

Exemple d'utilisation du Type BLOB

The screenshot shows a Windows application window titled "Form1". The window contains a data entry form with the following elements:

- A "Lister" button in the bottom-left corner.
- A "Quitter" button in the top-right corner.
- Form fields for:
 - Numéro: 134
 - Nom: Patoche
 - Prenom: Alain
 - Date inscription: 18 décembre 2014 (with a calendar icon)
- A "Choisir Image" button next to a red circular logo with a white stylized 'L' and the text "Col Lic".
- A set of navigation buttons: "Debut", ">>", "<<", and "FIN".
- Three stacked buttons: "vider les zones", "Ajouter", and "Modifier".

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Oracle.DataAccess.Client;
using System.IO;

namespace WindowsFormDataSet
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            OracleConnection conn = new OracleConnection();
            private DataSet monDataSet = new DataSet();
            string sql1 = "SELECT NUMAD, NOM, PRENOM, photo FROM ETUDIANTSINFO";
            string sql2 = "select numad, nom, prenom, dateinscription,photo from etudiantsinfo";
            string nomFichier;
```

```
private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        string chaineDeconnexion = "Data Source = primogene; User Id = user1; password = user1";
        conn.ConnectionString = chaineDeconnexion;
        conn.Open();
        MessageBox.Show(conn.State.ToString());
        // on liste dans les zones de texte
        afficherTxt();
    }

    catch (Exception sqlConn)
    {
        MessageBox.Show(sqlConn.Message.ToString());
    }
}
```

```
private void Deconnect_Click(object sender, EventArgs e)
{
    conn.Close();
    Application.Exit();
}
```

```

private void afficherTxt()
{
    try
    {
        OracleDataAdapter Adapter2 = new OracleDataAdapter(sql2, conn);
        if (monDataSet.Tables.Contains("resEtudiants"))
        {
            monDataSet.Tables["resEtudiants"].Clear();
        }
        Adapter2.Fill(monDataSet, "resEtudiants");
        Adapter2.Dispose();
        // on appelle la fonction lier pour faire
        // la liaison des données du DataSet avec les zones de text.
        lier();
    }

    catch (Exception exsql2)
    {
        MessageBox.Show(exsql2.Message.ToString());
    }
}

private void lier()
{
    textNumad.DataBindings.Add("text", monDataSet, "resEtudiants.numad");
    textNom.DataBindings.Add("text", monDataSet, "resEtudiants.nom");
    textPrenom.DataBindings.Add("text", monDataSet, "resEtudiants.Prenom");
    dateIns.DataBindings.Add("text", monDataSet, "resEtudiants.dateinscription");
    picPhoto.DataBindings.Add("image", monDataSet, "resEtudiants.photo", true);
}

```

```
private void dissocier()
{
    textNumad.DataBindings.Clear();
    textNom.DataBindings.Clear();
    textPrenom.DataBindings.Clear();
    dateIns.DataBindings.Clear();
    picPhoto.DataBindings.Clear();
    // effacer le contenu
    textNumad.Clear();
    textNom.Clear();
    textPrenom.Clear();
    dateIns.Value = DateTime.Now;
    picPhoto.Image = null;
}

private void debut_Click(object sender, EventArgs e)
{
    this.BindingContext[monDataSet, "resEtudiants"].Position = 0;
}

private void next_Click(object sender, EventArgs e)
{
    this.BindingContext[monDataSet, "resEtudiants"].Position += 1;
}

private void previous_Click(object sender, EventArgs e)
{
    this.BindingContext[monDataSet, "resEtudiants"].Position -= 1;
}

private void dernier_Click(object sender, EventArgs e)
{
    this.BindingContext[monDataSet, "resEtudiants"].Position =
    this.BindingContext[monDataSet, "resEtudiants"].Count - 1;
}
```

```

// affichage dans le DGV
private void ListerTout_Click(object sender, EventArgs e)
{
    afficherDGV();
}

private void afficherDGV()
{
    try
    {
        OracleDataAdapter Adapter1 = new OracleDataAdapter(sql1, conn);
        // on vérifie que le DataSet ne contient pas de Data Table de nom "ListeEtudiants"

        if (monDataSet.Tables.Contains("ListeEtudiants"))
        {
            monDataSet.Tables["ListeEtudiants"].Clear();
        }
        // on remplit le DataSet
        Adapter1.Fill(monDataSet, "ListeEtudiants");
        Adapter1.Dispose();
        //on fait une liaison des données entre le DGV et le DataSet pour "ListeEtudiants"
        BindingSource maSource;
        maSource = new BindingSource(monDataSet, "ListeEtudiants");
        DGVetudiants.DataSource = maSource;
    }
    catch (Exception exsql1)
    {
        MessageBox.Show(exsql1.Message.ToString());
    }
}

// dissocier les zones de texte== enlever le Bindings puis vider les zone.
private void vider_Click(object sender, EventArgs e)
{
    dissocier();
}

```

```

private void ajouter2_Click(object sender, EventArgs e)
{
    try
    {
        // la requête SQLajout est paramétrée. Elle a 4 paramètres.
        //les paramètres pour Oracle et C # sont précédés de deux points :
        string sqlajout = " insert into etudiantsinfo" +
            " (numad,nom,prenom,dateinscription,photo) values "+
            "(SeqEtu.nextval,:nom,:prenom,:dateinscription,:photo)";

        // On déclare les paramètres pour chaque paramètre de la requête

        OracleParameter oranom = new OracleParameter(":nom", OracleDbType.Varchar2, 30);
        OracleParameter oraprn = new OracleParameter(":prenom", OracleDbType.Varchar2, 40);
        OracleParameter oradate = new OracleParameter(":dateinscription", OracleDbType.Date);
        OracleParameter oraphoto = new OracleParameter(":photo", OracleDbType.Blob);
        // on affecte les valeurs aux paramètres.
        oranom.Value = textNom.Text;
        oraprn.Value = textPrenom.Text;
        oradate.Value = dateIns.Value;

        //on fait du streaming (lecture en continue du fichier BLOB
        // le résultat on le met dans une variable de type byte (octets).
        FileStream Streamp = new FileStream(nomFichier, FileMode.Open, FileAccess.Read);
        byte[] buffer1 = new byte[Streamp.Length];
        Streamp.Read(buffer1, 0, System.Convert.ToInt32(Streamp.Length));
        Streamp.Close();
        // on affecte le contenu de buffer1 à
        oraphoto.Value = buffer1;

        // En crée un Objet OracleCommand pour passer la requête à la bD
        OracleCommand oraAjout = new OracleCommand(sqlajout, conn);
        oraAjout.CommandType = CommandType.Text;
        // En utilisant la propriété Paramètres de OracleCommand, on spécifie les
        // Paramètre de la requête SQLajout.
    }
}

```

```
oraAjout.Parameters.Add(oranom);
oraAjout.Parameters.Add(oraprn);
oraAjout.Parameters.Add(oradate);
oraAjout.Parameters.Add(orphoto);

// on execute la requete
oraAjout.ExecuteNonQuery();

// on appelle la fonction dissociier pour pouvoir inserer une deuxième fois.
// cette fonction est définie plus haut.
dissocier();
}

catch (Exception exsqlajout)
{
    MessageBox.Show(exsqlajout.Message.ToString());
}
} //-----Fin de la fonction associée au bouton Ajouter .-----
```



```

// -----Début Traitement de l'image: Chercher un fichier image
private string RechercherFichier()
{
    OpenFileDialog fImage = new OpenFileDialog();
    fImage.Title = "sélectionner une image";
    fImage.CheckFileExists = true;
    fImage.InitialDirectory = @"C\";

    //fImage.InitialDirectory = Application.StartupPath;
    fImage.Filter = "Fichiers images (*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";
    fImage.FilterIndex = 1;
    fImage.RestoreDirectory = true;
    if (fImage.ShowDialog() == DialogResult.OK)
    {
        nomFichier = fImage.FileName;
        Bitmap bitmap1 = new Bitmap(nomFichier);
    }
    else
    {
        nomFichier = null;
    }
    return nomFichier;
} //-----Fin Rechercher Fichier -----

```

```

// code associé au bouton choisir image
private void ChercherImg_Click(object sender, EventArgs e)
{
    nomFichier = RechercherFichier();
    if (nomFichier != null)
    {
        picPhoto.Image = System.Drawing.Image.FromFile(nomFichier);
        picPhoto.ImageLocation = nomFichier;
    }
}

```

```

private void Modifier_Click(object sender, EventArgs e)
{
    try
    {
        string sqlmodif = "update etudiantsinfo set nom =:nom,photo=:photo where numad=:numad";
        OracleParameter oranum = new OracleParameter(":numad", OracleDbType.Int32);
        OracleParameter oranom = new OracleParameter(":nom", OracleDbType.Varchar2, 30);
        OracleParameter oraphoto = new OracleParameter(":photo", OracleDbType.Blob);
        // on affecte les valeurs aux paramètres.
        oranom.Value = textNom.Text;
        oranum.Value = textNumad.Text;

        //on fait du streaming (lecture en continue du fichier BLOB le résultat on le met dans une variable de type byte (octets).
        FileStream Streamp = new FileStream(nomFichier, FileMode.Open, FileAccess.Read);
        byte[] buffer1 = new byte[Streamp.Length];
        Streamp.Read(buffer1, 0, System.Convert.ToInt32(Streamp.Length));
        Streamp.Close();
        // ajout de la photo dans la BD.
        oraphoto.Value = buffer1;

        OracleCommand oraModif = new OracleCommand(sqlmodif, conn);
        oraModif.CommandType = CommandType.Text;
        // En utilisant la propriété Paramètres de OracleCommand, on spécifie les paramètres de la requête SQL.
        oraModif.Parameters.Add(oranom);
        oraModif.Parameters.Add(oraphoto);
        //remarquer que oranum est en dernier.
        oraModif.Parameters.Add(oranum);
        // on exécute la requête
        oraModif.ExecuteNonQuery();
        // on appelle la fonction dissocier pour pouvoir insérer une deuxième fois.
        dissocier();
    }
    catch (Exception exsqlModif)
    {
        MessageBox.Show(exsqlModif.Message.ToString());
    }
}
// fin modifier
}
}

```

La méthode `Stream.Read()`

Lit un bloc d'octets à partir du flux et écrit les données dans une mémoire tampon donnée. (Substitue [Stream.Read\(Byte\[\], Int32, Int32\)](#).)

buffer

Type : [System.Byte\[\]](#)

Tableau d'octets. Lorsque cette méthode est retournée, la mémoire tampon contient le tableau d'octets spécifié dont les valeurs comprises entre offset et (offset + count - 1) sont remplacées par les octets lus dans la source en cours.

offset

Type : [System.Int32](#)

Dans buffer, offset d'octet de base zéro à partir duquel commencer l'enregistrement des données lues dans le flux actuel.

count

Type : [System.Int32](#)

Nombre maximal d'octets à lire à partir du flux actuel.