

ADO.net pour Activex Data Objects

- Définitions:

ADO.NET est un ensemble de classes qui exposent des services standardisés d'accès aux données. Ces classes permettent donc aux programmeurs de concevoir des applications permettant de se connecter à des sources de données variées et, d'extraire, de manipuler et de mettre à jour ces données

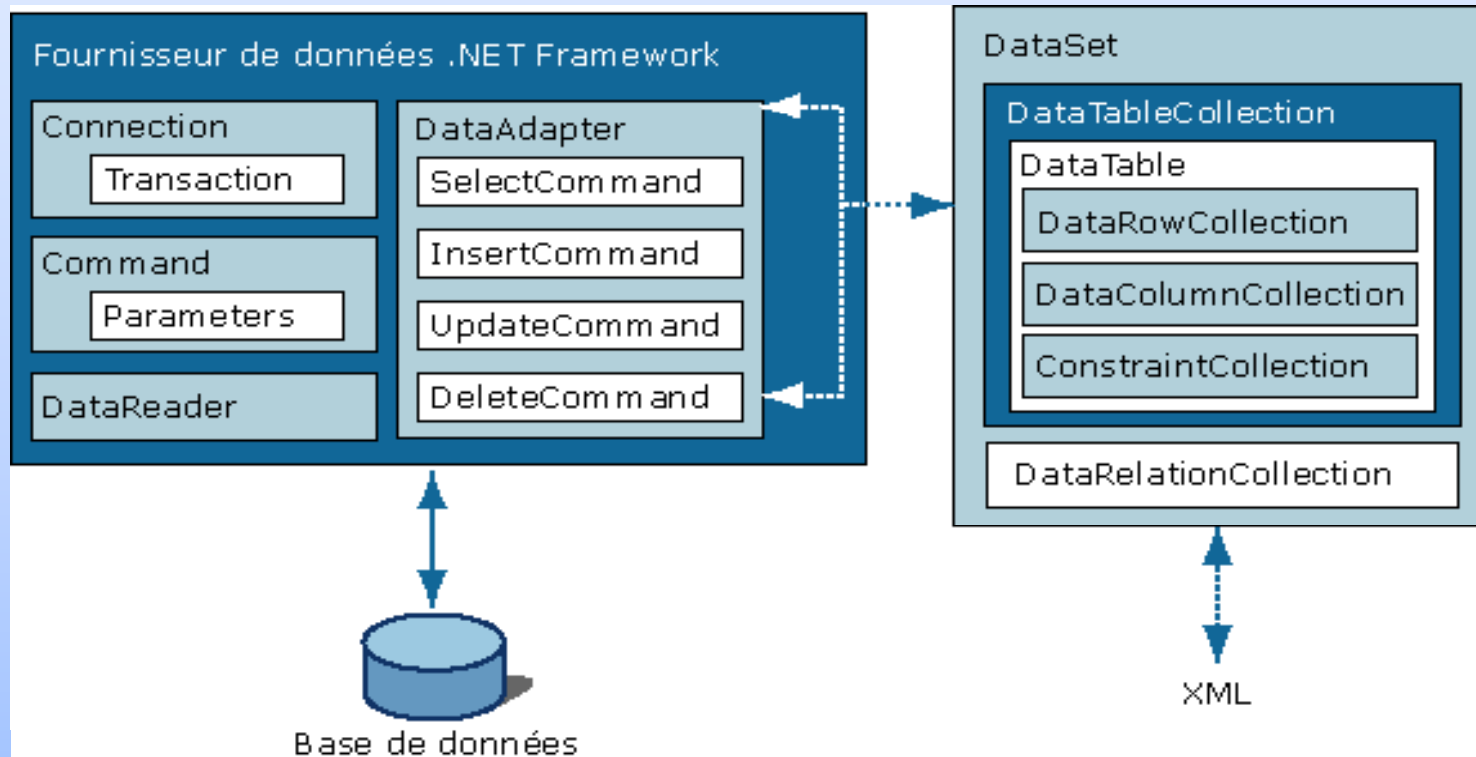
url: [http://msdn.microsoft.com/fr-fr/library/h43ks021\(v=VS.80\).aspx](http://msdn.microsoft.com/fr-fr/library/h43ks021(v=VS.80).aspx)

ADO.net

- **Caractéristiques: intègre les modèles :**
 - Connecté: applications traditionnelles
 - Non connecté: en utilisant un DataSet pour la représentation mémoire des données de la source de données (SGBD), et d'un DataAdapter qui gère la liaison Dataset-SGBD
- **Deux composants d'ADO.NET permettent d'accéder à des données et de les manipuler :**
 - Fournisseurs de données .NET Framework
 - Objet DataSet

ADO.net Architecture

Source : [http://msdn.microsoft.com/fr-fr/library/27y4ybxw\(VS.80\).aspx](http://msdn.microsoft.com/fr-fr/library/27y4ybxw(VS.80).aspx)



Les fournisseurs de données

Ce sont des composants explicitement conçus pour la manipulation des données et un accès aux données rapide, En ADO.NET, les principaux fournisseurs de données sont les suivants :

- Fournisseur de données « .NET Framework » pour SQL Server ;
- Fournisseur de données « .NET Framework » pour OLE DB;
- Fournisseur de données « .NET Framework » pour ODBC ;
- Fournisseur de données « .NET Framework » pour Oracle.
(abandonné par Microsoft)

Oracle et ADO.net

- Oracle Data Access Components (ODAC): offre des composants permettant d'accéder à des bases de données oracle par l'intermédiaire d'une application.net (ODP et OELDB)

- Site de téléchargement.

<http://www.oracle.com/technetwork/developer-tools/visual-studio/downloads/index.html>

- ODP.net(Oracle Data Provider) est un ensemble de classes permettant un accès aux données oracle.

ADO.net et oracle

- Pré-requis:

- Installation d'ODAC
- Ajout de référence dans visual studio. (C#).

La référence est: Oracle.DataAccess

- Ajout de l'espace de nom spécifique à ODP.net:
using Oracle.DataAccess.Client

- Les objets importants de ODP.net:

- **OracleConnection Class:** un objet OracleConnection représente une connexion à une base de données oracle.
- **OracleCommand Class:** un objet OracleCommand représente une commande SQL
- **OracleDataReader Class:** un objet OracleDataReader représente un flux de données forward-only, read-only.
- **OracleDataAdapter Class:** un objet OracleDataAdapter représente un objet a data provider .L'objet **DataAdapter** utilise des commandes pour mettre à jour la source de données après que des modifications aient été apportées au **DataSet**

L'objet OracleConnection

- Rôle : établit une connexion à la base de données oracle.
- Propriétés de la chaîne de connexion:
 - Data Source: définit la source de données oracle.
 - User Id: le nom d'un usager autorisé à accéder à la base de données
 - Password : le mot de passe de l'utilisateur autorisé.

Exemple: déclaration de la chaîne de connexion de nom ChaineConnexion

```
string ChaineConnexion = "Data Source=(DESCRIPTION="
+ "(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)"
+ "(HOST=205.237.246.251)(PORT=1521)))"
+ "(CONNECT_DATA=(SERVICE_NAME=ORCL)));"
+ "User Id=usager1;Password=oracle1;"
```

Voir à l'adresse : <http://www.connectionstrings.com/oracle#p16> pour les autres chaînes de connexion.

L'objet OracleConnection

- Méthodes importantes: de l'objet OracleConnection
 - Open: permet d'ouvrir une connexion
 - Close: permet de fermer la connexion.
 - CreateCommand: Crée et retourne un objet OracleCommand associé à OracleConnection.
 - D'autres méthodes à
http://download.oracle.com/docs/html/B28089_01/OracleConnectionClass.htm#i999634
- Constructeurs:
 - OracleConnection(): instancie un objet OracleConnection avec des valeurs par défaut.
 - OracleConnection(string): instancie un objet OracleConnection en utilisant la chaîne de connexion du provider (odp)

L'objet **OracleConnection**

- Établir une connexion en étapes:
 1. Déclarer une chaine de connexion : `ChaineConnexion`
 2. Instancier un objet `OracleConnection(ChaineConnexion)`:
`objConnexion`
 3. Associer ta chaine de connexion à ton `objConnexion`
 4. Ouvrir la connexion.

L'objet OracleConnection

- Exemple :

```
string ChaineConnexion = "Data Source=(DESCRIPTION="
    + "(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)"
    + "(HOST=205.237.246.251)(PORT=1521)))"
    + "(CONNECT_DATA=(SERVICE_NAME=ORCL)));"
    + "User Id=usager1;Password=oracle1;";
OracleConnection conn = new OracleConnection(ChaineConnexion);
conn.ConnectionString = ChaineConnexion;
conn.Open();
```

L'objet OracleConnection

- Ou encore

```
string Usager = "usager1";  
string MotdePass = "oracle1";  
string maBase = "(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 205.237.246.251)"  
    + "(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = orcl)))";
```

```
string ChaineConnexion = "Data Source =" + maBase  
    + ";User ID =" + Usager + ";Password =" + MotdePass;
```

```
OracleConnection conn = new OracleConnection(ChaineConnexion);  
conn.ConnectionString = ChaineConnexion;  
conn.Open();
```

L'objet OracleCommand

- Un objet OracleCommand représente une instruction SQL, une procédure stockée ou un nom de table.
- Il est responsable de formuler des requêtes SQL et de les passer à la base de données.
- s'il y a un résultat, il est responsable de les retourner dans un OracleDataReader ou autre.

L'objet OracleCommand

- **Propriétés importantes:**
 - **CommandText:** Obtient ou définit l'instruction SQL ou la procédure stockée à exécuter sur la base de données.
 - **CommandType:** Obtient ou définit une valeur indiquant la manière dont la propriété **CommandText** doit être interprétée.
 - **Parameters:** spécifie les paramètres de l'instruction SQL ou de la procédure stockée
- **Méthodes importantes:**
 - **ExecuteNonQuery:** exécute une instruction SQL ou une commande définie avec **CommandType** et **CommandText** et retourne le nombre de lignes affectées. Utiliser avec **Insert**, **update** et **Delte**.

L'objet OracleCommand

- Méthodes importantes (suite):
 - ExecuteReader: Execute un oracle CommandText et retourne un OracleDataReader (méthode surchargée)
 - ExecuteScalar: retourne la première colonne de la première ligne du résultat:
 - Clone :Crée une copie de cet objet OracleCommand.
 - Pour les autres méthodes:
http://download.oracle.com/docs/html/B28089_01/OracleCommandClass.htm
- Exemple:

L'objet OracleCommand

Exemple1:

```
string sql2 = "update typelivre set description ='Journal' where codetype =2";  
OracleCommand objCommand = new OracleCommand(sql2,conn);  
objCommand.CommandType = CommandType.Text;  
conn.Open();  
System.Console.WriteLine("statut de la connexion" + conn.State);  
objCommand.ExecuteNonQuery();
```

Exemple 2:

```
string sql = "select * from sydept ";  
OracleCommand objCommad = new OracleCommand(sql, conn);  
conn.Open();  
System.Console.WriteLine("statut de la connexion" + conn.State);  
OracleDataReader objRead = objCommad.ExecuteReader();  
Suite du code
```

L'objet OracleDataReader

- Définition: un objet OracleDataReader représente un ensemble d'enregistrements en lecture seulement et en avant uniquement (forward-only, read-only)
- Particularité: l'objet OracleDataReader est créé par l'exécution de la méthode ExecuteReader() de l'objet Command.
- C'est objet utilisé en mode connecté uniquement

http://download.oracle.com/docs/html/B28089_01/OracleDataReaderClass.htm#i1003252

L'objet OracleDataReader

- Quelques méthodes importantes:
 - Close: ferme l'objet DataReader
 - Dispose :libère les ressources utilisées par l'objet.
 - Read :lit l'enregistrement suivant
 - GetInt32: retourne la valeur de la colonne spécifiée sur 32 bits
 - GetInt16, GetInt64
 - GetString: retourne la valeur de la colonne spécifiée sous forme de chaîne de caractères
 - GetDateTime: retourne la valeur de la colonne spécifiée sous forme date.
 - GetName: retourne le nom de la colonne spécifié

L'objet **OracleDataReader**

- Quelques propriétés importantes:
 - **FieldCount**: retourne le nombre de colonne dans une ligne en cours:
 - **HasRows** : indique si **OracleDataReader** a une ou plusieurs lignes.
 - **RecordsAffected**: Obtient le nombre de lignes modifiées, insérées ou supprimées par l'exécution de l'instruction SQL.
 - **IsClosed**: Indique si **OracleDataReader** est fermé.

Voir aussi:

http://download.oracle.com/docs/html/B28089_01/OracleDataReaderClass.htm#i1003252

L'objet OracleDataReader

- Création de l'objet OracleDataReader:

```
OracleDataReader objRead = objCommad.ExecuteReader();
```

- Lecture du OracleDataReader

```
while (objRead.Read())  
{  
    Console.WriteLine("{0} - {1} - {2}",  
        objRead.GetInt32(0), objRead.GetString(1),  
        objRead.GetString(2));  
}
```

Exemple

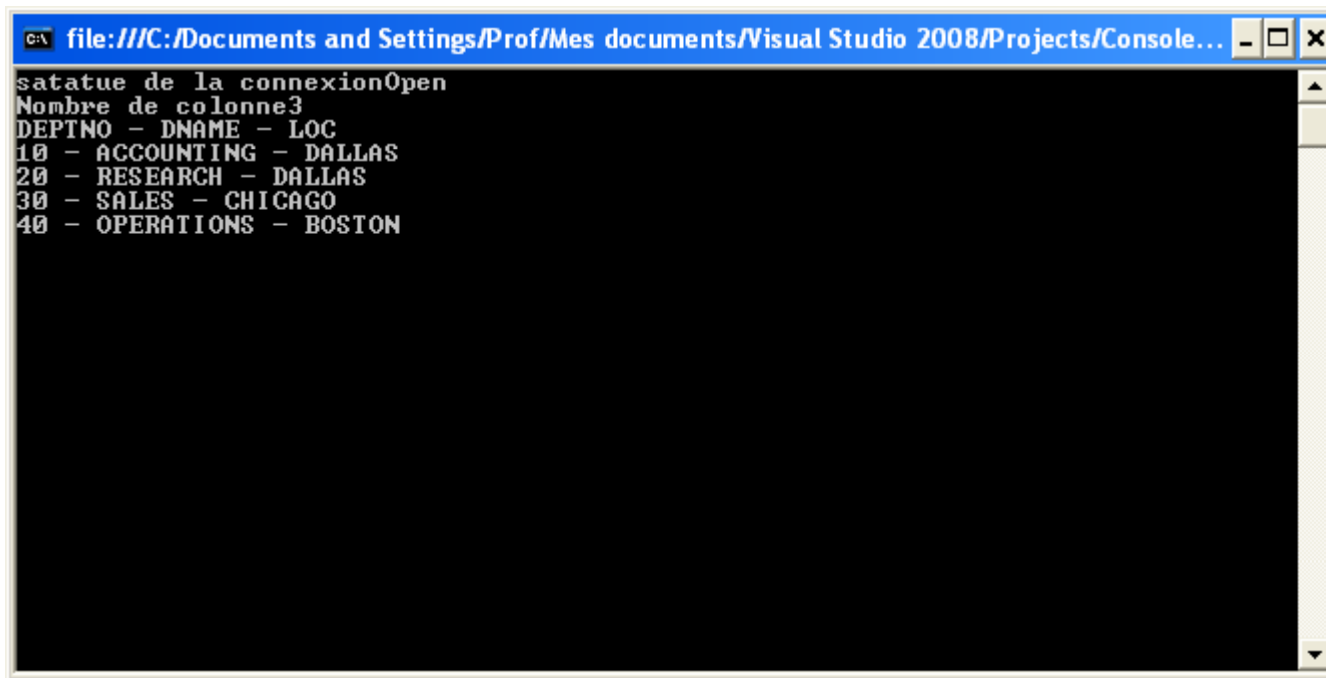
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using Oracle.DataAccess.Client;

namespace BD1Oracleconsole
{
    class Program
    {
        static void Main(string[] args)
        {
            string ChaineConnexion = "Data Source=(DESCRIPTION="
            + "(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=172.17.200.251)(PORT=1521)))"
            + "(CONNECT_DATA=(SERVICE_NAME=ORCL)));";
            + "User Id=usager1;Password=oracle1;";
        }
    }
}
```

Exemple(suite)

```
OracleConnection connexion = new OracleConnection(ChaineConnexion);
connexion.ConnectionString = ChaineConnexion;
    string sql = "select * from sydept ";
    OracleCommand objCommad = new OracleCommand(sql, connexion);
        connexion.Open();
            System.Console.WriteLine("satatue de la connexion" + connexion.State);
OracleDataReader objRead = objCommad.ExecuteReader();
int nbColonne = objRead.FieldCount;
System.Console.WriteLine("Nombre de colonne" + nbColonne);
Console.WriteLine("{0} - {1} - {2}",
    objRead.GetName(0), objRead.GetName(1), objRead.GetName(2));
    while (objRead.Read())
    {
        Console.WriteLine("{0} - {1} - {2}",
            objRead.GetInt32(0), objRead.GetString(1), objRead.GetString(2));
    }
    objRead.Close();
    connexion.Close();
    System.Console.ReadLine();
    }
}
```

Résultat



```
C:\ file:///C:/Documents and Settings/Prof/Mes documents/Visual Studio 2008/Projects/Console... - □ ×
satatue de la connexionOpen
Nombre de colonne3
DEPTNO - DNAME - LOC
10 - ACCOUNTING - DALLAS
20 - RESEARCH - DALLAS
30 - SALES - CHICAGO
40 - OPERATIONS - BOSTON
```