Databaseからデータ・グリッドに従業員データを取得するWebアプリケーションを構築します。その後、ログイン・コントロールを追加し、認可されたWebユーザーに対してのみこの従業員情報へのアクセスを許可することでこのアプリケーションを保護します。最後に、Oracle Providers for ASP.NETを使用して、認可されたWebユーザーを作成します。このWebユーザーは、アプリケーションによる認証のためにOracle Database内に格納されます。

このチュートリアルを始める前に

Oracle Developer ToolsでASP.NETアプリケーションを構築するには、次の各項で説明されているセットアップを実行する必要がある場合があります。

- [「Oracle Databaseへの接続」](#)
- Webサイト認証を使用する予定がある場合は、[「認証用のWebサイトの有効化」](#)に示すように、次のセットアップを実行する必要があります。
 - [「ユーザーの作成および権限の付与」](#)
 - [「すべてのOracle Providers for ASP.NETの構成」](#)
 - [「接続文字列の設定」](#)

Webサイトの作成およびデータベースへの接続

この項では、Oracle Databaseからデータを取得するASP.NET Webサイトを作成する方法について説明します。 Webサイトでは、ASP.NET GridViewにデータが表示され、ユーザーは結果を参照できるようになります。

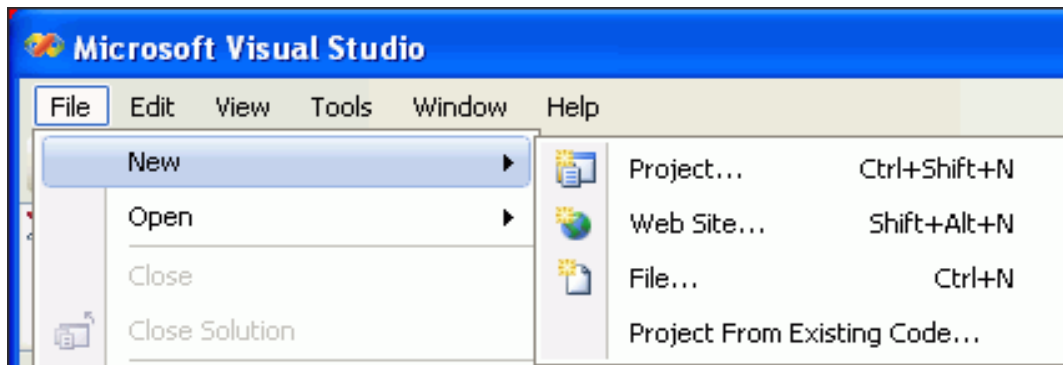
この項の内容は次のとおりです。

- [ASP.NET Webサイトの作成](#)
- [データソースの作成](#)

ASP.NET Webサイトの作成

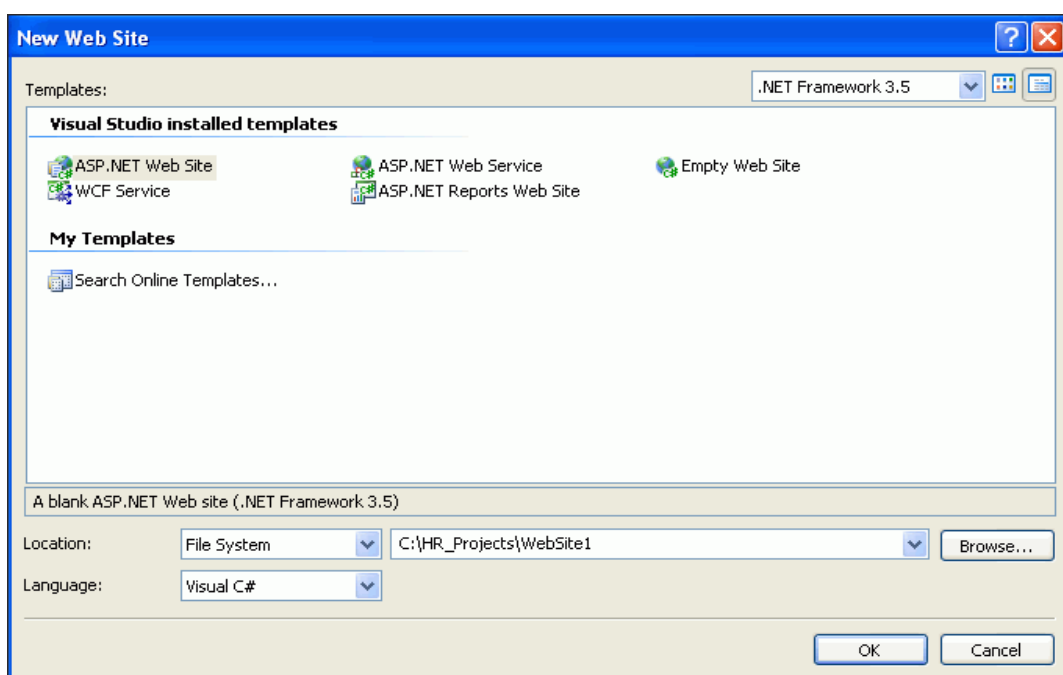
ASP.NET Webサイトをグリッドで作成するには、次の手順を実行します。

1. Visual Studioを起動します。
2. 「File」メニューから「New」、「Web Site...」の順に選択します。



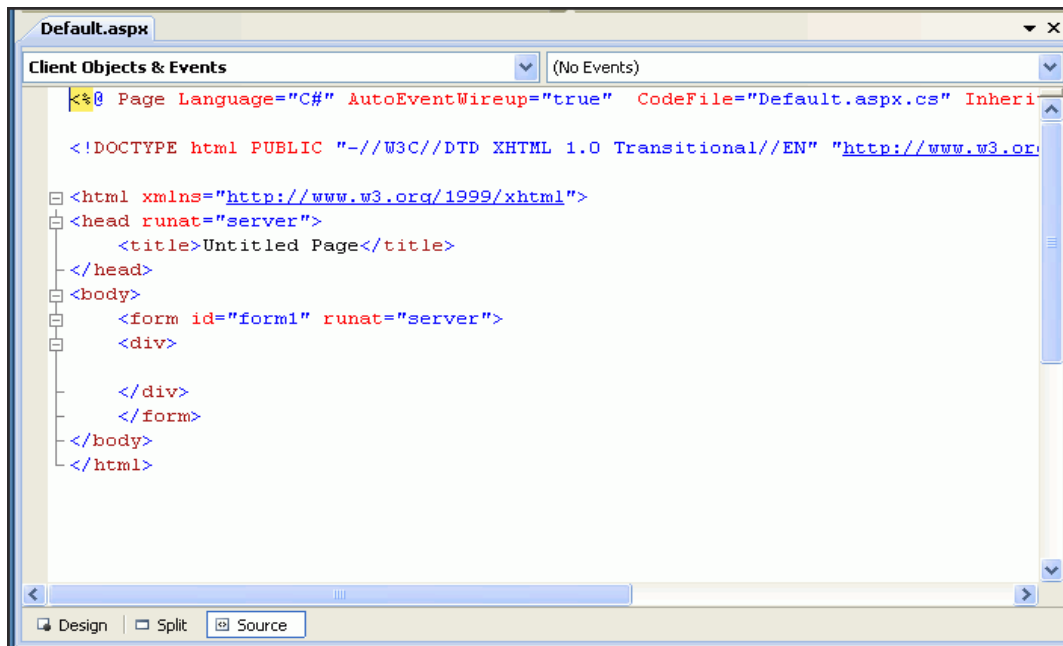
[aspnt01.gifの説明](#)

3. 「New Web Site」から、「ASP.NET Web Site」を選択し、Webサイト用のディレクトリの場所を入力するか、または参照します。「OK」をクリックします。



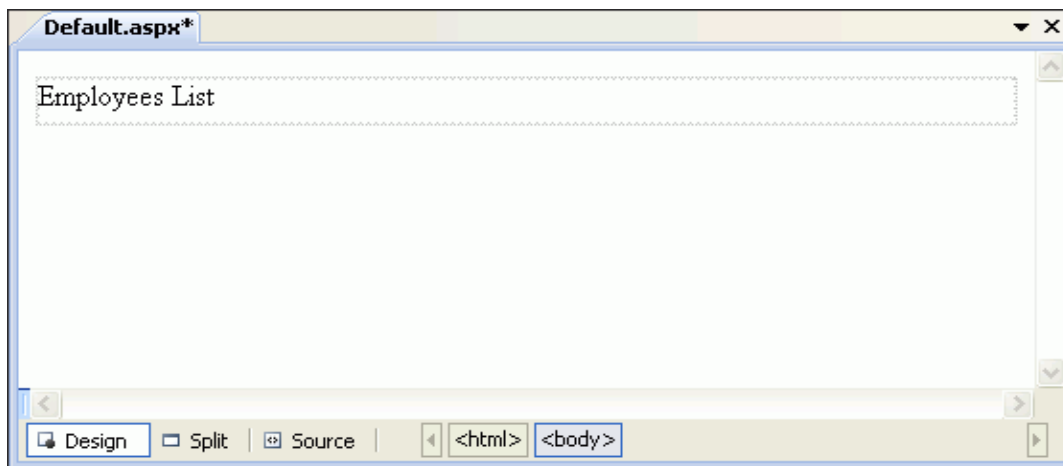
[aspnt02.gifの説明](#)

4. 「Default.aspx」タブで、画面下部の「Design」アイコンをクリックします。



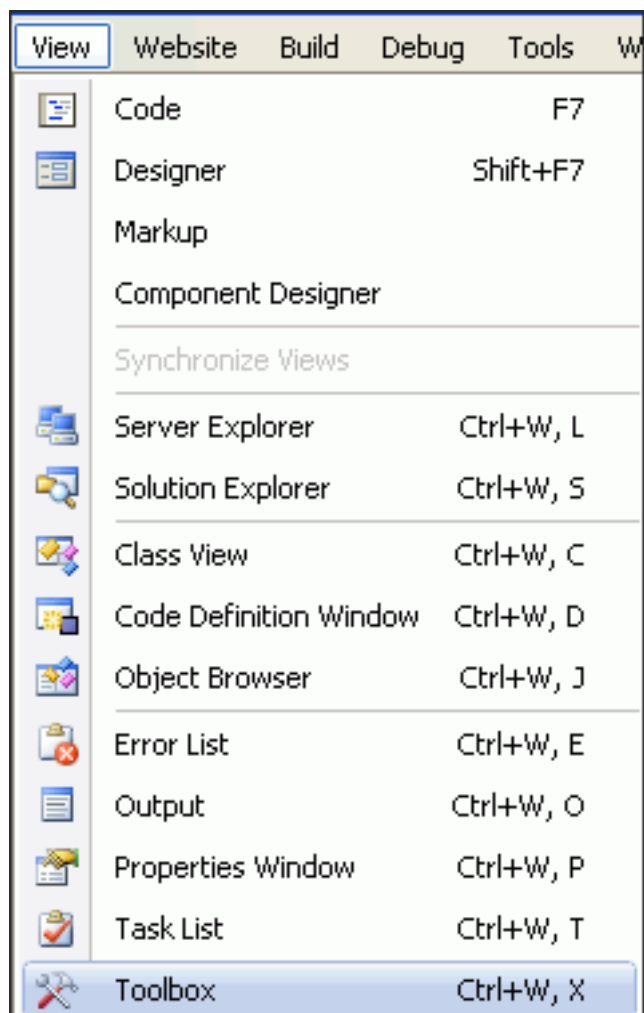
[aspnt03.gifの説明](#)

5. 点線の四角形として表示される<div>要素内に、Employees Listなどのタイトルを入力します。



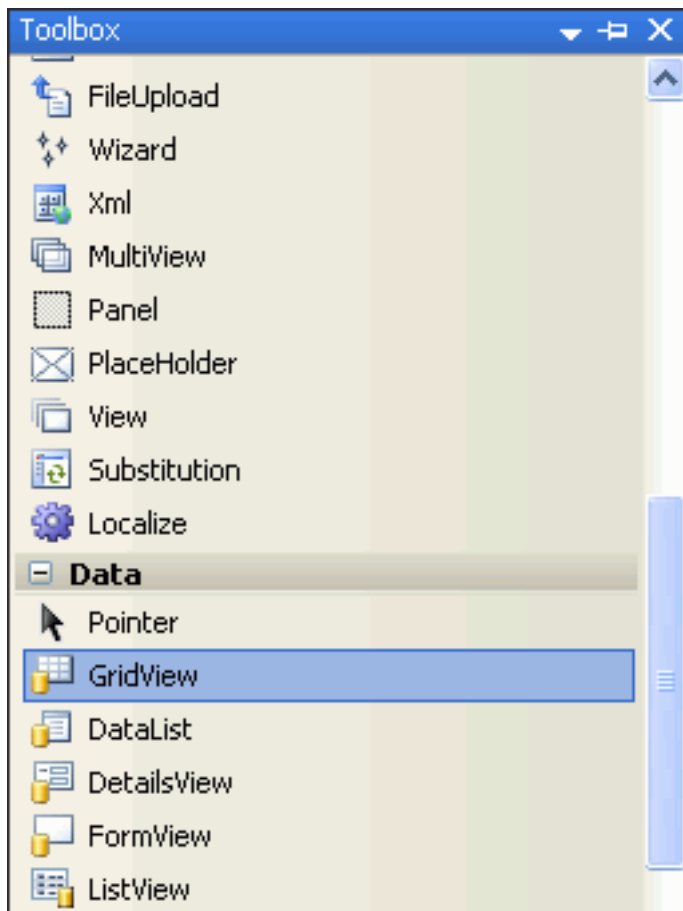
[aspnt04.gifの説明](#)

6. 「View」メニューから「Toolbox」を選択します。



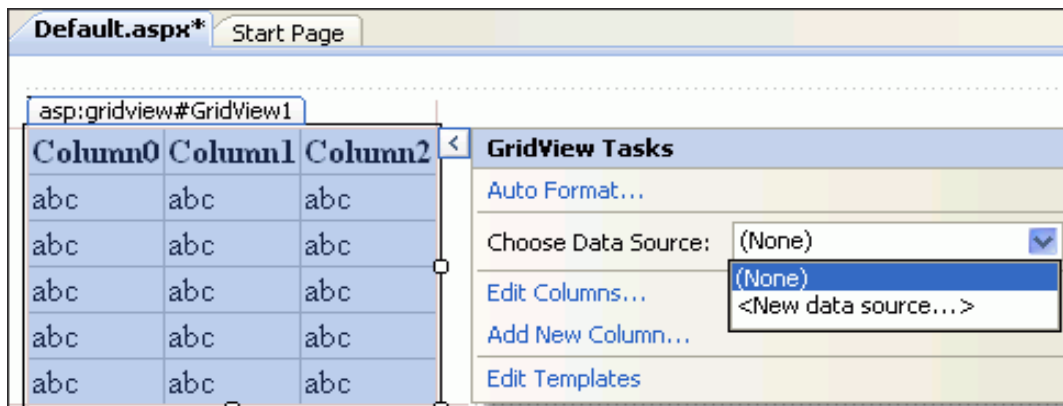
[aspnt05.gifの説明](#)

7. 「Data」グループを展開し、「GridView」コントロールを、デザイナの<div>というラベル付いた点線の四角形内にドラッグします。



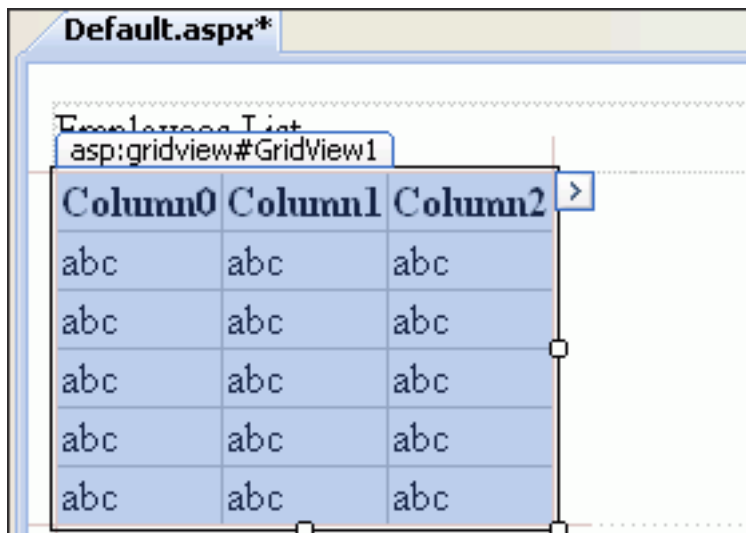
[aspnt06.gifの説明](#)

8. ダミーのタイトルと内容が含まれているグリッドが表示されたら、右側の「GridView Tasks」リストを確認します。



[aspnt08.gifの説明](#)

このタスク・リストが表示されない場合は、グリッドを選択してから右側の>記号をクリックします。



asnpt07.gifの説明

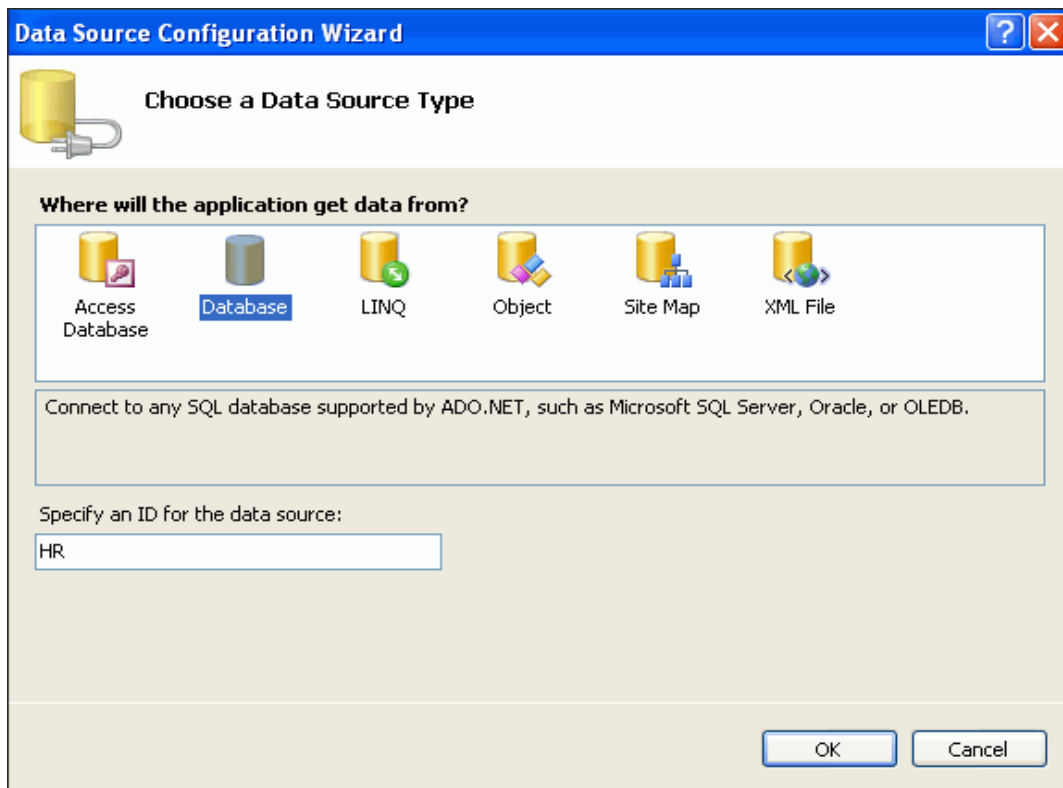
データソースの作成

データソースを作成するには、次の手順を実行します。

1. 前述の項（手順8）に示されているように、「GridView Tasks」の「Choose Data Source」リストから「<New data source...>」を選択します。

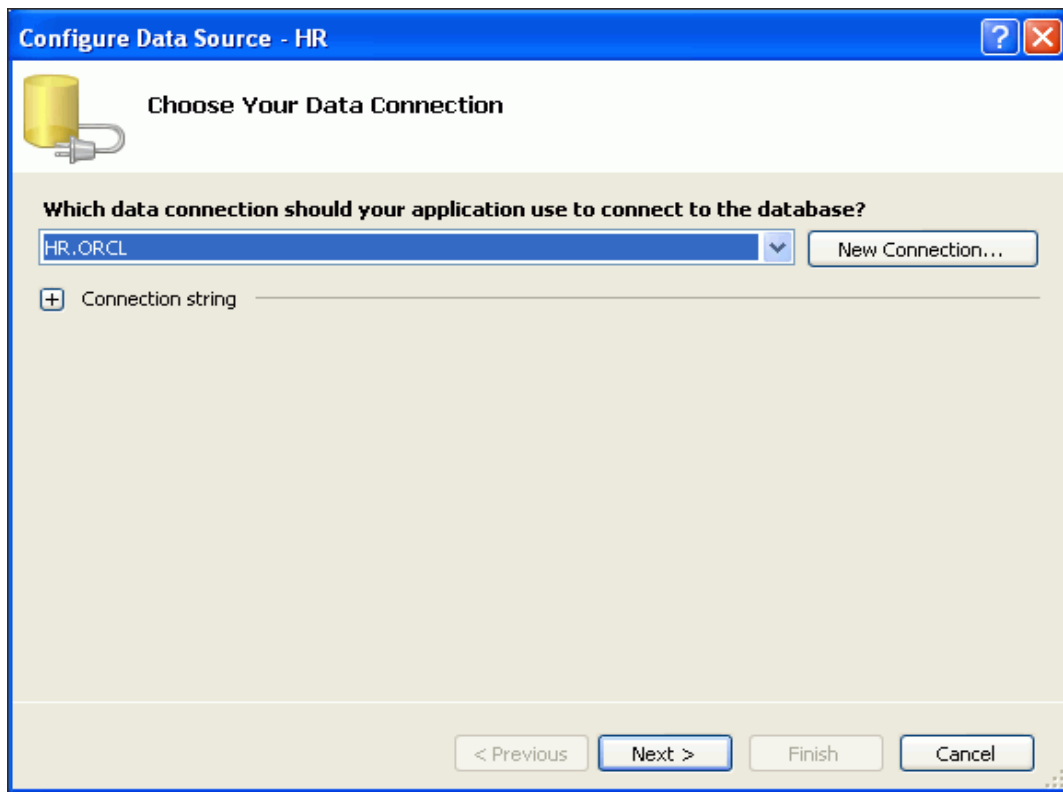
「Data Source Configuration Wizard」が起動されます。

2. 「Database」を選択します。 このデータソースのIDとしてHRを入力します。「OK」をクリックします。



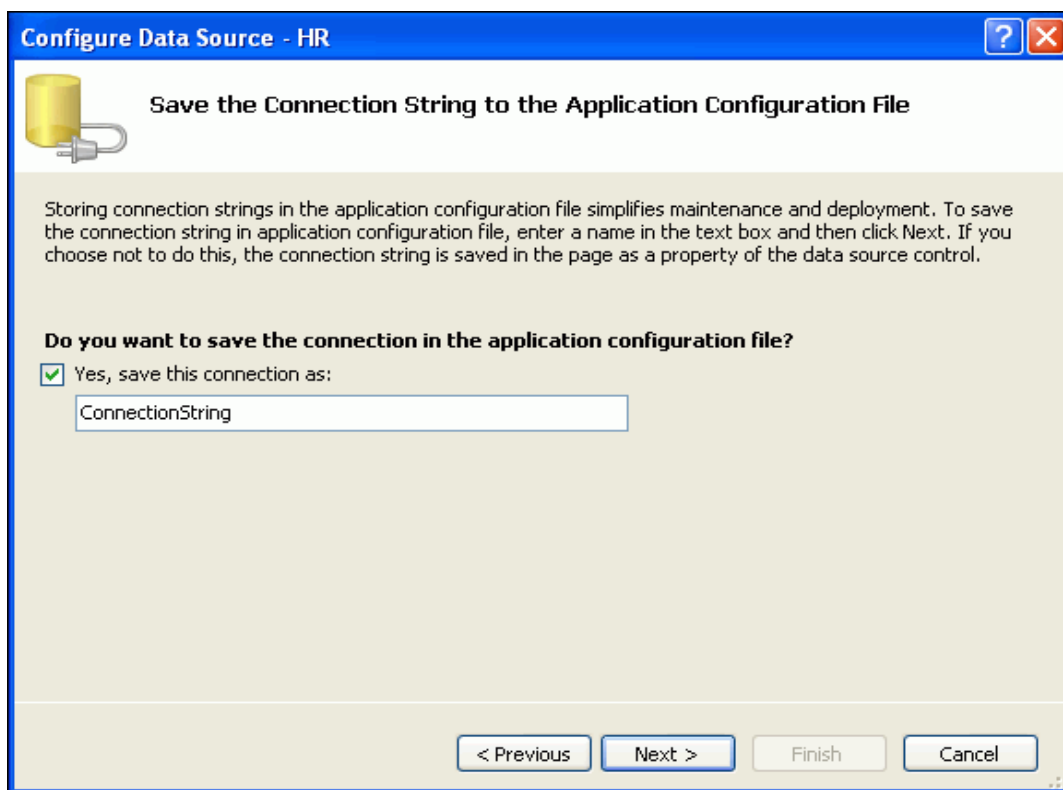
aspnt09.gifの説明

3. 下矢印をクリックして、リストから「HR.ORCL」を選択します。「Next」をクリックします。



[aspnt10.gifの説明](#)

4. 「Next」をクリックして、接続文字列をアプリケーション構成ファイルに保存します。

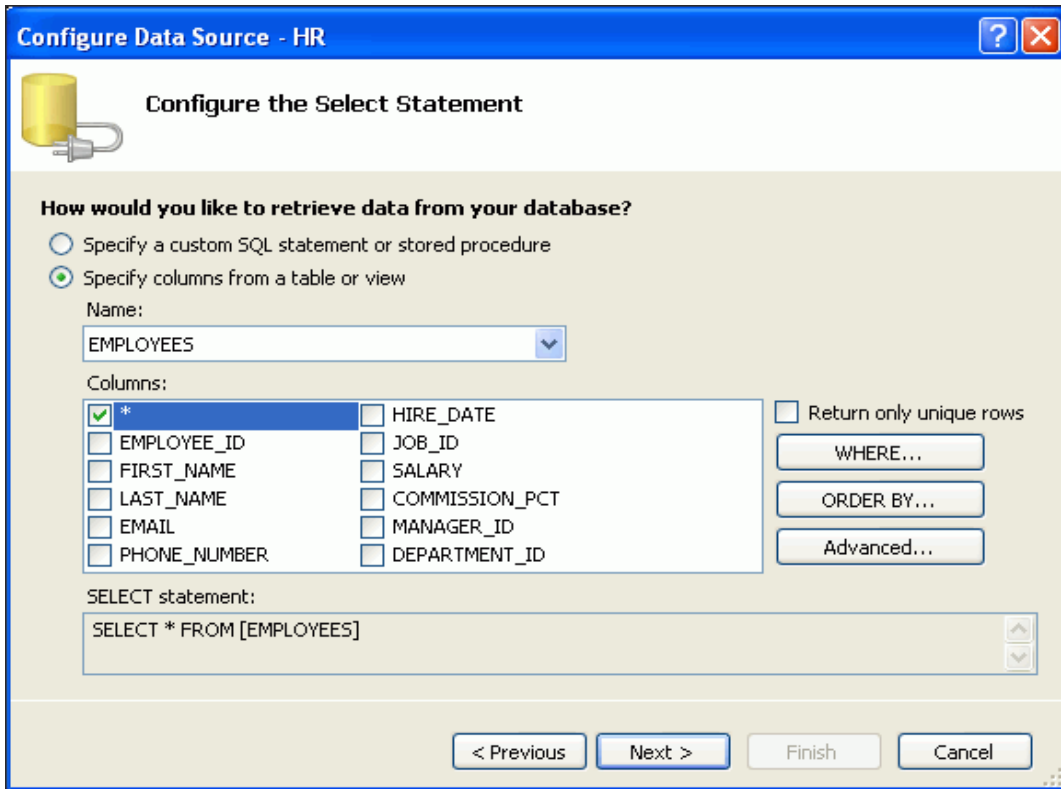


[aspnt11.gifの説明](#)

5. 「Name」リストからEMPLOYEES表を選択します。「Columns」リストで、アスタリスク (*) の横にあるボックスを選択します。

このように選択すると、EMPLOYEES表のすべての行を戻すようにOracleに指示されます。SELECT * FROM EMPLOYEESと入力した場合と同様です。

「Next」をクリックします。



Configure Data Source - HR

Configure the Select Statement

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name:

Columns:

<input checked="" type="checkbox"/> *	<input type="checkbox"/> HIRE_DATE
<input type="checkbox"/> EMPLOYEE_ID	<input type="checkbox"/> JOB_ID
<input type="checkbox"/> FIRST_NAME	<input type="checkbox"/> SALARY
<input type="checkbox"/> LAST_NAME	<input type="checkbox"/> COMMISSION_PCT
<input type="checkbox"/> EMAIL	<input type="checkbox"/> MANAGER_ID
<input type="checkbox"/> PHONE_NUMBER	<input type="checkbox"/> DEPARTMENT_ID

☐ Return only unique rows

WHERE...

ORDER BY...

Advanced...

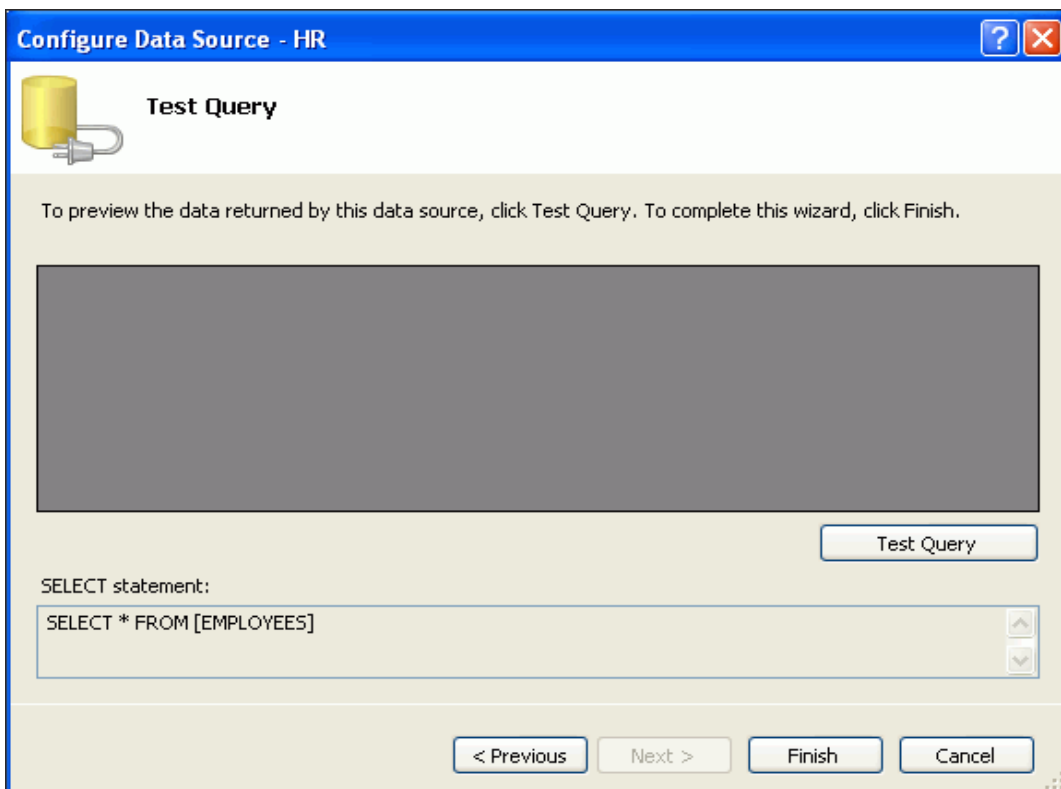
SELECT statement:

SELECT * FROM [EMPLOYEES]

< Previous Next > Finish Cancel

[aspnt12.gifの説明](#)

- 「Test Query」をクリックします。



Configure Data Source - HR

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

SELECT statement:

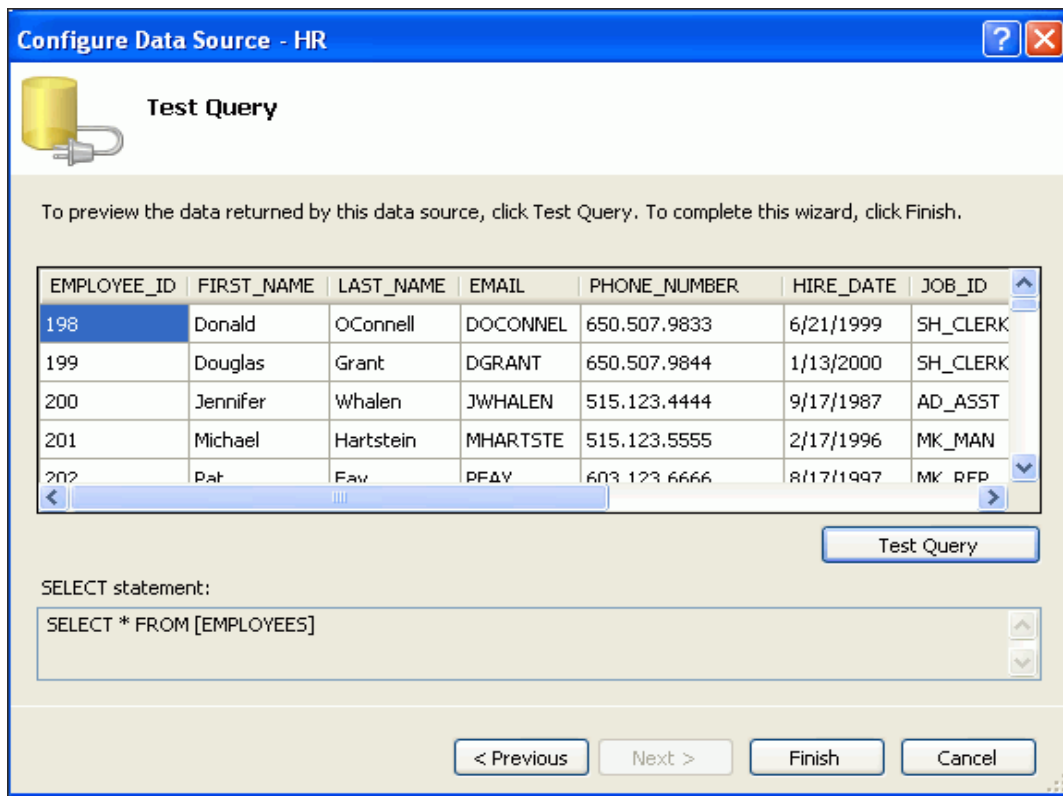
SELECT * FROM [EMPLOYEES]

Test Query

< Previous Next > Finish Cancel

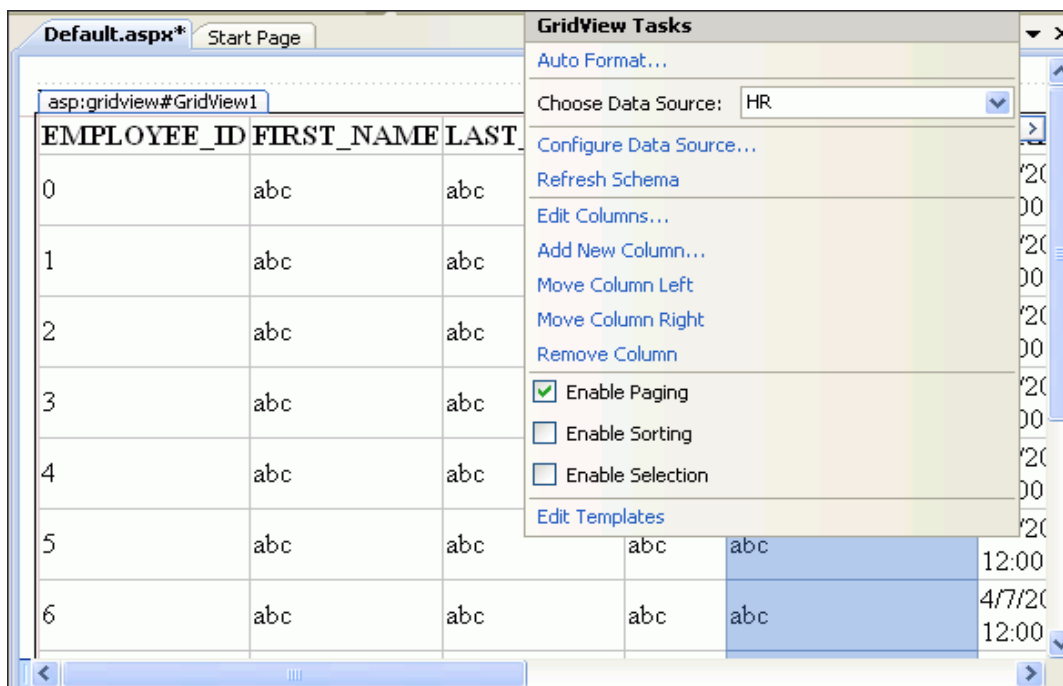
[aspnt13.gifの説明](#)

- 「Test Query」の結果が表示されたら、「Finish」をクリックします。



[aspnt14.gifの説明](#)

- 「GridViewTasks」から、「Enable Paging」を選択します。このタスク・リストが表示されない場合は、グリッドを選択してから右側の>記号をクリックします。 右にスクロールさせる必要がある場合があります。



[aspnt15.gifの説明](#)

- 「View」メニューから「Solution Explorer」を選択し、Webサイトを選択します。 右クリックして「Build Web Site」を選択します。 ステータス・バーに成功または失敗が表示されます。
- 「View」メニューから「Debug」を選択し、「Start Without Debugging」を選択します。

次のようなブラウザ・ウィンドウが表示され、問合せでリクエストしたデータが表示されます。 このページの左下の番号を使用すると、ページを指定して結果を表示できます。

Address  http://localhost:1458/WebSite1/Default.aspx					
Employees List					
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
198	Donald	OConnell	DOCONNEL	650.507.9833	6/21/1999 12:00:00 AM
199	Douglas	Grant	DGRANT	650.507.9844	1/13/2000 12:00:00 AM
200	Jennifer	Whalen	JWHALEN	515.123.4444	9/17/1987 12:00:00 AM
201	Michael	Hartstein	MHARTSTE	515.123.5555	2/17/1996 12:00:00 AM
202	Pat	Fay	PFAY	603.123.6666	8/17/1997 12:00:00 AM
203	Susan	Mavris	SMAVRIS	515.123.7777	6/7/1994 12:00:00 AM
204	Hermann	Baer	HBAER	515.123.8888	6/7/1994 12:00:00 AM
205	Shelley	Higgins	SHIGGINS	515.123.8080	6/7/1994 12:00:00 AM
206	William	Gietz	WGIEZT	515.123.8181	6/7/1994 12:00:00 AM
100	Steven	King	SKING	515.123.4567	6/17/1987 12:00:00 AM
1 2 3 4 5 6 7 8 9 10 ...					

[aspnt16.gifの説明](#)

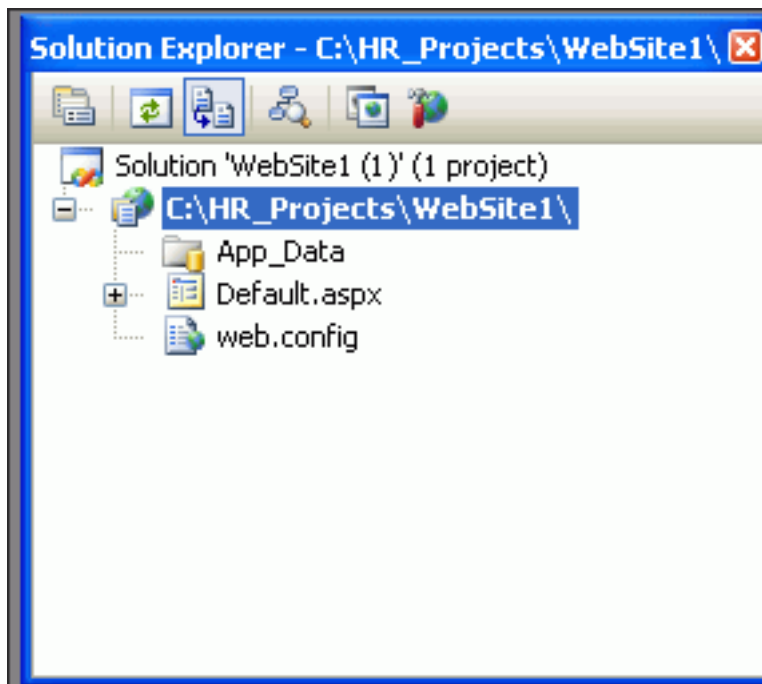
11. ブラウザを閉じます。

認証用のWebサイトの有効化

この項では、従業員データにアクセスできるユーザーを制限するために、Webサイト認証を追加する方法について説明します。 ASP.NETログイン・コントロールを使用して認証を行い、Oracle Providers for ASP.NETで作成および格納されたユーザーと照合します。

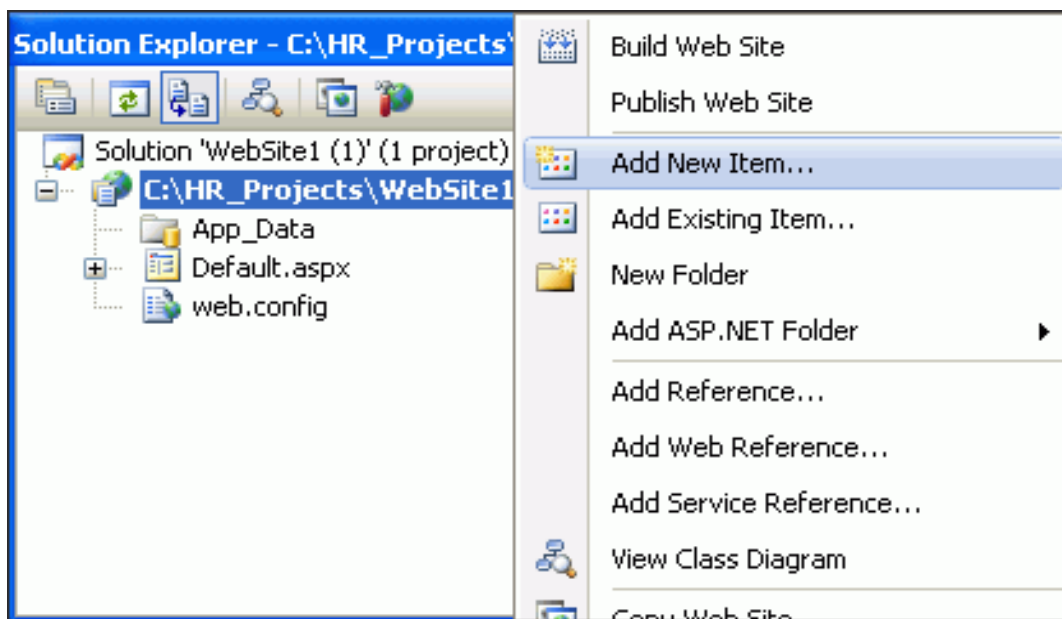
次の項の「[Oracle Providers for ASP.NETの有効化および軽量Webユーザーの作成](#)」では、構築したASP.NETアプリケーションを使用し、認可されたユーザーが情報にアクセスできるように従業員データを保護します。 Oracle Providers for ASP.NETを使用して、アプリケーションのWebユーザーを作成します。 このWebユーザーは、ログイン・コントロールを使用してWebアプリケーションを起動し、資格証明が正しい場合は従業員情報にアクセスできます。

1. 前の項で作成したWebサイトを再度開きます。
2. 「View」、「Solution Explorer」の順に選択し、このWebサイトをクリックします。



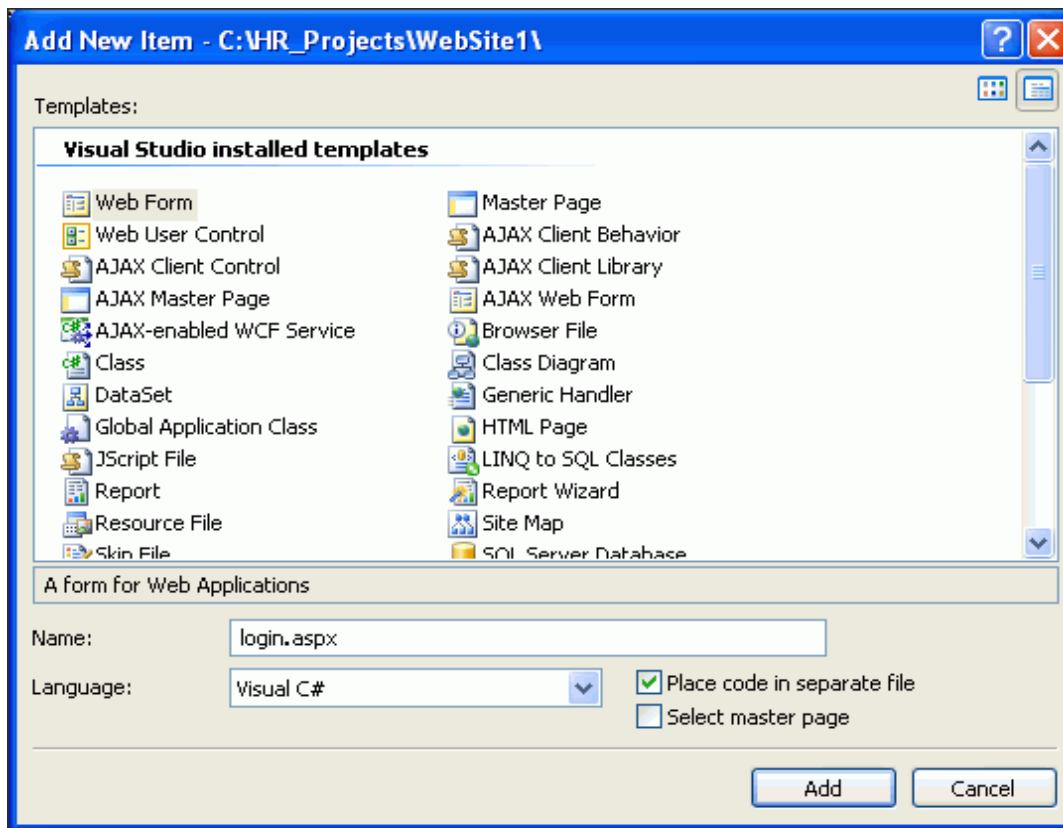
[aspnt17.gifの説明](#)

3. このWebサイトを右クリックして「Add New Item」を選択します。



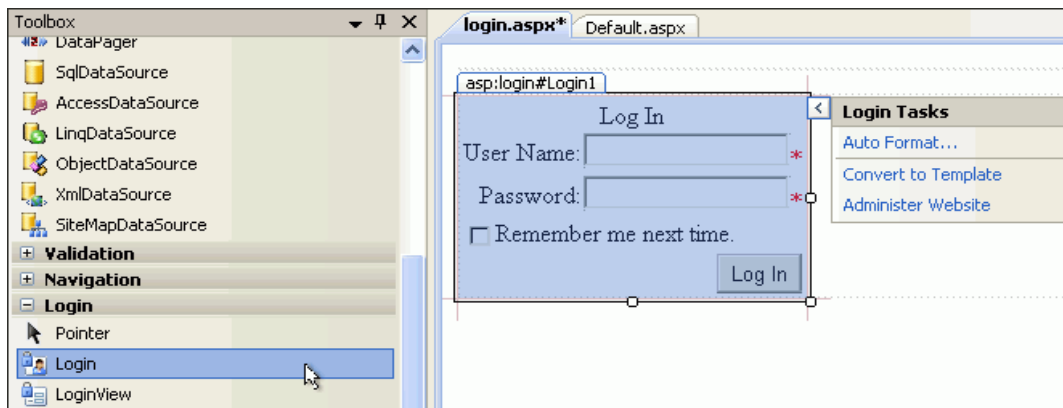
[aspnt18.gifの説明](#)

4. 「Web Form」を選択し、名前としてlogin.aspxを入力し、「Add」をクリックします。



[aspnt19.gifの説明](#)

5. 「login.aspx」ページが表示されたら、「Design」タブに切り替えます。
6. 「View」メニューから「Toolbox」を開き、「Login」セクションを展開して、ログイン・コントロールをフォームの<div>というラベルの付いた点線の四角形内にドラッグ・アンド・ドロップします。

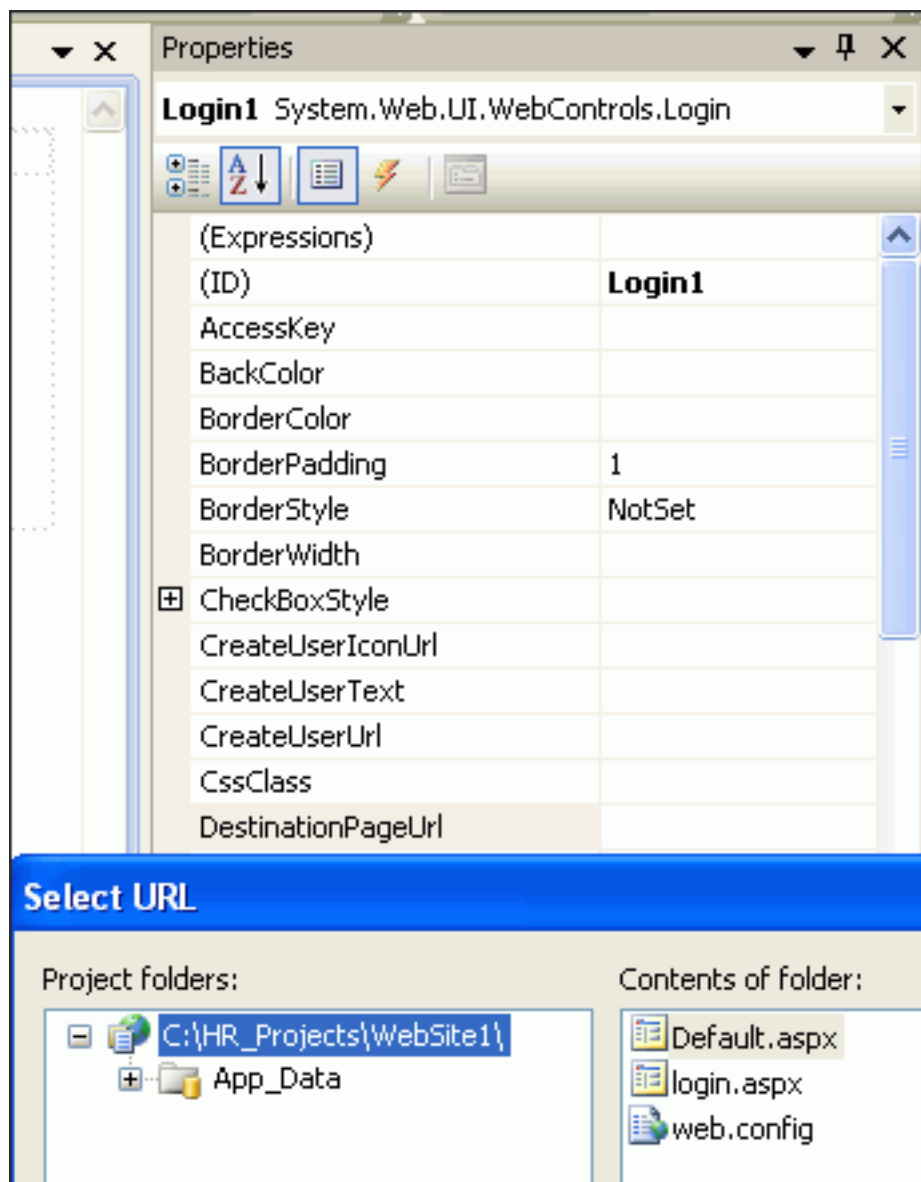


[aspnt20.gifの説明](#)

これは、標準のASP.NETログイン・コントロールで、Oracle Databaseに格納されているユーザー・ログイン資格証明を取得して確認できます。

7. ログイン・コントロールを右クリックし、「Properties」を選択します。「DestinationPageUrl」に対して、Default.aspxを選択するか、または入力します。

ログインに成功すると、従業員データが含まれている「Default.aspx」ページが表示されます。 ログインに失敗すると、ログイン・ページに戻ります。



[aspnt21.gifの説明](#)

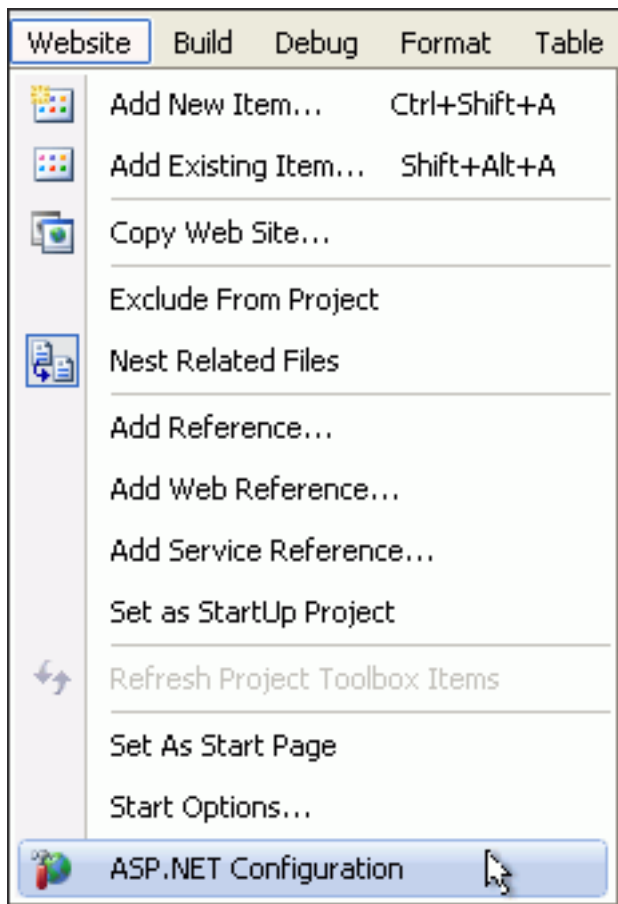
Oracle Providers for ASP.NETの有効化および軽量Webユーザーの作成

この項では、ASP.NET Webサイト管理ツールを使用して次の作業を実行します。

- Oracle ASP.NETプロバイダを使用するようにWebサイトに指示します。
- このWebサイトに固有の新しいWebユーザーを作成して、[「認証用のWebサイトの有効化」](#)でこのサイトに追加した認証機能を実際に使用してみます。

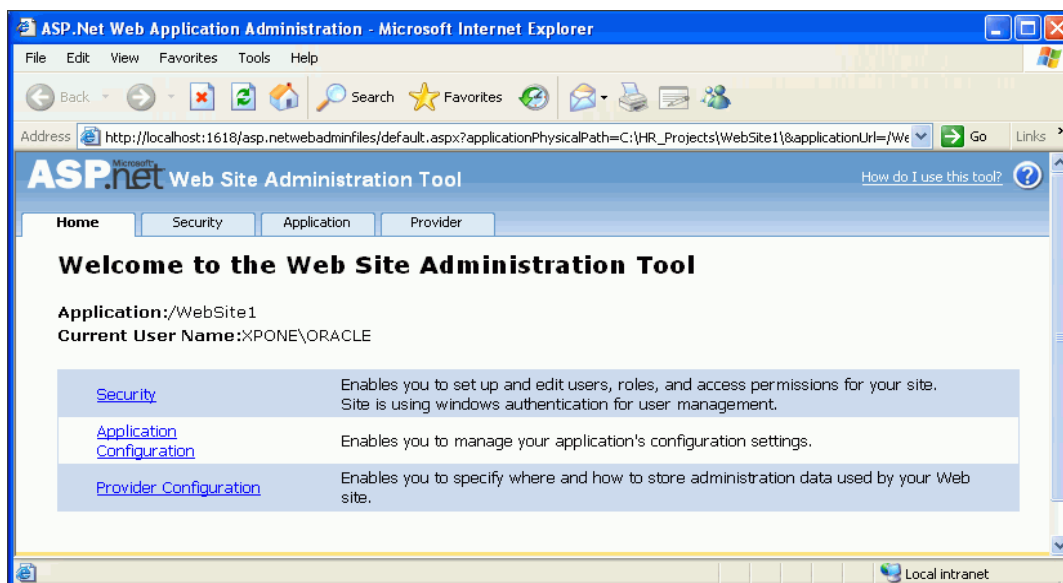
Oracle ASP.NETプロバイダを使用するようにWebサイトに指示し、新しいWebサイト・ユーザーを作成するには、次の手順を実行します。

1. Visual Studioで、「Website」、「ASP.NET Configuration」の順に選択します。



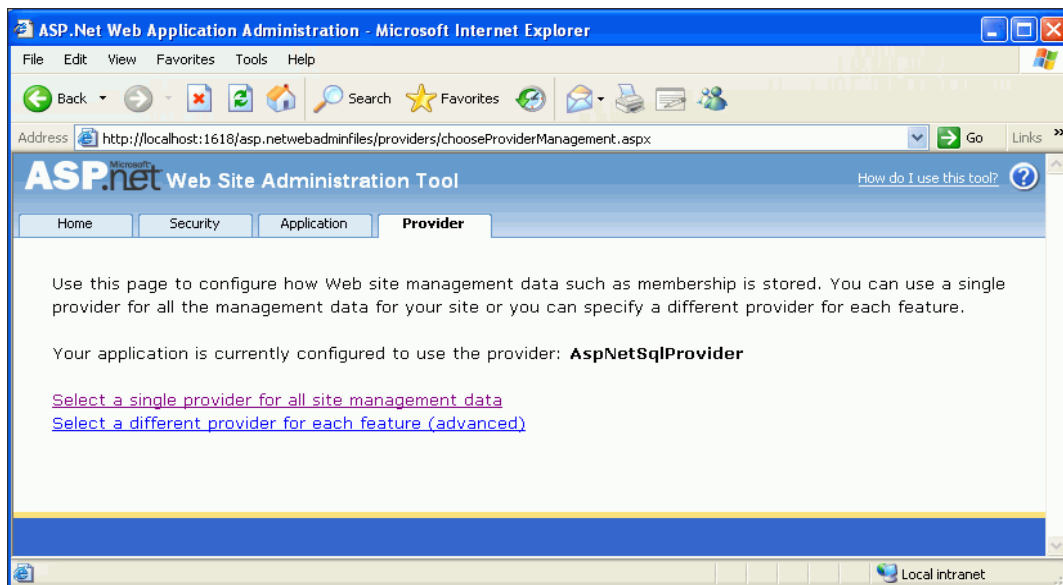
[aspnt23.gifの説明](#)

2. 「ASP.NET Web Site Administration Tool」が表示されたら、「Provider」タブを選択します。



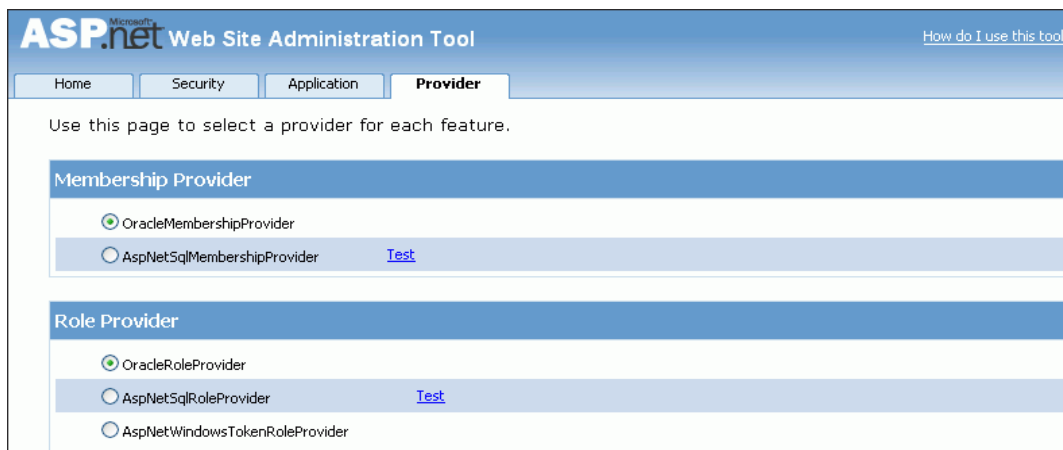
[aspnt24.gifの説明](#)

3. 「Provider」 ページで、2つ目のリンクの「Select a different provider for each feature (advanced)」を選択します。



[aspnt25.gifの説明](#)

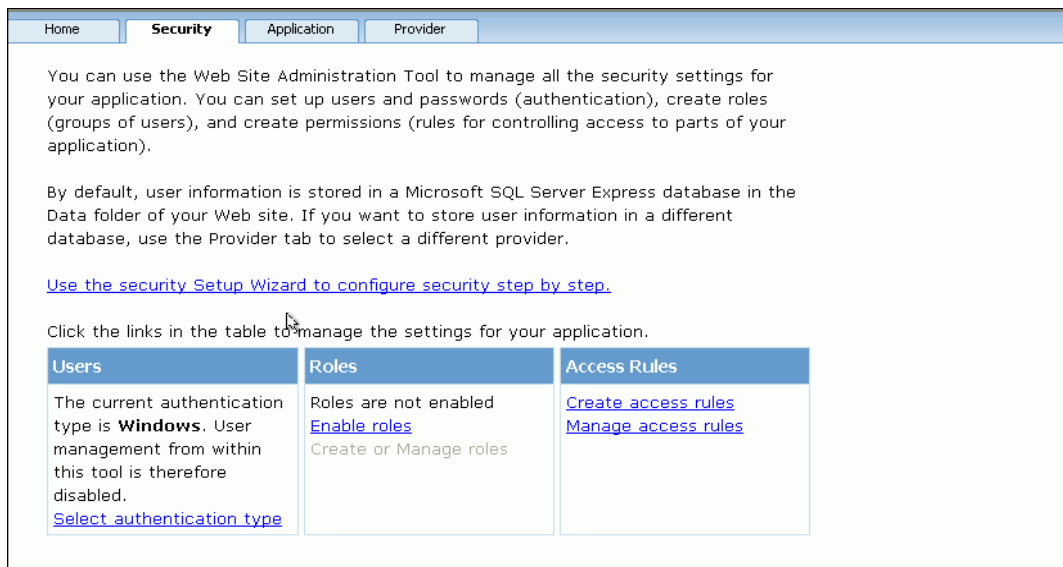
4. 「Provider」ページが再度表示されたら、「Membership Provider」および「Role Provider」でOracleバージョンが選択されていない場合は選択します。



[aspnt26.gifの説明](#)

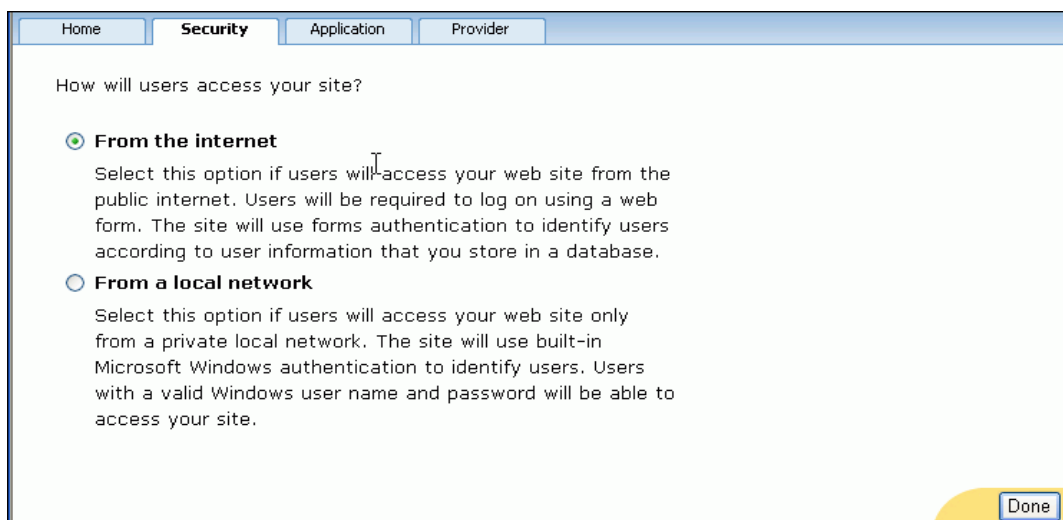
5. 「Security」タブに移動し、「Users」の下で「Select authentication type」をクリックします。

デフォルトでは、ASP.NETサイトはユーザーの識別にWindows認証を使用します。現在構築中のWebサイトでは、サイト固有のログインおよびパスワードでユーザーを識別します。このため、このサイトではログインおよびパスワードの使用が想定されるように構成する必要があります。



[aspnt27.gifの説明](#)

- 「Security」ページが再度表示されたら、「From the internet」を選択して「Done」をクリックします。



[aspnt28.gifの説明](#)

- 「Security」タブが再度表示され、「Users」の下に新しいリンクが表示されたら、「Create user」を選択します。

Home **Security** Application Provider

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

[Use the security Setup Wizard to configure security step by step.](#)

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 0 Create user Manage users Select authentication type	Roles are not enabled Enable roles Create or Manage roles	Create access rules Manage access rules

[aspnt29.gifの説明](#)

8. 「Create User」セクションで、Webサイトへのアクセスを許可するユーザーの情報を次のように入力します。英数字以外の文字を1文字含む7文字以上のパスワードを入力します。

「Create User」をクリックします。

Home **Security** Application Provider

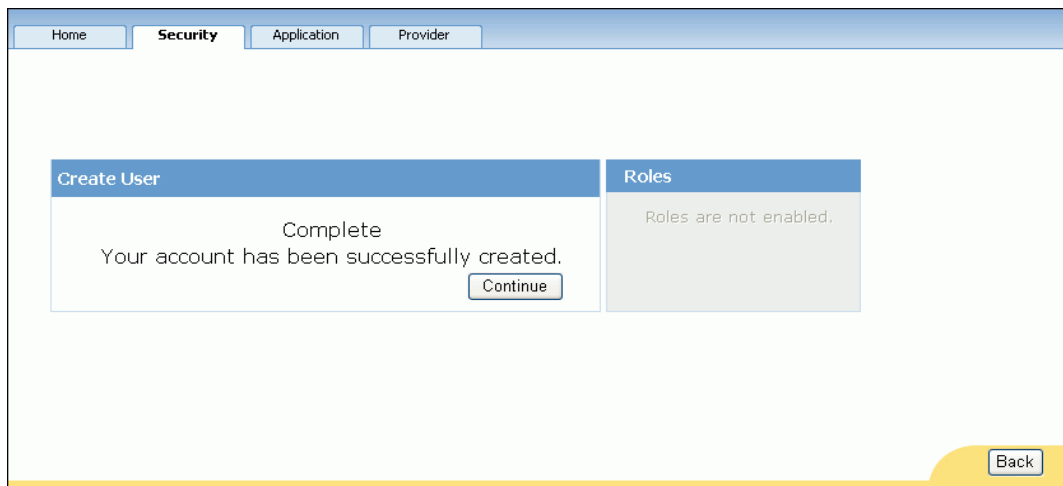
Add a user by entering the user's ID, password, and e-mail address on this page.

Create User	Roles
Sign Up for Your New Account User Name: <input type="text" value="Anne"/> Password: <input type="password" value="●●●●●●"/> Confirm Password: <input type="password" value="●●●●●●"/> E-mail: <input type="text" value="anne@example.com"/> Security Question: <input type="text" value="what is your pet's name"/> Security Answer: <input type="text" value="bingo"/> <input type="button" value="Create User"/>	Roles are not enabled.

☒ Active User

[aspnt30.gifの説明](#)

9. アカウントが正常に作成されたことを示す「Security」ページが表示されたら、「Security」タブをクリックします。



[aspnt31l.gifの説明](#)

注意：この画面のその他のオプションには、他のユーザーの作成、別のタブへの移動などがあります。

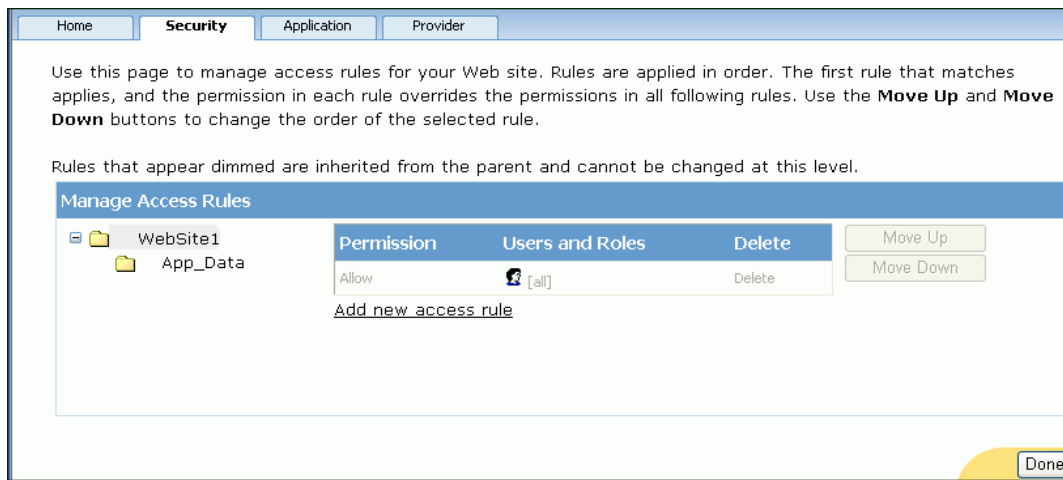
10. 「Security」メイン・ページが再度表示されたら、「Access Rules」の下の「**Manage access rules**」を選択します。



[aspnt31a.gifの説明](#)

注意：「Users」には現在1人のユーザーが存在します。

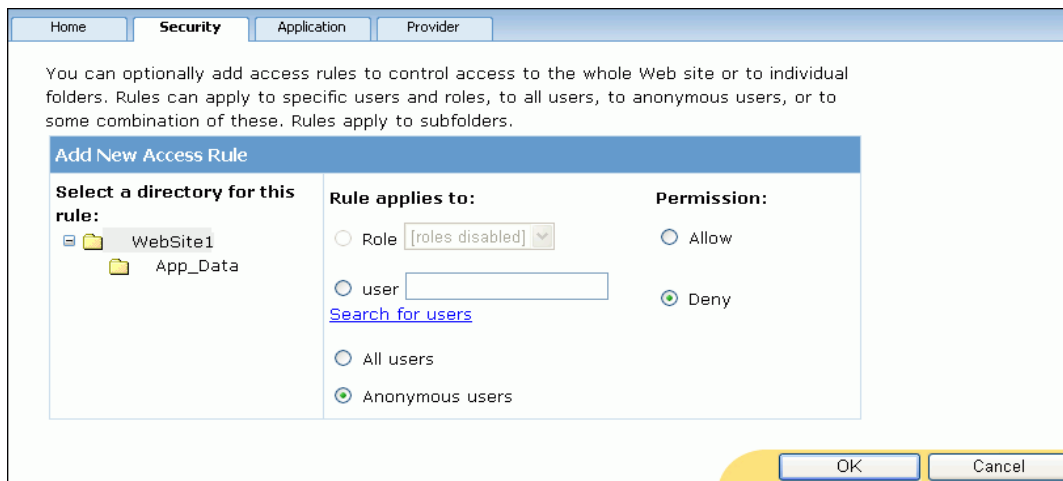
11. 「Security」タブに「Manage Access Rules」セクションが表示されたら、「**Add new access rule**」をクリックします。



[aspnt32.gifの説明](#)

12. 「Anonymous users」および「Deny」を選択し、「OK」をクリックします。

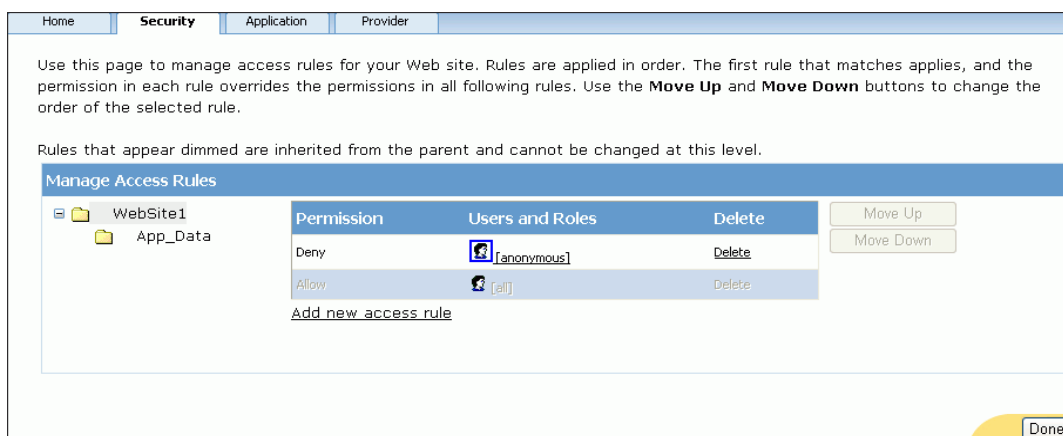
デフォルトでは、このWebサイトへの匿名アクセスは有効です。前述の設定によって、匿名アクセスが無効化されWebサイトが保護されています。これで、従業員データを表示できるのは認証済ユーザーのみです。



[aspnt33.gifの説明](#)

13. 「Security」ページには、Webサイトが匿名ユーザーによるアクセスを拒否したことが示されます。

「Done」をクリックします。



[aspnt34.gifの説明](#)

14. ブラウザを閉じます。

Webサイト認証のテスト

このWebサイトに固有のWebユーザーを作成したため、Webサイトはこのユーザーによる従業員データへのアクセスを許可し、匿名ユーザーを含む他のユーザーによるアクセスを拒否します。

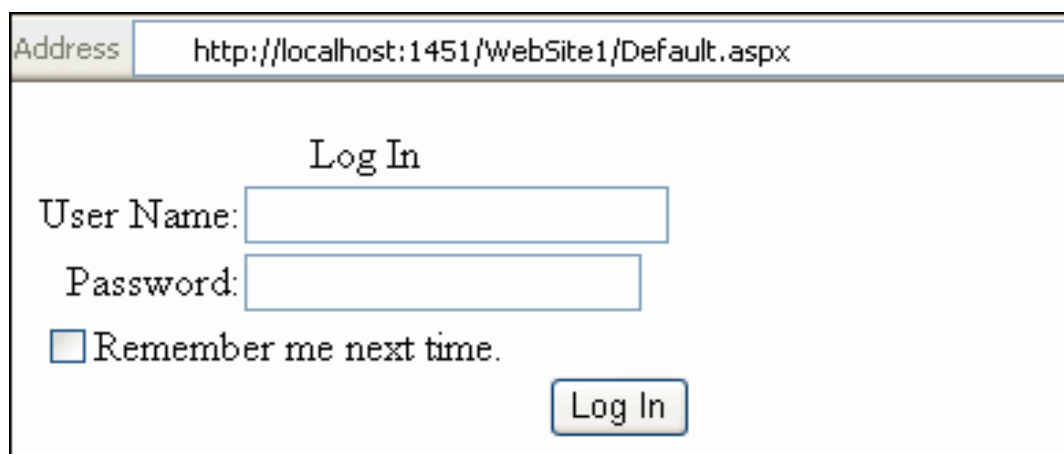
この項では従業員データへのアクセスを、匿名ユーザー、認可されていないユーザー、認可されているがパスワードが誤っているユーザー、認可されていてパスワードも正しいユーザーとして試みます。 Webサイトが従業員データへのアクセスを許可するのは、最後のシナリオのみです。

注意:

ASP.NETプロバイダ・ユーザーとして10分間に5回以上連続して無効なパスワードを入力すると、アカウントはロックされるため、認可されていないユーザーはパスワードを推測してアクセスすることができなくなります。Oracle Membership Providerでは、machine.configファイルまたはweb.configファイルで変更できるプロパティを介してこのようなセキュリティ対策を設定します。プロパティは、MaxInvalidPasswordAttempts（デフォルトの試行回数は5回）およびPasswordAttemptWindow（デフォルトの時間は10分）です。

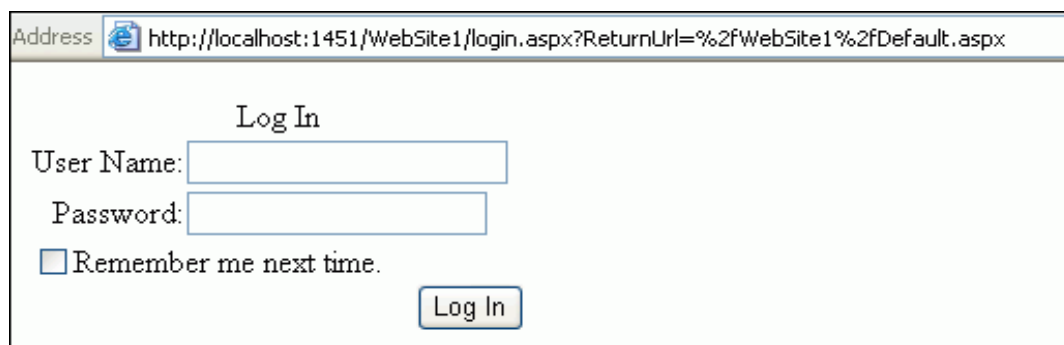
アカウントがロックされた場合は、UnlockUserメソッドをコールするとユーザーをロック解除できます。

1. 「Debug」メニューから「Start Without Debugging」を選択し、ログインWebページが表示されたら、URLの末尾をlogin.aspxではなくDefault.aspxに変更して[Enter]キーを押します。



[aspnt39.gifの説明](#)

アクセスは拒否され、ログイン・ページにリダイレクトされます。このことは、匿名ユーザーはWebサイトを参照できず、資格証明を持つユーザーのみがアクセスできることを示しています。



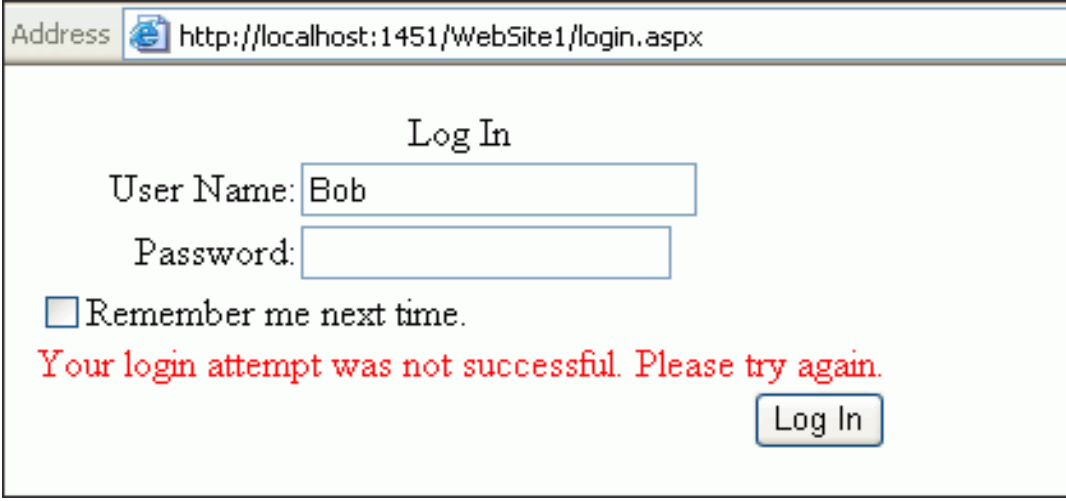
[aspnt40.gifの説明](#)

認証メカニズムを試してみる場合は、この手順を繰り返したり、様々なパターンを試すこともできます。いずれのパターンでも、新しいブラウザを起動するか、ブラウザ・キャッシュを消去します。ブラウザではWebページがキャッシュされるため、Default.aspxに再びアクセスした場合にこのWebページが表示されても、それはキャッシュされたものであることがあります。これは意図された動作ではありません。Webページでは、ASP.NETプロバイダ認証プロセスが行われる必要があります。そのためには、新しいブラウザ・インスタンスを使用するか、またはブラウザのキャッシュを消去します。

2. login.aspxから後のURLテキストを削除します。 これによって、URLは、ユーザーが最初にサイトにアクセスしたときの元の状態に戻ります。

ユーザー名Bobを入力し、英数字以外の文字を1文字含む7文字以上のパスワードを入力します。

「Log In」をクリックします。



The screenshot shows a web browser window with the address bar displaying 'http://localhost:1451/WebSite1/login.aspx'. The page content includes a 'Log In' heading, a 'User Name:' label with a text box containing 'Bob', a 'Password:' label with an empty text box, and a checkbox labeled 'Remember me next time.' Below these is a red error message: 'Your login attempt was not successful. Please try again.' At the bottom right is a 'Log In' button.

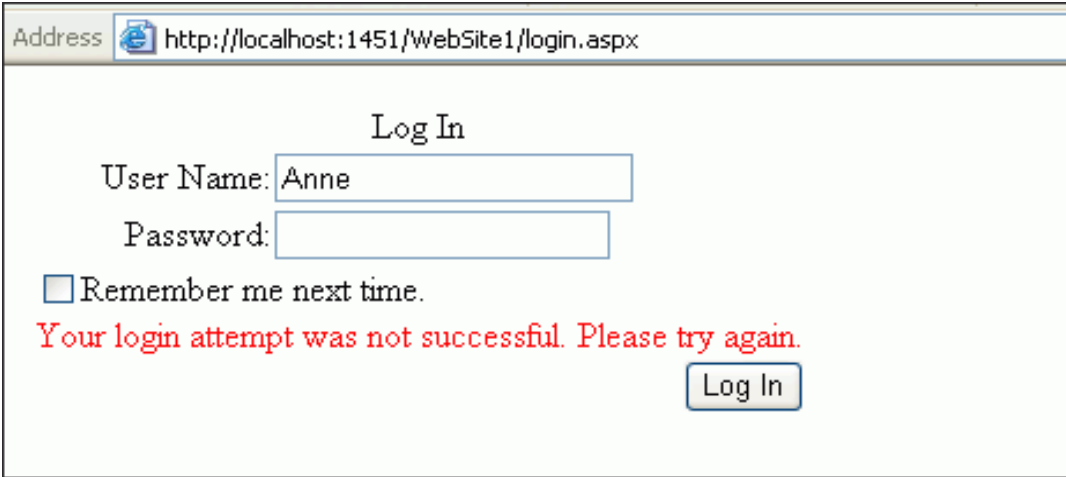
[aspnt35.gifの説明](#)

ページには、「Your login attempt was not successful. Please try again.」というメッセージが再度表示されます。

Bobは認可されたユーザーではありません。 Webサイトは、正確にこのユーザーのアクセスを拒否します。

3. ユーザー名Anneを入力し、そのWebサイト・ユーザーの誤ったパスワードを入力します。

「Log In」をクリックします。



The screenshot shows a web browser window with the address bar displaying 'http://localhost:1451/WebSite1/login.aspx'. The page content includes a 'Log In' heading, a 'User Name:' label with a text box containing 'Anne', a 'Password:' label with an empty text box, and a checkbox labeled 'Remember me next time.' Below these is a red error message: 'Your login attempt was not successful. Please try again.' At the bottom right is a 'Log In' button.

[aspnt38.gifの説明](#)

スクリーンショットに示されているように、このユーザーはアクセスを拒否され、コントロールはOracle Membership Providerに格納されている資格証明でこのユーザーの資格証明を検証できませんでした。

4. Webサイト・ユーザーの正しいパスワードを入力します。

「Log In」をクリックします。

従業員データが表示されます。ここでは、データにアクセスできるのが認可されたユーザーのみであることが示されます。このように、Oracle Providers for ASP.NETでは、非常に簡単な方法でWebサイトにセキュリティが提供されます。

これで、データ駆動型ASP.NET Webアプリケーションの構築が完了しました。このアプリケーションによって、認証が実行され、データベースから従業員データが取得されます。

8 .NETストアド・プロシージャの開発とデプロイ

この章の内容は次のとおりです。

- [.NETストアド・プロシージャの概要](#)
- [共通言語ランタイム・サービスの開始](#)
- [SYSDBAとしての接続の作成](#)
- [Oracleプロジェクトの作成](#)
- [.NETストアド・ファンクションおよびプロシージャの作成](#)
- [.NETストアド・ファンクションおよびプロシージャのデプロイ](#)
- [.NETストアド・ファンクションおよびプロシージャの実行](#)
- [問合せウィンドウでの.NETストアド・プロシージャの実行](#)

.NETストアド・プロシージャの概要

.NETストアド・プロシージャは、SQL文またはPL/SQL文を含む.NET言語で記述されたメソッドまたはプロシージャです。

カスタム・ストアド・プロシージャおよびファンクションは、C#やVB.NETなどの.NETに準拠した任意の言語を使用して記述できます。また、これらの.NETストアド・プロシージャは、他のPL/SQLストアド・プロシージャまたはJavaストアド・プロシージャと同様にデータベースで使用できます。.NETストアド・プロシージャは、PL/SQLパッケージ、プロシージャ、ファンクションおよびトリガーからコールできます。また、SQL文からコールすることも、PL/SQLプロシージャまたはファンクションをコールできる任意の場所からコールすることもできます。

この章の例を実行するには、Oracle Database Extensions for .NET (.NETストアド・プロシージャを記述できるデータベース・オプション) をデータベースにインストールして構成しておく必要があります。

この章では、.NETストアド・プロシージャをアプリケーションで使用およびデプロイする方法を説明します。

共通言語ランタイム・サービスの開始

.NETストアド・プロシージャを使用するには、最初に共通言語ランタイム・エージェント (OraClrAgentサービス) を開始する必要があります。このサービスはデフォルトでは開始されません。このサービスは、クライアント上ではなくOracle Database上にあることに注意してください。

注意:

OraClrAgntは、OracleOracleHomeNameClrAgntという名前で「コントロール パネル」の「サービス」からアクセスできます。OracleHomeNameはOracleホームです。

共通言語ランタイム・サービスを開始するには、次の手順を実行します。

1. 「スタート」メニューから「すべてのプログラム」、「管理ツール」、「サービス」の順に選択します。
2. 「サービス」ウィンドウで「拡張」タブをクリックします。

サービスのリストをスクロールし、「OracleOracleHomeNameClrAgnt」を選択します。

3. 「サービスの開始」ハイパーリンクをクリックします。

「サービス コントロール」ウィンドウに、OracleClrAgentを開始していることが表示されます。

4. 「サービス コントロール」ウィンドウが閉じたら、OracleClrAgentの状態が「開始」に変わっていることを確認します。

SYSDBAとしての接続の作成

次に、SYSDBAとしてデータベース接続を作成する必要があります。これにより、Oracleプロジェクトをデプロイできるようになります。

注意:

このタスクを実行するには、SYSDBAとしての管理者権限が必要です。

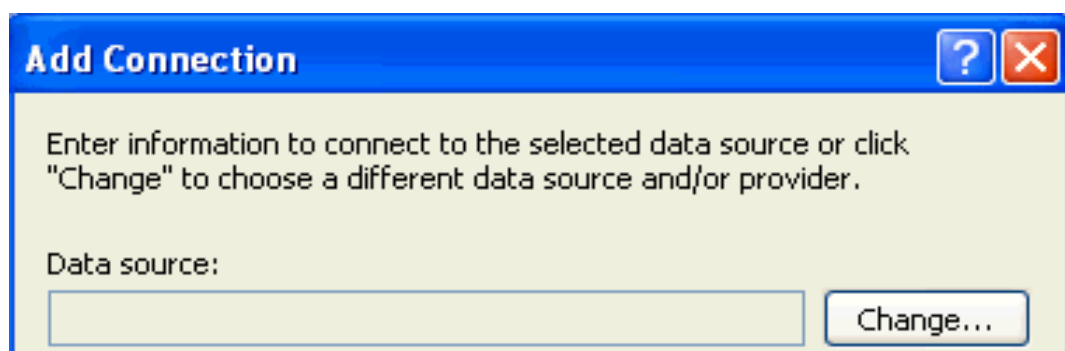
注意:

Enterprise Managerを使用してsysアカウントのパスワードを設定する場合は、『Oracle Database 2日データベース管理者』の管理者アカウントおよび管理者権限に関する項を参照してください。

ODTでデータベース接続を作成するには、次の手順を実行します。

1. 「View」メニューから「Server Explorer」を選択します。
2. Server Explorerで、「Data Connections」を右クリックします。
3. 「Add Connection」を選択します。
4. 「Add Connection」ウィンドウが表示されたら、「Data source」に「Oracle Database (Oracle ODP.NET)」と表示されているかどうかを確認します。

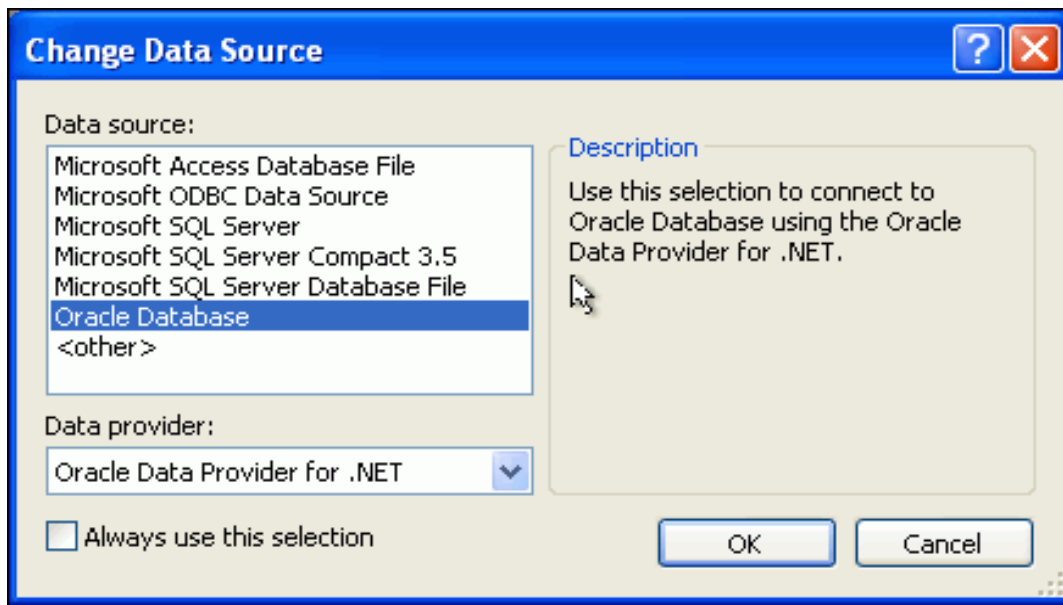
表示されている場合は、手順6に進みます。



[addconnection1a.gifの説明](#)

「Data source」に「Oracle Database (Oracle ODP.NET)」と表示されていない場合は、「Change」を選択します。

「Change Data Source」ウィンドウが表示されます。



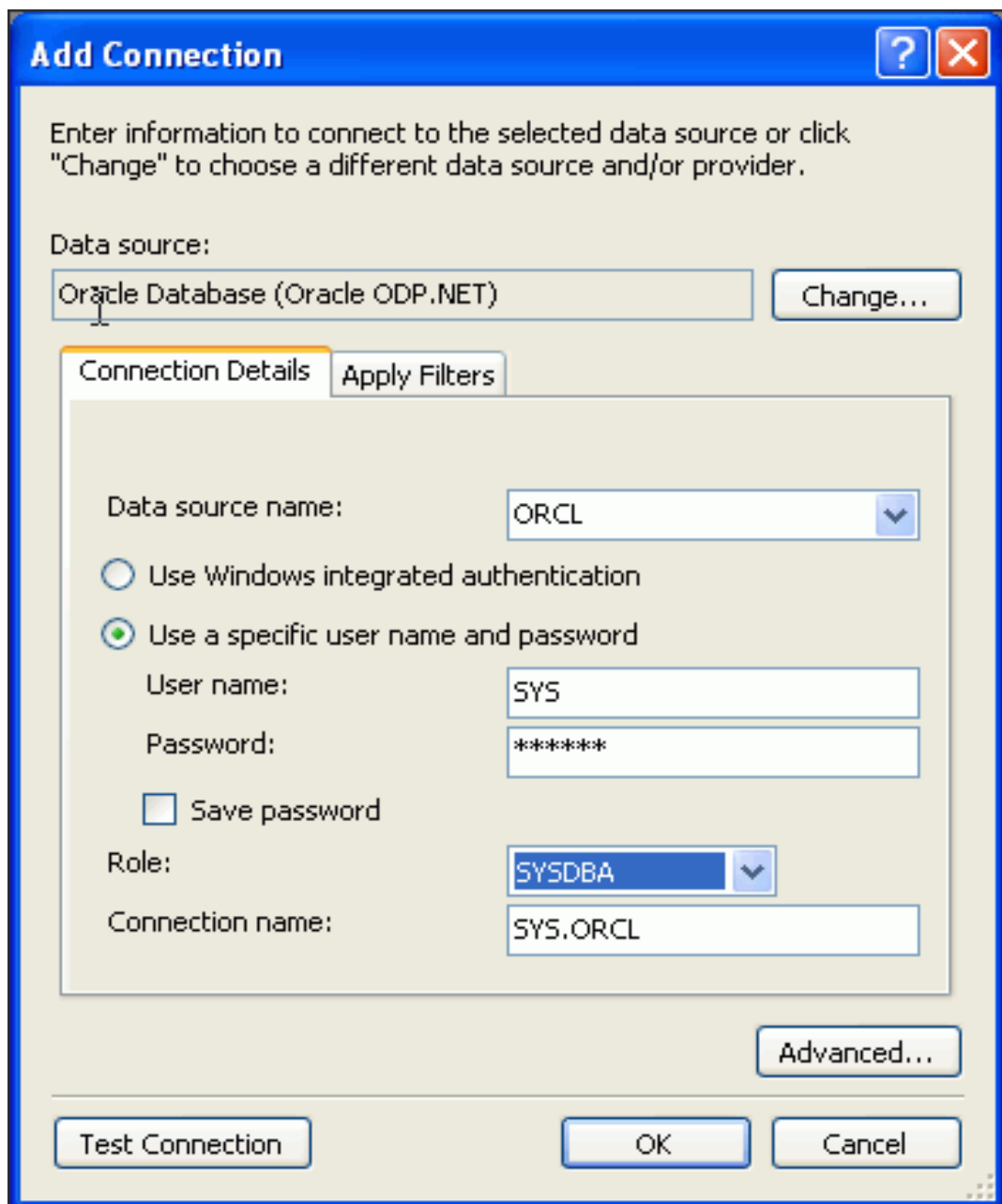
[addconnection2a.gifの説明](#)

5. 「Oracle Database」を選択してから、「Oracle Data Provider for .NET」を選択します。
6. 「Add Connection」ウィンドウでは、次のように指定します。
 - 「User name」にはsysと入力します。
 - 「Password」には、sysアカウントのロック解除および設定を行った管理者により設定されたパスワードを入力します。

Enterprise Managerを使用してsysアカウントのパスワードを設定する場合は、『Oracle Database 2日でデータベース管理者』の管理者アカウントおよび管理者権限に関する項を参照してください。

- 「Role」がSysdbaに設定されていることを確認します。

「Connection name」は「Data source name」および「User name」の値から自動生成されます。



[deploy03.gifの説明](#)

7. 「Add Connection」ウィンドウで、「OK」をクリックします。

これで、「Server Explorer」ウィンドウにSYS.ORCL接続が含まれます。

Oracleプロジェクトの作成

.NETでストアド・プロシージャを使用するには、ストアド・プロシージャを保持するための新しいOracleプロジェクトを最初に作成する必要があります。

.NETストアド・プロシージャ用のプロジェクトを作成するには、次の手順を実行します。

1. 「File」メニューから「New」、「Project」の順に選択します。

「New Project」ダイアログ・ボックスが表示されます。

2. 「Project Types」で、作成するプロジェクトの種類を選択します。

- Visual C#:

「Visual C#」、「Database」の順に選択し、「Templates:」の下で「Oracle Project」を選択します。

「Name:」にHR_DeployStored_CSと入力します。

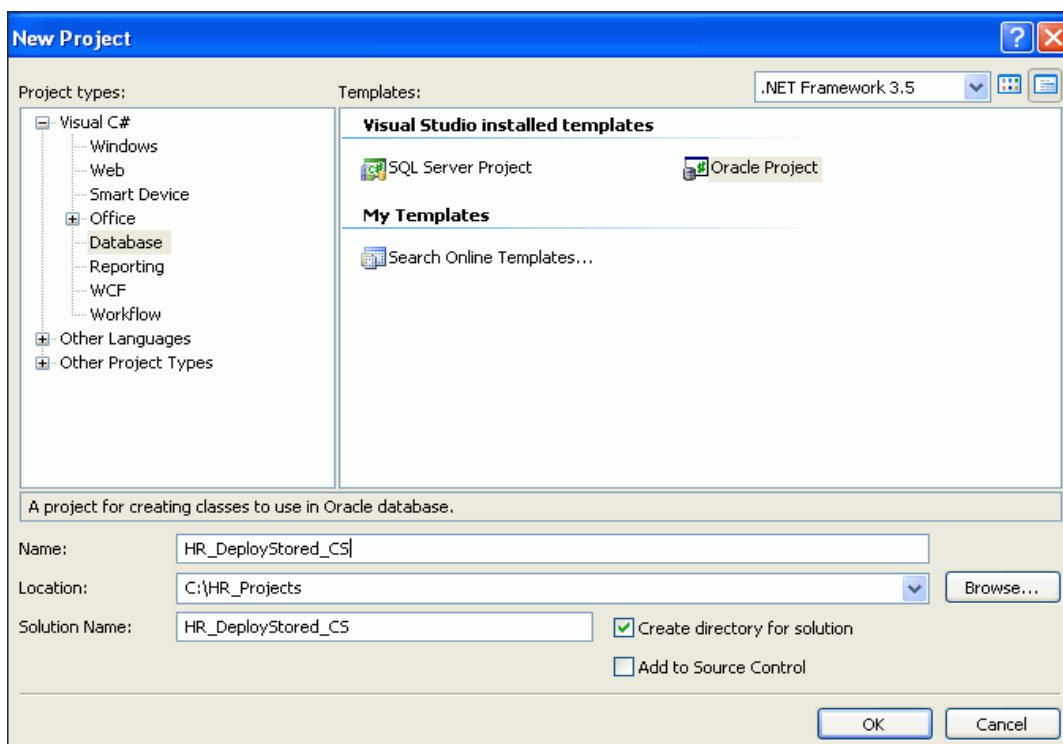
- Visual Basic:

「Other Languages」、「Visual Basic」、「Database」の順に選択し、「Templates:」の下で「Oracle Project」を選択します。

「Name:」にHR_DeployStored_VBと入力します。

3. 「Location:」にC:\HR_Projectsと入力します。

4. 「OK」をクリックします。



[deploy01.gifの説明](#)

.NETストアド・ファンクションおよびプロシージャの作成

これで、.NETストアド・プロシージャを作成する準備ができました。

.NETストアド・プロシージャを作成するには、次の手順を実行します。

1. 「Solution View」で、プロジェクトの「Class1.cs」タブまたは「Class1.vb」タブを選択します。
2. [「名前空間ディレクティブの追加」](#)の説明に従って、言語ごとに次の名前空間ディレクティブを追加します。

Visual C#:

```
using Oracle.DataAccess.Client;  
using Oracle.DataAccess.Types;
```

Visual Basic:

```
Imports Oracle.DataAccess.Client
Imports Oracle.DataAccess.Types
```

3. [「参照の追加」](#)の説明に従って、Oracle.DataAccess.dllに参照を追加します。
4. 次に示すように、getDepartmentno()メソッドをClass1宣言にコピーします。

Visual C#

```
public static int getDepartmentno(int employee_id)
{
    int department_id = 0;

    // Get a connection to the db
    OracleConnection conn = new OracleConnection();
    conn.ConnectionString = "context connection=true";
    conn.Open();

    // Create and execute a command
    OracleCommand cmd = conn.CreateCommand();
    cmd.CommandText = "select department_id from employees where employee_id = :1";
    cmd.Parameters.Add(":1", OracleDbType.Int32, employee_id,
        ParameterDirection.Input);
    OracleDataReader rdr = cmd.ExecuteReader();

    while(rdr.Read())
        department_id=rdr.GetInt32(0);

    rdr.Close();
    cmd.Dispose();

    // Return the employee's department number
    return department_id;
}
```

Visual Basic:

```
Public Shared Function getDepartmentno(ByVal employee_id As Integer) As Integer
    Dim department_id As Integer = 0

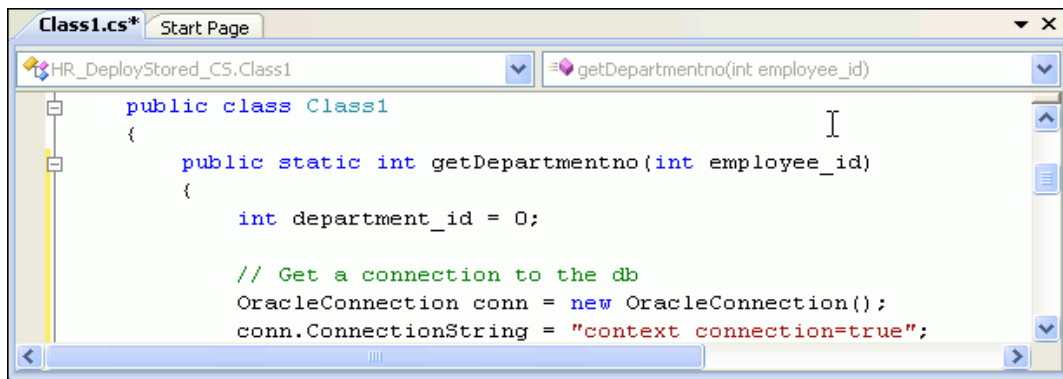
    ' Get a connection to the db
    Dim conn As OracleConnection = New OracleConnection()
    conn.ConnectionString = "context connection=true"
    conn.Open()

    ' Create and execute a command
    Dim cmd As OracleCommand = conn.CreateCommand()
    cmd.CommandText = "select department_id from employees where employee_id = :1"
    cmd.Parameters.Add(":1", OracleDbType.Int32, employee_id, ParameterDirection.Input)
    Dim rdr As OracleDataReader = cmd.ExecuteReader()

    While rdr.Read()
        department_id = rdr.GetInt32(0)
    End While

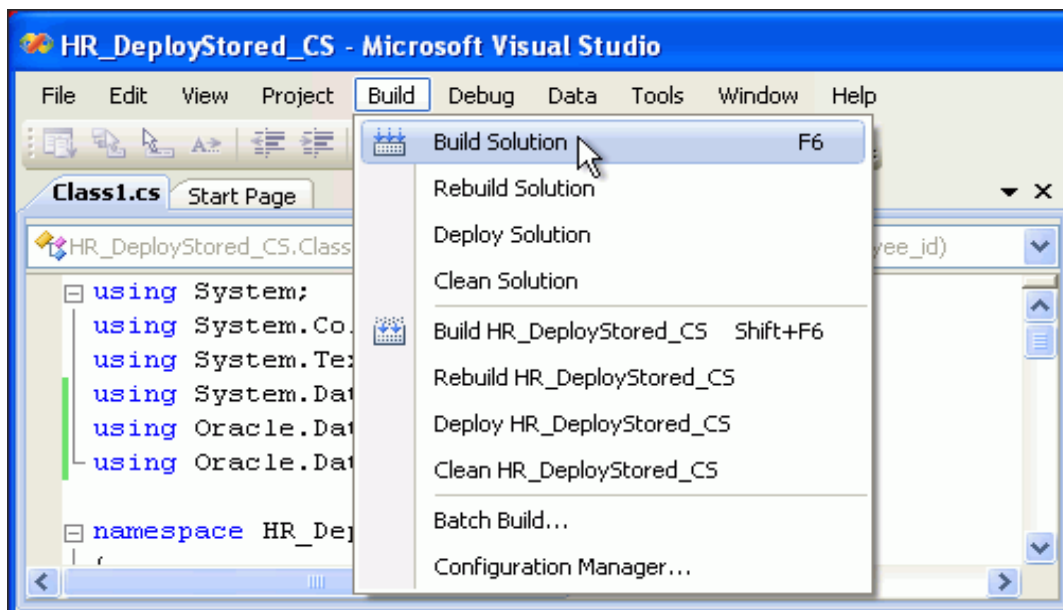
    rdr.Close()
    cmd.Dispose()

    ' Return the employee's department number
    Return department_id
End Function
```

deploy04.gifの説明

5. Class1を保存します。
6. 「Build」メニューから「Build Solution」を選択します。



deploy05.gifの説明

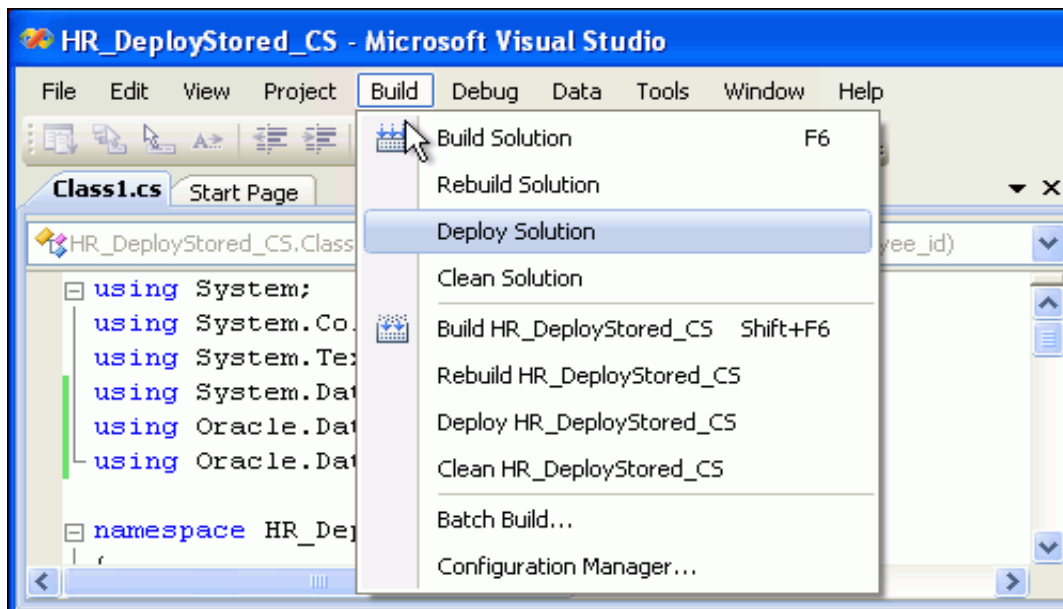
7. 正しくビルドされたことが「Output」ウィンドウに示されていることを確認し、このウィンドウを閉じます。

.NETストアド・ファンクションおよびプロシージャのデプロイ

これまでの手順により、[「.NETストアド・ファンクションおよびプロシージャの作成」](#)で作成した.NETストアド・プロシージャをデプロイできる状態になっています。

.NETストアド・プロシージャをデプロイするには、次の手順を実行します。

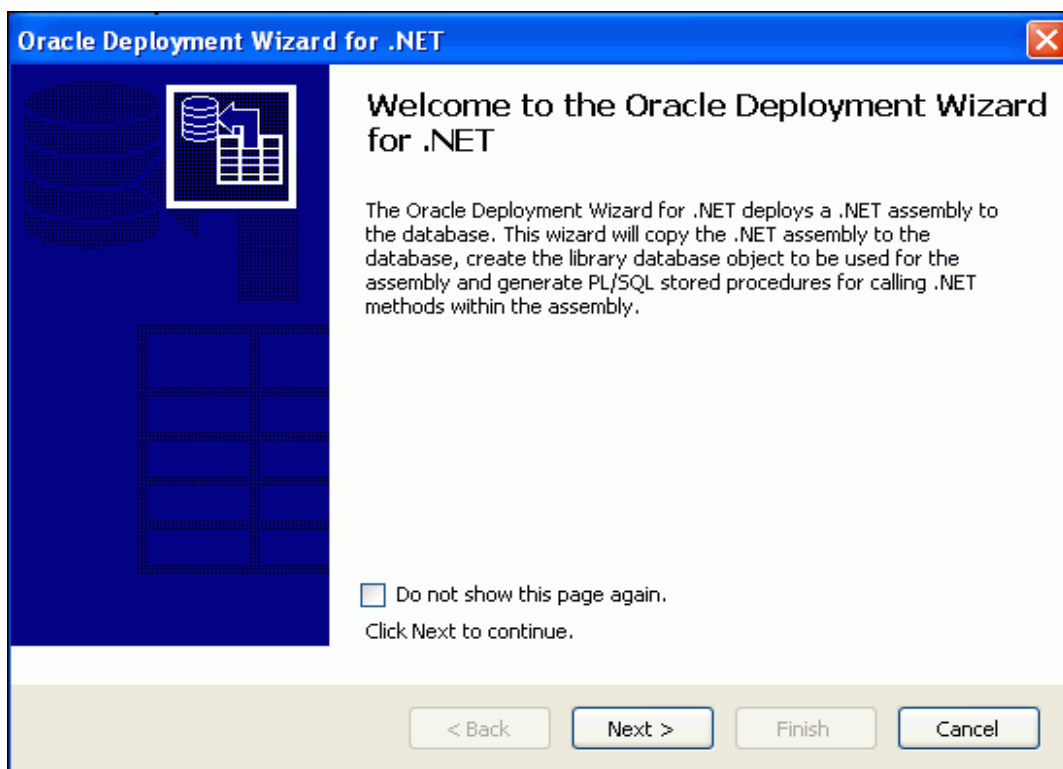
1. 「Build」メニューから「Deploy Solution」を選択します。



[deploy06.gifの説明](#)

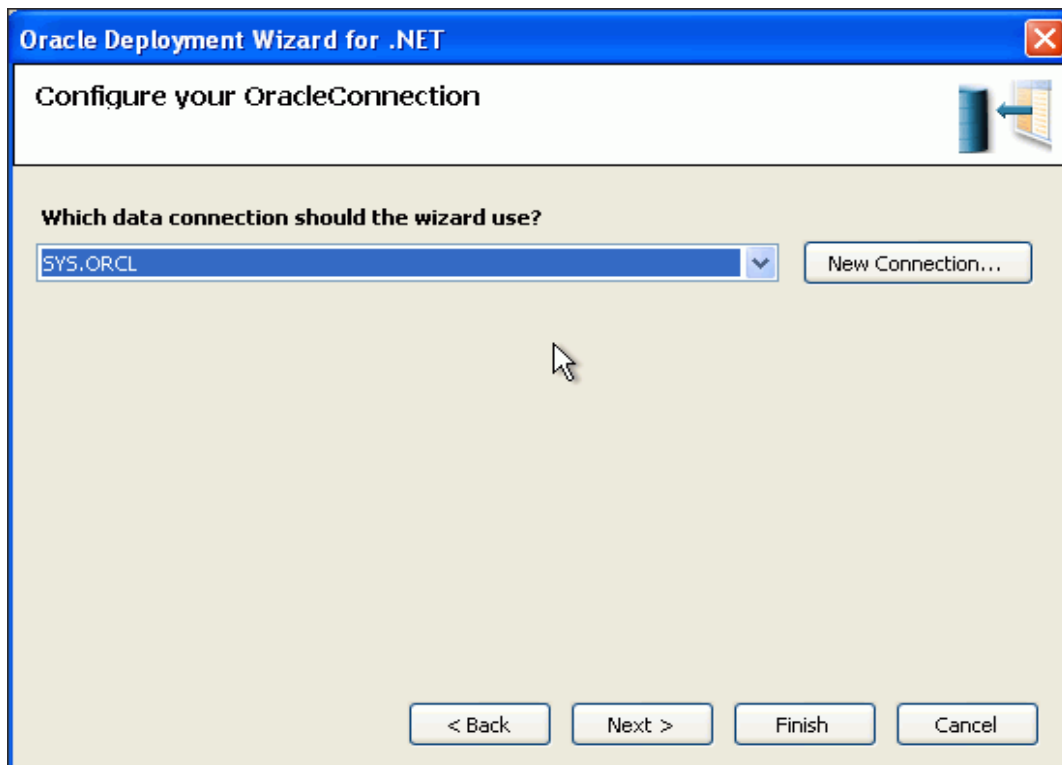
「.NET用のOracleデプロイメント・ウィザード」ウィンドウが表示されます。

2. 「.NET用のOracleデプロイメント・ウィザード」ウィンドウで「次へ」をクリックします。



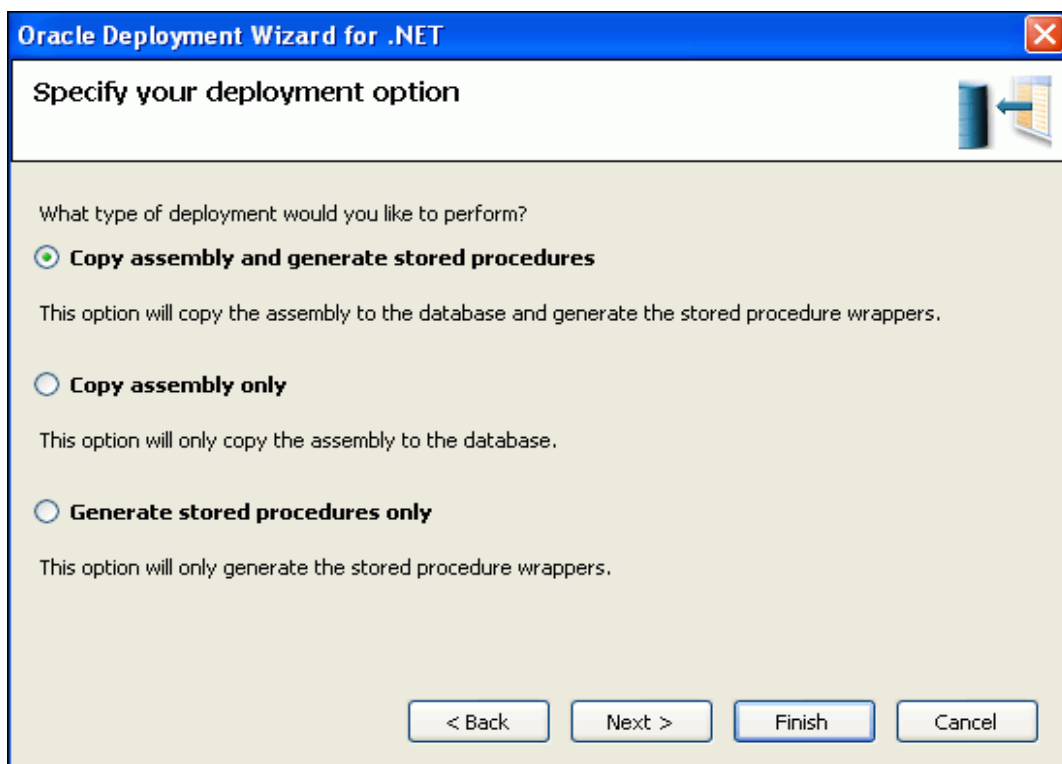
[deploy07.gifの説明](#)

3. 「OracleConnectionを構成します」ウィンドウで「次へ」をクリックします。



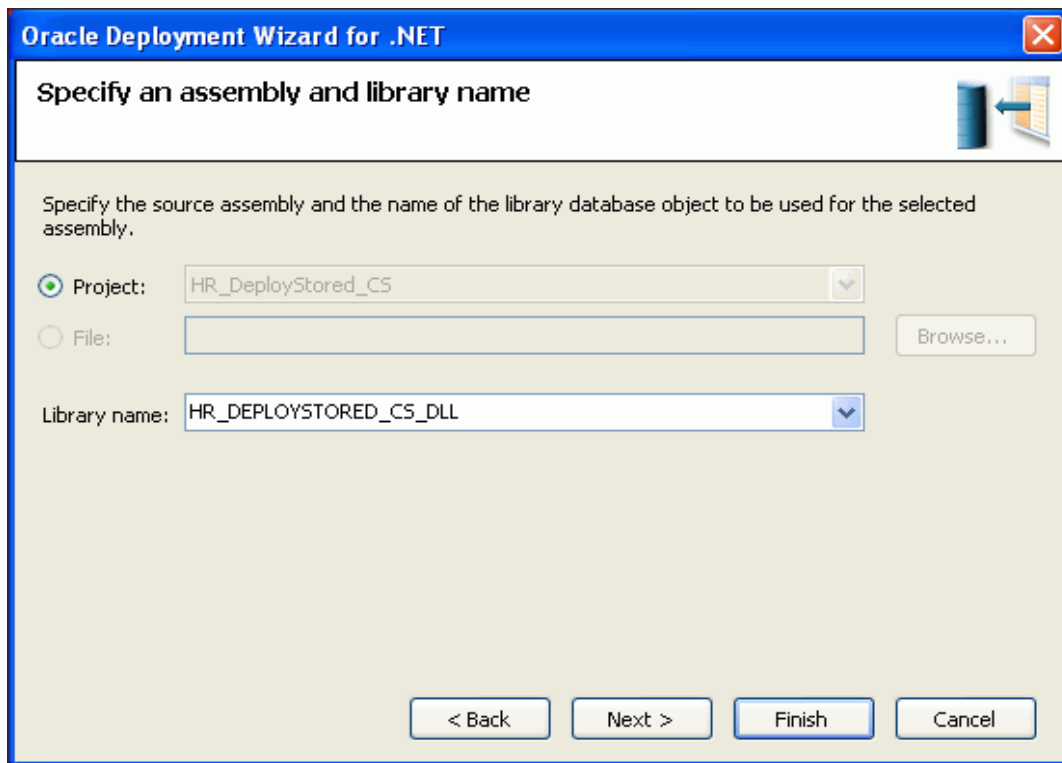
[deploy08.gifの説明](#)

4. 「デプロイメント・オプションを指定します」ウィンドウで、最初のオプション（「アセンブリをコピーしてストアド・プロシージャを生成」）が選択されていることを確認し、「次へ」をクリックします。



[deploy10.gifの説明](#)

5. 「アセンブリおよびライブラリ名を指定します」ウィンドウでデフォルトを受け入れ、「次へ」をクリックします。

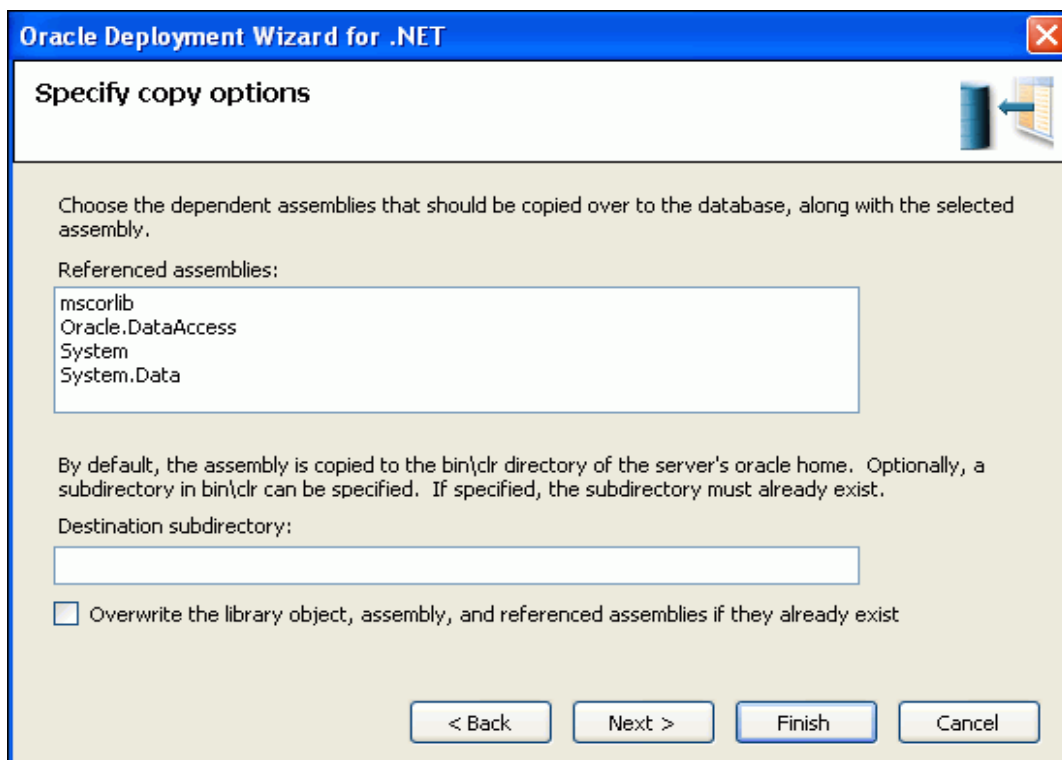


[deploy11.gifの説明](#)

6. 「コピー・オプションの指定」ウィンドウでデフォルトを受け入れ、「次へ」をクリックします。

Visual Basic:

Visual Basicを使用している場合は、Microsoft.VisualBasicアセンブリも参照アセンブリとして表示されます。

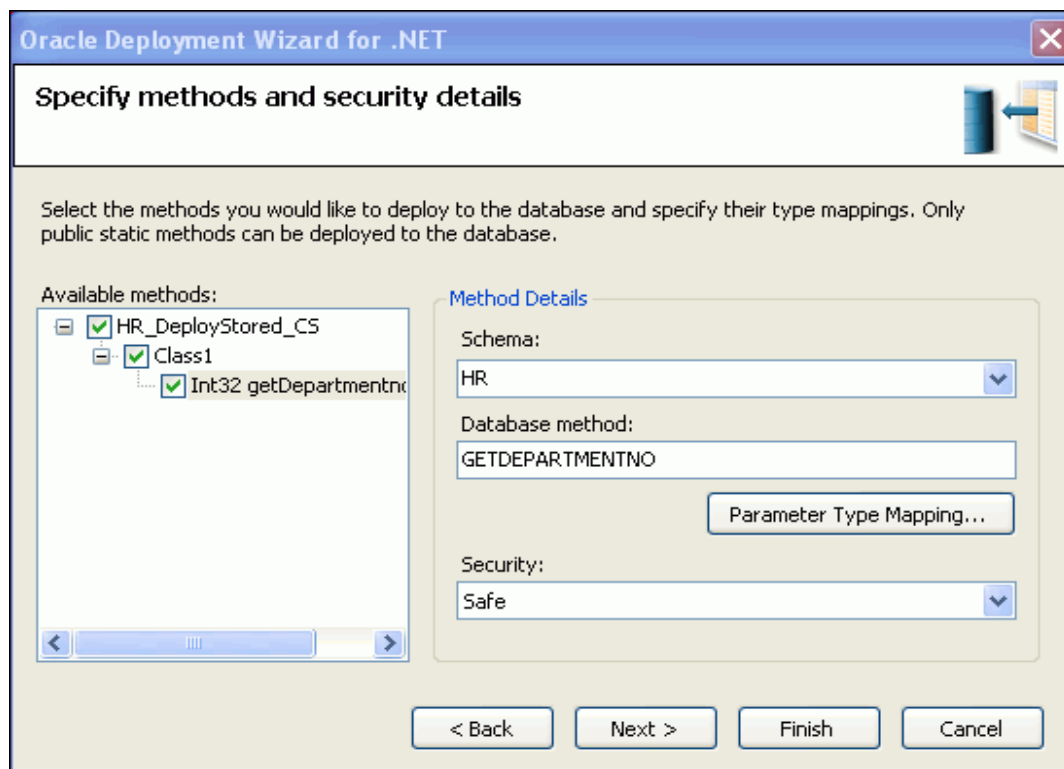


[deploy12.gifの説明](#)

7. 「メソッドおよびセキュリティ詳細の指定」ウィンドウで、「使用可能なメソッド」にある「HR_DeployStored_CS」または「HR_DeployStored_VB」を開き、「Class1」を開いて「getDepartmentno()」メソッドを選択します。

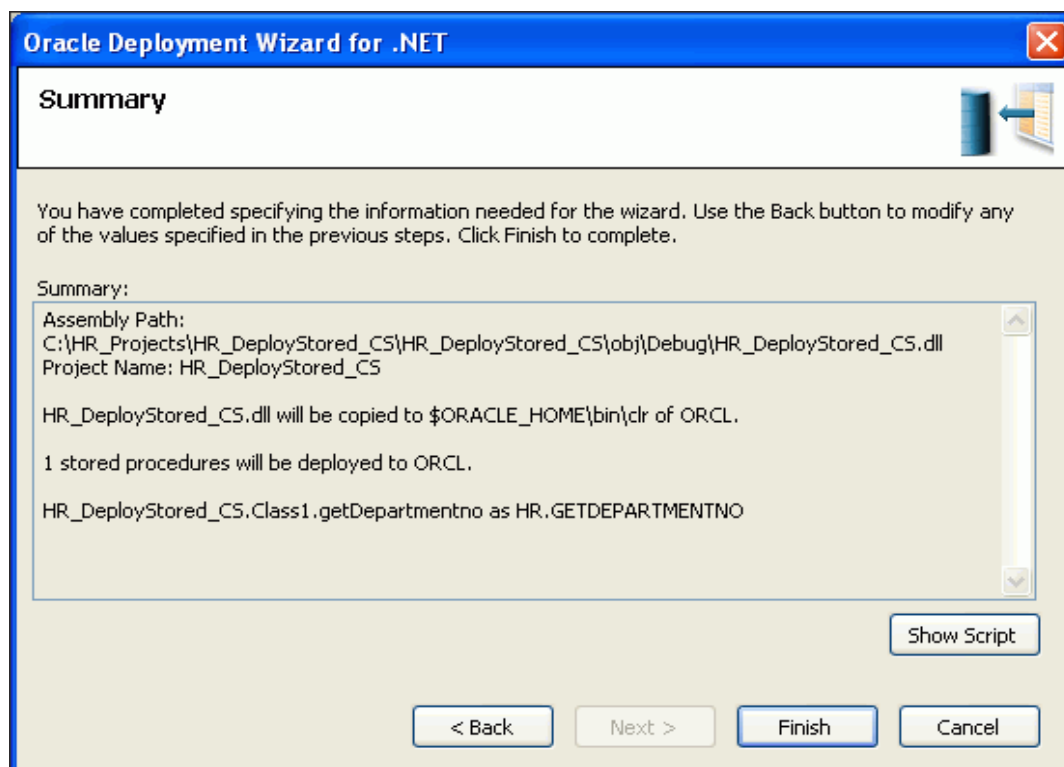
「メソッドの詳細」で、スキーマ・リストから「HR」を選択します。

「次へ」をクリックします。



deploy13.gifの説明

8. 「サマリー」ウィンドウで「終了」をクリックします。



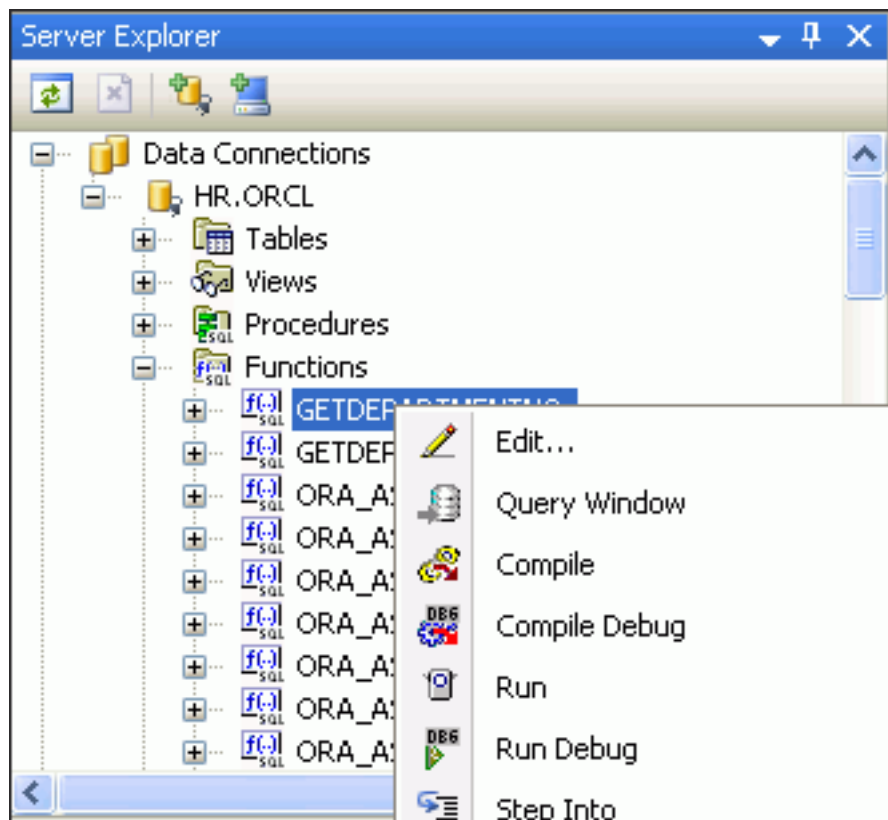
deploy14.gifの説明

.NETストアド・ファンクションおよびプロシージャの実行

これで、前の項でデプロイした.NETストアド・プロシージャを実行する準備ができました。

.NETストアド・プロシージャを実行するには、次の手順を実行します。

1. Server Explorerで、HR.ORCL接続をオープンします。「Functions」を開きます。「GETDEPARTMENTNO」を右クリックして「Run」を選択します。

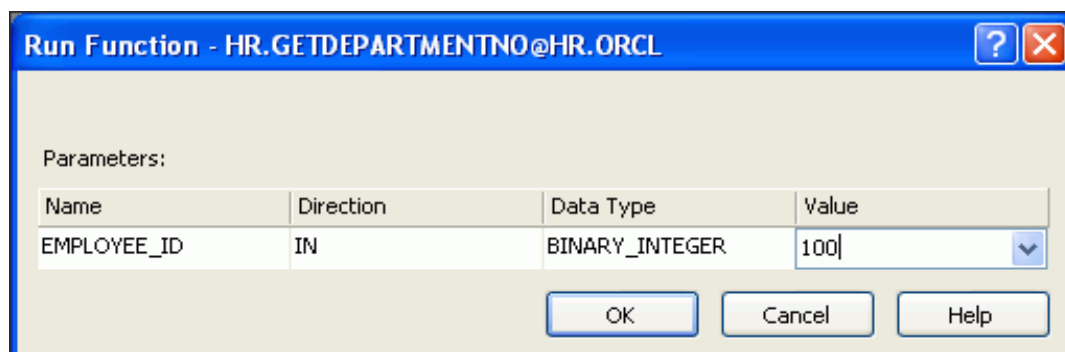


[deploy15.gifの説明](#)

「Run Function」ウィンドウが表示されます。

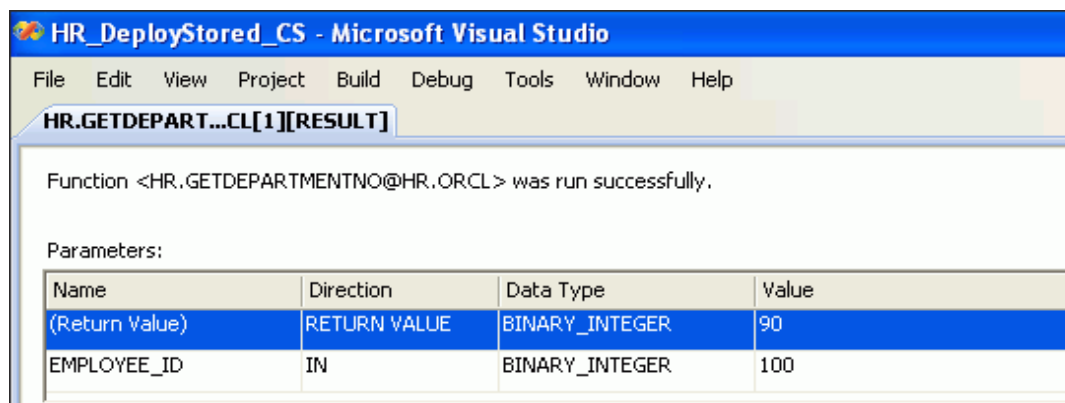
2. 「Run Function」ウィンドウで、「EMPLOYEE_ID」の「Value」に100を入力します。

「OK」をクリックします。



[deploy16.gifの説明](#)

3. 部門の戻り値が90であることを確認します。これは、EMPLOYEE_ID 100は部門90に含まれると示します。

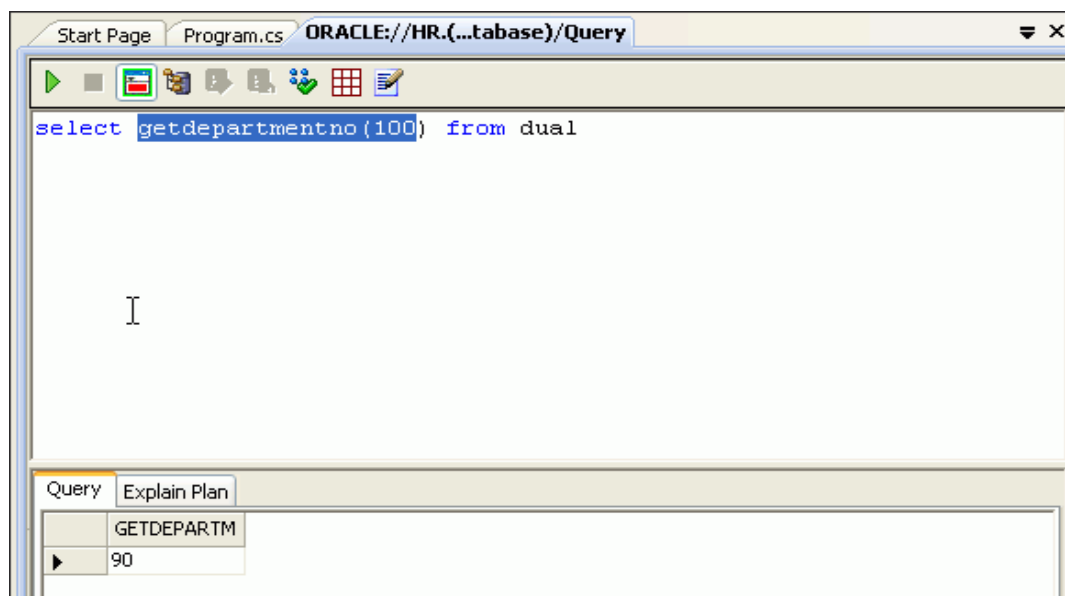


[deploy17.gifの説明](#)

問合せウィンドウでの.NETストアド・プロシージャの実行

作成したばかりの.NETストアド・プロシージャは、Server Explorerから実行する他に、ODT問合せウィンドウを使用して実行することもできます。

1. Server ExplorerでHR.ORCLスキーマを開きます。
2. 「Functions」を開き、「GETDEPARTMENTNO」を選択します。
3. 右クリックして「Query Window」を選択します。
4. `Select getdepartmentno(100) from dual`と入力します。
5. ツールバーから「Execute」をクリックします。



[querywindow.gifの説明](#)

9 グローバリゼーション・サポートの組み込み

この章の内容は次のとおりです。

- [グローバル・アプリケーションの概要](#)
- [.NET Frameworkを使用したグローバル・アプリケーションの開発](#)
- [ユーザーの正しいローカル規則でのデータ表示](#)
- [.NETとOracle Databaseのロケール環境の同期化](#)
- [Oracle Data Provider for .NETでのクライアント・グローバリゼーションのサポート](#)

参照:

- 『Oracle Data Provider for .NET開発者ガイド』の第8章「Oracle Data Provider for .NETグローバリゼーション・クラス」
- 『Oracle Database 2日で開発者ガイド』のグローバル環境での作業に関する項
- Microsoft社の.NET国際化対応に関するインターネット・サイト (<http://msdn.microsoft.com/en-us/goglobal/bb688096.aspx>)

グローバル・アプリケーションの概要

この章では、.NETでのOracle Databaseを使用するグローバル・アプリケーションの開発について説明します。ここでは、ロケール認識の開発やユーザーのロケールによる文化的な表記規則を使用したデータ表示など、グローバルにデプロイできるアプリケーションを開発するための基本的なタスクを扱います。また、Oracle Data Provider for .NETで使用できるグローバリゼーション・サポート機能についても説明します。

様々なロケールをサポートするグローバル対応アプリケーションを作成するには、正しい開発手順が必要です。

ロケールとは、各国語およびその言語が話されている地域を意味します。アプリケーション自体で、ユーザーのロケール・プリファレンスを認識し、ユーザーが期待する文化的な表記規則に従ってコンテンツを表示する必要があります。正しい日付書式や数値書式など、該当するロケールの特性を使用してデータを表示することが重要です。Oracle Databaseは完全に国際化されており、グローバル・アプリケーションの開発およびデプロイに対応したグローバル・プラットフォームを提供しています。

.NET Frameworkを使用したグローバル・アプリケーションの開発

グローバル対応アプリケーションを計画する際には、次の2つの主なタスクを考える必要があります。

- **グローバリゼーション**は、様々な文化に適応できるアプリケーションを設計するプロセスです。
- **ローカリゼーション**は、特定の文化に合わせてリソースを翻訳するプロセスです。

.NET FrameworkのSystem.Globalization名前空間には、言語、国および地域、カレンダー、日付の書式パターン、通貨、数値および文字列のソート順序など、文化に関連する情報を定義するクラスが含まれています。これらのクラスによりグローバル対応アプリケーションの開発プロセスが簡略化されます。そのため、ユーザーの文化を表すCultureInfoオブジェクトをSystem.Globalization名前空間に含まれるメソッドに渡すと、一連の正しいルールとデータが適用されます。

.NET Frameworkではリソースの作成とローカライゼーションもサポートされており、リソースをパッケージ化したりデプロイするためのモデルが提供されています。アプリケーションのリソースを特定の文化用にローカライズすることで、翻訳版のアプリケーションの開発がサポートされます。 .NET Frameworkの基本クラス・ライブラリでは、アプリケーション・リソースを構築および操作するための様々なクラスをSystem.Resources名前空間で提供しています。

ユーザーの正しいローカル規則でのデータ表示

アプリケーションのデータはユーザーが期待するとおりに表示される必要があり、そうでない場合は意味が誤って解釈される可能性があります。たとえば、12/11/05は、アメリカ合衆国では2005年12月11日を意味しますが、イギリスでは2005年11月12日を意味します。数値書式と通貨書式にも同様の違いがあります。たとえば、ピリオド (.) は、アメリカ合衆国では小数点セパレータですが、欧州では千単位のセパレータです。

各言語には、それぞれ独自のソート規則があります。アルファベットの文字の順序に従う言語や、文字の画数に従う言語、単語の発音に従う言語などがあります。ユーザーが慣れている言語順序でソートされていないデータを表示すると、情報の検索が難しくなり、時間がかかることがあります。

アプリケーション・ロジックおよびデータベースから取得されるデータの量によっては、アプリケーション・レベルよりもデータベース・レベルでデータを書式設定した方が適切な場合があります。 Oracle Databaseには、ユーザーのローカル・プリファレンスがわかっている場合に、データの表現を調整する機能が多く用意されています。

SQL*Plusへの接続

次の例のいくつかでは、SQL*Plusを使用して、SYSやSYSTEMなどのデータベース管理者権限を持つユーザーとして接続する必要があります。

参照:

詳細は、『Oracle Database 2日 でデータベース管理者』のユーザー・アカウントのロックおよびロック解除に関する項を参照してください。

Oracleの日付書式の使用

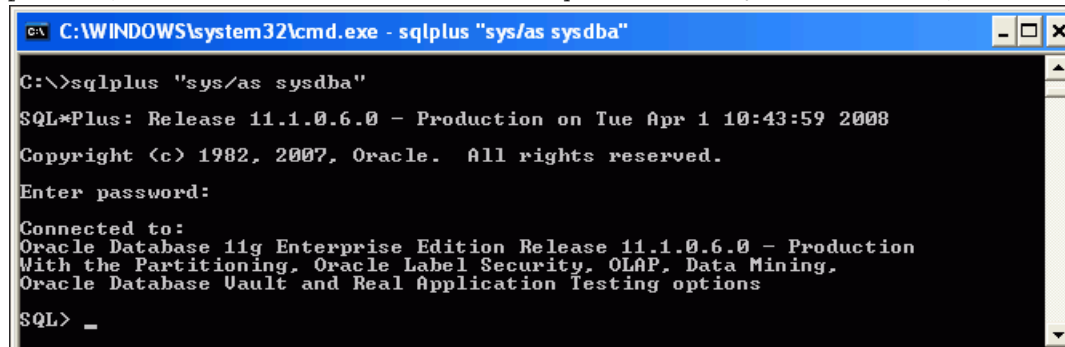
Oracle Databaseの日付表示書式には、標準の書式、短い書式、長い書式の3種類があります。次の手順では、アメリカ合衆国とドイツでの短い日付書式と長い日付書式の違いを示します。

Oracleの日付書式を変更するには、次の手順を実行します。

1. Windowsのコマンド・プロンプトから次のコマンドを入力します。

```
C:\>sqlplus "sys as sysdba"
Enter password:passwd
```

passwdは、データベースのインストール時に設定したSysパスワードです。文字を入力しても、パスワードは表示されません。



[nls00.gifの説明](#)

2. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_TERRITORY=america NLS_LANGUAGE=american;
```

「セッションが変更されました。」というメッセージが表示されます。

パラメータを現行の設定に設定しても問題はありません。セキュリティ上必要な場合に行うことはできます。現行の設定を確認するには、次のように入力します。

```
SQL> select * from v$nls_parameters;
```

または

```
select * from v$nls_parameters where parameter = 'NLS_LANGUAGE';
```

3. SQLプロンプトで次の問合せを入力します。

```
SQL> SELECT employee_id "ID",  
SUBSTR (first_name,1,1)||'. '||last_name "Name",  
TO_CHAR (hire_date, 'DS') "Short Hire",  
TO_CHAR (hire_date, 'DL') "Long Hire Date"  
FROM hr.employees  
WHERE employee_id < 105;
```

現在はhrではなくsysでログインしているため、hrスキーマのemployees表にアクセスするには、hr.employeesを使用する必要があります。ことに注意してください。

問合せ結果は、手順1で指定した米語書式で戻されます。

[nls01.gifの説明](#)

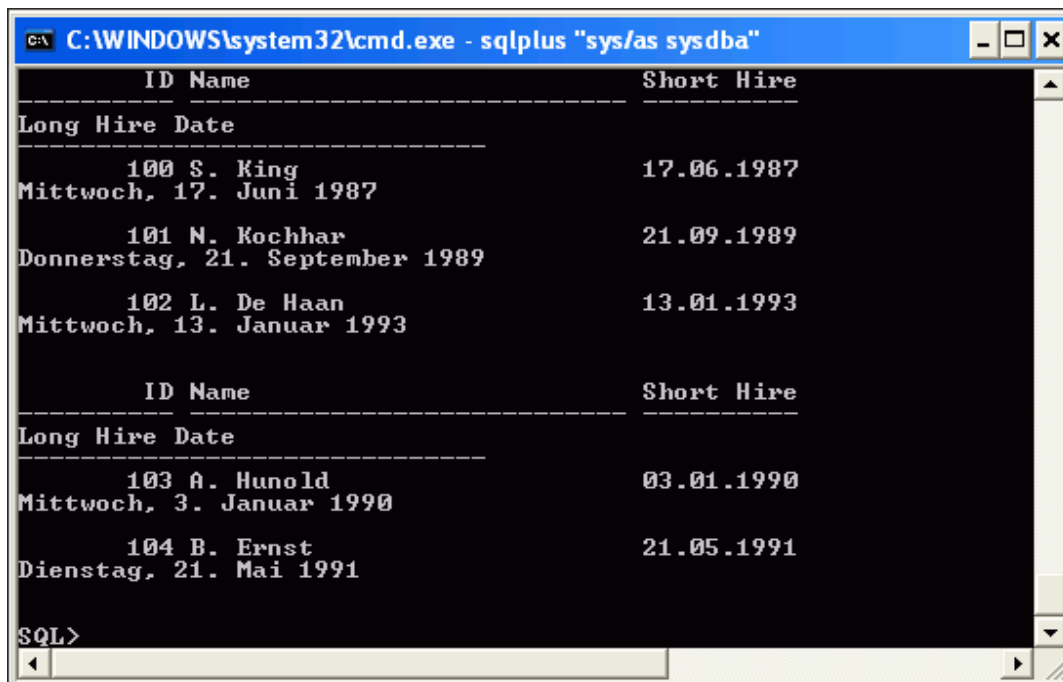
4. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_TERRITORY=germany NLS_LANGUAGE=german;
```

「セッションが変更されました。」というメッセージが表示されます。

5. SQLプロンプトで、手順3の問合せを入力します。

問合せ結果は、手順4で指定したドイツ語書式で戻されます。



[nls02.gifの説明](#)

Oracleの数値書式の使用

小数点文字およびグループ・セパレータにも違いがあります。次の手順では、アメリカ合衆国とドイツでのこれらの違いを示します。

Oracleの数値書式を変更するには、次の手順を実行します。

1. SQLプロンプトで次のコマンドを入力します。

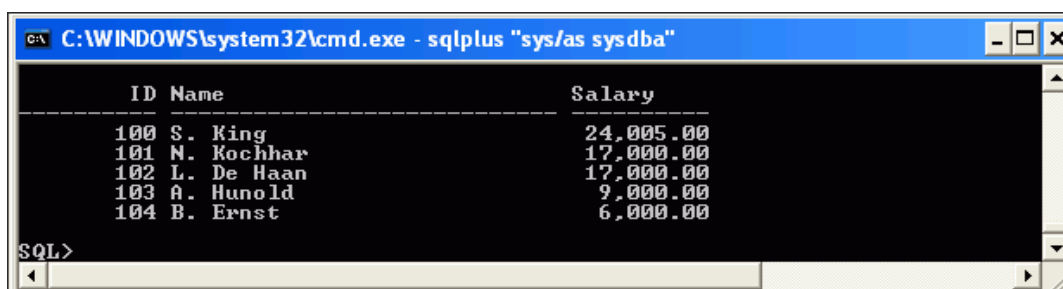
```
SQL> ALTER SESSION SET NLS_TERRITORY=america NLS_LANGUAGE=american;
```

「セッションが変更されました。」というメッセージが表示されます。

2. SQLプロンプトで次の問合せを入力します。

```
SQL> SELECT employee_id "ID",
SUBSTR (first_name,1,1)||'. '||last_name "Name",
TO_CHAR (salary, '99G999D99') "Salary"
FROM hr.employees
WHERE employee_id < 105;
```

問合せ結果は、手順1で指定した米語書式で戻されます。



[nls03.gifの説明](#)

3. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_TERRITORY=germany;
```

「セッションが変更されました。」というメッセージが表示されます。

4. SQLプロンプトで、手順2の問合せを入力します。

問合せ結果は、手順3で指定したドイツ語書式で戻されます。

[nls04.gifの説明](#)

Oracleの言語ソートの使用

スペインでは、伝統的にch、llおよびñが独自の文字として扱われ、それぞれc、lおよびnの後に順序付けられています。次の手順では、Chen、ChungおよびColmenaresという従業員名にスペイン語のソートを使用した場合の結果を説明します。

Oracleの言語ソートを変更するには、次の手順を実行します。

1. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_SORT=binary;
```

「セッションが変更されました。」というメッセージが表示されます。

2. SQLプロンプトで次の問合せを入力します。

```
SQL> SELECT employee_id "ID",  
       last_name "Name"  
FROM hr.employees  
WHERE last_name LIKE 'C%'  
ORDER BY last_name;
```

問合せ結果は、手順1で指定したバイナリ・ソート順で戻されます。

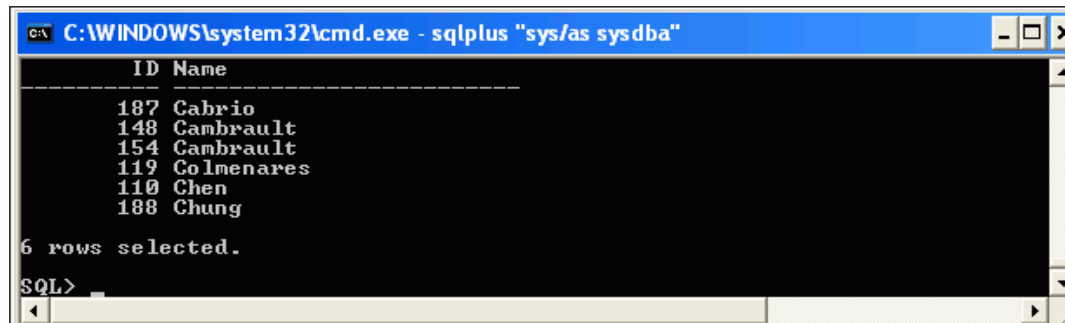
[nls05.gifの説明](#)

3. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_SORT=spanish_m;
```

「セッションが変更されました。」というメッセージが表示されます。

4. SQLプロンプトで、手順2の問合せを入力します。
5. 問合せ結果は、手順3で指定したスペイン語のソート順で戻されます。



[nls06.gifの説明](#)

Oracleのエラー・メッセージ

NLS_LANGUAGEパラメータでは、データベースのエラー・メッセージの言語を制御することもできます。次の手順で示すとおり、SQL問合せを実行する前にこのパラメータを設定することで、ローカル言語固有のエラー・メッセージが戻されるようになります。

OracleのNLS言語パラメータを変更するには、次の手順を実行します。

1. SQLプロンプトで次のコマンドを入力します。

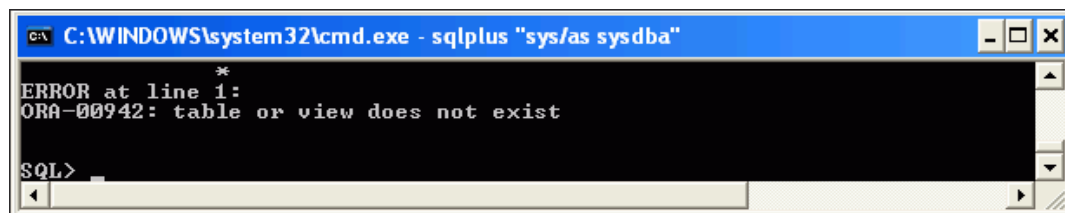
```
SQL> ALTER SESSION SET NLS_LANGUAGE=american;
```

「セッションが変更されました。」というメッセージが表示されます。

2. SQLプロンプトで次の問合せを入力します。

```
SQL> SELECT * FROM managers;
```

問合せ結果として、手順1で指定した言語でエラー・メッセージが戻されます。



[nls07.gifの説明](#)

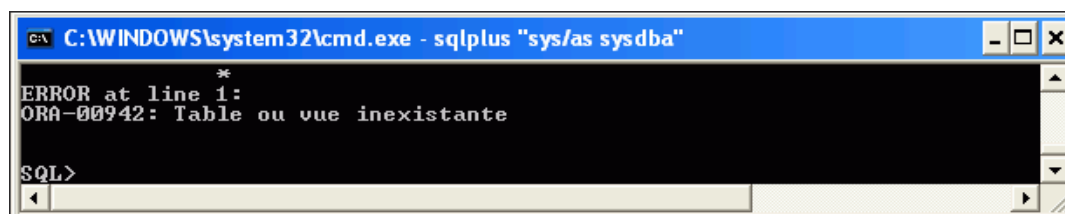
3. SQLプロンプトで次のコマンドを入力します。

```
SQL> ALTER SESSION SET NLS_LANGUAGE=french;
```

「セッションが変更されました。」というメッセージが表示されます。

- SQLプロンプトで、手順2の問合せを入力します。

問合せ結果として、手順3で指定した言語でエラー・メッセージが戻されます。



[nls08.gifの説明](#)

- 言語、ローカルおよびソート設定をリセットし、元の値に戻します。

.NETとOracle Databaseのロケール環境の同期化

グローバル・アプリケーションを開発するときは、データベースとクライアントのユーザー・ロケール設定を常に同期させます。同期させない場合は、文化によって異なる情報が矛盾した状態でアプリケーションから表示されることがあります。たとえば、.NETアプリケーションでは、SQL操作を実行する前に、アプリケーション・ユーザーのCulture IDを正しいNLS_LANGUAGEパラメータ値およびNLS_TERRITORYパラメータ値にマップする必要があります。

[表9-1](#)に、一般的に使用されるロケールの.NET環境とOracle環境での定義を示します。

表9-1 一般的なNLS_LANGUAGEおよびNLS_TERRITORYパラメータ

Culture	Culture ID	NLS_LANGUAGE	NLS_TERRITORY
中国語（中華人民共和国）	zh-CN	SIMPLIFIED CHINESE	CHINA
中国語（台湾）	zh-TW	TRADITIONAL CHINESE	TAIWAN
英語（アメリカ合衆国）	en-US	AMERICAN	AMERICA
英語（イギリス）	en-GB	ENGLISH	UNITED KINGDOM
フランス語（カナダ）	fr-CA	CANADIAN FRENCH	CANADA
フランス語（フランス）	fr-FR	FRENCH	FRANCE
ドイツ語	de	GERMAN	GERMANY
イタリア語	it	ITALIAN	ITALY
日本語	ja	JAPANESE	JAPAN
韓国語	ko	KOREAN	KOREA

ポルトガル語（ブラジル）	pt-BR	BRAZILIAN PORTUGUESE	BRAZIL
ポルトガル語	pt	PORTUGUESE	PORTUGAL
スペイン語	es	SPANISH	SPAIN

Oracle Data Provider for .NETでのクライアント・グローバリゼーションのサポート

Oracle Data Provider for .NETを使用すると、Oracle Databaseで定義されている文化によって異なる表記規則を使用して、適切な文字列書式、日付、時間、金額、数値、ソート順序を保持したり、カレンダーをサポートするなど、文化によって異なるデータをアプリケーションで操作できます。デフォルトのグローバリゼーション設定は、クライアントのNLS_LANGパラメータによって決まります。このパラメータは、ローカル・コンピュータのWindowsレジストリで定義されます。OracleConnection.Openメソッドにより接続が確立されるときに、NLS_LANGパラメータの値で指定されたグローバリゼーション・パラメータを使用してセッションが暗黙的にオープンされます。

クライアント・グローバリゼーション設定

クライアント・グローバリゼーション・パラメータ設定は読み取り専用であり、アプリケーションの存続期間中は変わりません。OracleGlobalizationオブジェクト・プロパティを変更しても、セッションまたはスレッドのグローバリゼーション設定は変更されません。次の項では、セッションおよびスレッド・レベルでグローバリゼーション設定を変更する方法を説明します。

.NETアプリケーションでは、OracleGlobalization.GetClientInfo()静的メソッドをコールすることでグローバリゼーション設定を取得できます。次のOracleGlobalizationのサンプル・コードは、.NETの値の一部を取得する方法を示しています。

Visual C#:

```
using System;
using Oracle.DataAccess.Client;

class ClientGlobalizationSample
{
    static void Main()
    {
        OracleGlobalization ClientGlob = OracleGlobalization.GetClientInfo();
        Console.WriteLine("Client machine language: " + ClientGlob.Language);
        Console.WriteLine("Client charset: " + ClientGlob.ClientCharacterSet);
    }
}
```

Visual Basic:

```
Imports System
Imports Oracle.DataAccess.Client

Class ClientGlobalizationSample
    Shared Sub Main()
        Dim ClientGlob As OracleGlobalization = OracleGlobalization.GetClientInfo()
        Console.WriteLine("Client machine language: " + ClientGlob.Language)
        Console.WriteLine("Client charset: " + ClientGlob.ClientCharacterSet)
    End Sub
End Class
```

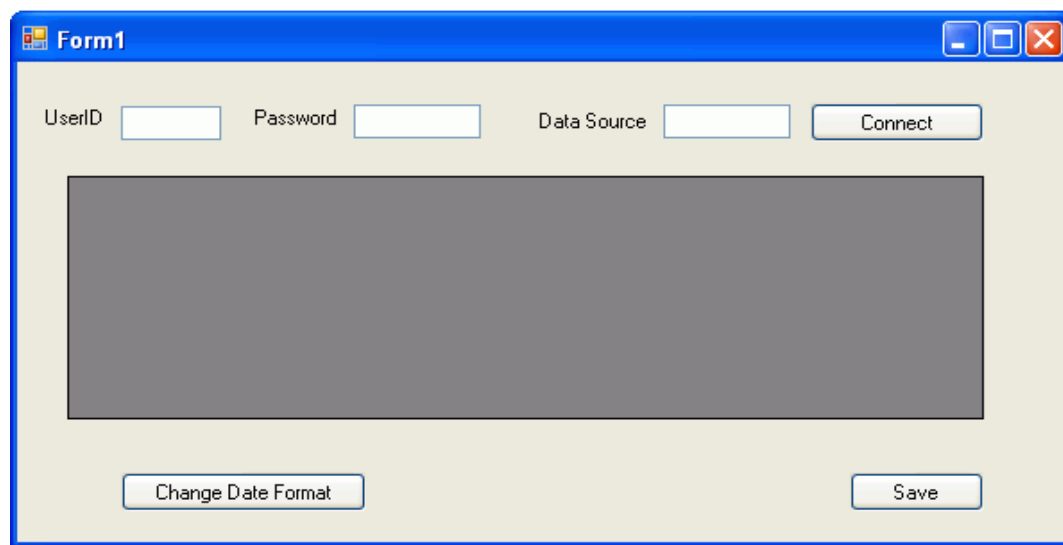
セッション・グローバリゼーション設定の使用

セッション・グローバリゼーション・パラメータは、最初はクライアント・グローバリゼーション設定と同じですが、変更が可能です。セッション・パラメータを変更するには、データベース接続を確立する必要があります。その後、OracleConnectionオブジェクトのGetSessionInfo()メソッドをコールしてセッション・グローバリゼーション設定を取得します。次に、必要に応じてグローバリゼーション設定を変更し、SetSessionInfo(OracleGlobalization)メソッドを使用してOracleConnectionオブジェクトに設定を保存します。

グローバル化・セッション設定を指定するには、次の手順を実行します。

1. アプリケーションHR_Connect_CSまたはHR_Connect_VBを開きます。
2. [付録B「フォームのコピー」](#)の手順に従って、[第4章](#)の最後で完成させたForm3.xxのコピーを作成し、Form5.xxという名前を付けます。
3. プロジェクトのForm1を開き、設計ビューに切り替えます。
4. 「View」メニューから「Toolbox」を選択します。
5. 「Toolbox」の「Windows Forms」の下にある「Button」をForm1にドラッグ・アンド・ドロップします。
6. 新しい「Button」を右クリックして「Properties」を選択します。「Properties」ウィンドウが表示されます。
7. 「Properties」ウィンドウで、次のプロパティを設定します。
 - 「Appearance」の下で、「Text」をChange Date Formatに変更します。
 - 「Design」の下で、「(Name)」をdate_changeに変更します。

Form1の外観は次のようになります。



[nls09.gifの説明](#)

「Properties」ウィンドウで「Events」（稲妻のアイコン）をクリックすると、date_change_Click()が日付ボタンの「Event」として表示されます。

8. 作成したばかりの新しいdate_change_Click()メソッドを開き、次のコードを追加して日付書式を標準のDD-MON-RRからYYYY-MM-DDに変更し、DataSetを更新します。

Visual C#:

```
si.DateFormat = "YYYY-MM-DD";  
conn.SetSessionInfo(si);  
  
ds.Clear();  
da.Fill(ds);  
departments.DataSource = ds.Tables[0];
```


Visual Basic:

```
si.DateFormat = "YYYY-MM-DD"  
conn.SetSessionInfo(si)  
  
ds.Clear()  
da.Fill(ds)  
departments.DataSource = ds.Tables(0)
```

ds.Clear()をコールすると、変更したデータがポストされる前に古い結果が消去されることに注意してください。

また、siクラス変数の宣言およびセッション・グローバリゼーション情報の取得は、手順10および手順11で行います。

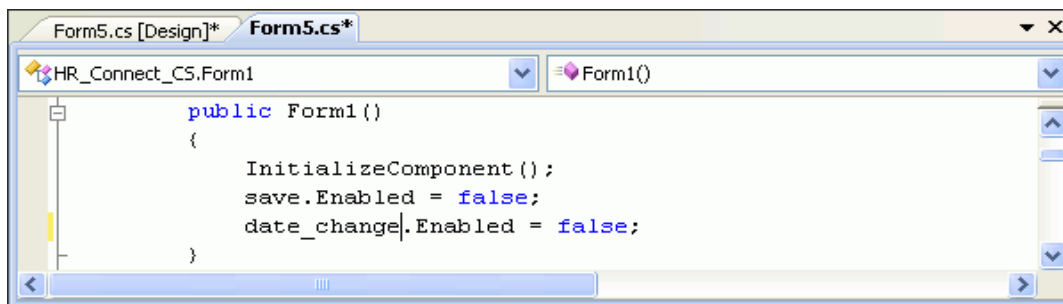
9. 該当するメソッド内に、次のコードを追加します。

Visual C#: Form1()メソッドに追加

```
date_change.Enabled = false;
```

Visual Basic: Form1_Loadメソッドに追加

```
date_change.Enabled = false
```



[nls10.gifの説明](#)

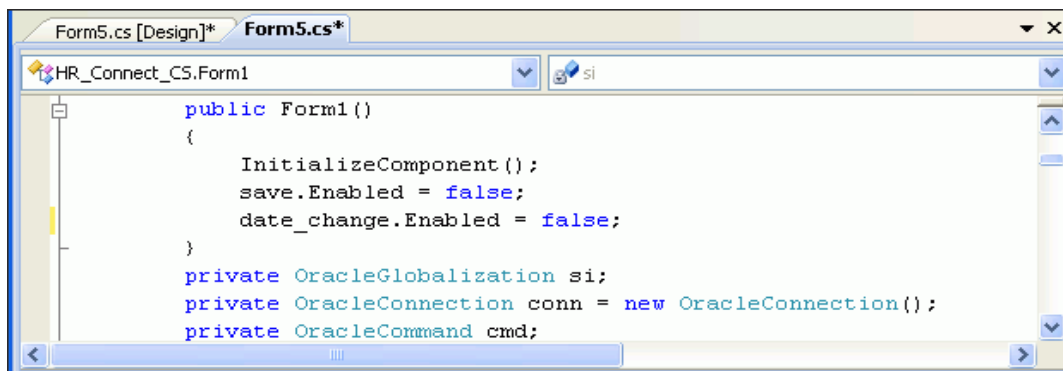
10. 次のコードを使用して、既存のForm1クラス宣言のpublic Form1()ブロックの直後に次のクラス変数を追加します。

Visual C#:

```
private OracleGlobalization si;
```

Visual Basic:

```
private si As OracleGlobalization
```



[nls11.gifの説明](#)

11. `connect_Click()` メソッドのTryブロックに、次の操作を実行するコードを追加します。

- OracleGlobalizationオブジェクトの値の取得
- EMPLOYEES表からのデータの取得（新しい問合せ）
- 「Change Date Format」 ボタンの有効化

変更したコードは太字で示しています。

Visual C#:

```
conn.Open();
connect.Enabled = false;

si = conn.GetSessionInfo();

string sql = "select employee_id, first_name, last_name, TO_CHAR(hire_date)" +
    " ¥"Hire Date¥" from employees where employee_id < 105";
cmd = new OracleCommand(sql, conn);
cmd.CommandType = CommandType.Text;

da = new OracleDataAdapter(cmd);
cb = new OracleCommandBuilder(da);
ds = new DataSet();

da.Fill(ds);

departments.DataSource = ds.Tables[0];

save.Enabled = true;
date_change.Enabled = true;
```

Visual Basic:

```
conn.Open()
connect.Enabled = false

si = conn.GetSessionInfo()

Dim sql As String = "select employee_id, first_name, last_name, " & _
    "TO_CHAR(hire_date) ""Hire Date"" from employees where employee_id < 105"
cmd = new OracleCommand(sql, conn)
cmd.CommandType = CommandType.Text

da = new OracleDataAdapter(cmd)
cb = new OracleCommandBuilder(da)
ds = new DataSet()

da.Fill(ds)

departments.DataSource = ds.Tables[0]

save.Enabled = true
date_change.Enabled = true
```

12. Form1を保存します。

13. [F5]キーボード・ショートカットを使用してアプリケーションを実行します。

アプリケーションがデータベースに正しく接続されると、データ・グリッドに問合せ結果が移入されます。

Form1

UserId: Password: Data Source:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	Hire Date
▶	100	Steven	King	17-JUN-87
	101	Neena	Kochhar	21-SEP-89
	102	Lex	De Haan	13-JAN-93
	103	Alexander	Hunold	03-JAN-90
	104	Bruce	Ernst	21-MAY-91
*				

[nls12.gifの説明](#)

- 「Change Date Format」をクリックします。

Form1

UserId: Password: Data Source:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	Hire Date
▶	100	Steven	King	1987-06-1
	101	Neena	Kochhar	1989-09-2
	102	Lex	De Haan	1993-01-1
	103	Alexander	Hunold	1990-01-0
	104	Bruce	Ernst	1991-05-2
*				

[nls13.gifの説明](#)

日付書式が元のDD-MON-RRからYYYY-MM-DDに変更されることを確認します。

- アプリケーションを閉じます。

スレッドベースのグローバリゼーション設定

スレッドベースのグローバリゼーション・パラメータ設定は、各スレッドに固有です。これらの設定は、最初はクライアント・グローバリゼーション・パラメータと同じですが、プログラムで変更できます。 ODP.NET型と文字列との間で変換が行われるときに、適用可能であればスレッドベースのグローバリゼーション・パラメータが使用されます。

スレッド・ベースのグローバリゼーション・パラメータ設定を取得するには、OracleGlobalizationクラスのGetThreadInfo()静的メソッドをコールします。SetThreadInfo()静的メソッドをコールすると、スレッドのグローバリゼーション設定が設定されます。

文化によって異なるデータを操作する場合、ODP.NETクラスおよび構造体は、OracleGlobalization設定のみを使用します。 .NETスレッドの文化情報は使用されません。 アプリケーションで.NET型のみを使用する場合は、OracleGlobalizationの設定は影響しません。 ただし、ODP.NET型と.NET型の間で変換を行う場合は、該当する箇所ではOracleGlobalizationの設定が使用されます。

注意:

System.Threading.Thread.CurrentThread.CurrentCultureプロパティを変更しても、スレッドまたはセッションのOracleGlobalization設定は変わりません。その逆も同様です。

A Oracle Databaseインスタンスの起動および停止

データベースの停止や再起動は頻繁に必要な操作です。

Oracle Databaseインスタンスを起動するには、次の手順を実行します。

1. 「スタート」ボタンから、「プログラム」、「管理ツール」、「サービス」の順に選択し、**OracleServiceDatabaseName**を選択します。DatabaseNameはtnsnames.oraファイルで示されているデータベースのservice_nameです。詳細は、[「NET接続の別名の構成」](#)参照してください。
2. 左側のパネルで、リンクをクリックして、サービスを開始します。
3. データベース・サービスが開始され、「データベースの起動」ウィンドウが表示されます。「OracleService service was started successfully」というメッセージが表示されるまで、操作は行わないでください。

Oracle Databaseインスタンスを停止するには、次の手順を実行します。

1. 「スタート」ボタンから、「プログラム」、「管理ツール」、「サービス」の順に選択し、**OracleServiceDatabaseName**を選択します。
2. 左側のパネルで、リンクをクリックして、サービスを停止します。
3. データベースの停止が開始され、「データベースの停止」ウィンドウが表示されます。「OracleService service was stopped successfully」というメッセージが表示されるまで、操作は行わないでください。

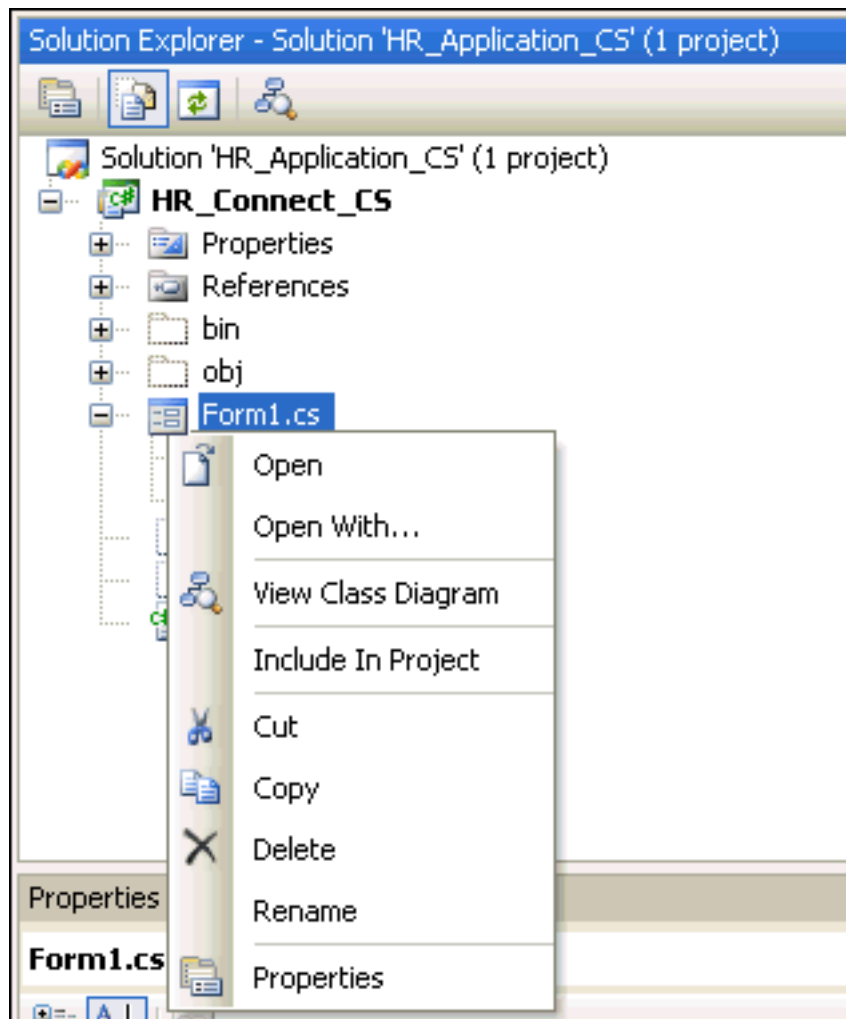
B フォームのコピー

Oracleを使用したアプリケーション開発の様々な側面について学習するために、このアプリケーションを使用する際は、フォームをコピーして再利用することが必要になる場合があります。

既存のフォームのコピーを作成するには、次の手順を実行します。

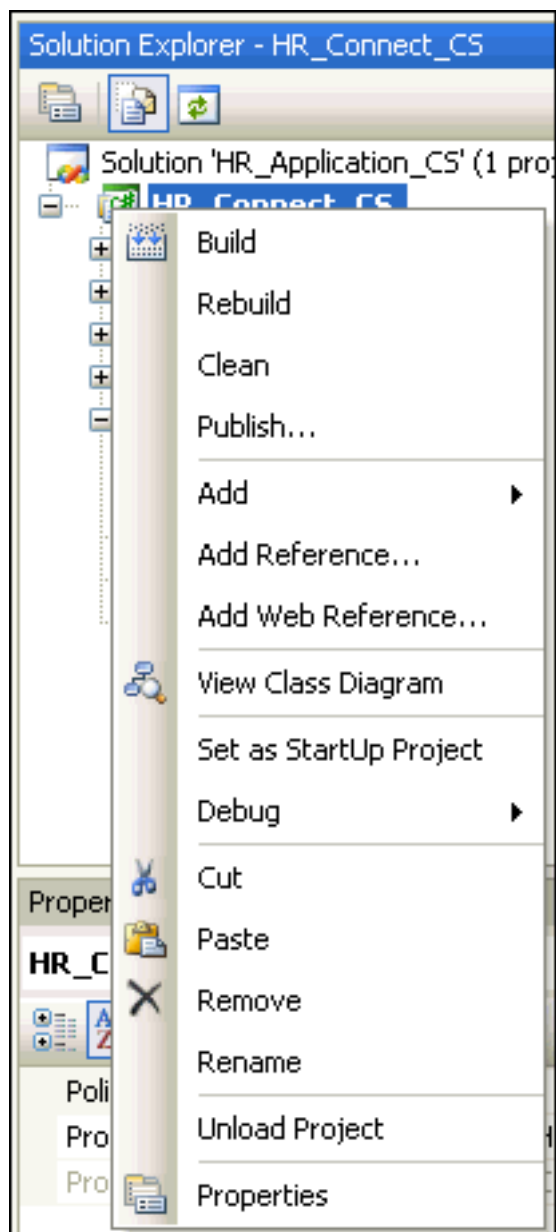
1. Solution Explorerで、コピーする必要があるForm1.xxまたはその他のファイルを右クリックします。「Copy」を選択します。

Form1.xxがSolution Explorerに表示されない場合は、「Project」メニューから、「Show All Files」を選択します。



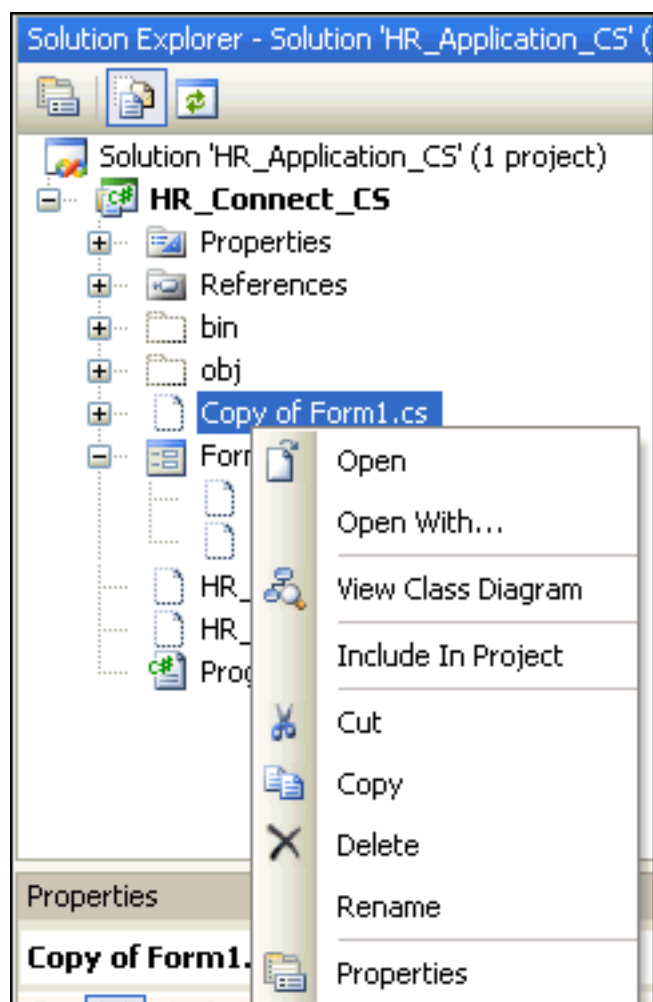
[copy1.gifの説明](#)

2. HR_Connect_CSまたはその他のプロジェクトを右クリックします。「Paste」を選択します。



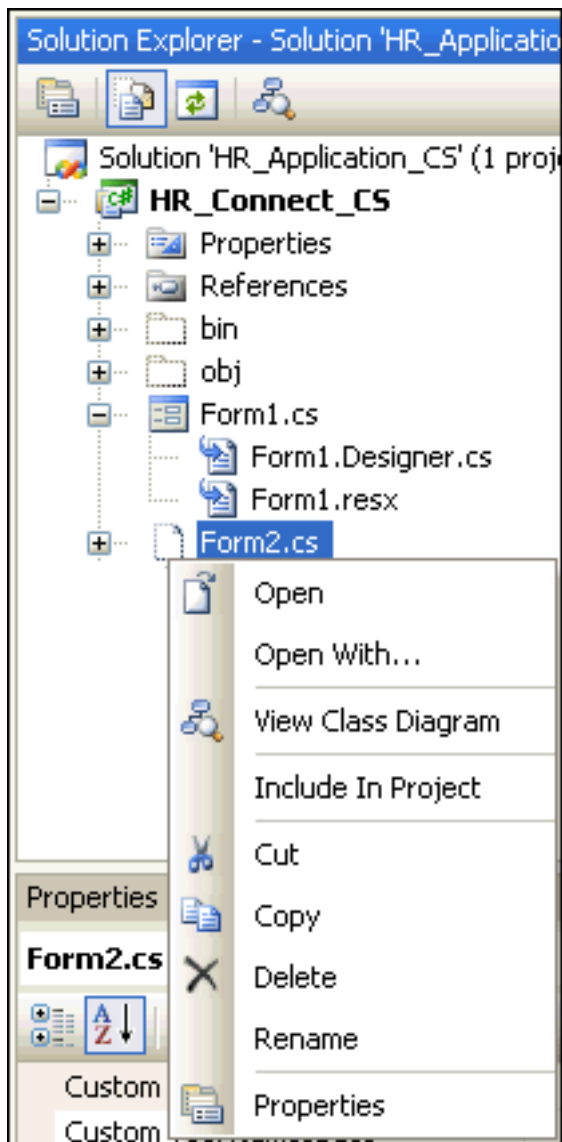
[copy2.gifの説明](#)

3. 「Copy of Form1.cs」を右クリックします。「Rename」を選択します。フォームの名前を**Form2.cs**に変更します。



[copy3.gifの説明](#)

4. 「Form2.cs」を右クリックして、「Include In Project」を選択します。



[copy4.gifの説明](#)

5. 「Form1.cs」を右クリックして、「Exclude From Project」を選択します。

これらの手順を逆に実行することで、プロジェクトにフォームを含めたり、プロジェクトからフォームを除外することができます。

注意:
通常、このプロセスは問題なく行われます。問題が発生した場合は、「Build」メニューから「Rebuild Solution」を実行する方法を試してください。

索引

記号 [A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [あ](#) [か](#) [さ](#) [た](#) [な](#) [は](#) [ま](#) [や](#) [ら](#)

記

.NETアセンブリ, [\[1\]](#)
.NET型, [\[1\]](#)
.NET言語, [\[1\]](#)
.NETストアド・ファンクションおよびプロシージャ
作成, [\[1\]](#)
実行, [\[1\]](#)
デプロイ, [\[1\]](#)
.NETストアド・プロシージャ, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
デプロイメント, [\[1\]](#)

A

Add() メソッド, [\[1\]](#)
ALTER TABLE, [\[1\]](#)
ASP.NET Webサイトの管理ツール, [\[1\]](#)
ASP.NETの構成, [\[1\]](#)
ASP.NETのチュートリアル, [\[1\]](#)
ASP.NETのチュートリアルの開始前, [\[1\]](#)
ASP.NETユーザー・スキーマ, [\[1\]](#)
ASPNET_DB_USER, [\[1\]](#)

C

C#文
使用, [\[1\]](#)
case文, [\[1\]](#)
CLR（共通言語ランタイム）, [\[1\]](#)
CommandTypeプロパティ, [\[1\]](#)
CultureInfoオブジェクト, [\[1\]](#)
Cultureパラメータ（ID）, [\[1\]](#)
CurrentCultureパラメータ, [\[1\]](#)

D

DataGridクラス, [\[1\]](#)
DataGridコントロール, [\[1\]](#)
DataReaderクラス, [\[1\]](#)
DataSetクラス, [\[1\]](#)
更新, [\[1\]](#)

Data Source Configuration Wizard, [\[1\]](#)
Default.aspx, [\[1\]](#)
Directionプロパティ, [\[1\]](#)
Dispose() メソッド, [\[1\]](#)

E

Enterprise Manager, [\[1\]](#), [\[2\]](#)
Errorプロパティ, [\[1\]](#)
Exceptionクラス, [\[1\]](#)
ExecuteReader() メソッド, [\[1\]](#)

F

FCL (Frameworkクラス・ライブラリ) , [\[1\]](#)
「File」メニュー, [\[1\]](#), [\[2\]](#)
Finallyブロック, [\[1\]](#)
Form1, [\[1\]](#)
form1.cs, [\[1\]](#)
form1.vb, [\[1\]](#)
Frameworkクラス・ライブラリ (FCL)
定義, [\[1\]](#)

G

GetSessionInfo() メソッド, [\[1\]](#)
GetThreadInfo() メソッド, [\[1\]](#)
GridViewコントロール, [\[1\]](#)

H

HRスキーマ, [\[1\]](#)

I

Imports文, [\[1\]](#)
「Indexes」タブ, [\[1\]](#)
InstallOracleASPNETCommon.sql, 構成, [\[1\]](#)

L

ListBox, [\[1\]](#)
login.aspx, [\[1\]](#)

M

machine.config, [\[1\]](#)
Microsoft .NET Framework
 定義, [\[1\]](#)
Microsoft Visual Studio, [\[1\]](#)
Microsoft社の国際化対応
 URL, [\[1\]](#)

N

Nameプロパティ, [\[1\]](#)
NET接続, [\[1\]](#)
 「New Package」 ウィンドウ, [\[1\]](#), [\[2\]](#)
 「New Project」 ダイアログ・ボックス, [\[1\]](#)
NLS_LANGUAGEパラメータ, [\[1\]](#), [\[2\]](#), [\[3\]](#)
NLS_LANGパラメータ, [\[1\]](#), [\[2\]](#)
NLS_SORTパラメータ, [\[1\]](#)
NLS_TERRITORYパラメータ, [\[1\]](#), [\[2\]](#), [\[3\]](#)
NLSエラー・メッセージの設定, [\[1\]](#)
NLSソート順序, [\[1\]](#)
NLSの数値書式
 設定, [\[1\]](#)

O

ODAC (Oracle Data Access Components) , [\[1\]](#)
ODP.NET型, [\[1\]](#)
ODTの「Grant/Revoke Privileges Wizard」, [\[1\]](#)
ODTを使用したASP.NETアプリケーションの構築, [\[1\]](#)
Open() メソッド, [\[1\]](#), [\[2\]](#)
OracleClrAgentサービス, [\[1\]](#)
OracleCommandクラス, [\[1\]](#), [\[2\]](#), [\[3\]](#)
 ストアド・プロシージャの使用, [\[1\]](#)
 「OracleConnection」 ウィンドウの構成, [\[1\]](#)
OracleConnectionクラス, [\[1\]](#), [\[2\]](#)
 GetSessionInfo() メソッド, [\[1\]](#)
 Open() メソッド, [\[1\]](#)
OracleDataAccess.dll, [\[1\]](#)
Oracle Data Access Components (ODAC) , [\[1\]](#)
 ダウンロード, [\[1\]](#)
Oracle Database, [\[1\]](#)
 インストール, [\[1\]](#)
 ドキュメント・ライブラリ, [\[1\]](#)

- Oracle Database Extensions for .NET
 - アップグレード, [\[1\]](#)
 - インストール, [\[1\]](#)
- Oracle Databaseインスタンスの起動, [\[1\]](#)
- Oracle Databaseインスタンスの停止, [\[1\]](#)
- Oracle Data Provider for .NET, , [\[1\]](#)
 - 使用, [\[1\]](#)
- Oracle Data Provider for .NET (ODP.NET)
 - インストール, [\[1\]](#)
 - グローバリゼーション, [\[1\]](#)
 - 定義, [\[1\]](#)
- OracleDataReaderクラス, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
- OracleDbTypeプロパティ, [\[1\]](#), [\[2\]](#)
- Oracle Deployment Wizard for .NET, [\[1\]](#), [\[2\]](#)
- Oracle Developer Tools
 - インストール, [\[1\]](#)
 - 機能
 - Oracle Data Window, [\[1\]](#)
 - Oracle Query Window, [\[1\]](#)
 - PL/SQLエディタ, [\[1\]](#)
 - ウィザード, [\[1\]](#)
 - ダイナミック・ヘルプ, [\[1\]](#)
 - デザイン, [\[1\]](#)
 - ドラッグ・アンド・ドロップ, [\[1\]](#)
 - 使用, [\[1\]](#)
 - 定義, [\[1\]](#)
- OracleErrorCollectionクラス, [\[1\]](#)
- OracleErrorクラス, [\[1\]](#)
- OracleExceptionクラス, [\[1\]](#), [\[2\]](#)
- OracleGlobalization
 - GetClientInfo() メソッド, [\[1\]](#)
 - クラス, [\[1\]](#)
- OracleGlobalizationクラス
 - GetThreadInfo() メソッド, [\[1\]](#)
 - SetThreadInfo() メソッド, [\[1\]](#)
- OracleParameterCollectionクラス, [\[1\]](#)
- OracleParameterクラス, [\[1\]](#), [\[2\]](#)
- Oracle Providers for ASP.NET, [\[1\]](#)
 - カスタマイズ, [\[1\]](#)
 - 個別に構成, [\[1\]](#)
 - セットアップ, [\[1\]](#)
 - 有効化, [\[1\]](#)
- Oracle Providers for ASP.NETのカスタマイズ, [\[1\]](#)
- Oracle Providers for ASP.NETの構成
 - 個別, [\[1\]](#)
 - すべて, [\[1\]](#)
- Oracle Providers for ASP.NETのセットアップ, [\[1\]](#)
- Oracle Providers for ASP.NETの有効化, [\[1\]](#)
- OracleRefCursorクラス, [\[1\]](#)
- OracleService, [\[1\]](#)
- Oracle Universal Installer (OUI) , [\[1\]](#)
- ORACLE_BASE\ORACLE_HOME, [\[1\]](#)
- Oracleのエラー・メッセージ, [\[1\]](#)
- Oracleの言語ソート, [\[1\]](#)
- Oracleの数値書式, [\[1\]](#)
- Oracleの日付書式, [\[1\]](#)
- Oracleプロジェクト
 - 作成, [\[1\]](#)
- Oracleプロジェクトの作成, [\[1\]](#)
- OraProvCfg, [\[1\]](#)
- OUI (Oracle Universal Installer) , [\[1\]](#)

P

PACKAGE型, [\[1\]](#)
ParameterName, [\[1\]](#)
PL/SQLストアド・プロシージャ
 ODP.NET, [\[1\]](#)
 REF CURSOR, [\[1\]](#), [\[2\]](#)
 概要, [\[1\]](#)
 定義, [\[1\]](#)
PL/SQLパッケージ
 インタフェース, [\[1\]](#)
 概要, [\[1\]](#)
 定義, [\[1\]](#)
 本体, [\[1\]](#)

R

REF CURSOR
 PL/SQLストアド・プロシージャ, [\[1\]](#), [\[2\]](#)
 PL/SQLデータ型, [\[1\]](#)
 アクセシビリティ, [\[1\]](#)
 概要, [\[1\]](#)
 定義, [\[1\]](#)
 割当て, [\[1\]](#), [\[2\]](#)
「Run Function」 ウィンドウ, [\[1\]](#)

S

Saveコマンド, [\[1\]](#)
SELECT文
 単純, [\[1\]](#)
 バインド変数, [\[1\]](#)
Server Explorer, [\[1\]](#), [\[2\]](#)
 使用, [\[1\]](#)
service_name, [\[1\]](#)
SetThreadInfo() メソッド, [\[1\]](#)
Sizeプロパティ, [\[1\]](#)
SQL*Plus, [\[1\]](#)
 接続, [\[1\]](#)
sqlnet.ora, [\[1\]](#)
SQLでの.NETプロシージャの実行, [\[1\]](#)
SQL問合せ, [\[1\]](#)
SQLのプレビュー, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
SQL文の文字列, [\[1\]](#)
Start Without Debugging, [\[1\]](#)
SYSDBA
 接続, [\[1\]](#)
System.Globalization, [\[1\]](#)
System.Resources, [\[1\]](#)
System.Threading.Thread.CurrentThread.CurrentCultureパラメータ, [\[1\]](#)

T

Textプロパティ, [\[1\]](#)
tnsnames.ora, [\[1\]](#)
 構成, [\[1\]](#)
Toolbox, [\[1\]](#)
Try-Catch-Finallyエラー処理, [\[1\]](#)
Try-Catch-Finallyブロック, [\[1\]](#)
Tryコード・ブロック, [\[1\]](#), [\[2\]](#)

U

user_sourceビュー, [\[1\]](#)
using文, [\[1\]](#)

V

Valueプロパティ, [\[1\]](#)
 「View」メニュー, [\[1\]](#)
Visual Basic (VB) 文
 Imports, [\[1\]](#)
Visual Studio, [\[1\]](#)
 バージョン, [\[1\]](#)

W

Webサイト
 作成, [\[1\]](#)
 データベースへの接続, [\[1\]](#)
Webサイト認証, [\[1\]](#)
 テスト, [\[1\]](#)
Webサイトの作成, [\[1\]](#)
Webサイトのデータベースへの接続, [\[1\]](#)
Webユーザー
 作成, [\[1\]](#)
Webユーザーの作成, [\[1\]](#)
Windowsレジストリ, [\[1\]](#), [\[2\]](#)

アカウント
 ロック解除, [\[1\]](#), [\[2\]](#)
アカウントのロック解除, [\[1\]](#), [\[2\]](#)
イベント

クリック, [\[1\]](#)
エラー処理
 ODP.NET, [\[1\]](#)
 ODP.NETの例外, [\[1\]](#)
 Oracle, [\[1\]](#)
 Try-Catch-Finally, [\[1\]](#)
エラー・メッセージ, [\[1\]](#)

外部キー, [\[1\]](#)
共通言語ランタイム (CLR)
 エージェント, [\[1\]](#)
 サービス
 開始, [\[1\]](#)
 定義, [\[1\]](#)
クライアント・グローバリゼーション設定, [\[1\]](#)
クラス変数, [\[1\]](#)
クリック・イベント, [\[1\]](#)
グローバリゼーション
 セッション情報, [\[1\]](#)
 定義, [\[1\]](#)
グローバリゼーション・サポート
 ODP for .NET, [\[1\]](#)
 クライアント, [\[1\]](#)
 .NET Framework, [\[1\]](#)
グローバル・アプリケーション
 開発, [\[1\]](#)
 概要, [\[1\]](#)
警告
 エラー処理, [\[1\]](#)
結果セット, [\[1\]](#)
権限
 付与, [\[1\]](#)
権限の付与, [\[1\]](#)
言語ソート, [\[1\]](#)
構成スクリプト, [\[1\]](#)
コードとデザイナの切替え, [\[1\]](#)
コード表示, [\[1\]](#)
コード・ファイルの名前, [\[1\]](#)
コピー・オプション・ウィンドウの指定, [\[1\]](#)
コマンド
 使用, [\[1\]](#)
 問合せ, [\[1\]](#)
コントロール, [\[1\]](#)
 DataGrid, [\[1\]](#)
 Label, [\[1\]](#), [\[2\]](#)
 Listbox, [\[1\]](#)
 Textbox, [\[1\]](#)
 Toolbox, [\[1\]](#)
 ボタン, [\[1\]](#)

索引
 作成, [\[1\]](#)

追加, [\[1\]](#)
プロパティ, [\[1\]](#)
サービス, [\[1\]](#)
サマリー
 デプロイメント, [\[1\]](#)
参照
 追加, [\[1\]](#)
参照の追加, [\[1\]](#)
サンプル・スキーマ, [\[1\]](#)
サンプル・データ, [\[1\]](#)
自動ネーミング, [\[1\]](#)
主キー
 列, [\[1\]](#)
スキーマ
 表示, [\[1\]](#), [\[2\]](#)
スキーマ・オブジェクト, [\[1\]](#), [\[2\]](#)
スキーマの表示, [\[1\]](#), [\[2\]](#)
ストアド・プロシージャ
 実行, [\[1\]](#)
 定義, [\[1\]](#)
 保持を目的にしたOracleプロジェクトの作成, [\[1\]](#)
スレッドベースのグローバリゼーション設定, [\[1\]](#)
制約
 タブ, [\[1\]](#)
 追加, [\[1\]](#)
 プロパティ, [\[1\]](#)
セキュリティ, [\[1\]](#)
設計ビュー, [\[1\]](#)
セッション・グローバリゼーション設定, [\[1\]](#)
接続, [\[1\]](#)
 hr, [\[1\]](#)
 SYSDBAとして, [\[1\]](#), [\[2\]](#)
 オープン, [\[1\]](#)
 構築, [\[1\]](#)
 詳細, [\[1\]](#)
 新規, [\[1\]](#)
 追加, [\[1\]](#)
 データソース名, [\[1\]](#), [\[2\]](#), [\[3\]](#)
 特定のユーザー名とパスワード, [\[1\]](#)
 名前, [\[1\]](#)
 破棄, [\[1\]](#)
 パスワード, [\[1\]](#), [\[2\]](#)
 ユーザー名, [\[1\]](#), [\[2\]](#)
 ユーザー名とパスワード, [\[1\]](#)
 ルール, [\[1\]](#)
接続記述子, [\[1\]](#)
接続コントロール, [\[1\]](#)
接続のオープン, [\[1\]](#)
接続の作成, [\[1\]](#)
接続の別名, [\[1\]](#)
接続文字列
 ASP.NET用の設定, [\[1\]](#)
ソリューション, [\[1\]](#)
ソリューションの再作成, [\[1\]](#)

た

ダイアログ・ボックス
 新しいプロジェクト, [\[1\]](#)
単純な問合せ, [\[1\]](#)

チュートリアル, [\[1\]](#)
テキストボックス・コントロール, [\[1\]](#)
デザイナ, [\[1\]](#)
デザイナとコードの切替え, [\[1\]](#)
テスト
 Webサイト認証, [\[1\]](#)
データ・グリッド, [\[1\]](#)
データソース名, [\[1\]](#), [\[2\]](#)
データ入力コントロール, [\[1\]](#)
データの更新
 バインド変数, [\[1\]](#)
データの削除, [\[1\]](#)
データの取得
 Oracleから, [\[1\]](#)
 valueメソッド, [\[1\]](#)
 アクセッサの型, [\[1\]](#)
 単純な問合せ, [\[1\]](#)
 バインド変数, [\[1\]](#)
 複数の値, [\[1\]](#)
 複数の行, [\[1\]](#)
 複数の列, [\[1\]](#)
 ループ, [\[1\]](#)
データの挿入, [\[1\]](#)
データのバインド, [\[1\]](#)
データ・プロバイダ, [\[1\]](#)
 Oracle Data Provider for .NET, [\[1\]](#)
データベース・エラー・メッセージ, [\[1\]](#)
デフォルトのロール, [\[1\]](#)
デプロイメント・オプション・ウィンドウの指定, [\[1\]](#)
 .NETプロシージャの実行, [\[1\]](#)
問合せウィンドウ
問合せウィンドウでの.NETプロシージャの実行, [\[1\]](#)
問合せ作業領域
 定義, [\[1\]](#)
問合せのパフォーマンス, [\[1\]](#)
ドキュメント・ライブラリ, [\[1\]](#)
匿名ユーザー
 拒否, [\[1\]](#)

な

名前空間ディレクティブ, [\[1\]](#)
認証
 Webサイト, [\[1\]](#)
認証用のWebサイトの有効化, [\[1\]](#)

は

バインド変数
 位置, [\[1\]](#)
 名前, [\[1\]](#)
パスワード
 保存, [\[1\]](#)
パッケージ
 新規, [\[1\]](#), [\[2\]](#)
パッケージ・インタフェース, [\[1\]](#)

パッケージ本体, [\[1\]](#)
日付書式, [\[1\]](#)
 変更, [\[1\]](#)
ビュー
 user_source, [\[1\]](#)
 設計, [\[1\]](#)
 表の設計, [\[1\]](#)
表
 新しいリレーショナル, [\[1\]](#)
 グリッド, [\[1\]](#)
 作成, [\[1\]](#)
 新規, [\[1\]](#)
 制約, [\[1\]](#)
 追加, [\[1\]](#)
 制約プロパティ, [\[1\]](#)
 制約名, [\[1\]](#)
 単純な問合せ, [\[1\]](#)
 追加, [\[1\]](#)
 データの取得, [\[1\]](#)
 データの追加, [\[1\]](#)
 問合せ, [\[1\]](#)
 リレーショナル, [\[1\]](#)
 レコード, [\[1\]](#)
表の設計ウィンドウ, [\[1\]](#)
表の設計ビュー, [\[1\]](#)
フィルタの適用, [\[1\]](#), [\[2\]](#)
フォーム, [\[1\]](#)
フォームのコピー, [\[1\]](#)
フォームの名前, [\[1\]](#)
プロジェクト
 参照の追加, [\[1\]](#)
 種類
 Visual Basic, [\[1\]](#)
 Visual C#, [\[1\]](#)
 新規, [\[1\]](#)
 ソリューション, [\[1\]](#)
プロパティ
 Direction, [\[1\]](#)
 OracleDbType, [\[1\]](#)
 OracleDbTypeプロパティ, [\[1\]](#)
 ParameterName, [\[1\]](#)
 Size, [\[1\]](#)
 Value, [\[1\]](#)
 エラー, [\[1\]](#)
プロパティ・ウィンドウ, [\[1\]](#)
文
 case, [\[1\]](#)
 Imports, [\[1\]](#)
 解析, [\[1\]](#)
 最適化, [\[1\]](#)
 再利用, [\[1\]](#)
 使用, [\[1\]](#)
文化的な表記規則, [\[1\]](#)
文化によって異なるデータ, [\[1\]](#)
別名
 データベース, [\[1\]](#)
変数宣言, [\[1\]](#)
ボタン・コントロール, [\[1\]](#)

メソッド
 Add(), [\[1\]](#)
 Dispose(), [\[1\]](#)
 Open(), [\[1\]](#)
メソッドおよびセキュリティ詳細ウィンドウの指定, [\[1\]](#)
メソッド・パラメータ
 定義, [\[1\]](#)
 バインド, [\[1\]](#)
メニュー
 File, [\[1\]](#), [\[2\]](#)
 View, [\[1\]](#)
メモリーの位置, [\[1\]](#)

ユーザー
 作成, [\[1\]](#)
 ロケール設定, [\[1\]](#)
 ロール, [\[1\]](#), [\[2\]](#)
ユーザー・アカウントのロック解除
 Oracle Databaseのインタフェース, [\[1\]](#)
ユーザー・インタフェース
 設計, [\[1\]](#)
ユーザー・インタフェースの設計, [\[1\]](#)
ユーザー・スキーマ
 ASPNET_DB_USER, [\[1\]](#)
ユーザーの作成, [\[1\]](#)
ユーザーのローカル規則, [\[1\]](#)

ラベル・コントロール, [\[1\]](#), [\[2\]](#)
例
 名前, [\[1\]](#)
レコード, [\[1\]](#)
 追加, [\[1\]](#)
ローカライズ
 リソース, [\[1\]](#)
ロケール
 定義, [\[1\]](#)
 同期化, [\[1\]](#)
 特性, [\[1\]](#)
 認識, [\[1\]](#)
ロック, [\[1\]](#), [\[2\]](#)
ロール
 ユーザーのデフォルト, [\[1\]](#)

この図は、OUIの「ようこそ」画面を示しています。ここには、「製品の削除」、「バージョン情報」、「ヘルプ」、「インストールされた製品」、「戻る」、「次へ」、「インストール」および「取消」の各ボタンがあります。

この図は、本文で説明されている2つのオプションを示しています。ここには、「製品の言語」、「ヘルプ」、「戻る」、「次へ」、「インストール」および「取消」の各ボタンがあります。

この図は、.NET製品の現在のインストール先フォルダを表示しています。ここには、パスを変更する場合の「参照」ボタンと、「ヘルプ」、「インストールされた製品」、「戻る」、「次へ」、「インストール」、「取消」の各ボタンがあります。

この図は、インストーラにより、使用中の環境がインストールする製品の最低要件を満たしていることが検証されたことを示しています。また、「チェック」、「タイプ」および「ステータス」というタイトルの付いたグリッドを表示しています。選択されている項目は、「Oracleホームのパスで空白を確認中」です。「タイプ」には「自動」、「ステータス」には「成功しました」がそれぞれ表示されています。

この図は、Oracle Data Access Components for Oracle Clientのインストール画面を示しています。必要なコンポーネントは本文にリストされています。チェック・ボックスのある「コンポーネント」列と、「インストール状況」列があります。「すべて開く」、「すべて閉じる」、「すべて選択」、「すべて選択解除」、「ヘルプ」、「インストールされた製品」、「戻る」、「次へ」、「インストール」および「取消」の各ボタンがあります。「必須の依存コンポーネントを含むすべてのコンポーネントを表示。」のチェック・ボックスがあります。

この図は、「サマリー」ウィンドウを表示しています。ここには、「Oracleベース: E:¥app¥Oracle」などの「グローバルな設定」、「製品の言語」、「必要な領域」および「新規インストール」が示されています。また、「ヘルプ」、「インストールされた製品」、「戻る」、「次へ」、「インストール」および「取消」の各ボタンがあります。

この図は、Server Explorerのツリーを示しています。SYS. ORCLに関するメニューが開いて「Query Window」が選択されています。

このスクリーンショットは、SQLコマンド「create user aspnet_db_user identified by your_password」が表示されている問合せウィンドウを示しています。 下の問合せの結果ウィンドウには「The command(s) successfully completed.」と表示されています。

この図は、Server Explorerのツリーを示しています。SYS. ORCLに関するメニューが開いて「Privileges...」が選択されています。

このウィンドウには、「スキーマ」（リストからSYSが選択されている）、「オブジェクト・タイプ」（リストからUSERが選択されている）、および「オブジェクト名」（無効、リストからANONYMOUSが選択されている）が表示されています。

「ユーザー/ロール」セクションがあります。「ユーザー」が選択され、リストから「ASPNET_DB_USER」が選択されています。その下に「権限」セクションがあります。2つのリストがあります。左側の「使用可能な権限」には選択可能なすべての権限が含まれています。右側の「付与された権限」には選択された権限が含まれています。リストの間には矢印があり、これをクリックすることでリスト間で項目を移動できます。

一番下には、「プレビューSQL」、「OK」、「取消」および「ヘルプ」の各ボタンがあります。

このスクリーンショットは、Run SQL*Plus Script機能を示しています。「File Information」のグループには、参照ボタンでSQL*Plusスクリプトのフルパスを指定するためのフィールドがあります。「Connection Information」のグループには、データベース接続情報としてSYS.ORCLが指定されており、参照用の「New Connection」ボタンと、「DATA SOURCE=ORCL;DBAPRIVILEGE=SYSDBA;PERSIS SECURITY」が表示された「Connection String」セクションがあります。その下に読取り専用の「User input area」があります。実行スクリプト・バーと「Run」ボタンおよび「Cancel」ボタンもあります。

このスクリーンショットは、ORACLE_BASE¥¥ORACLE_HOME¥ASP.NET¥sqlディレクトリのInstallAllOracleASPNETProviders.sqlファイルが、開くために選択された状態を示しています。

このスクリーンショットは、「Connection Properties」ダイアログを示しています。最初のフィールドは「Data source」で、「Oracle Database (Oracle ODP.NET)」が入力されています。この設定を変更するボタンがあります。

また、「Connection Details」タブがあり、選択されたデータソース名が「ORCL」になっています。

「Use Windows integrated authentication」オプションの選択が解除され、「Use a specific user name and password」オプションが選択された状態で、ユーザー名にはASPNET_DB_USER、パスワードには*****が表示されています。

「Save Password」チェックボックス（選択されていない状態）、その下に「Role」（リストから「Default」が選択された状態）があります。また、「Connection name」にはASPNET_DB_USER.ORCLが指定されています。

一番下には「Advanced」、「Test Connection」、「OK」および「Cancel」の各ボタンがあります。

このスクリーンショットは、Run SQL*Plus Script機能を示しています。「File Information」のグループには、参照ボタンでSQL*Plusスクリプトのフルパスを指定するためのフィールドがあります。「Connection Information」のグループには、データベース接続情報としてASPNET_DB_USER. ORCLが指定されており、参照用の「New Connection」ボタンと、「DATA SOURCE=ORCL:PERSIS SECURITY INFO=False;USER」が表示された「Connection String」セクションがあります。その下に読取り専用の「User input area」があります。実行スクリプト・バーと「Run」ボタンおよび「Cancel」ボタンもあります。

この図は、Microsoft Visual Studioの開始ウィンドウを表示しています。左上に「Recent Projects」セクション、左中央に「Getting Started」セクション、左下に「Visual Studio Headlines」セクション、中央に「Visual Studio Developer News」ページ、右上に「Solution Explorer」、「Properties」ウィンドウがあります。

この図は、図の前の本文で説明されているフィールドを表示しています。左側にプロジェクト・タイプ、右側にテンプレートが表示されています。プロジェクトの場所を特定する「Browse」ボタン、「Create directory for solution」および「Add to Source Control」の各チェック・ボックスがあり、「OK」ボタンと「Cancel」ボタンがあります。

この図は、図の前の本文で説明されている新しいC#プロジェクトの最初のフォームを示しています。

この図は、この手順の操作結果を示しています。この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図は、ツリー構造の「Solution Explorer」を示しています。最初のブランチは「Properties」で、次に「References」があります。「References」が開かれて、「Oracle.DataAccess」、「System」など多数のコンポーネントが表示されています。

この図は、「View」メニューを表示しています。使用可能なコマンドには、「Code」、「Designer」、「Server Explorer」、「Solution Explorer」および「Class View」があります。

この図は、C#アプリケーションの宣言部を表示しています。本文で説明されている2つのusing文の他に、多くの文が表示されています。

この図は、Visual Basicアプリケーションの宣言部を表示しています。本文で説明されている2つのImports文の他に、多くの文が表示されています。

この図は、「Toolbox」を示しています。多数のコントロールが含まれています。最初に、閉じている状態の「All Windows Forms」があり、次に開かれている状態の「Common Controls」があります。そこには「Pointer」、「Button」、「CheckBox」、「Label」、「ListBox」など多数のコントロールが表示されています。

この図の説明は、図の前の本文を参照してください。

この図は、表示されるメニューを示しています。このメニューには、「View Code」、「Bring to Front」、「Send to Back」、「Lock Controls」などの各コマンドが含まれています。

この図は、「Text」が選択され、テキスト・プロパティが「User ID」である「Properties」ウィンドウを示しています。

この図は、図の前の本文で説明されているように、この手順の操作結果を示しています。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図は、「Build Solution」、「Rebuild Solution」、「Clean Solution」、「Batch Build」などのコマンドがある「Build」メニューを示しています。

この図は、「Build」からの「Output」ウィンドウを示しています。次のテキストが表示されています。

```
Compile complete -- 0 errors, 0 warnings
```

```
HR_Connect_CS -> C:\¥HR_Projects¥HR_Connect_CS¥HR_Connect_CS¥bin¥Debug¥
```

```
HR_Connect_CS.exe
```

```
=====Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

この図は、「Windows」、「Start Debugging」、「Start Without Debugging」、「Attach to Process」、「Exceptions...」、「Step Into」などのコマンドがある「Debug」メニューを示しています。

この図は、リクエストされた項目を入力するフィールド、および「Connect」というラベルの付いたボタンのあるフォームを示しています。この図については、図の前の本文を参照してください。

この図には、本文で説明されている例外エラー・テキストが含まれており、大きなXマーク、および「Details」、「Continue」、「Quit」の各ボタンがあります。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」、「Department」の各フィールドのあるフォームを示しています。「Departments」フィールドにはdepartmentsと入力されています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」、「Department」の各フィールドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorcl、「Department」にはAdministrationがそれぞれ入力されています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」、「Department」の各フィールドのあるフォームを示しています。ラベルのない大きなフィールドが1つあります。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」、「Department」の各フィールドと、リスト・フィールドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorcl、「Department」には50がそれぞれ入力されています。リスト・フィールドには「Shipping」と表示されています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」、「Department」の各フィールドと、リスト・フィールドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorcl、「Department」には50がそれぞれ入力されています。リスト・フィールドには「Administration」、「Marketing」「Purchasing」「Human Resources」が表示されています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」の各フィールドのあるフォームを示しています。ラベルのない大きなフィールドが1つあります。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」の各フィールドと、グリッドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorclがそれぞれ入力されています。また、グリッドには「DEPARTMENT_ID」、「DEPARTMENT_NAME」、「MANAGER_ID」、「LOCATION_ID」が表示されています。最初の行は、10、Administration、200、1700です。2行目は、20、Marketingなどで、50まで表示されています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」の各フィールドのあるフォームを示しています。ラベルのない大きなフィールドが1つと、「Save」ボタンがあります。

この図は前の項の最後にあるものと同じですが、ここには「Save」ボタンがあります。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」の各フィールドと、グリッドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorclがそれぞれ入力されています。また、グリッドには「DEPARTMENT_ID」、「DEPARTMENT_NAME」、「MANAGER_ID」、「LOCATION_ID」が表示されています。最初の行は、10、Administration、200、1700です。2行目には、20、Marketingなどが表示されています。最後の行は、5、Community Outreach、空白、1700です。さらに入力が続いています。

この図は、「Connect」ボタン、および「User ID」、「Password」、「Data Source」の各フィールドと、グリッドのあるフォームを示しています。「User ID」にはhr、「Password」には*****、「Data Source」にはorclがそれぞれ入力されています。また、グリッドには「DEPARTMENT_ID」、「DEPARTMENT_NAME」、「MANAGER_ID」、「LOCATION_ID」が表示されています。最初の行は、5、Community Outreach、空白、1700です。2行目は、10、Administration、200、1700です。3行目には、20、Marketingなどが表示されています。

この図は、前のショットと同じビューを表示していますが、マウスが一番左の列の上にあり、削除するためにその行全体が選択されています。

この図の説明は、図の前の本文を参照してください。

この図は、空の「Data source:」フィールドと「Change」ボタンを示しています。

この図はデータソースのリストを示しています。「Oracle Database」が選択されています。この下には「Data provider」のリストがあり、「Oracle Data Provider for .NET」が表示されています。「Always use this selection」チェック・ボックス、「OK」ボタンおよび「Cancel」ボタンがあります。

このスクリーンショットは、「Add Connection」ダイアログを示しています。最初のフィールドは「Data source」で、「Oracle Database (Oracle ODP.NET)」が入力されています。この設定を変更するボタンがあります。

また、「Connection Details」タブがあり、選択されたデータソース名がorclになっています。

「Use Windows integrated authentication」オプションの選択が解除され、「Use a specific user name and password」オプションが選択された状態で、ユーザー名にはHR、パスワードには*****が入力されています。

「Save Password」チェックボックス（選択された状態）、その下に「Role」（リストから「Default」が選択された状態）があります。また、「Connection name」にはHR.orclが指定されています。

一番下には「Advanced」、「Test Connection」、「OK」および「Cancel」の各ボタンがあります。

この図は、「Apply Filters」タブが選択された状態の「Add Connection」画面を示しています。左側には「Available Schema」があり、Anonymousなどが表示されています。右側には「Displayed Schemas」があります。これには、HRスキーマが含まれています。2つのスキーマ・リストの間には、スキーマを相互に移動するための矢印があります。「Display Public Synonyms」チェック・ボックスがあります。「Advanced...」、「Test Connection」、「OK」および「Cancel」の各ボタンがあります。

この図は、「Test connection succeeded.」のメッセージを示しています。「OK」ボタンがあります。

この図は「Server Explorer」を示しています。「Data Connections」が選択され、「HR. ORCL」接続が開かれています。

この図は、「HR. ORCL」接続が開かれた「Server Explorer」を示しています。「Tables」が選択され、メニューが表示されています。メニューのリストには、「New Relational Table...」、「New Object Table...」、「New XML Table...」、「Import Table...」、「Query Window」、「Privileges...」、「Refresh」および「Properties」が表示されています。

この図は、表のデザイナーを示しています。一番上に「Schema Name」があり、リストから「HR」が選択されています。また、「Table type」はRelationalが指定されています。「Columns」、「Constraints」、「Indexes」、「Storage」という4つのタブがあります。「Columns」タブが選択されています。ここには「Name」と「Data Type」というヘッダーの付いた列領域があり、その下に「Add」ボタンと「Remove」ボタンがあります。列の順番を変更する上向き/下向き矢印があり、その右側には「Column Properties」領域があります。その領域には「Name」、「Data Type」および「Default」を入力するフィールドがあります。一番下に、「Preview SQL」ボタンと「Save」ボタンがあります。

この図は、説明されている表のデザイナを示しています。「Columns」リストでは、「Name」にLAST_NAME、「Data Type」にVARCHAR2が指定されています。「Column Properties」では、「Name:」にLAST_NAME、「Data Type:」にVARCHAR2が指定され、「Default:」は空白です。「Allow null」ボックスが選択されており、「Size:」は30に指定され、「Length semantic」はDefaultが選択されています。

この図には、次のコードが含まれています。

```
CREATE TABLE "DEPENDENTS" (  
  "LAST_NAME" VARCHAR2(30) NULL,  
  "FIRST_NAME" VARCHAR2(20) NULL,  
  "BIRTH_DATE" DATE NULL,  
  "RELATIONSHIP" VARCHAR2(20) NULL,  
  "EMPLOYEE_ID" NUMBER(6,0) NOT NULL,  
  "DEPENDENT_ID" NUMBER(6,0) NOT NULL);
```

この図は、説明されている表のデザイナを示しています。「Columns」リストが増えています。このリストには、「LAST_NAME」の他に、「FIRST_NAME」、「BIRTH_DATE」、「RELATIONSHIP」、「EMPLOYEE_ID」、「DEPENDENT_ID」が追加されています。「Column Properties」では、「Name:」にDEPENDENT_ID、「Data Type:」にNUMBERが指定され、「Default:」は空白です。「Allow null」ボックスの選択が解除されており、「Size:」は5に指定され、「Length semantic」は0が選択されています。

この図は、「Indexes」タブが選択された状態の「Table Designer」を示しています。「Name:」がDependents_Indexで「Type:」がB-treeの索引が1つあります。右側の「Index Properties」には、「Name:」にDependents_Index、「Type:」にB-treeが指定されています。「Unique」と「Reverse」のチェック・ボックスがあります。「Index Keys:」というセクションがあり、「Key」にはDependent_ID、「Order」にはAscendingがそれぞれ指定されています。「Index properties」の下に「Add」ボタンと「Remove」ボタンがあり、これを使用してプロパティをさらに追加できます。

この図には、次のコードが含まれています。

```
CREATE INDEX "DEPENDENTS_INDEX"  
ON "DEPENDENTS" (  
    "DEPENDENT_ID");
```

この図は「Server Explorer」ツリーを示しています。「HR. ORCL」スキーマの「Tables」、「Indexes」、「DEPENDENTS_INDEX」が開かれています。

この図は、「Constraints」タブが選択された状態の「Table Designer」画面を示しています。関連性のない2つのシステム索引と、この例の索引（「Name:」がEMPLOYEES_FK、「Type:」がForeign Key）が1つあります。右側には「Constraints Properties」があり、「Type:」がForeign KeyのEMPLOYEES_FKが表示されており、「Deferrable」、「Enabled」（選択された状態）、「Validate」（選択された状態）の各チェック・ボックスがあります。無効化された「Execution」フィールドがあります。「Select the foreign key constraint」というセクションがあり、「Table:」にはEMPLOYEES、「Constraint:」にはEMP_EMP_ID_PKが選択されています。「Association:」というグリッドがあり、「Referenced Column:」にはEMPLOYEE_ID、「Local Column:」にはEMPLOYEE_IDと表示されています。「On delete」というリストがあり、「Cascade」が選択されています。

この図は、前の制約と、「Name:」がDEPENDENTS_PKで「Type:」がPrimary Keyである新しい制約を示しています。「Constraint Properties」では、「Enabled」と「Validate」が選択されています。また、「Primary key columns:」にはDEPENDENT_IDが指定されており、「Using index:」にはDEPENDENTS_INDEXが選択されています。

この図には、次のコードが含まれています。

```
ALTER TABLE "DEPENDENTS" ADD (  
  CONSTRAINT "EMPLOYEES_FK"  
    FOREIGN KEY ("EMPLOYEEID")  
      REFERENCES "EMPLOYEES" (EMPLOYEE_ID)  
    ON DELETE CASCADE  
  ENABLE  
  VALIDATE);  
ALTER TABLE "DEPENDENTS" ADD (  
  CONSTRAINT "DEPENDENTS_PK"  
    PRIMARY KEY("DEPENDENTID")  
  USING INDEX "DEPENDENTS_INDEX"  
  ENABLE  
  VALIDATE);
```

「Add to Project」ボタン（無効化された状態）と「OK」ボタンがあります。

この図は、この手順の操作結果を示しています。この図の説明は、図の前の本文を参照してください。

この図は、「HR. ORCL」の「Tables」から「DEPENDENTS」が開かれた「Server Explorer」ツリーを示しています。表のメニューが開かれて、「Design...」、「Retrieve Data...」、「Add Trigger...」、「Query Window」、「Generate Script...」、「Generate Create Script to Project...」、「Privileges..」、「Copy」、「Delete」、「Refresh」および「Properties」が表示されています。

この図は、DEPENDENTS表の新しいデータと同じデータが含まれたグリッドを示しています。

この図は、次のメッセージを示しています。「Your project currently has no data sources associated with it. Add a new data source, then data-bind items by dragging from this window on forms or existing controls.」その下に、「Add New Data Source」というラベルの付いたリンクがあります。

この図の説明は、図の前の本文を参照してください。「Choose a Data Source Type」の下に、「Where will the application get data from?」というラベルの付いたフィールドがあります。選択用のアイコンがあり、「Database」、「Service」、「Object」とラベルが付いています。このフィールドの下には領域があり、アイコンを選択すると、それぞれの説明が表示されます。現在はデータベース・アイコンが選択され、次の説明が表示されています。「Lets you connect to a database and choose the database objects for your application. This option creates a dataset.」「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

この図については、本文を参照してください。データ接続リストの後に、他の選択可能な接続を検索するための参照ボタンがあります。その下に、秘密データが接続文字列に表示されることを許可するかどうかの確認メッセージがあります。「No」と「Yes」の選択オプションがあります。その下に、「Connection string」というラベルの付いたプラス記号 (+) があります。これを開くと、接続文字列が表示されます。

この図の説明は、図の前の本文を参照してください。

この図の説明は、図の前の本文を参照してください。「Tables」のツリーが表示され、「DEPENDENTS (HR)」が選択されています。その下には「DataSet name」というフィールドがあり、そこにはtableDependentsと入力されています。さらにその下には、「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

この図は、「Data Sources」ウィンドウを示しています。このウィンドウには「tableDependents」から開かれたツリーと、その下には「DEPENDENTS」表があります。

この図は、空のグリッドがある「Form1」を示しています。この列には、「LAST_NAME」、「FIRST_NAME」、「BIRTH_DATE」、「RELATIONSHIP」、「EMPLOYEE_ID」および「DEPENDENT_ID」が表示されています。この下には、本文でリストされているコンポーネントのラベル付きアイコンがあります。

この図は、接続名にHR. ORCLが指定された「Oracle Server Login」を示しています。パスワードはマスクされています。「Save password」チェック・ボックス、および「OK」、「Cancel」、「Help」の各ボタンがあります。

この図は、「Server Explorer」ツリーを示しています。「HR. ORCL」が開かれ、「Packages」ノードのコンテキスト・メニューが開かれています。このメニューには、「New Package...」、「Privileges...」、「Refresh」および「Properties」の各コマンドが表示されています。

この図は「New Package」ウィンドウをしています。このウィンドウの「Schema name:」フィールドにはHR、「Package name:」フィールドにはHR_DATA、「Authentication identifier:」フィールドにはDEFINERがそれぞれ指定されています。選択された状態の「Generate empty package body」チェック・ボックスがあります。その下には、パッケージにメソッドを追加するためのフィールドがあります。メソッドを追加するには、このフィールドの下にある「Add」ボタンをクリックします。メソッドの「Edit」および「Remove」セクションもあります。一番下には、「Preview SQL」、「OK」、「Cancel」および「Help」の各ボタンがあります。

この図は、「Add Method」ウィンドウを示しています。このウィンドウには3つのフィールドがあり、「Method name:」にはGETCURSORS、「Method type:」にはProcedureが指定されています。無効化されたフィールドの「Return type:」にはNUMBERが表示されています。本文で説明されているように、「Parameters:」領域ではパラメータの追加と削除を行い、「Parameter Details」フィールドではパラメータの詳細を追加します。

この図は、「HR. ORCL」が開かれた「Server Explorer」ツリーを示しています。「Packages」の下の「HR. DATA」が選択され、コンテキスト・メニューが表示されています。このメニューには、「Edit Package Specification」、「Edit Package Body...」、「Compile」、「Compile Debug」などの各コマンドが表示されています。

この図は、「HR. ORCL」が開かれた「Server Explorer」ツリーを示しています。「Packages」の下の「HR. DATA」が開かれ、「GETCursors」が選択されてコンテキスト・メニューが表示されています。このメニューには、「Run」、「Run Debug」、「Step Into」などのコマンドが表示されています。

この図は「Run Procedure」ウィンドウを示しています。このウィンドウには、「Parameters」というタイトルのグリッドがあります。その下には複数のフィールドがあり、「Name」にはDEP_ID、「Direction」にはIN、「Data Type」にはNUMBERが指定され、「Value」にはリストから60が選択されています。「OK」、「Cancel」、「Help」の各ボタンがあります。

この図は、「Parameters」というタイトルのグリッドを示しています。その下には複数のフィールドがあり、「Name」にはDEP_ID、「Direction」にはIN、「Data Type」にはNUMBERが指定され、「Value」にはリストから60が選択されています。2番目のデータ行が選択されています。このデータ行には、EMPLOYEES_C、OUT、REF CURSORが指定されており、「Value」には<Click here for Details.>という指示が表示されています。

その下にある別のグリッドには、選択した行で<Click here for Details.>をクリックするとアクティブ化される、問合せ結果が表示されています。

この図は、「Parameters」というタイトルのグリッドを示しています。その下には複数のフィールドがあり、「Name」にはDEP_ID、「Direction」にはIN、「Data Type」にはNUMBERが指定され、「Value」にはリストから60が選択されています。2番目のデータ行には、EMPLOYEES_C、OUT、REF CURSORが指定されており、「Value」には<Click here for Details.>という指示が表示されています。3番目のデータ行が選択されています。このデータ行には、DEPENDENTS_C、OUT、REF CURSORが指定されており、「Value」には<Click here for Details.>という指示が表示されています。

その下にある別のグリッドには、選択した行で<Click here for Details.>をクリックするとアクティブ化される、問合せ結果が表示されています。

この図は、「Form1」のフィールドとデータを示しています。「UserID」にはHR、「Password」にはXXXXX、「Data Source」にはORCLがそれぞれ入力されており、無効化された「Connect」ボタンがあります。その下には、「Salary」、「Commision_PC」、「Manager_ID」、「Department_ID」などを表示するグリッドがあります。さらにその下に「Save」ボタンがあります。

このスクリーンショットはVisual Studioの「File」メニューを示しています。「New」が開かれて「Project...」、「Web Site...」、「File...」および「Project from Existing Code...」が表示されています。

これは「New Web Site」のスクリーンショットです。「Templates:」という見出しがあります。また、「Visual Studio Installed templates」には「ASP.NET Web Site」、「ASP.NET Web Service」、「Empty Web Site」などのリストが表示され、「My Templates」には「Search online Templates...」が表示されています。

これらのリストの下には「Location:」があり、リストから「File System」および「C:\%HR_Projects\WebSite1」が選択されています。また、「Browse」ボタンがあります。その下の「Language」フィールドではリストから「Visual C#」が選択されています。一番下に「OK」ボタンと「Cancel」ボタンがあります。

このスクリーンショットは、Visual Studio IDE内のDefault.aspxのソース・コードを示しています。

ヘッダーで、リストから「Client Objects and Events」が選択されています。 その下には、タイトルがUntitled Pageのhtmlセクションを含むコードが示されています。 一番下には、「Source」から「Design」または「Spilt」ビューに切り替えるためのタブがあります。

これは、設計ビューのDesign.aspxのスクリーンショットです。 メイン・ウィンドウには点線の四角形があり、Employees Listという文字列が含まれています。

このスクリーンショットはVisual Studioの「View」メニューを示しています。「Code」、「Designer」、「Markup」などが含まれています。メニューの一番下の「Toolbox」が選択されています。

このスクリーンショットは「Toolbox」を示しており、「Data」グループが開かれて「GridView」が選択されています。

このスクリーンショットは前の図と同様ですが、右矢印がある点が異なります。「GridView Tasks」のリストが表示されています。「Choose Data Source」フィールドがあり、「(None)」と、その下に「<New data source...>」が表示されています。

このスクリーンショットは、Column0、Column1などダミーのタイトルと、ダミーの内容のabcが含まれるグリッドを示しています。列タイトルの横に表示される右矢印を使用すると「GridView Tasks」リストが表示されます。

このスクリーンショットは「Data Source Configuration」ウィザードを示しています。「Choose a Data Source Type」でデータソースの種類の選択が要求されます。「Access Database」、「Database」、「LINQ」、「Object」、「Site Map」および「XML file」のうちから選択します。「Database」が選択されています。これを選択した場合、このリストの下に「Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.」と表示されます。さらに下にデータソースのIDを入力するフィールドがあり、HRが指定されています。一番下に「OK」ボタンと「Cancel」ボタンがあります。

このスクリーンショットは、「Choose Your Data Connection」でデータ接続の選択が要求されることを示しています。 リストから「HR. ORCL」が選択されています。 新しい接続を指定するための「New Connection」ボタンもあります。

一番下に「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

このスクリーンショットでは、アプリケーション構成ファイルに保存する接続文字列が要求されます。最初に「Yes, save this connection as:」チェック・ボックスがあり、このフィールドに「ConnectionString」が入力されています。その下には「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

このスクリーンショットの冒頭には「Configure the Select Statement」と表示されています。「How would you like to retrieve data from your database?」と尋ねられ、2つのオプションが示されます。1つは「Specify a custom SQL statement or store procedure」です。もう1つは「Specify columns from a table or view」で、「Names」に表またはビューのリストが含まれています。名前として「EMPLOYEES」が表示されています。その下に、選択する列が示されます。現在はアスタリスク (*) が選択されています。その他に、EMPLOYEE_ID、FIRST_NAME、LAST_NAMEなどが含まれています。

その下に、本文で説明されているSELECT文が示されます。「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

このスクリーンショットは、「Test Query」を示しています。最初に「To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.」が表示されます。

その下には、空のフィールドがあります。その下に「Test Query」ボタンがあり、さらに下には本文で説明されているSELECT文が示されています。

「Previous」、「Next」、「Finish」、「Cancel」の各ボタンがあります。

これも「Test Query」画面ですが、空だったフィールドにこの画面ではEMPLOYEE_ID、FIRST_NAME、LAST_NAMEなどの従業員関連の列と、先頭から5人の従業員のデータが表示されています。

このスクリーンショットは再び「GridView」を示しています。ここでは、Column01などではなく、EMPLOYEES表のEMPLOYEE_ID、FIRST_NAMEなどの列名が表示されています。「GridView Tasks」ページが表示され、実行可能ないくつかのタスクが示されています。「Enable Paging」（選択された状態）、「Enable Sorting」（選択されていない状態）、「Enable Selection」（選択されていない状態）の各チェック・ボックスがあります。

このスクリーンショットは、本文で説明されているブラウザを示しています。 このページの1行目は「Employees List」です。 一番下に、23456などの番号が表示され、表示するページ間を変更できます。

このスクリーンショットはSolution Explorerを示しています。「Solution WebSite1」、「C:¥HR_Projects¥WebSite1」が順に開かれて、「App_Data」、「Default.aspx」および「web.config」が表示されています。

このスクリーンショットでは、「C:¥HR_Projects¥WebSite1」が選択されてメニューの「Add New Item...」が選択されています。

このスクリーンショットは、「Add New Item - C:\¥HR_Projects¥WebSite1¥」を示しています。「Templates:」という見出しがあります。また「Visual Studio Installed templates」には、「Web Form」、「Web User Control」、「Master Page」などのリストが表示されます。

その下の「Name」フィールドには、login.aspxが入力されています。「Language」フィールドではリストから「Visual C#」が選択されています。「Place code in separate file」（選択された状態）および「Select master page」（選択されていない状態）の2つのチェック・ボックスがあります。一番下に「Add」ボタンと「Cancel」ボタンがあります。

このスクリーンショットの説明は、本文を参照してください。 このログイン・コントロールには、「Login In」という見出しおよび「User Name」と「Password」の入力フィールドが含まれています。「Remember me next time」チェック・ボックスおよび「Login In」ボタンがあります。

また、右側には矢印があり、これをクリックすると「Login Tasks」に「Auto Format...」、「Convert to Template」および「Administer Website」が表示されます。

このスクリーンショットは「Properties Windows」を示しており、その中の「DestinationPageUrl」が選択されています。 その下には、「Select URL」という別のウィンドウがあります。 「Project folder」の内容として、Default.aspx（必ず選択されている）、login.aspxおよびweb.configが示されています。

このスクリーンショットは「Website」メニューを示しており、その一番下の「ASP.NET Configuration」が選択されています。

このスクリーンショットは「ASP.NET Web Site Administration Tool」を示しています。「Home」、「Security」、「Application」および「Provider」タブが表示されています。 前面の「Home」タブには他のタブへのリンクがあります。

このスクリーンショットは「Provider」ページを示しています。「Select a single provider for all site management」および「Select a different provider for each feature (advanced)」の2つのリンクがあります。

このスクリーンショットには2つの見出しがあります。1つ目は「Membership provider」で「OracleMembershipProvider」および「AspNetSqlMembershipProvider」の2つの選択肢があります。2つ目の見出しは「Role Provider」で「OracleRoleProvider」、「AspNetSqlRoleProvider」および「AspNetWindowsTokenRoleProvider」の3つの選択肢があります。

このスクリーンショットは「Security」ページを示しています。「Users」という見出しの下には「Select authentication type」リンクが1つ、「Roles」という見出しの下には「Enable roles」リンクが1つ、「Access Roles」という見出しの下には「Create access rules」と「Manage access rules」の2つのリンクがあります。

このスクリーンショットは、「How will users access your site?」と尋ねる画面を示しています。「From the internet」と「From a local network」の2つのオプションがあります。一番下に「Done」ボタンがあります。

このスクリーンショットは、前にも示した「Security」ページですが、ここでは「Users」という見出しの下に「Existing Users: 0」と示され、「Create user」および「Manage users」の2つのリンクがあります。

このスクリーンショットは「Security」ページの「Create User」セクションを示しています。「User Name」、「Password」、「Confirm Password」、「E-mail」、「Security Question」、「Security Answer」の各フィールドがあります。「Active user」チェック・ボックスおよび「Create User」ボタンがあります。このスクリーンショットには、「Create User」セクションの横にもう1つのセクションがあります。「Roles」セクションです。表示されるテキストは「Roles are not enabled」だけです。

この「Security」ページの「Create User」見出しの下には、「Complete. Your account has been successfully completed.」というテキストが表示されています。「Continue」ボタンがあります。「Roles」セクションもありますが依然として「Roles are not enabled」と示されたままです。

このスクリーンショットは前にも示した「Security」ページですが、ここでは「Users」という見出しの下に「Existing Users: 1」と示されています（以前は0）。「Create user」および「Manage users」の2つのリンクがあります。

このスクリーンショットは「Manage Access Rules」セクションを示しています。ツリー・ビューの「WebSite1」が開かれて「App_Data」フォルダが表示されています。このツリー・ビューの横にルールを示すパネルがあります。「Permission」にはallow、「User and Roles」にはall、「Delete」にはDeleteが表示されています。

その下に、「Add new access rule」へのリンクがあります。

このスクリーンショットは「Add New Access Rule」セクションを示しています。このセクションはいくつかの部分に分かれています。まず「Select directory for this rule:」があります。ツリー・ビューの「WebSite1」が開かれて「App_Data」フォルダが表示されています。2つ目の部分は「Rule applies to:」で、「Role」（リストを含む）または「User」（入力フィールドを含む）を選択するオプションがあります。その下に「Search for users」へのリンクがあります。その下に「All users」または「Anonymous users」を選択するオプションがあります。最後の部分は「Permission:」で、「Allow」および「Deny」（選択された状態）オプションがあります。一番下に「OK」ボタンがあります。

このスクリーンショットの説明は、本文を参照してください。

このスクリーンショットでは、「Address」に、末尾がDefault.aspxであるアドレス（http://localhost:1451/WebSite1/Default.aspx）が示されています。実際のページではログインが示されます。「User Name」および「Password」の入力フィールド、「Remember me next time」チェック・ボックス（選択されていない状態）、および「Log In」ボタンがあります。

このスクリーンショットの「Address」のアドレスは末尾がDefault.aspxですが、[http://localhost:1451/WebSite1/login.aspx?ReturnUrl= %2fWebSite1%2fDefault.aspx](http://localhost:1451/WebSite1/login.aspx?ReturnUrl=%2fWebSite1%2fDefault.aspx)に変更されています。 実際のページは、前のスクリーンショットと同じです。

このスクリーンショットは直前のショットとほぼ同じですが、本文で説明されているメッセージが表示され、アドレスがhttp://localhost:1451/WebSite1/login.aspxに戻っている点が異なります。

「User Name」はAnneで、パスワードは表示されていません。このログイン画面には、「Your login attempt was not successful. Please try again.」というメッセージが表示されます。

この図の説明は、図の前の本文を参照してください。下部に、「Advanced...」、「Test Connection」、「OK」および「Cancel」の各ボタンがあります。

この図の左側には「Project types」が表示され、「Visual C#」の下にある「Database」が選択されています。その右側では、「Visual Studio installed templates」の下で「Oracle Project」が選択されています。下部には、本文で説明されているように、「Name」、「Location」、「Solution」の各フィールドがあります。「Location」には参照ボックスがあります。選択された「Create directory for solution」チェック・ボックスと、選択解除された「Add to Source Control」チェック・ボックスがあります。最後に「OK」ボタンと「Cancel」ボタンがあります。

この図はClass1.csのコードを示しています。ここには、パブリック・クラスClass1の最初の部分が表示されています。その下には、上にリストされているgetDepartmentnoファンクション用のコードの最初の6行が表示されています。

この図は、「Build Solution」コマンドが選択された状態の「Build」メニューを示しています。

この図は、「Deploy Solution」コマンドが選択された状態の「Build」メニューを示しています。

これは、Oracle Deployment Wizardの最初のウィンドウです。

「このページを再表示しない。」を選択するボックスがあります。また、「戻る」、「次へ」、「終了」および「取消」の各ボタンもあります。

このウィンドウには、ウィザードでどのデータ接続を使用するかを尋ねるメッセージが表示されています。現行の接続SYS. ORCLが表示されています。

「新規接続」ボタンがあります。

また、「戻る」、「次へ」、「終了」および「取消」の各ボタンもあります。

このウィンドウでは、デプロイメント・オプションを指定する必要があります。このオプションには、「アセンブリをコピーしてストアード・プロシージャを生成」（選択されている状態）、「アセンブリのみコピー」、「ストアード・プロシージャのみの生成」があります。

また、「戻る」、「次へ」、「終了」（グレーアウトされている状態）および「取消」の各ボタンもあります。

このウィンドウでは、アセンブリおよびライブラリ名を指定する必要があります。

最初に、ソース・アセンブリの場所を指定します。「プロジェクト」のオプションが選択されています。

次に、選択したアセンブリに使用するライブラリ・データベース・オブジェクトの名前を指定します。ドロップダウン・リストには「HR_DEPLOYSTORED_CS_DLL」が表示されています。

また、「戻る」、「次へ」、「終了」および「取消」の各ボタンもあります。

このウィンドウでは、データベースにコピーする必要がある依存アセンブリを選択します。ここにリストされているアセンブリは、mscorlib、Oracle.DataAccess、SystemおよびSystem.Dataです。

次に、接続先がデフォルトのbin/clrディレクトリでない場合は、bin/clrディレクトリの接続先を指定します。この例では、このフィールドは空白です。

その下にチェック・ボックスがあり、ファイルがすでに存在する場合にそれを上書きするかどうかを選択できます。

「戻る」、「次へ」、「終了」、「取消」の各ボタンがあります。

このウィンドウでは、メソッドとセキュリティの詳細を指定します。左側のテキスト・ボックスに使用可能なメソッドが表示され、「HR_DeployedStored_CS」、その下の「Class1」、および「Int32 GetDepartmentNo」が選択されています。右側に「メソッドの詳細」というタイトルがあり、その下の「スキーマ」のドロップダウン・リストに、選択しているスキーマとして「HR」が表示されています。その下に「データベース・メソッド」というテキスト・フィールドがあり、「GETDEPARTMETNTNO」が表示されています。その下に、「パラメータ・タイプ・マッピング」というラベルの付いたボタンがあります（これをクリックすると、その画面に移動します）。

その下に、「セキュリティ」を設定するためのドロップダウン・リストがあります。デフォルト・レベルの「安全」が入力されています。「戻る」、「次へ」、「終了」、「取消」の各ボタンがあります。

「サマリー」ウィンドウが表示され、ウィザードに必要な情報の指定が完了したことが示されています。

HR_DeployedStored_CS.dllへの完全なアセンブリ・パスおよびプロジェクト名HR_DeployedStored_CSを含むサマリーが表示されています。

このスクリーンショットでは、HR_DeployedStored_CS.dllがローカル・データベースの\$Oracle_HOME¥bin¥clrディレクトリにコピーされ、1つのストアド・プロシージャがローカル・データベースにデプロイされることが示されています。

「戻る」、「次へ」、「終了」、「取消」の各ボタンの他に、「スクリプトの表示」ボタンがあります。

この図の説明は、図の前の本文を参照してください。

この図は、「Parameters」というラベルの付いたグリッドを示しています。この列とデータは、「Name」列がEMPLOYEE_ID、「Direction」列がIN、「Data Type」列がBINARY_INTEGER、「Value」が100です。また、「OK」、「Cancel」、「Help」の各ボタンがあります。

この図は、本文で説明されている情報が追加されたグリッドを示しています。

この図は、問合せウィンドウを示しています。入力セクションには、本文で説明されているSELECT文が表示されており、問合せ結果セクションには、その結果の90が表示されています。

この図は、本文で示されているコマンドと、SQL*Plusが起動したときに表示される標準的なレスポンス・メッセージを表示しています。レスポンス・メッセージには、リリース番号と著作権情報が表示されます。続いて、説明されているようにパスワードの入力が要求されます。次に、「Connected to: Oracle DatabaseEdition Releaseに接続されました。」というレスポンス・メッセージが表示されます。その後の行には、SQL*PlusプロンプトのSQL>が表示されます。

この図は、IDが105未満の従業員を表示しています。最初の行のデータは、「ID」列が100、「Name」列がS. King、「Short Hire」列が6/17/1987、「Long Hire Date」列がWednesday, June 17, 1987です。

この図は、IDが105未満の従業員を表示しています。最初の行のデータは、「ID」列が100、「Name」列がS. King、「Short Hire」列が17.06.1987、「Long Hire Date」列がMittwoch, 17 Juni, 1987です。

この図は、IDが105未満の従業員の給与を表示しています。最初の行のデータは、「ID」列が100、「Name」列がS. King、「Salary」列が24,005.00です。

この図は、IDが105未満の従業員の給与を表示しています。最初の行のデータは、「ID」列が100、「Name」列がS. King、「Salary」列が24.005,00です。

この図は、Cで始まる名前とそのIDのリストを表示しています。名前は、Cabrio、Cambrault、Cambrault、Chen、Chung、Colmenaresの順に表示されています。

この図は、Cで始まる名前とそのIDのリストを表示しています。名前は、Cabrio、Cambrault、Cambrault、Colmenares、Chen、Chungの順に表示されています。

この図は、次のメッセージを表示しています。

行%dでエラーが発生しました。: ORA-00942: 表またはビューが存在しません。

この図は、次のメッセージを表示しています。行%dでエラーが発生しました。: ORA-00942: Table ou vue inexistante.

この図は、「Form1」を示しています。このフォームには、「UserId」、「Password」、「Data Source」の各ラベルと各フィールドがあります。「Connect」、「Save」、「Change Date Format」の各ボタンがあります。また、このフォームが実行されたときに結果を含めるための大きな空のフィールドがあります。

この図は、HR_Connect_CS.Form1のコード・ウィンドウを示しています。このウィンドウには、次の行が表示されています。

```
public Form1()  
{  
    InitializeComonent();  
    Save.Enabled = false;  
    date_change.Enabled = false;  
}
```

この図は、HR_Connect_CS.Form1のコード・ウィンドウを示しています。このウィンドウには、次の行が表示されています。

```
public Form1()
{
    InitializeComponent();
    Save.Enabled = false;
    date_change.Enabled = false;
}
private OracleGlobalization si;
private OracleConnection conn = new OracleConnection();
private OracleCommand cmd;
```

この図は、IDが105未満の従業員を表示しています。最初の行のデータは、「EMPLOYEE_ID」列が100、「FIRST_NAME」列がSteven、「LAST_NAME」列がKING、「Hire Date」列が17-Jun-87です。

この図は、IDが105未満の従業員を表示しています。最初の行のデータは、「EMPLOYEE_ID」列が100、「FIRST_NAME」列がSteven、「LAST_NAME」列がKING、「Hire Date」列が1987-06-17です。

この図は、HR_CONNECT_CSプロジェクトのSolution Explorerを示しています。ここでは「Form1.cs」が選択され、開かれています。コンテキスト・メニューが開かれ、「Open」、「Open With...」、「View Class Diagram」などのコマンドが表示されています。

この図は、「HR_CONNECT_CS」プロジェクトが選択された状態のSolution Explorerを示しています。コンテキスト・メニューが開かれ、「Build」、「Rebuild」、「Clean」、「Publish」、「Add」などのコマンドが表示されています。

この図は、HR_CONNECT_CSプロジェクトのSolution Explorerを示しています。ここでは「Copy of Form1.cs」が選択され、開かれています。コンテキスト・メニューが開かれ、「Open」、「Open With...」、「View Class Diagram」、「Include in Project」などのコマンドが表示されています。

この図は、HR_CONNECT_CSプロジェクトのSolution Explorerを示しています。ここでは「Form2.cs」が選択され、開かれています。コンテキスト・メニューが開かれ、「Open」、「Open With...」、「View Class Diagram」、「Include in Project」などのコマンドが表示されています。