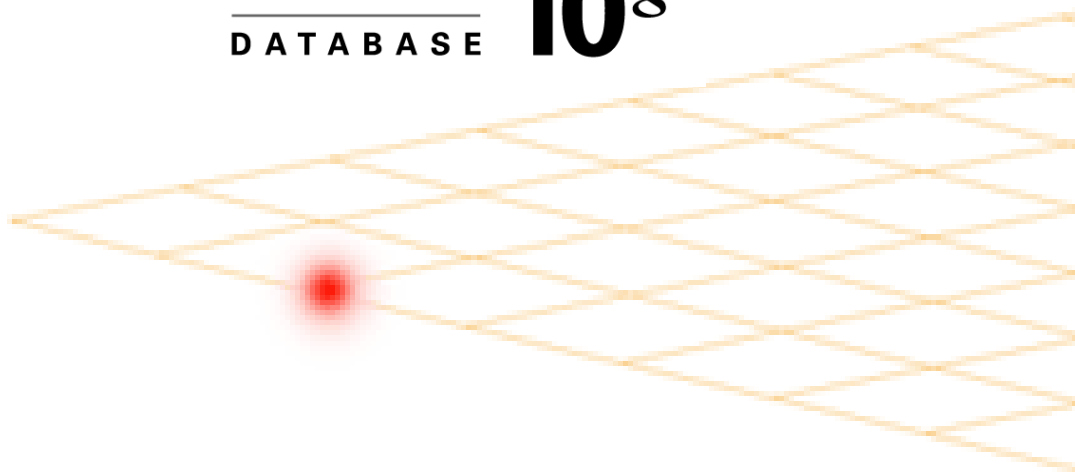


GrapeCity コンポーネント活用

- .NET 開発 with GrapeCity -

「InputMan 編」

ORACLE®
DATABASE 10^g



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

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ス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 InputMan 編ということで、グレースシティ社が提供している InputMan と Oracle が提供している .NET から Oracle データベースへ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 (本書)
3. Spread 編
4. FlexGrid 編

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

1. InputMan とは
2. サンプルアプリケーションのインストール
3. InputMan 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



InputMan とは

InputMan はテキスト、日付・時刻、数値、マスク、コンボなどの目的に合わせて最適化された入力インタフェースコントロールのセットです。XP テーマと Office ライクなアクティブフラットの 2 つのスタイルにより、洗練された外観を簡単に実現できます。これらのコントロールに搭載された機能の多くは、プロパティを設定するだけで実現できるため、コーディング量と開発工数を大幅に削減します。エンドユーザーが感じる迷いや陥りやすい誤りを未然に防ぐ設計の InputMan for .NET は、自然でスムーズな入力を可能にし、アプリケーションの基本となる入力インタフェースの構築を強力に支援します。

InputMan のより詳細な情報はグレースィティ社の以下の URL を参照してください。

グレースィティ社 - InputMan 製品情報

http://www.grapecity.com/japan/support/database/P7_307.htm

今回は InputMan の 2006 年 1 月時点で最新版のバージョンである、「InputMan for .NET 3.0J」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/inputman/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「InputManDemo」、「Config」という 2 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。

InputManDemo

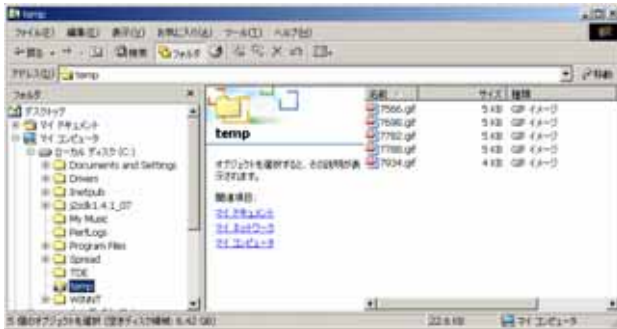
VB.NET で作成された Windows アプリケーションが格納されています。

Config

サンプルアプリケーションで使用するテーブルなどの自動作成スクリプトが格納されています。

サンプルアプリケーションを実行させるためには、まず Oracle データベース側に今回使用するサンプルスキーマの設定をおこないます。sample.zip を解凍すると作成される

「Config」フォルダ内の「gen_inputman.sql」を SQL*PLUS から実行します。今回のサンプルでは画像データを Oracle データベースに格納し、InputMan のコントロールで表示しています。このため、その画像を Oracle データベースに読み込む必要があるため、「Config」フォルダ内の「Images」フォルダに格納されている画像ファイル(gif ファイル)を「c:\Temp」フォルダにコピーしてください。「gen_inputman.sql」は「C:\Temp」フォルダにコピーされた画像ファイルを Oracle データベースに取り込みます。



画像ファイルをコピーしたら、「Config」フォルダ内の「gen_inputman.sql」を SQL*PLUS などから以下のように実行してください。この SQL スクリプトの実行は、リモート実行ではなく、ターゲットのデータベースサーバーで実行します。実行する際に、ユーザー名/パスワードを求められますので、その際には SYSTEM ユーザーとそのパスワードを入力してください。

```
C:\>sqlplus /nolog

SQL*Plus: Release 10.2.0.1.0 - Production on 火 2月 14 09:12:05 2006

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

SQL> @gen_inputman

*-----*
* please enter password of system *
*-----*

Enter password:
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。
```

-<省略> -

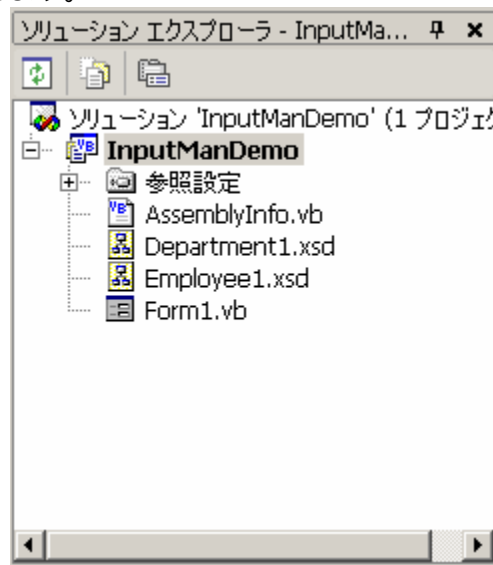
PL/SQL プロシージャが正常に完了しました。

Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production

With the Partitioning, OLAP and Data Mining options との接続が切断されました。

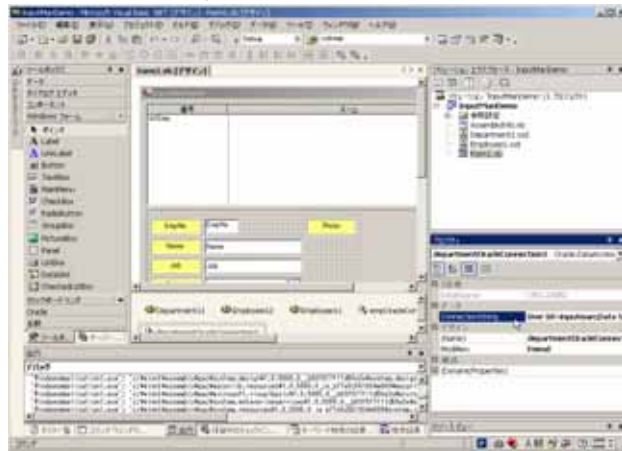
サンプルアプリケーションの動作確認

サンプルアプリケーションを実行させるための Oracle のデータベース・オブジェクトを作成したら、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。ダウンロードファイル(sample.zip)を解凍後に作成される InputManDemo ディレクトリ下にある、VS.NET のソリューションファイル「InputManDemo.sln」を VS.NET から開いてください。InputManDemo.sln ファイルには「InputManDemo」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。



今回のサンプルアプリケーションを実行するには、ODP.NETがインストールされ、VS.NETから参照設定が適切におこなわれている必要があります。ODP.NETのインストール方法などは、「意外と簡単!? .NETでオラクル」スマートクライアント編の[「Oracle Data Provider for .NETのインストール」](#)を参照してください。ODP.NETをVS.NETから利用するための準備が整いましたら次にODP.NETから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。InputManDemoプロジェクト内にある、Form1.vbのデザイン画面を開き、「empOracleConnection1」コントロールと

「departmentOracleConnection1」コントロールの「ConnectionString」プロパティを変更してください。



初期状態では以下のように設定されています。

```
User Id=inputman;Data Source=ORCL10GR2;Password=oracle;
```

「User Id」では「inputman」と指定していますが、こちらは先ほど説明した「Config」フォルダ内の「gen_inputman.sql」を実行していれば、「inputman」ユーザーが作成されています。「Data Source」は Oracle データベースへ接続するためのデータソース名となります。接続文字列を適切に設定したら、次に ASP.NET WEB アプリケーションを実行してみましょう。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。

The screenshot shows a Windows application window titled "InputManDemo". At the top, there is a table with columns "番号" (Number) and "名前" (Name). The table contains three rows: (7698, BLAKE), (7499, ALLEN), and (7504, MILLER). Below the table, there are several input fields and controls:

- EmpNo:** 7698
- Name:** BLAKE
- Job:** MANAGER
- Salary:** 2860
- Mgr:** JONES
- Dept:** SALES
- Hire Date:** 1981年5月 (with a calendar view showing the date 5/2)
- Commission:** 0
- Photo:** A blue 3D person icon.

At the bottom, there is a "データ操作" (Data Operation) section with buttons for "レコードの移動" (Move Record), "最初" (First), "前" (Previous), "次" (Next), "最後" (Last), and "データの更新" (Update Data).

以上のように InputMan のコントロールを使用した Windows フォームの画面が確認できます。

InputMan から ODP.NET の利用

では、実際に InputMan 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と入力をおこなってみましょう。

DataSet と InputMan の連携

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 InputMan に連携して表示しています。実際の手順は以下のようになります。

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。

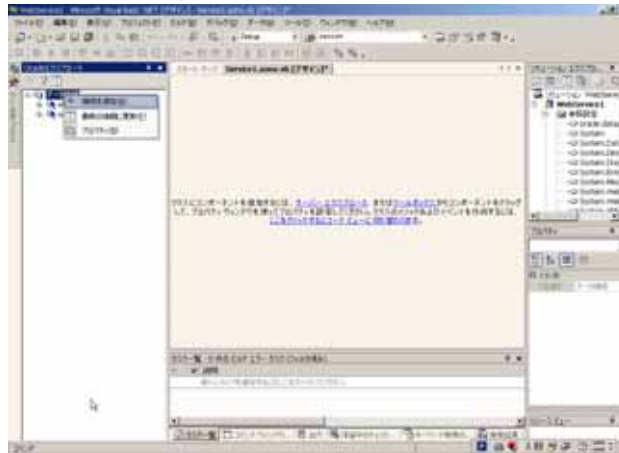
1. Oracle データベースへの接続の設定

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

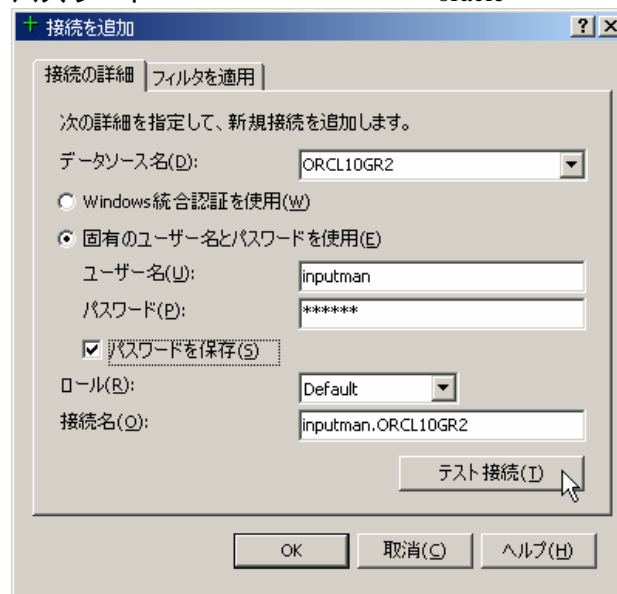
Oracle エクスプローラの「データ接続」を右クリックし、接続の追加(A)を選択します。



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。

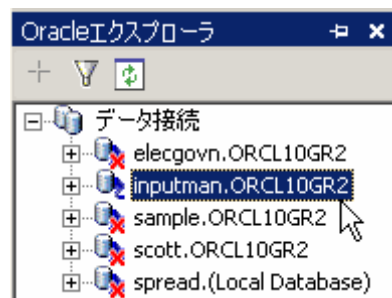
データソース名	<データソース名>
ユーザー名	inputman
パスワード	oracle



データソース名は Network Configuration Assistant(以下、netca)で設定されたデータソース名を指定します。ユーザー名は「inputman」、パスワードに「oracle」と指定します。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。

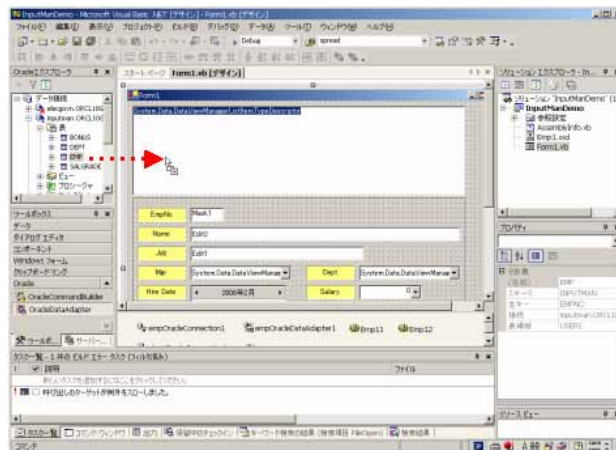


接続が失敗した場合は、再度接続文字列、ユーザー名、パスワードの内容が正しいか確認をおこなってください。テストが終了後、「接続を追加」ウィンドウで「OK」ボタンをクリックすると、「Oracle エクスプローラ」に接続情報が追加されます。



2. DataAdapter の設定

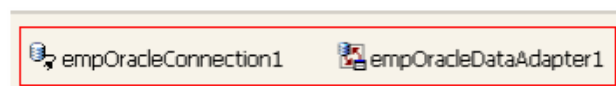
Oracle データベースへの接続情報の設定が完了したら、次に DataAdapter の設定をおこない、DataSet へ格納する情報の設定をおこないます。DataAdapter の設定も ODT を使用します。Oracle エクスプローラから対象表を選択し、デザインウィンドウにドラッグ・アンド・ドロップします。



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のが、デザインウィンドウで確認できます。



今回は、以下の2つのテーブルから情報を取得しています。

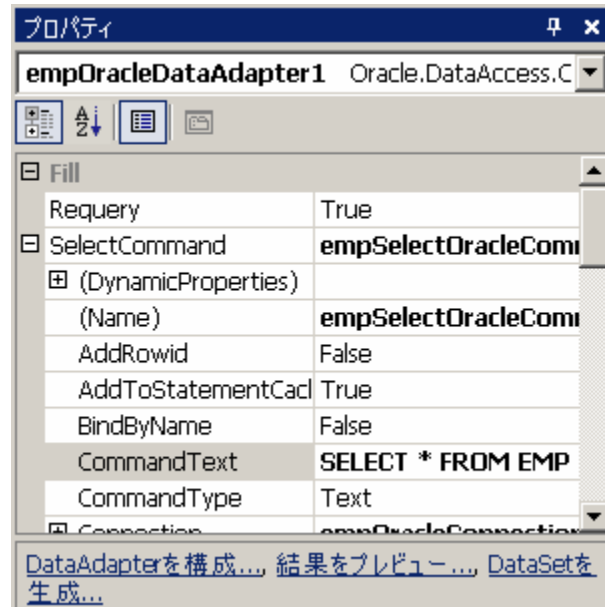
EMPLOYEE(社員情報テーブル)

DEPARTMENT(部署テーブル)

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 SelectCommand、CommandText プロパティの内容は以下のようなドラッグ・アンド・ドロップされた単一テーブルの SQL が設定されています。

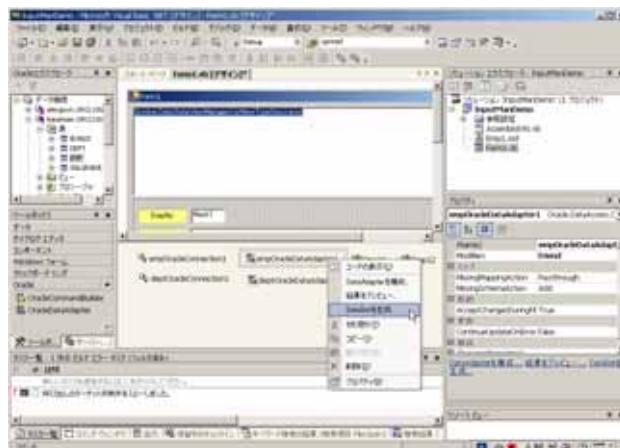
```
SELECT * FROM EMPLOYEE
```

SQL の変更は直接 CommandText プロパティの内容を変更します。



3. DataSet へのデータの格納

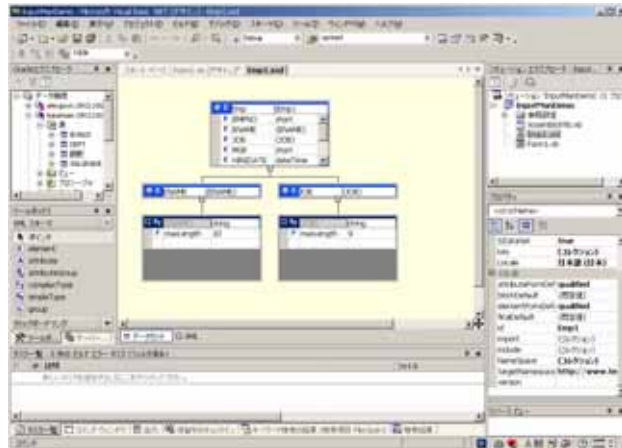
DataAdapter の設定が完了したら、次に DataSet にデータを格納するための手順について説明します。上記作業で作成した DataAdapter オブジェクトを右クリックし、「DataSet を生成 ...」を選択します。



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。



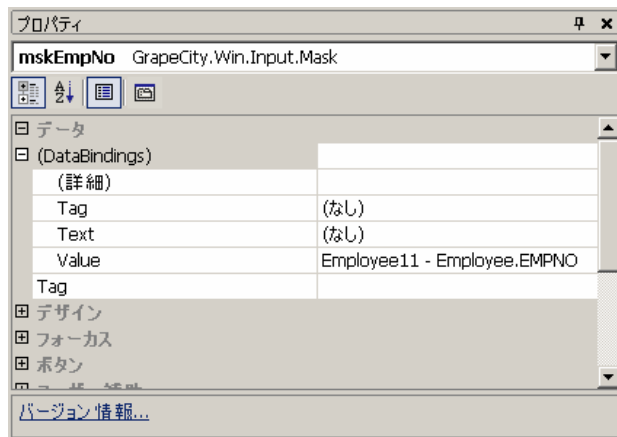
ODT により作成された DataAdapter から DataSet へのデータの格納は以下のようなコードになります。

```
empOracleDataAdapter1.Fill(Emp11.Emp)
```

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のが確認できたと思います。今回のサンプルアプリケーションも ODT を使用して開発されていますので、今まで説明した内容はサンプルアプリケーションを参照することにより確認できます。次に DataSet の内容を InputMan の各コンポーネントを使用して表示してみましょう。

DataSet と InputMan コンポーネントの連携

DataSet に格納された内容を InputMan コンポーネントに連携するには、InputMan コンポーネントの「DataBindings」プロパティを設定します。「DataBindings」プロパティの設定はコードで記述することも可能ですが、プロパティウィンドウで設定することも可能です。



コードで記述する場合は、DataSet オブジェクトの内容をテキストコントロールの DataBindings プロパティに設定します。DataBindings プロパティが参照する ControlBindingsCollection オブジェクトの Add メソッドの引数には、次のように設定します。

第 1 引数：データをバインドするプロパティ名 (文字列型)

第 2 引数：対象となる DataSet オブジェクト

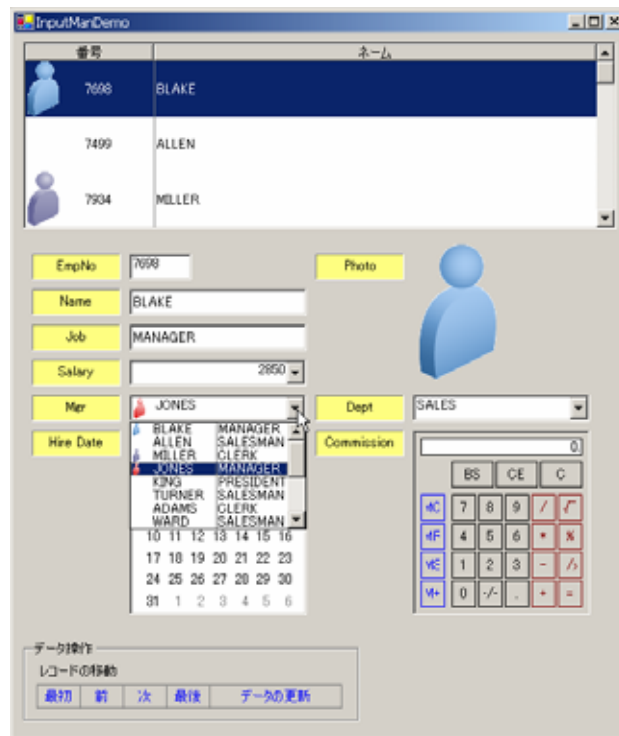
第 3 引数：データをバインドするテーブルと列名 (文字列型)

```
edtName.DataBindings.Add("Text", Me.Employee11, "Employee.ENAME")
```

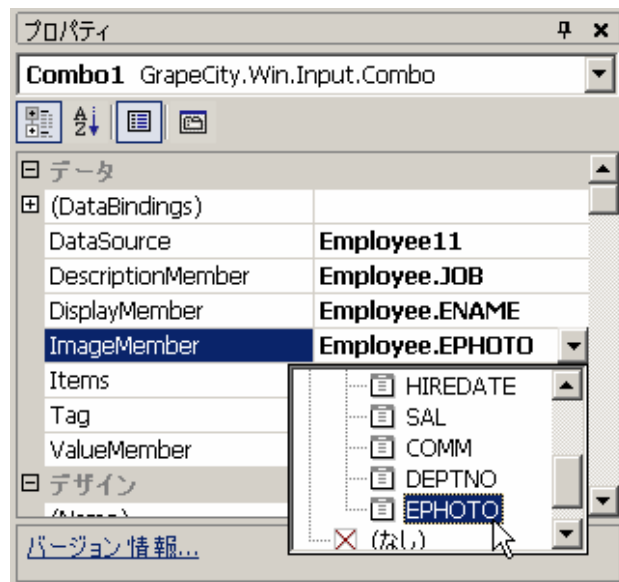
DataBindings プロパティの詳細については、VS.NET のヘルプの「Control.DataBindings プロパティ」を参照してください。

BLOB 型フィールドへの対応

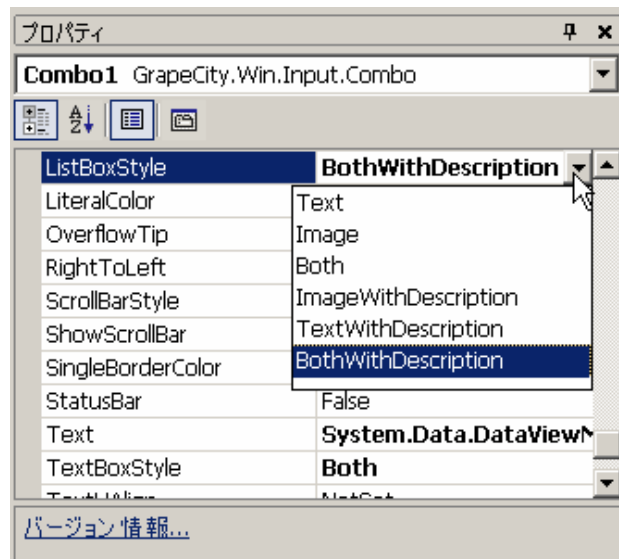
画像などを扱う BLOB フィールドの内容も InputMan のコンポーネントで表示することが可能です。今回のサンプルアプリケーションでは実際に BLOB フィールドに格納された社員のイメージ情報を InputMan のリストボックスコントロールとコンボボックスコントロールで表示しています。



「employee」テーブルの画像が格納された BLOB フィールドである「ephoto」列を表示するには、まず InputMan のリストボックス、もしくはコンボボックスの「ImageMember」プロパティに「ephoto」列を指定します。



次に、ListBoxStyle プロパティと Text プロパティの値を画像が表示できるように設定します。



以上のように InputMan の標準の機能を利用することで、簡単に Oracle データベース内に格納されている BLOB フィールドの画像を表示できるのが確認できたかと思います。

ListboxStyle と TextBoxStyle プロパティの内容はそれぞれ以下ようになります。

プロパティ値 - GrapeCity.Win.Input.ListBoxStyle 型

(既定値 : ListBoxStyle.Text)

ListboxStyle の値	説明
0 - Image	項目の画像だけを表示します。
1 - Text	Text 項目の内容だけを表示します。
2 - Both	Both 項目の画像と内容を表示します。
3 - ImageWithDescription	ImageWithDescription 項目の画像と説明を表示します。
4 - TextWithDescription	TextWithDescription 項目の内容と説明を表示します。
5 - BothWithDescription	BothWithDescription 項目の画像、内容、および説明を表示します。

プロパティ値 - GrapeCity.Win.Input.TextBoxStyle 型

(既定値 : GrapeCity.Win.Input.TextBoxStyle.TextOnly)

ListboxStyle の値	説明
0 - ImageOnly	テキストボックスに画像だけが表示されます。
1 - TextOnly	テキストボックスに文字列だけが表示されます。
2 - Both	テキストボックスに画像と文字列の両方が表示されます。

各プロパティの詳細については InputMan のマニュアルを参照してください。

Oracle フィールド属性の自動設定

今回のサンプルでは、「dynamicSetProperties()」メソッドを利用して、「employee」テーブルのフィールド属性を取得し、InputMan のプロパティに設定しています。文字列フィールドの場合は、入力文字列の長さを設定するために InputMan の Edit コンポーネント MaxLength プロパティの値を設定しています。また、数値列に関しては、Oracle の数値列から桁数を取得し、InputMan の Number コンポーネント Format プロパティにセットしています。具体的な手順は以下のようになります。

GetSchemaTable メソッドを使用して列情報を取得

ODP.NET に含まれている「OracleDataReader」オブジェクトの GetSchemaTable メソッドを使用して、列スキーマ情報を取得する方法について説明します。列スキーマは、フィールド プロパティとも呼ばれます。列スキーマ情報には、列に関する次のような情報が含まれます。

- 名前
- データ型
- サイズ
- その列が主キーのフィールドかどうか
- その列がオートナンバー (AutoIncrement) 型のフィールドかどうか

GetSchemaTable メソッドを利用したコードは今回のサンプルコードを参照してください。

```
' =====
' テーブルのスキーマを動的に取得する
' =====
Private Function getDynamicSchema() As DataTable

    " テーブルスキーマ情報の取得

    Dim aCMD As OracleCommand

    Dim rdr As OracleDataReader

    aCMD = empOracleDataAdapter1.SelectCommand
    aCMD.Connection.Open()
    rdr = aCMD.ExecuteReader()

    getDynamicSchema = rdr.GetSchemaTable
```

```
aCMD.Connection.Close()
```

```
End Function
```

列スキーマ情報により、各コントロールのフォーマットを設定
 GetSchemaTable メソッドを使用して取得した列スキーマ情報をもとに、各コントロールのフォーマットプロパティを設定します。

```
' =====
' テーブルスキーマにより、各コントロールのフォーマットを設定する
' =====
Private Sub dynamicSetProperties()

    Dim schemaTable As DataTable
    Dim datarowTmp As DataRow

    Dim NumericPrecision As Integer
    Dim NumericScale As Integer
    Dim str As String

    " EMP テーブルのスキーマを取得する
    schemaTable = getDynamicSchema()

    " ENAME
    datarowTmp = schemaTable.Rows(1)
    Me.edtName.MaxLength = datarowTmp(2)

    " JOB
    datarowTmp = schemaTable.Rows(2)
    Me.edtJob.MaxLength = datarowTmp(2)

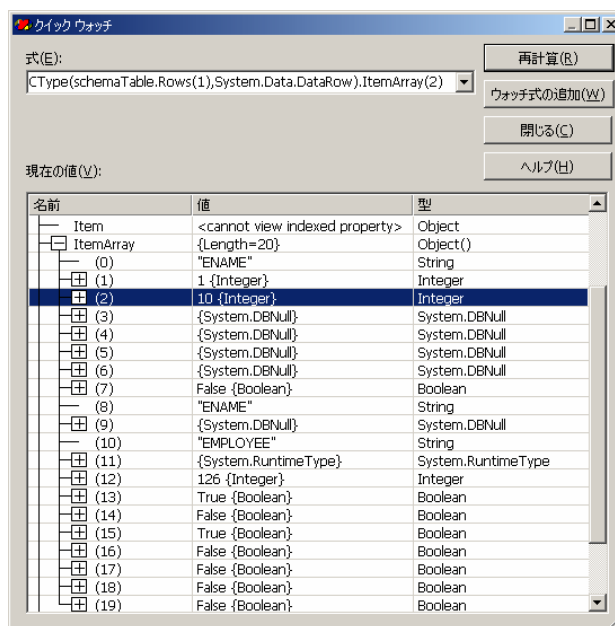
    " SALARY
    datarowTmp = schemaTable.Rows(5)
    NumericPrecision = datarowTmp(3)
    NumericScale = datarowTmp(4)
    If NumericScale = 0 Then
```

```

str = Space(NumericPrecision)
Else
str = Space(NumericPrecision) + "." + Space(NumericScale)
End If
str = str.Replace(" ", "#")
Me.numSalary.Format.Digit = str
End Sub

```

列スキーマ情報を格納した「schemaTable」オブジェクトの Rows プロパティには、「employee」テーブルの列数分の情報が格納されています。Rows プロパティ中の中に詳細な列情報が格納されており、その情報を取得してプロパティの値をセットしています。



GetSchemaTable メソッドを使用して取得した列スキーマ情報についての詳細は、VS.NET のヘルプの「GetSchemaTable メソッド」を参照してください。以上のように、Oracle データベース上に格納されたスキーマ情報をもとに、InputMan の入力プロパティの値をセットし、フィールド定義情報にもとづいた入力項目を作成することが可能です。

SafeMapping について

InputMan のコンポーネントと Oracle データベースのデータを連携する場合に、

今回のサンプルで紹介したように DataSet を利用します。この場合、OracleDataAdapter を使用して Oracle データベースのデータを DataSet に取り込みますが、Oracle 固有のデータ型は、暗黙的に Oracle のデータ型に最適な .NET のデータ型に変換されます。しかし、双方のデータ型の特性から DataSet にデータを取り込んで処理する際に、データが失われる危険性が存在しています。具体的には、以下の表の Oracle データ型が該当します。

	データ型		最小値	最大値
1	Oracle	NUMBER	最大 38 精度	
	.NET	Decimal	最大 28 精度	
2	Oracle	DATE	紀元前 4712 年 1 月 1 日	紀元 9999 年 12 月 31
	.NET	Datetime	紀元 0001 年 1 月 1	紀元 9999 年 12 月 31
3	Oracle	TIMESTAMP	紀元前 4712 年 1 月 1 日 00:00:00.000000000	紀元 9999 年 12 月 31 23:59:59.999999999
	.NET	Datetime	紀元 0001 年 1 月 1 00:00:00.0000000	紀元 9999 年 12 月 31 23.59.59.9999999
4	Oracle	INTERVAL DAY TO SECOND	-999999999 23:59:59.999999999	+999999999 23:59:59.999999999
	.NET	TimeSpan	-10675199 02:48:05.4775808	+10675199 02:48:05.4775808

このような場合に、ODP.NET では、OracleDataAdapater の SafeMapping プロパティを使用することにより、Oracle データ型を DataSet において、.NET の String 型、もしくは Oracle 書式での .NET の byte[] で表すことにより、データ損失の危険性を防ぐことが可能です。しかしながら、表の値をみても分かる通り、多くの業務アプリケーションにおいては問題となる場合がないと考えられます。よって今回は DATE 型を例にした SafeMapping のみ解説いたします。

```

-<省略> -
cmd.CommandText = "SELECT empno, hiredate FROM employee"
Dim da As New OracleDataAdapter(cmd)
da.SafeMapping.Add("HIREDATE", GetType(System.String))

Dim ds As New DataSet
If ds.Tables.Contains("employee") Then ds.Tables("employee").Clear()
da.Fill(ds, "employee")

```

-<省略> -

設計時に列名が不明な場合は、アスタリスク(*)を使用すると、データベース型のすべての列を安全な.NET タイプにマップできます。

```
da.SafeMapping.Add("*", GetType(System.String))
```

その他の型の詳細に関しましては、ODP.NET のマニュアル「Oracle Data Provider for .NET 開発者ガイド」を参照してください。

メモ: SafeMapping にて指定する列名に関しては、必ず大文字で設定する必要があります。
また、SELECT 文で別名を指定している場合は、その別名を適切に指定してください。

例) 別名を指定している場合

```
SELECT empno, hiredate as 入社日 FROM emp  
da.SafeMapping.Add("入社日", GetType(System.String))
```

今まで説明したように、ODP.NET と InputMan で使用するコンポーネントの親和性は非常に優れています。ODP.NET を利用することにより Oracle データベースに高速にアクセス可能なアプリケーションを開発することが可能です。また、ODT を利用すればアプリケーションの開発生産性を高め、さらに InputMan を組み合わせることにより洗練された外観とインタフェースを簡単に実現にできます。是非お試しください。