



Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

Les bases fondamentales de Microsoft Synchronization Framework For ADO .NET

Version 1.0



James RAVAILLE

<http://blogs.dotnet-france.com/jamesr>



Sommaire

1	Introduction.....	3
1.1	Présentation	3
1.2	Pré-requis	3
2	Architecture de Microsoft Synchronization Framework	4
2.1	Présentation	4
3	Présentation détaillée de Synchronization Framework For ADO .NET	5
3.1	Présentation générale	5
3.2	Composants de synchronisation	5
3.3	L'agent de synchronisation.....	5
3.4	Les fournisseurs de synchronisation	6
3.5	Les sens de synchronisation	6
3.6	Gestion des conflits de données	7
4	Conclusion	8

1 Introduction

1.1 Présentation

Dans ce cours, nous allons étudier les bases fondamentales de Synchronization Framework For ADO .NET, un sous-ensemble de Microsoft Synchronization Framework. Il vous permettra d'acquérir les connaissances qui vous seront nécessaires, pour vous permettre de réaliser une application, synchronisant deux bases de données entre elles. Ce sujet sera traité dans le prochain chapitre intitulé « Premiers pas avec Microsoft Synchronization Framework For ADO .NET ».

1.2 Pré-requis

Pour lire ce cours, nous vous conseillons d'avoir lu :

- Le cours d'introduction à Microsoft Synchronization Framework.
- Les cours sur les bases fondamentales d'ADO .NET.

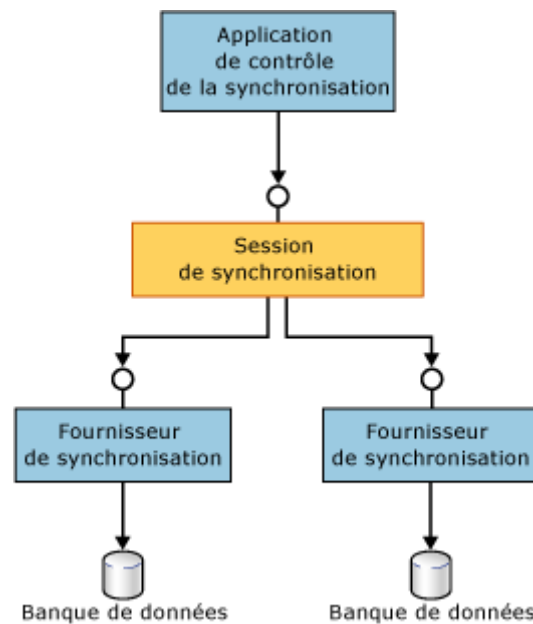
Tous ces cours sont publiés sur Dotnet-France.



2 Architecture de Microsoft Synchronization Framework

2.1 Présentation

Voici une architecture de haut niveau de Microsoft Synchronization Framework :



Cette architecture permet de synchroniser des sources de données locales ou distantes (ici appelées « Banques de données »), et quelque soit le type du poste de travail sur laquelle l'application .NET s'exécute. Avec Synchronization Framework For ADO .NET, une source de données sera vue par l'application de contrôle de la synchronisation comme la base de données « cliente », et l'autre comme la base de données « serveur ».

Les fournisseurs de synchronisation, propres à chaque type de magasin de données, permettent d'instaurer un couplage faible, entre les composants de synchronisation de données (agissant lors de la synchronisation) et les sources de données. Ils permettent de fournir les métadonnées et les données des sources de données. La session de synchronisation correspond à une opération de synchronisation de données des deux sources de données. Elle communique automatiquement l'état de la synchronisation, les erreurs, les conflits de données l'application l'ayant créée. Nous verrons qu'il existe des sens (directions) de synchronisation permettant de désigner si la synchronisation est uni-directionnelle ou bi-directionnelle, ... Quant à l'application de contrôle de synchronisation, elle permet de « piloter » la synchronisation : démarrage, arrêt, gestion des erreurs, gestion des conflits...

3 Présentation détaillée de Synchronization Framework For ADO .NET

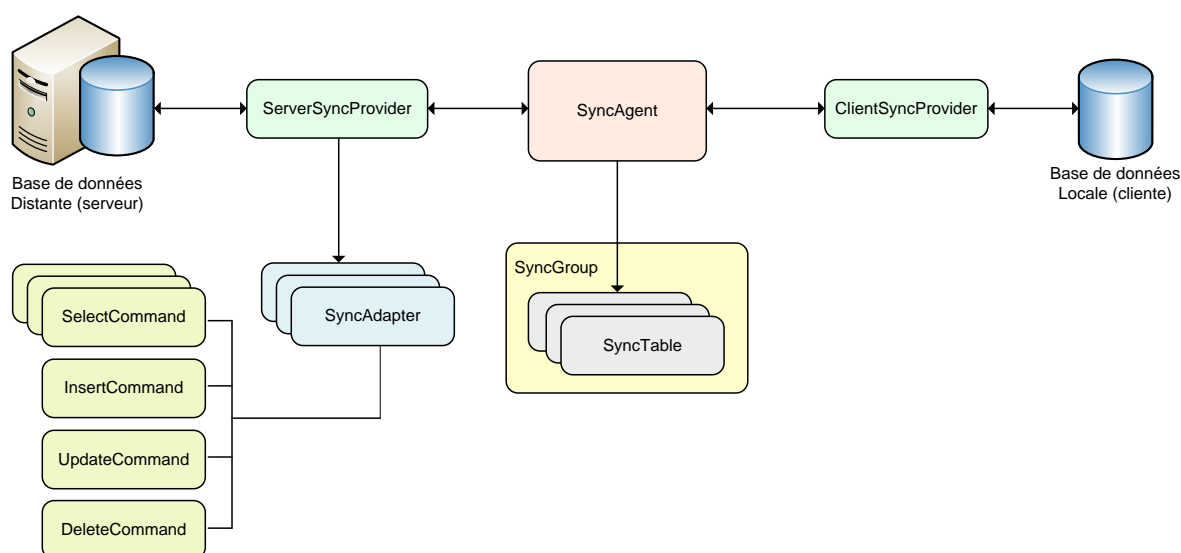
3.1 Présentation générale

Synchronization Framework For ADO .NET est un sous-ensemble de Microsoft Synchronization Framework, permettant de synchroniser deux sources de données de même types ou « compatibles », au travers de providers de données ADO .NET.

Synchronization Framework For ADO .NET sera sans nul doute principalement utilisé dans les applications de gestion de données dites « nomades », afin que des applications utilisant une base de données « déportée » puissent synchroniser leurs données avec une base de données distante.

3.2 Composants de synchronisation

Voici les composants fondamentaux de Synchronization Framework For ADO .NET, sur lesquels nous allons nous appuyer pour comprendre son fonctionnement :



3.3 L'agent de synchronisation

L'agent de synchronisation (*SyncAgent*) joue un rôle essentiel lors des opérations de synchronisation. Il joue un rôle de chef d'orchestre, qui va diriger les différentes étapes de la synchronisation, lors de l'exécution de la session de synchronisation. Pour accéder aux bases de données, il utilisera les fournisseurs de synchronisations (aussi appelé providers).

L'agent de synchronisation possède une collection d'objets *SyncTable*, un par table à synchroniser. Les tables peuvent être regroupées dans des regroupements de tables nommés *SyncGroup*. Par défaut, une table est toujours rattachée à un *SyncGroup*.

Le fournisseur de synchronisation côté serveur possède une collection de *SyncAdapter*, un par table à synchroniser. Ils ne sont pas sans rappeler les objets de type

DataAdapter du modèle objet ADO .NET. Chacun de ces *SyncAdapter* possède des propriétés contenant des objets Command (ADO .NET) permettant :

- De sélectionner les enregistrements à ajouter, à mettre à jour, à supprimer ;
- D'ajouter, modifier, supprimer un enregistrement dans la table ;
- ...

3.4 Les fournisseurs de synchronisation

L'agent de synchronisation possède deux fournisseurs de synchronisation :

- Un fournisseur client, appelé *ClientSyncProvider*. Il permet d'accéder à une source de données « locale ».
- Un fournisseur distant, appelée *ServerSyncProvider*. Il permet d'accéder à la base de données distante.

3.5 Les sens de synchronisation

Par défaut, l'application de contrôle de la synchronisation permet uniquement de mettre à jour les données de la base de données cliente. Les données de la base de données distante ne sont pas modifiées.

Pour définir les sens de synchronisation, Synchronization Framework For ADO .NET propose une énumération : *Microsoft.Synchronization.Data.SyncDirection*. Ses différents éléments sont les suivants :

Valeur	Libelle	Description
0	DownloadOnly	Seules les données de la base de données locale sont mises à jour. Il s'agit du sens de synchronisation par défaut.
1	UploadOnly	Seules les données de la base de données distante sont mises à jour.
2	Bidirectional	Les données des bases de données locale et distante sont mises à jour.
3	Snapshot	Seules les données de la base de données locale sont mises à jour. A chaque synchronisation, les données de la base de données locale à synchroniser sont intégralement mises à jour.

Vous pouvez définir un sens de synchronisation sur chacune des tables à synchroniser, au travers de la propriété *SyncDirection* des objets de type *SyncTable* de l'agent de synchronisation.



3.6 Gestion des conflits de données

Synchronization Framework For ADO .NET propose des mécanismes de gestion de conflits de données, pouvant survenir lors de synchronisation de données entre une base de données distante et une base de données locale. Ils permettent de définir :

- Soit de manière automatique les données à conserver.
- Soit de permettre à l'utilisateur d'intervenir afin qu'il puisse choisir.



4 Conclusion

Ce cours vous a permis de découvrir les bases fondamentales de Synchronization Framework For ADO .NET.

Dans le prochain chapitre, nous réaliserons une application permet de le mettre en œuvre, afin de synchroniser une base de données distantes (SQL Server 2008) avec une base de données locales (fichier de base de données SQL Server CE - .sdf).