



Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

Présentation du Framework Entity

Version 1.1



James RAVAILLE

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

Sommaire

1	Introduction.....	3
1.1	Présentation	3
1.2	Les évolutions du Framework .NET	3
1.3	Rôle du Framework Entity dans le développement d’applications.....	3
2	Entity Data Model.....	5
2.1	Présentation générale	5
2.2	Présentation détaillée	6

1 Introduction

1.1 Présentation

Ce cours présente le Framework Entity dans les grandes lignes, et vous propose de le mettre rapidement en œuvre, au travers d'un exemple pratique. Cette présentation pratique permet de vous fournir les connaissances de base, afin de pouvoir vous permettre d'évoluer par vous-même dans l'utilisation du Framework Entity.

Dans les cours suivants sur le Framework Entity publiés sur Dotnet-France, vous pourrez découvrir différents aspects et concepts, qui vous permettront d'approfondir vos connaissances.

1.2 Les évolutions du Framework .NET

Depuis, la sortie de la première version d'ADO .NET, Microsoft propose de nouveaux composants, permettant d'améliorer l'accès aux données dans les applications .NET :

- Framework .NET 1.0/1.1 : contient les briques fondamentales d'ADO .NET, avec les classes permettant d'accéder, lire, et gérer des données contenues dans une base de données (connection, command, DataReader, DataAdapter, DataTable, DataSet, CommandBuilder, ...).
- Framework .NET 2.0 : ADO .NET évolue. Le Framework .NET propose de nouvelles classes de plus haut niveau offrant plus de possibilités dans l'accès aux données, et nous permettant d'écrire moins de code dans nos applications (TableAdapter, DataSource, ...).
- Framework .NET 3.0 : extensions du Framework 2.0, proposant les extensions suivantes : Windows Communication Foundation (WCF), Windows Presentation Foundation (WPF), Windows Workflow Foundation(WF), et CardSpace.
- Framework .NET 3.5 : Microsoft fait évoluer les langages C# et VB .NET, de manière à leur ajouter de nouveaux concepts (initialiseur d'objets, méthodes d'extension, inférence de type, type anonyme, expression Lambda, ...). L'introduction de ces concepts ont permis d'intégrer dans le jeu instructions des langages, des instructions de requêtage, permettant de requêter différentes sources de données : grappes d'objets (Linq For Object), flux XML (Linq For XML), base de données SQL Server (Linq For SQL, Linq For Entities).

1.3 Rôle du Framework Entity dans le développement d'applications

Dans la conception de leurs applications, les développeurs calquaient fortement le diagramme de classe métier de leur application, sur le schéma de base de données. La conception d'un schéma de base de données relationnelle, distinguent suivant les trois phases suivantes :

- Conception du modèle conceptuel de données (MCD). Le concepteur modélise les entités de la base de données et leurs relations.
- Conception du modèle logique de données (MLD). A partir du MCD, le concepteur normalise entités et relations sous la forme de tables, de clés étrangères et de contraintes d'intégrité.
- Conception du modèle physique de données. A partir du MLD, le concepteur va définir des options de stockage des données, les index, ... dans la base de données.

L'Entity Framework est une technologie intégrée à ADO .Net, qui permet de développer des applications de gestion de données, sans se soucier de la conception du schéma de base de données devant contenir les données de l'application. Le développeur ne sera plus obligé de baser la conception de son application sur le modèle logique de données, mais sur le modèle conceptuel, prenant ainsi de la distance avec la manière dont les données sont stockées dans la base de données.

2 Entity Data Model

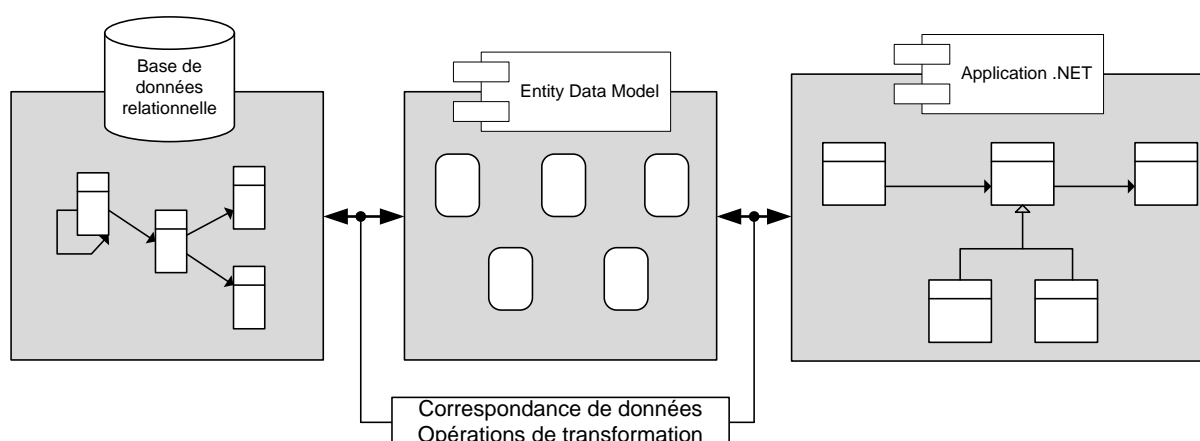
2.1 Présentation générale

Lors de la création d'une application, un schéma de base de données est établi. Il permet de persister dans le temps, toutes les données contenues dans les objets manipulés par l'application. Ces données sont organisées de manière relationnelle (entité/relation) et normalisée.

Puis le un diagramme de classes est créé, permettant de gérer les données contenues dans la base de données. Ce diagramme contient des classes, qui sont reliées entre elles par des relations : des relations d'association et des relations d'héritage.

La conception du diagramme de classes peut être proche du diagramme de base de données, mais elle présente néanmoins des différences :

- Les noms des classes peuvent être différents.
- Les noms des attributs et leurs types peuvent être différents.
- Un arbre d'héritage de classes peut permettre de créer des objets qui seront enregistrées dans une ou plusieurs tables de la base de données. Il n'existe alors pas de correspondance une table -> une classe.
- ...



Entity Data Model est un composant du Framework Entity. Il permet de définir un modèle indépendant d'entités intermédiaires, entre le schéma de base de données et le diagramme de classes. Il permet de créer une couche d'abstraction entre ces deux diagrammes. Cette abstraction permet aux deux modèles d'évaluer de manière indépendante. Ces entités permettront d'assurer :

- La correspondance entre les deux diagrammes.
- Des transformations et traitements de données.

Pour interroger notre base de données, Microsoft a fait évoluer Linq pour créer Linq To Entities. Pour en apprendre plus, nous vous invitons à lire le chapitre publié sur ce sujet, sur Dotnet-France.

2.2 Présentation détaillée

Le Framework Entity constitue une couche d'abstraction, au travers d'entités définies dans un modèle, appelé Entity Data Model (EDM). Ce modèle est divisé en 3 parties :

- Modèle conceptuel : syntaxe CSDL au format XML (*Conceptual Schema Definition Language*) qui définit les entités et les associations entre elles. Il décrit les classes utilisées dans l'application pour accéder et gérer les données.
- Modèle logique : syntaxe SSDL (*Store Schema Definition Language*). Il définit les tables et l'ensemble des contraintes sur ces tables. Il présente ainsi une vue de la base de données.
- Schéma de liaison : syntaxe MSL (*Mapping Specification Language*) qui permet de faire le lien entre les 2 modèles précédents. Il réalise alors les opérations de mapping.

