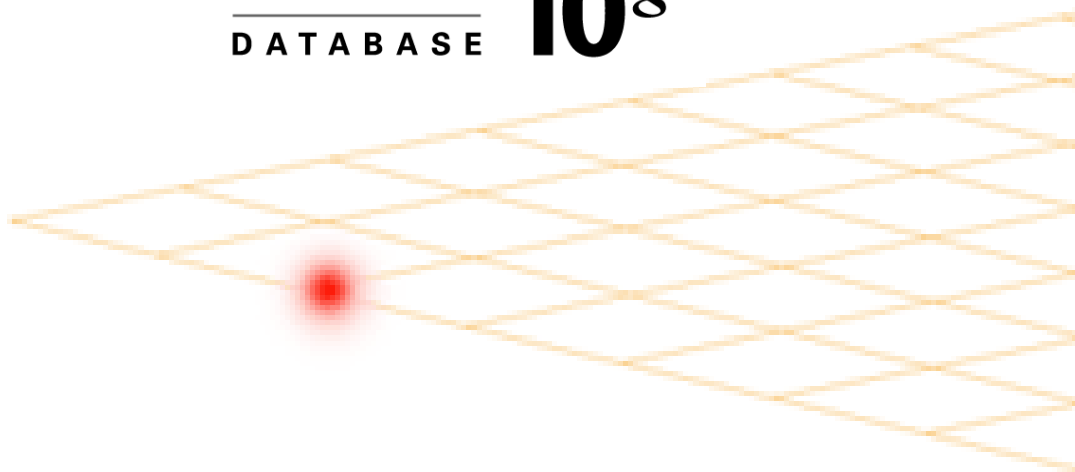


GrapeCity コンポーネント活用

- .NET 開発 with GrapeCity -

「SPREAD 編」

ORACLE®
DATABASE 10^g



Creation Date:	Nov. 30, 2005
Last Update:	Nov. 30, 2005
Version:	1.0



Document Control

Internal Use Only

Author

Hiroshi Ota

Change Logs

Date	Author	Version	Change Log
Nov. 30, 2005	Hiroshi Ota		Created.

Reviewers

Name	Position
------	----------

Approvals

<Approver 1> _____

<Approver 2> _____

Distribution

Copy No.	Name	Location
----------	------	----------

はじめに

「GrapeCity コンポーネント活用」シリーズは、Microsoft Visual Studio.NET と VB ユーザーや.NET ユーザーから高い支持を受けているグレープシティ社のコンポーネント製品を組み合わせた Oracle 対応アプリケーションの開発方法について説明します。

実際のサンプルアプリケーションを提供することにより、単なるコーディング Tips にとどまらず、より実践的なアプリケーション開発の資料として構成するようにしております。今回は SPREAD 編ということで、グレープシティ社が提供している SPREAD と Oracle が提供している.NET から Oracle データベースヘネイティブに接続するためのミドルウェアである「Oracle Data Provider for .NET(以下、ODP.NET)を利用したアプリケーションの開発方法について、実際のサンプルアプリケーションを動かしながら確認することができます。今回のサンプルアプリケーションの説明はポイントとなる部分のみの説明になりますので予めご了承ください。

「意外と簡単!? .NET で Oracle」シリーズが.NET 開発者でオラクルを利用したい方のシステム構築の一助になれば幸いです。

「GrapeCity コンポーネント活用」シリーズは以下の4つの構成を予定しています。

1. ActiveReports 編
2. InputMan 編
3. SPREAD 編 (本書)
4. FlexGrid 編

「GrapeCity コンポーネント活用」シリーズの「SPREAD 編」は、以下の3つの内容から構成しております。

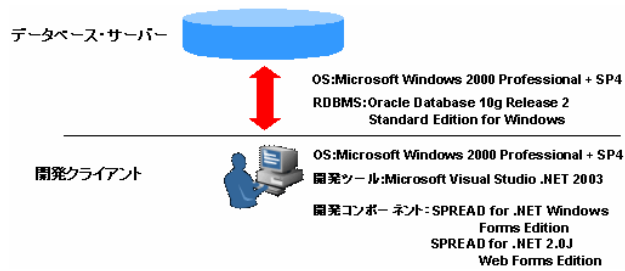
1. SPREAD とは
2. サンプルアプリケーションのインストール
3. SPREAD と ODP.NET を利用した開発

「GrapeCity コンポーネント活用」シリーズにおける開発環境

- ・ データベース・サーバー
 - OS : Microsoft Windows 2000 Professional + SP4
 - RDBMS : Oracle Database 10g Release 2 Standard Edition for Windows

・ 開発クライアント

- OS : Microsoft Windows 2000 Professional + SP4
- 開発ツール : Microsoft Visual Studio .NET 2003
- 開発コンポーネント : GrapeCity SPREAD for .NET Windows Forms Edition
SPREAD for .NET 2.0J Web Forms Edition



SPREAD とは

SPREAD とは、グレースシティ社から提供されている表計算コンポーネントになります。主に以下の特徴があります。

豊かな表現力

マルチヘッダ表示やセル結合、データに応じた自動マージなどの機能のほか、マルチシート、アルファブレンディングやカスタマイズ可能な罫線などが加わり、すぐれた表現力を持っています。

エンドユーザーにやさしい操作性

アルファベットや数字入力のほか、日本語入力のための、ひらがな・カタカナ・半角カタカナの入力モードを提供しています。半角英数字のみの入力制御もプロパティの設定だけで簡単に実現できます。テキストセルではそれぞれの入力モードに合わせてIMEモードも自動的に切り替わり、エンドユーザーに自然な入力環境を提供します。

また、列や行の固定表示とは別に、SPREAD シートを画面上で複数に分割することができるマルチスプリッタ機能やシート上の表示サイズを自由に変更できるズーム機能を装備します。

強力なデータベースサポート

ADO.NET とシームレスに連携するデータモデルの採用により、リレーションに応じたデータの階層表示や、高速なデータ接続を実現しました。本格的なデータベースアプリケーションの開発にも高いパフォーマンスを発揮します。また、.NET 環境から Oracle データベースへのネイティブ接続が可能な Oracle Data Provider for .NET(以下、ODP.NET)にも対応しています。

多彩な印刷機能

標準的なグリッド印刷や、印刷設定に応じて自動的に用紙方向やサイズ調整を行うスマート印刷、複数の SPREAD を同一ページに印刷できるユーザー定義印刷の3つの印刷機能を提供します。目的に合わせて適切な印刷方法を選択することができます。

このほか IDE と同様のプロパティウィンドウにより、SPREAD 本体、SPREAD シート、行、列、ヘッダおよびセルの各レベルにおけるプロパティの設定をすべてデザイナー上でおこなうことができます。ユーザーインタフェースデザインのためのコーディング作業時間を大幅に短縮できます。SPREAD のより詳細な情報はグレースシティ社の以下の URL を参照してください。

グレースシティ社 - .NET 製品情報

http://www.grapecity.com/japan/support/database/DOTNET_ProductList.htm

今回は SPREAD の 2006 年 1 月時点で最新版のバージョンである、「SPREAD for .NET Windows Forms Edition」と「SPREAD for .NET 2.0J Web Forms Edition」を使用しております。

サンプルアプリケーションのインストール

サンプルアプリケーションは以下のサイトよりダウンロードできます。

<http://otndnld.oracle.co.jp/easy/dotnet/spread/sample.zip>

ダウンロードファイル(sample.zip)を任意の一時フォルダに保存し、解凍してください。解凍しますと「CONFIG」、「WebForms」、「WindowsForm」という3つのフォルダが作成されます。それぞれのフォルダには以下の情報が格納されています。

CONFIG

サンプルアプリケーションを動作させるために必要なテーブルを作成するためのスクリプトが格納されています。

WebForms

ASP.NET で作成されたサンプルアプリケーションが格納されています。

WindowsForm

VB.NET で作成された Windows フォームでのサンプルアプリケーションが格納されています。

Oracle データベース側の準備

サンプルアプリケーションを動作させるために、Oracle データベース側にテスト用のテーブルを作成します。テストテーブルの作成と必要な初期データの作成は全てスクリプトで自動生成されます。自動生成のためのスクリプトは sample.zip ファイルを解凍後に作成されるフォルダ「CONFIG」下の「gen_SPREAD.sql」になりますので、こちらを「SQL*PLUS」などから実行します。以下、実行結果になります。

```
C:¥>sqlplus /nolog

SQL*Plus: Release 10.2.0.1.0 - Production on 木 2月 2 10:43:52 2006

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> @gen_SPREAD
```

```
system ユーザーのパスワードを入力してください:
```

```
接続されました。
```

```
ユーザーが作成されました。
```

```
権限付与が成功しました。
```

```
接続されました。
```

```
表が作成されました。
```

```
~ 途中省略 ~
```

```
1行が作成されました。
```

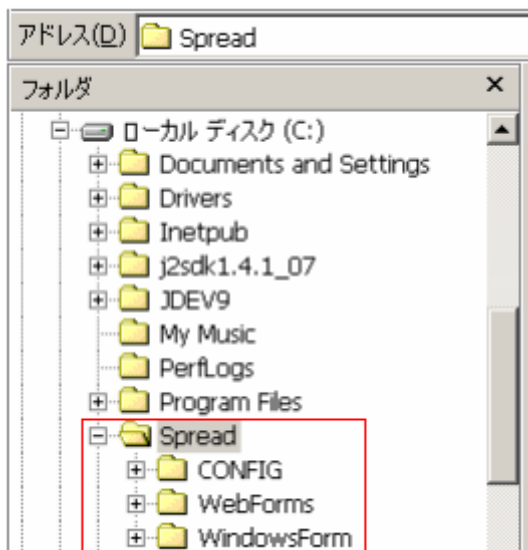
```
1行が作成されました。
```

```
SQL>
```

「gen_SPREAD.sql」を実行すると、最初に system ユーザーのパスワードを入力する必要がありますのでパスワードを入力してください。以上で Oracle データベース側に必要な情報が生成されました。次に IIS の設定をおこないます。

Internet Information Services(IIS)の設定 (アプリケーション・サーバー上での設定)

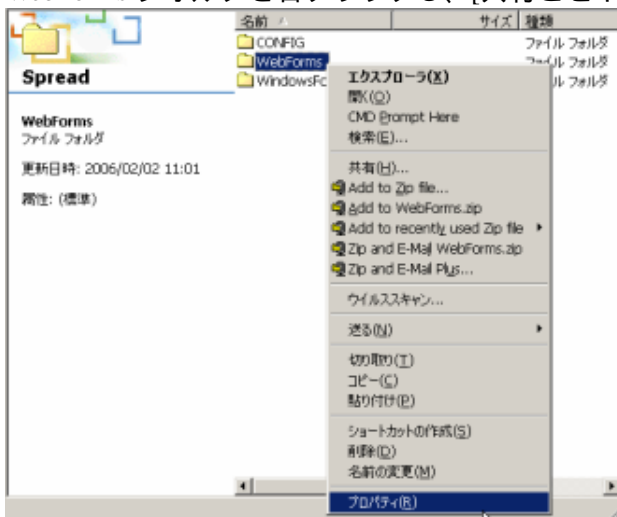
ダウンロードファイル (sample.zip) を解凍後に作成される WebForms ディレクトリを Internet Information Services(IIS)から参照できるように以下の手順に従って設定します。今回は sample.zip ファイルを「C:\SPREAD」ディレクトリに展開したという想定で説明をおこないます。解凍後のディレクトリ構成は以下のようになります。



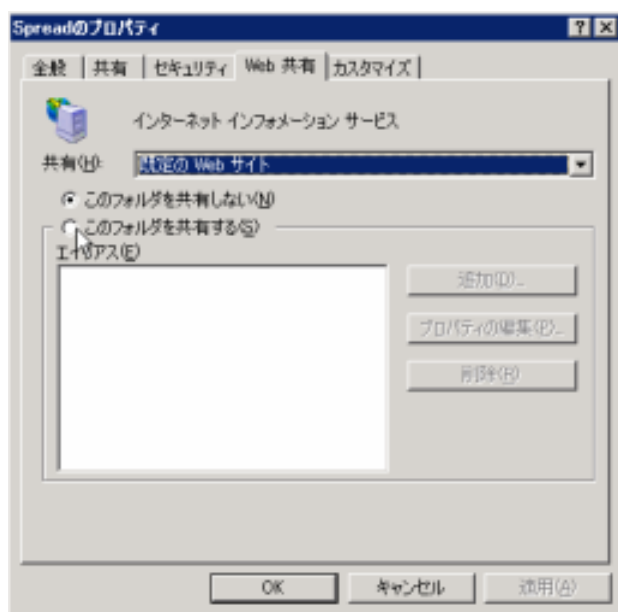
「WebForms」フォルダに格納されている ASP.NET WEB アプリケーションを IIS に登録する手順は以下ようになります。

1. 仮想ディレクトリの作成

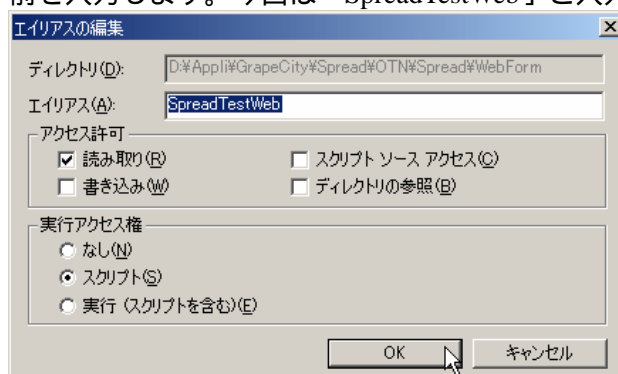
Windows エクスプローラを使用して、IIS から利用する仮想ディレクトリを作成します。WebForms フォルダを右クリックし、[共有とセキュリティ]をクリックします。



フォルダプロパティの [WEB 共有] タブの [共有] ボックスで、「仮想ディレクトリを作成する」を選択します。



[エイリアス編集] ダイアログボックスの [エイリアス] ボックスに、仮想ディレクトリの名前を入力します。今回は「SpreadTestWeb」と入力します。



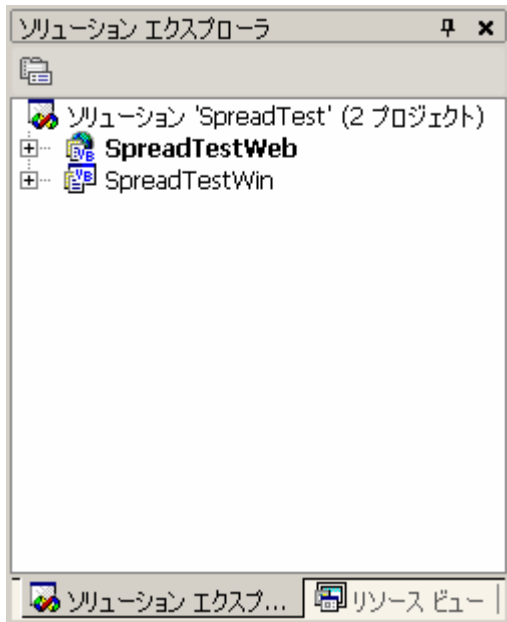
「OK」ボタンをクリックすることにより仮想ディレクトリの設定が完了します。

サンプルアプリケーションの動作確認

IIS 上での設定が終わりましたら、次にサンプルアプリケーションの動作確認をおこなってみましょう。サンプルアプリケーションを動作させるためには SPREAD のコンポーネントが既にインストールされている必要があります。こちらのインストール方法については製品のヘルプ、もしくは GrapeCity 社のサイトを参照してください。SPREAD の製品が正常にインストールされた環境で、ダウンロードファイル(sample.zip)を解凍後に作成される SPREAD ディレクトリ下にある、VS.NET のソリューションファイル「SpreadTest.sln」を VS.NET から開いてください。SpreadTest.sln ファイルには以下の 2 つのプロジェクトが含まれているのが、ソリューション・エクスプローラから確認できます。

SpreadTestWeb (ASP.NET WEB アプリケーション)

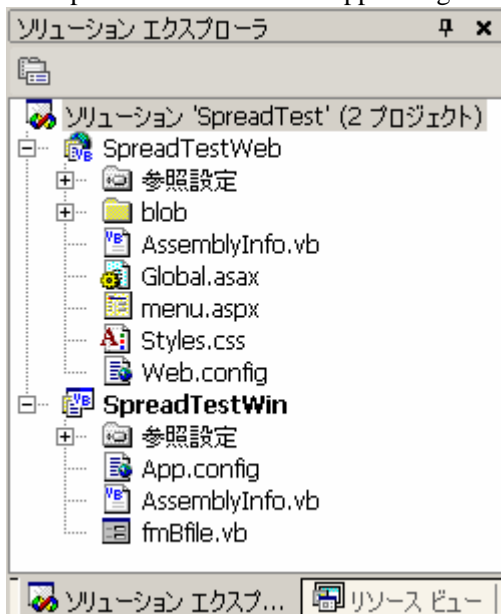
SpreadTestWin (Windows アプリケーション)



サンプルアプリケーションの動作確認をおこなう前に、Oracle データベースに接続するための接続文字列を変更します。接続文字列は、以下のプロジェクト毎にそれぞれ記述されています。

SpreadTestWeb - > Web.config

SpreadTestWin - > App.config



上記のファイルは XML で記述されています。<appSettings>タグに connectionString キーの値が格納されているので、この箇所の値を変更します。変更箇所は、data source の値を tnsnames.ora ファイルに記述された、ネットサービス名になります。

```
<add key="connectionString"
value="user id=spread; password=oracle; data source=orcl10gr2" />
```

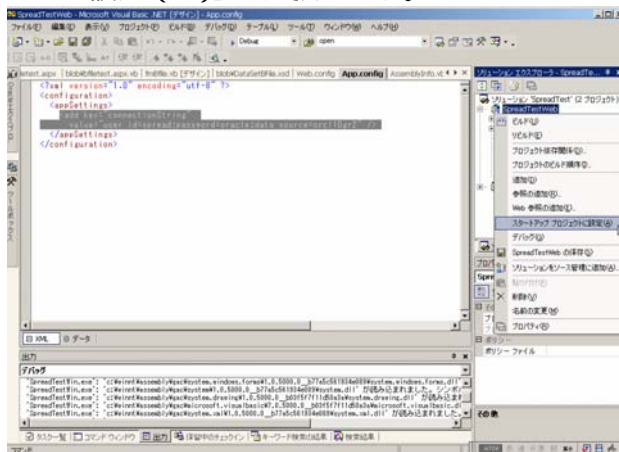
メモ：tnsnames.ora ファイルについての詳細は、以下のサイトを参照してください。

Oracle Database Net Services 管理者ガイド

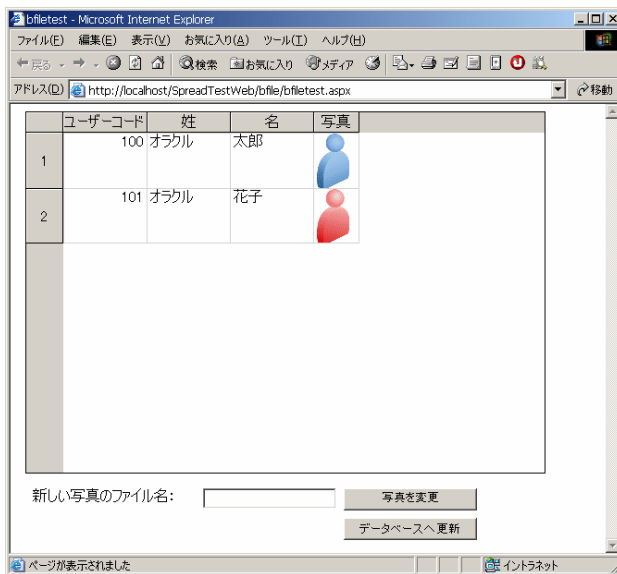
10g リリース 2 (10.2) ネーミング・メソッドの構成

http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_v8/nav/portal_3.htm

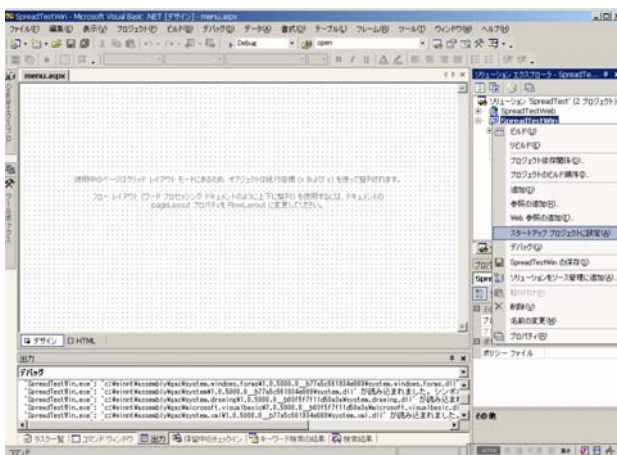
接続文字列を適切に設定したら、次にアプリケーションを実行してみましょう。はじめに ASP.NET で作成された WEB アプリケーションを実行します。VS.NET のソリューション・エクスプローラから「SpreadTestWeb」プロジェクトを右クリックし、「スタートアップ プロジェクトに設定 (A)」を選択します。



次に、VS.NET のメニュー「デバッグ(D)」 -> 「開始(S)」を選択して実行します。



以上のように SPREAD で作成された WEB ページが確認できます。次に Windows アプリケーションで確認をおこなってみましょう。VS.NET のソリューション・エクスプローラから「SpreadTestWin」プロジェクトを右クリックし、「スタートアップ プロジェクトに設定 (A)」を選択します。



次に、VS.NET のメニュー「デバッグ(D)」 -> 「開始(S)」を選択して実行します。



以上のように SPREAD で作成された Windows フォームの画面が確認できます。以上でサンプルアプリケーションの動作確認をおこないました。次に、SPREAD から ODP.NET を利用するための方法について、コードを交えて説明します。

SPREAD から ODP.NET の利用

では、実際に SPREAD のコンポーネントから ODP.NET を利用して Oracle に格納された情報の表示をおこなってみましょう。

DataSet の内容を SPREAD で表示

今回のサンプルアプリケーションでは、Oracle データベースからデータを取得し DataSet に格納、DataSet の内容を SPREAD で表示、修正がおこなうサンプルアプリケーションになります。SPREAD を利用するクライアントアプリケーションは、Windows アプリケーションと ASP.NET WEB アプリケーションになります。まずは単一の表を SPREAD で表示する手順を説明します。Windows アプリケーションでの説明になりますが、基本的な手順は ASP.NET WEB アプリケーションも同様になります。

DataSet にデータを格納

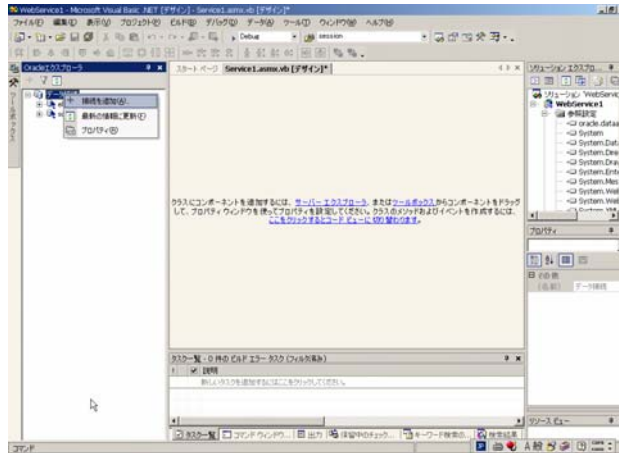
Oracle データベースからデータを取得し DataSet に格納する手順は以下のようになります。

1. Oracle データベースへの接続の設定

Oracle への接続するための接続の設定を行います。Oracle への接続を確立するためには、どの Oracle データベースへどんなユーザーで接続するかという情報を接続文字列として設定する必要があります。接続文字列はコードで記述することも可能ですが、Oracle Developer Tools for Visual Studio .NET(以下、ODT)を利用すると簡単です。ODT を利用した Oracle データベースへの接続について説明します。

Oracle エクスプローラにデータ接続の追加

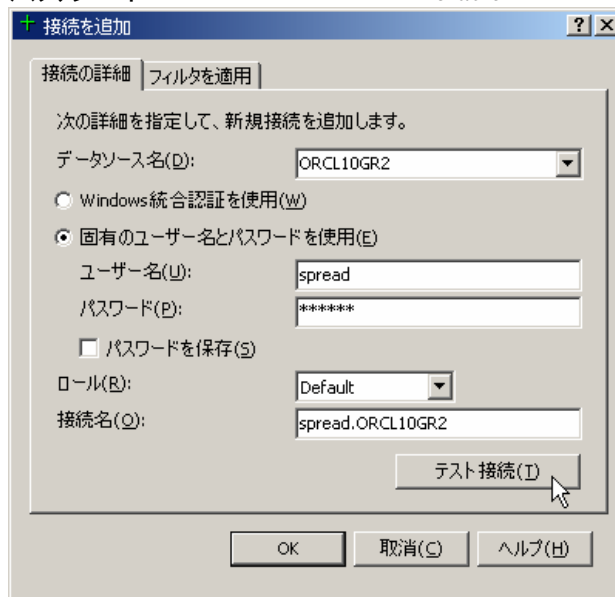
Oracle エクスプローラの「データ接続」を右クリックし、接続の追加(A)を選択します。



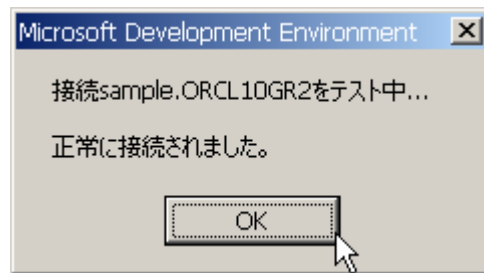
接続先情報の入力

Oracle データベースへの接続情報を入力します。

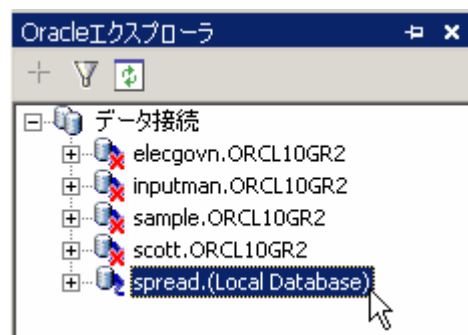
データソース名 <データソース名>
 ユーザー名 spread
 パスワード oracle



データソース名は tnsnames.ora ファイルに記述された、ネットサービス名になります。入力された情報が正しいか確認するために、「テスト接続(T)」をクリックすると、以下の画面が表示されます。

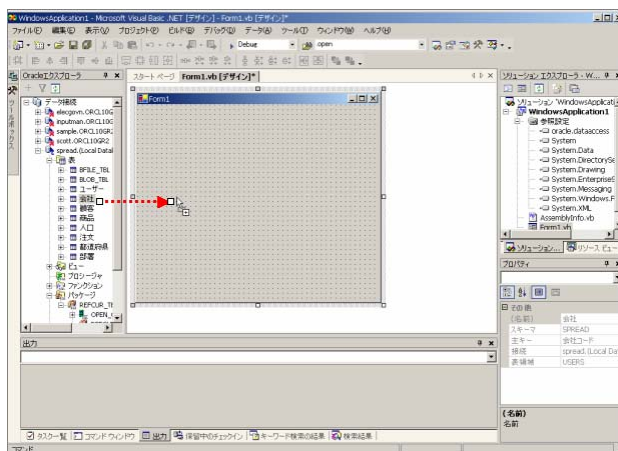


接続が失敗した場合は、再度接続文字列、ユーザー名、パスワードの内容が正しいか確認をおこなってください。テストが終了後、「接続を追加」ウィンドウで「OK」ボタンをクリックすると、「Oracle エクスプローラ」に接続情報が追加されます。



2. DataAdapter の設定

Oracle データベースへの接続情報の設定が完了したら、次に DataAdapter の設定をおこない、DataSet へ格納する情報の設定をおこないます。DataAdapter の設定も ODT を使用します。今回は、サンプルテーブルである「会社」テーブルを選択してデザインウィンドウにドラッグ・アンド・ドロップします。

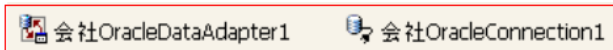


ドラッグ・アンド・ドロップ後に、パスワード保存のダイアログが表示されますので、「は

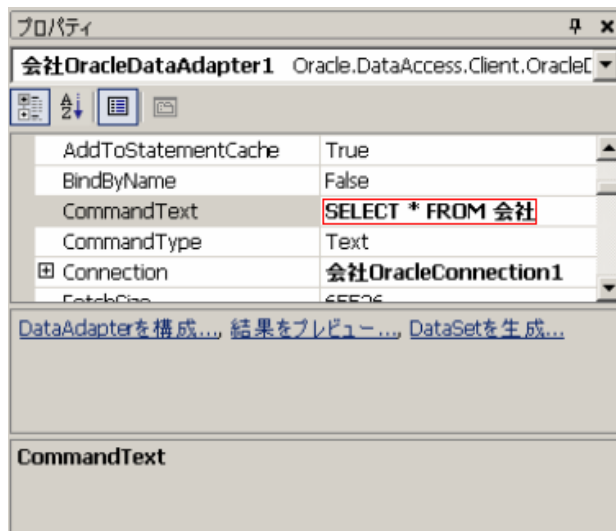
「はい(Y)」をクリックします。



以上の作業で DataAdapter の作成は完了です。DataAdapter オブジェクトと Connection オブジェクトが自動生成されていることが、デザインウィンドウで確認できます。

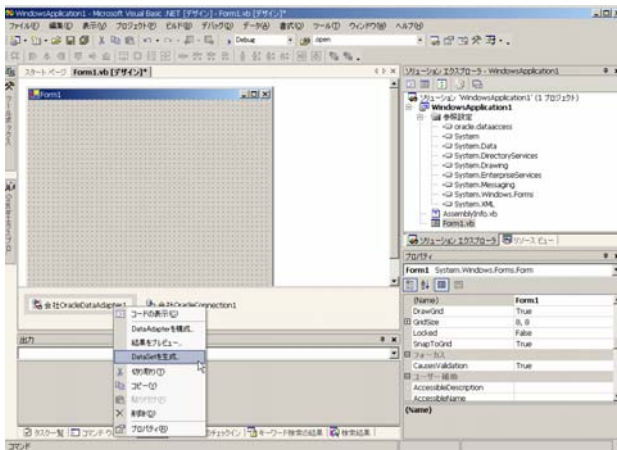


Oracle エクスプローラからテーブルをドラッグ・アンド・ドロップした場合には DataAdapter の SselectCommand、CommandText プロパティの内容はドラッグ・アンド・ドロップされた単一テーブルの SQL が設定されます。



3. DataSet へのデータの格納

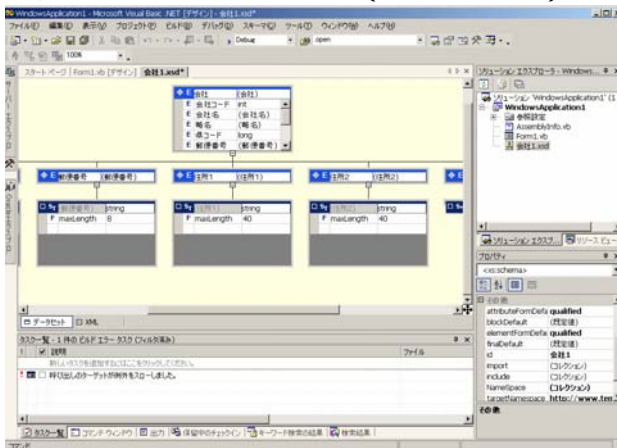
DataAdapter の設定が完了したら、次に DataSet にデータを格納するための手順について説明します。上記作業で作成した DataAdapter オブジェクトを右クリックし、「DataSet を生成 ...」を選択します。



以上の作業で DataSet オブジェクトが自動生成されたのがデザインウィンドウから確認できます。



また、ODT を使用して DataSet オブジェクトを生成すると、型付データセットとして生成され、XML スキーマファイル (xsd ファイル) も自動生成されます。



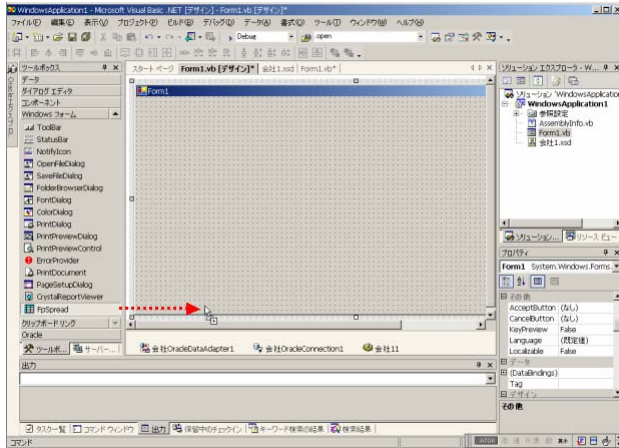
ODT により作成された DataAdapter から DataSet へのデータの格納は以下のようなコードになります。

```
会社 OracleDataAdapter1.Fill(会社 11.会社)
```

上記コードは DataAdapter の Fill メソッドを使用して DataSet にデータを格納しています。今までの説明で DataSet に格納するまでは ODT を利用することにより簡単におこなえることが確認できたと思います。次に DataSet の内容を SPREAD に表示してみましよう。

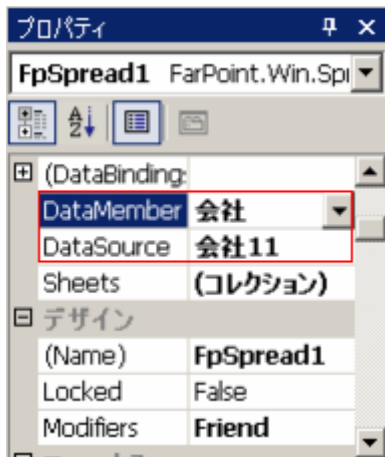
DataSet と SPREAD の連携

はじめに、SPREAD のコントロールを Windows フォームにドラッグ・アンド・ドロップします。



次に、SPREAD コントロールのプロパティを以下のように設定します。

- DataMember : 会社
- DataSource : 会社 11



VSメニューから、「デバッグ(D)」 -> 「スタート(S)」を選択して、アプリケーションを実行します。

	会社コード	会社名	略名	県コード	郵便番号
1	1	とまと青果		46	064-9999
2	2	榎木工所		46	061-9999
3	3	(株)北日本ペーパー商事		27	031-9999
4	4	(株)旭川鉄鋼工業		46	079-9999
5	5	宮城モロヘイヤ販売		9	983-9999
6	6	東北蜘蛛の巣漬物(株)		9	981-9999
7	7	(株)ローズ生花		9	981-9999
8	8	赤間雲母取引		37	166-9999
9	9	(株)高田ポック魚介		37	169-9999
10	10	(株)酒井鋸金リース		37	182-9999
11	11	(株)東京ベニヤ塗装	TBT	37	150-9999
12	12	江戸甘酒堂		37	108-9999
13	13	中央ヤマ研究所		37	164-9999
14	14	中川曲輪建設		37	103-9999
15	15	(株)石川腰痛薬品		37	194-9999

DataSet の内容が SPREAD に表示されるのが確認できます。

BLOB フィールドへの対応

画像などを扱う BLOB フィールドの内容も SPREAD で表示が可能です。今回のサンプルアプリケーションでは実際に BLOB フィールドに格納された画像を SPREAD で表示しています。ただ、BLOB フィールドに格納された画像データを SPREAD のセルに表示するのはコードを記述する必要があります。以下、Windows アプリケーション、ASP.NET WEB アプリケーションでそれぞれ BLOB フィールドを使用した場合のコーディングする際の注意点を説明します。

Windows アプリケーションでのコーディング

Windows アプリケーションで BLOB フィールドに格納された画像情報を SPREAD に表示するには以下のようなコードになります。

```
' SPREAD で、指定された行の写真を表示します
Private Sub ShowphotoInSpread(ByVal row As Integer)
    ' 画像データの存在チェック
    If Not IsDBNull(ds.BlobTbl.Rows(row).Item(3)) Then
        'DataSet から画像データを Byte 配列として取得
        Dim imgData As Byte() = CType(ds.BlobTbl.Rows(row).Item(3), Byte())

        '画像をセルに表示
        Dim stmPic As New MemoryStream(imgData)
        Dim imgType As New FarPoint.Win.Spread.CellType.GeneralCellType
        imgType.BackgroundImage = _
            New FarPoint.Win.Picture(Image.FromStream(stmPic))
        FpSpread1.Sheets(0).Cells(row, 3).CellType = imgType
    End If
End Sub
```

```
End If
End Sub
```

リスト.1 frmBlob.vb - Sub ShowphotoInSpread

SPREAD のセルには画像を背景として表示する機能があります。BackgroundImage プロパティに画像情報を設定すると SPREAD のセル内に画像を表示することが可能です。



ASP.NET WEB アプリケーションでのコーディング

今度は、BLOB フィールドの画像情報を ASP.NET WEB アプリケーションで表示させてみましょう。以下のようなコードになります。

```
' Spread で、指定された行の写真を表示します
Private Sub ShowphotoInSpread(ByVal row As Integer)
    ' 画像データの存在チェック
    If Not IsDBNull(ds.BlobTbl.Rows(row).Item(3)) Then
        ' DataSet から画像データを Byte 配列として取得
        Dim imgData As Byte() = CType(ds.BlobTbl.Rows(row).Item(3), Byte())

        ' 画像をファイルとして出力
        Dim strUserCode As String = get_user_code_by_row(row)
        Dim imgfilePhyPath As String = Server.MapPath(".") & "¥" & _
            strUserCode & ".gif"
        Dim imgfileLogPath As String = strUserCode & ".gif"
        Dim imgfile As New System.IO.FileStream(imgfilePhyPath, IO.FileMode.Create)
        imgfile.Write(imgData, 0, imgData.Length)
        imgfile.Close()
```

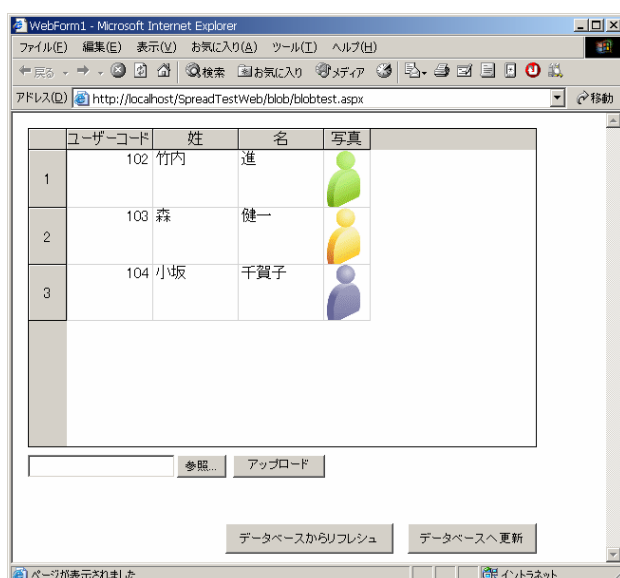
```

'画像をセルに表示
FpSpread1.ActiveSheetView.Cells(row, 3).CellType = _
    New FarPoint.Web.Spread.ImageCellType(imgfileLogPath)
FpSpread1.ActiveSheetView.Cells(row, 3).Text = ""
End If
End Sub

```

リスト.2 blobtest.aspx.vb - Sub ShowphotoInSpread

ASP.NET WEB アプリケーションの場合は、BLOB フィールドに格納された画像データを SPREAD に表示することができません。一度 BLOB フィールドの画像をファイルとして保存し、SPREAD の ImageCellType プロパティで画像ファイルの URL をフルパスで指定する必要があります。



今まで説明した BLOB フィールドによる画像ファイルの表示は、BFILE フィールドでも同様になります。Windows アプリケーションでは、そのまま画像を SPREAD のセルに表示できますが、ASP.NET WEB アプリケーションでは一旦ファイルに保存してから、SPREAD のセルに表示する必要がありますので注意してください。

カーソル変数(REF CURSOR)の利用

カーソル変数は、PL/SQL パッケージと様々なクライアントの間で問い合わせ結果セットを渡すために使用されます。また、PL/SQL ブロックを使用して複数のホスト・カーソル変数を 1 回のラウンドトリップでオープン/クローズすることが可能ですので、ネットワークの通信量を削減することもできます。では実際に、カーソル変数を使用して複数の結果セットを取得し、SPREAD に表示してみましょう。

PL/SQL パッケージの作成

まず、「ユーザー」「会社」テーブルのカーソル変数を返す、PL/SQL パッケージを作成します。こちらの PL/SQL パッケージはサンプルスクリプト SQL である、「gen_spread.sql」を実行していた場合は、既に作成されています。

```
CREATE OR REPLACE PACKAGE SPREAD.REFCUR_TEST AS
  TYPE UserCurTyp IS REF CURSOR RETURN ユーザー%ROWTYPE;

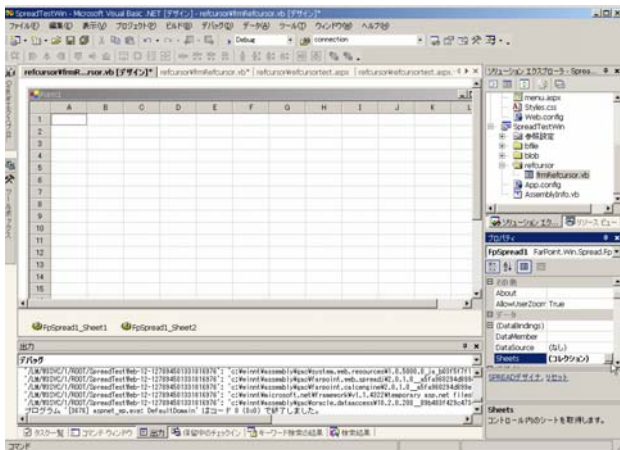
  CURSOR c1 IS SELECT * FROM 会社;
  TYPE CompCurTyp IS REF CURSOR RETURN c1%ROWTYPE;

  PROCEDURE open_cv(user_cv IN OUT UserCurTyp, comp_cv IN OUT CompCurTyp);
END;
/

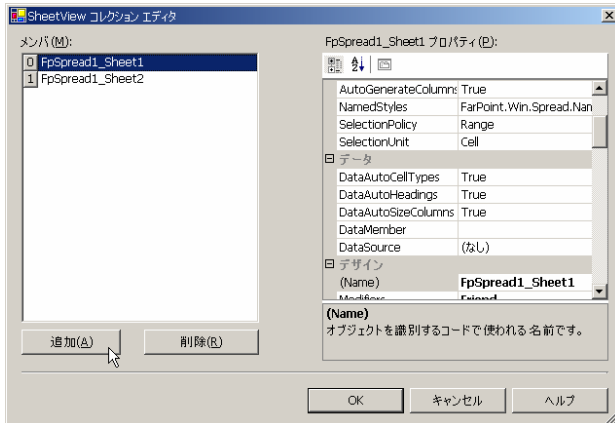
CREATE OR REPLACE PACKAGE BODY SPREAD.REFCUR_TEST AS
  PROCEDURE open_cv(user_cv IN OUT UserCurTyp, comp_cv IN OUT CompCurTyp) IS
  BEGIN
    OPEN user_cv FOR SELECT * FROM ユーザー;
    OPEN comp_cv FOR SELECT * FROM 会社;
  END open_cv;
END;
/
```

SPREAD 側の設定

今回は2つのテーブルを1つの SPREAD コントロールで表示します。SPREAD コントロールのプロパティから、「Sheets」プロパティの「...」ボタンをクリックします。



デフォルトで1つのシートメンバが作成されているので、「追加(A)」ボタンをクリックして、シートを追加します。合計2つのシートメンバを作成します。



コードの記述

最後に、上記で作成したフォームのロードイベントにカーソル変数のデータを SPREAD に表示するためのコードを記述します。コードの内容は以下のようになります。

```
Imports Oracle.DataAccess.Client
Imports Oracle.DataAccess.Types
Imports System.Configuration

Public Class fmRefcursor
    Inherits System.Windows.Forms.Form

    #Region " Windows フォーム デザイナで生成されたコード "
        ~ 省略 ~
    #End Region
```

```
Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load

    'データベースへの接続情報
    Dim conn As New OracleConnection( _
        ConfigurationSettings.AppSettings("connectionString"))

    'コマンドの準備
    Dim cmd As New OracleCommand("REFCUR_TEST.open_cv", conn)
    cmd.CommandType = CommandType.StoredProcedure

    'REF CURSOR パラメータのバインド
    cmd.Parameters.Add("user_cv", _
        OracleDbType.RefCursor, ParameterDirection.Output)
    cmd.Parameters.Add("comp_cv", _
        OracleDbType.RefCursor, ParameterDirection.Output)

    'DataSet へのセット
    Dim ds As New DataSet
    Dim da As New OracleDataAdapter(cmd)
    da.Fill(ds, "data")

    'データの表示
    With FpSpread1.Sheets(0)
        .DataSource = ds
        .DataMember = "data"
    End With

    With FpSpread1.Sheets(1)
        .DataSource = ds
        .DataMember = "data1"
    End With

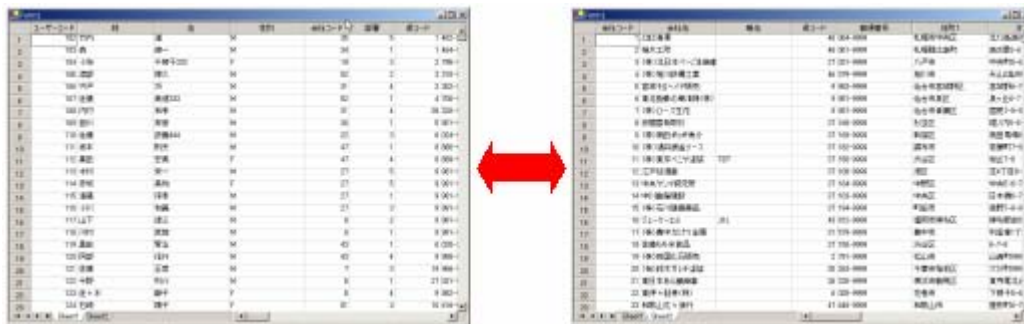
    conn.Close()
    conn.Dispose()
```

```

End Sub
End Class
    
```

リスト.3 frmRefcursor.vb - Sub Form1_Load

上記コードを記述したら、実行すると2つのシートをもった SPREAD コントロールが確認できます。



以上のように、Oracle の機能と SPREAD の機能を組み合わせることにより柔軟なアプリケーションを開発できます。是非、お試しください。