



## Module 1 Développer l'interface d'une application informatique

Composants accès données	Séance	S01	Activité	A-001
--------------------------	--------	-----	----------	-------

Cette activité d'apprentissage vous fera découvrir les classes du framework d'accès aux données de l'architecture logicielle .Net désigné sous l'acronyme ADO.Net.  
Dans ce premier atelier nous utiliserons uniquement les classes mises en œuvre dans le mode d'accès « connecté ».

Sommaire de l'activité proposée :

<b>1</b>	<b>Présentation .....</b>	<b>2</b>
1.1	L'objet Connexion.....	2
1.2	Gérer les connexions.....	2
<b>2</b>	<b>L'objet Commande associée à des requêtes .....</b>	<b>3</b>
2.1	Les commandes qui ne retournent pas de jeu d'enregistrements .....	3
2.2	Les commandes qui retournent un jeu d'enregistrements.....	5

# 1 Présentation

Au cours de ce premier atelier, vous allez découvrir les techniques de programmation de composants d'accès aux données en mode connecté.

L'application cliente est une application Windows.

Il vous est proposé de découvrir les différentes classes d'ADO Net étape par étape en complétant les formulaires Windows mis à votre disposition sur le site.

Vous utiliserez le fournisseur de données propre à SQL Server tout au long des étapes de l'atelier.

Les classes de ce fournisseur de données (Data Provider) se trouvent dans l'espace de noms System.Data.SqlClient.

Les données seront extraites de la base de données Comptoir Anglais

## 1.1 L'objet Connexion

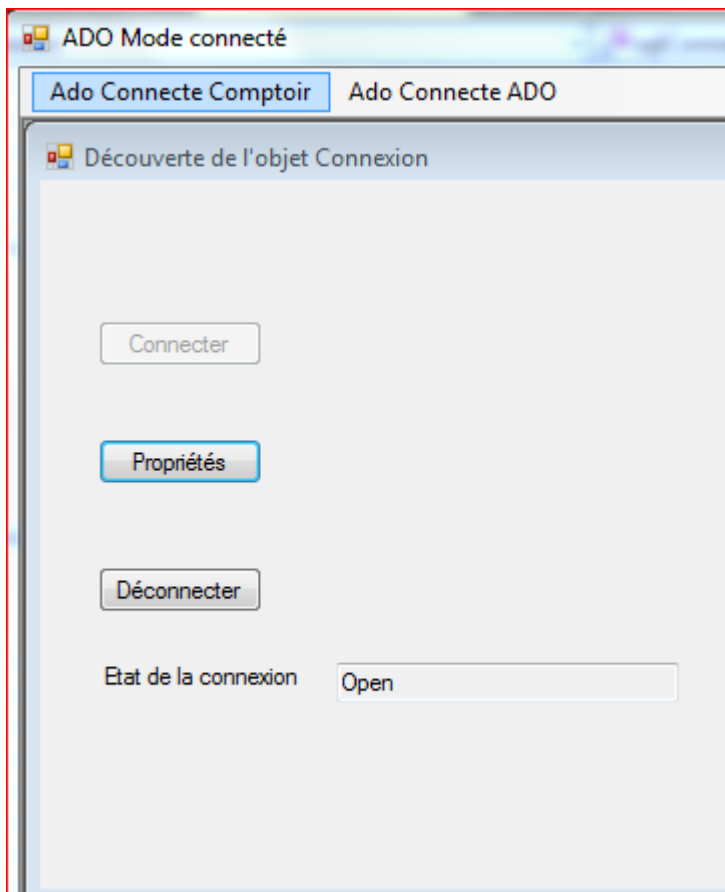
Consultez le document focus sur la génération de la chaîne de connexion et le chapitre 6 du support de cours « Ado.Net Mode Connecté ».

Pour définir la chaîne de connexion et comprendre les notions d'instance et de protocoles de transport, consultez le document à l'adresse suivante <https://metis.afpa.fr/mod/resource/view.php?id=8281738>.

Vous allez ici découvrir les principales méthodes et propriétés de l'objet connexion. Celui constitue la base de tout échange entre l'application client et le serveur de données.

## 1.2 Gérer les connexions

Première étape avant toute possibilité d'interaction avec les objets de la base de données, créer et ouvrir une connexion sur le serveur.



Modifiez le formulaire Connexion et programmez les actions prévues sur les différents boutons de commande présents.

**Connecter** : Création et ouverture de la connexion. Pour extraire la propriété ConnectionString :

```
Properties.Settings.Default  
.ADO_NetConnectionString
```

**Propriétés** : Affichage des propriétés de la connexion, au choix, dans une boîte de messages.

**Déconnecter** : Fermeture de la connexion si celle-ci est ouverte.

**Etat de la connexion** : Se met à jour sur changement d'état de la connexion. Voir événement de changement d'état.

## 2 L'objet Commande associée à des requêtes

C'est à travers cet objet, associé à une connexion valide et ouverte, que vont pouvoir s'opérer les échanges de données entre le serveur et l'application cliente.

Vous référez au chapitre 7 du support de cours « Ado.Net Mode Connecté ».

### 2.1 Les commandes qui ne retournent pas de jeu d'enregistrements

Dans ce premier exemple nous allons programmer des commandes qui :

- Retournent une valeur unique (scalaire)
- Exécutent une requête SQL et renvoient le nombre de lignes affectées.

Elles peuvent nécessiter la transmission et/ou la réception de paramètres.

L'objet commande permet d'exécuter des scripts sql et des procédures stockées.

Dans ce premier exemple, nous nous contenterons d'exécuter des scripts.

Ado Connecte Comptoir Ado Connecte ADO

Découverte de l'objet Commande

Requête avec réponse scalaire Execute Scalar

Sans paramètre : Nombre de clients

Avec paramètre : Nombre de commandes de ALFKI Résultat 89

Résultat requête suivante

Requête sans jeu de résultats Execute Non Query

Résultat requête suivante Nombre de lignes affectées

Vous devez programmer les actions sur les différents boutons de commande et définir les commandes ad'hoc en fonction de la nature des opérations.

1. La première est la forme la plus simple : Cette requête retourne le nombre de clients, donc une valeur scalaire.
2. La deuxième nécessite la définition d'un paramètre en entrée et est de type SELECT ... Where Colonne = @Paramètre.
3. Pour la troisième, vous avez le choix de la définition du script à exécuter. La seule limite reste que son exécution ne doit renvoyer qu'une valeur scalaire.
4. Pour la quatrième requête, programmez une requête de mise à jour ou d'insertion. Vous devez afficher le nombre de lignes affectées après appel de la méthode ExecuteNonQuery.

ADO Mode connecté - [Découverte de l'objet Commande]

Ado Connecte Comptoir Ado Connecte ADO

Requête avec réponse scalaire Execute Scalar

Sans paramètre : Nombre de clients 89

Avec paramètre : Nombre de commandes de ALFKI 6

Résultat requête suivante 2

select COUNT(distinct P.productid) from products P where P.ProductID not in (select ProductID from [Order Details])

Requête sans jeu de résultats Execute Non Query

Résultat requête suivante 14

Update Products  
set UnitPrice +=1  
where CategoryID in (select CategoryID from Categories  
where CategoryName like 'sauces%');

Exemple programmé :

Dans tous les cas, pensez à vérifier si des valeurs d'exception ne sont pas susceptibles d'entraîner des erreurs d'exécution.

## 2.2 Les commandes qui retournent un jeu d'enregistrements

Nous allons maintenant nous intéresser aux commandes dont la méthode, `ExecuteReader`, nous renverra un jeu de données qui pourra être traité par le biais d'un `DataReader`.

Programmez le formulaire en traitant un jeu de données qui contient des valeurs de différents types (chaîne, date, numérique,...)

Découverte de l'objet DataReader pour extraire des valeurs

Commande Paramétrée avec IdClient

ALFKI

```
select Top 5 C.CustomerID,C.CompanyName,C.ContactName,O.OrderID,O.OrderDate
from Customers C inner Join Orders O
on C.CustomerID = o.CustomerID
Where c.CustomerID = @CustomerID
order by OrderDate desc
```

Client ALFKI Nom ALFREDS FUTTERKISTE Contact Bost Cde 11011 Date mercredi 9 avril 2008  
Client ALFKI Nom ALFREDS FUTTERKISTE Contact Bost Cde 10952 Date dimanche 16 mars 2008  
Client ALFKI Nom ALFREDS FUTTERKISTE Contact Bost Cde 10835 Date mardi 15 janvier 2008  
Client ALFKI Nom ALFREDS FUTTERKISTE Contact Bost Cde 10702 Date samedi 13 octobre 2007  
Client ALFKI Nom ALFREDS FUTTERKISTE Contact Bost Cde 10692 Date mercredi 3 octobre 2007