

Le gestionnaire de connexions OLEDB

Object Linking and Embedding Database

Objectifs

- Fournir un accès homogène aux SGBDR, grâce à des pilotes (*drivers*) appropriés
- Offrir une abstraction des SGBDR cibles (indépendance de l'application)
- Permettre l'exécution de requêtes SQL
- Proposer une mise en œuvre simple

OleDb se présente sous la forme d'une API, développée par Microsoft®. Initialement destinée à remplacer ODBC (Open Database Connectivity).

Le pendant pour Oracle est JDBC (Java Database Connectivity), qui est spécifique à Java (voir diaporama JDBC).

Architecture OLEDB

OLEDB repose sur une architecture consommateur-producteur (*consumer-provider*) :

- **Application (consommateur)** : Effectue le traitement et invoque les fonctions OLEDB pour envoyer des instructions SQL et récupérer les résultats.
- **Pilote (producteur)** : Traite les appels de fonction OLEDB, soumet les demandes SQL à une source de données spécifique et retourne les résultats à l'application.
- **Source de données** : Comprend les données auxquelles l'utilisateur souhaite accéder, ainsi que le système d'exploitation, le SGBD et la plateforme réseau associés (le cas échéant) utilisés pour accéder au SGBD.

Accès aux données : principe

Le principe général est le suivant :

1. Connexion à la base
2. Création d'une commande (requête sql)
3. Exécution de la commande
4. Lecture des résultats

La connexion à la base cible est réalisée via une chaîne de connexion. Par exemple, pour info-dormeur (avec authentification SQL Server) :

```
using System.Data.OleDb;
...
// nom de la base
string nomBase = "MusiqueSQL";
// chaîne de connexion (info-dormeur)
string ChaineBd = "Provider=SQLOLEDB;" +
    "Data Source=INFO-DORMEUR;Initial Catalog=" +
    nomBase + ";Uid=ETD;Pwd=ETD";
// Connexion à la base de données
OleDbConnection dbConnection = new OleDbConnection(ChaineBd);
dbConnection.Open();
```

Connexion à la base

2/2

... et sur un serveur local (avec authentication Windows) :

```
using System.Data.OleDb;
```

```
...
```

```
// nom de la base
```

```
string nomBase = "MusiqueSQL";
```

```
// chaîne de connexion (SQL Server local)
```

```
string ChaineBd = "Provider=SQLOLEDB;" +
```

```
    "Data Source=DESKTOP-FATLC4L;Initial Catalog=" +
```

```
    nomBase + ";Integrated Security=SSPI";
```

```
// Connexion à la base de données
```

```
OleDbConnection dbConnection = new OleDbConnection(ChaineBd);
```

```
dbConnection.Open();
```

L'acquisition de données peut se faire de façon « statique » par une requête simple :

```
string sql = "Select Nom_Musicien, Prénom_Musicien, " +  
             "Année_Naissance FROM Musicien";  
OleDbCommand cmd = new OleDbCommand(sql, dbConnection);  
OleDbDataReader reader = cmd.ExecuteReader();  
while (reader.Read()) {  
    string nom = reader.GetString(0);  
    string prénom = ""; int année = 0;  
    if (!reader.IsDBNull(1)) prénom = reader.GetString(1);  
    if (!reader.IsDBNull(2)) année = reader.GetInt32(2);  
    Console.WriteLine(nom + " " + prénom + " (" +  
        année.ToString()+")");  
}  
reader.Close();
```

L'accès peut également se faire de façon « dynamique » via une requête paramétrée :

```
string initiale = "W";
string sql2 = "Select Nom_Musicien, Prénom_Musicien FROM " +
    "Musicien " + "WHERE Nom_Musicien Like '" +
    initiale + "%' ORDER BY Nom_Musicien";
OleDbCommand cmd2 = new OleDbCommand(sql2, dbConnection);
OleDbDataReader reader2 = cmd2.ExecuteReader();
while (reader2.Read()) {
    // Code de traitement
}
reader2.Close();
```


Accès aux méta-données

La méthode `GetOleDbSchemaTable()` permet d'obtenir les méta-données d'un `OleDbConnection`. Elle renvoie un `DataTable` qui décrit les métadonnées des tables de la base concernée.

La méthode `GetSchemaTable()` permet d'obtenir les méta-données d'un `OleDbDataReader`. Elle renvoie un `DataTable` qui décrit les métadonnées des colonnes de la table concernée.

Voir sur Moodle la section « Dictionnaire de données » pour en savoir plus...

1. Application console

1. Création d'une application console C# sous Visual Studio. **Attention à bien choisir Application console (.NET Framework)**
2. Vérification des paramètres de connexion pour la base MusiqueSQL sur le serveur info-dormeur (phase de connexion)
3. Affichage des noms et prénoms des musiciens.
4. Affichage des noms et prénoms des musiciens dont l'initiale du nom est choisie par l'utilisateur.

2. Application Windows Forms (.NET Framework)

1. À l'aide de deux `ListBox`, afficher la liste des *compositeurs* et, pour un compositeur sélectionné, de ses œuvres

- connexion (appelée dans `Form1()`)
- création des classes `Musiciens` et `Œuvres` (cf. cours Moodle « Langages évolués », Chap. 7)
- Chargement des musiciens (`Form1()`)
- Si musicien sélectionné : chargement de ses œuvres (dans `listBox1_SelectedIndexChanged`)

2. Zone de texte pour filtrer le choix du compositeur..