

Faculté des Sciences de Monastir
Département des Sciences de l'Informatique

Notes de cours sur

La technologie .NET

Par : Karim Kalti

SOMMAIRE

- Introduction à la technologie .NET
- Le Langage C#.NET
- Les IHM en .NET
- Accès aux données (ADO.NET)
- Architectures des applications Web dynamiques
- ASP.NET
- Les services Web
- Les serveurs de média (audio et vidéo streaming)
- Développement et déploiement d'applications Web multimédia.

INTRODUCTION AU FRAMEWORK .NET

Qu'est ce que le Framework .NET :

D'après Microsoft, le Framework .NET est une nouvelle plate-forme informatique qui simplifie le développement d'applications dans l'environnement fortement distribué d'Internet. (*Source MSDN*)

Principaux objectifs

Le Framework .NET est conçu pour remplir les objectifs suivants : (*Source MSDN*)

- Fournir un environnement cohérent de programmation orientée objet que le code objet soit stocké et exécuté localement, exécuté localement mais distribué sur Internet ou exécuté à distance.
- Fournir un environnement qui garantit l'exécution sécurisée de code.
- Fournir aux programmeurs un environnement cohérent qui permet de développer une grande variété de types d'applications comme les applications Windows, les services Windows, les applications embarquées et les applications Web.
- Générer toutes les communications à partir des normes d'industries pour s'assurer que le code basé sur le Framework .NET peut s'intégrer à n'importe quel autre code. (forte utilisation du XML)

L'architecture du Framework .NET

Le Framework .NET possède une architecture composée de trois principales couches :

- La CLS qui est une spécification visant à garantir l'interopérabilité des langages de programmation en .NET.
- La bibliothèque de classes .NET (API.NET): qui permet de développer des applications .NET
- La Common Language Runtime (CLR) : qui permet d'exécuter les applications .NET



La Common Language Specification (CLS)

- La *Common Language Specification* est une normalisation qui doit être respecté et implémenté par tout langage de programmation qui se veut capable de créer des applications .NET.
- Cette spécification s'intéresse à la normalisation d'un certain nombre de fonctionnalités comme par exemple la manière d'implémenter les types de données, les classes, les délégués, la gestion des événements, etc ...
- La CLS a pour objectif de garantir l'interopérabilité entre les langages .NET. En effet, deux langages conformes à la CLS peuvent facilement échanger des données entre eux (ils implémentent au bas niveau de la même manière ces données). Il est possible par exemple de créer une classe dans un langage conforme à la CLS qui hérite d'une autre classe définie dans un autre langage conforme à la CLS.

La bibliothèque de classes .NET

- La bibliothèque de classes .NET est une API complètement orientée objet qui offre la possibilité de développer des applications allant des traditionnelles applications à ligne de commande ou à interface graphique utilisateur (GUI, Graphical User Interface) jusqu'aux applications qui exploitent les dernières innovations fournies par ASP.NET, comme les services Web XML et les Web Forms.
- La version 1.0 du .NET Framework comporte une bibliothèque d'environ 9000 classes et 270000 méthodes.
- Ces classes et leurs méthodes couvrent presque tous les besoins du développement d'applications (Accès aux données, programmation réseau, développement d'interfaces graphiques, développement Web, fonctions mathématiques, manipulation des chaînes, des dates ...).
- Ces classes sont regroupées d'une manière thématique et hiérarchique en espaces de noms.

La bibliothèque .NET : une API de programmation multi-langages

Langages classiques de programmation

- Les langages classiques : chaque langage possède sa syntaxe de base (liste de mots-clés plus les règles d'écriture)
- Un langage est toujours accompagné d'une bibliothèque de fonctions. Ces fonctions enrichissent les possibilités du langage.
- Exemples : fonctions mathématiques, fonctions de manipulation des chaînes de caractères, fonctions graphiques, fonctions systèmes, ...

Langage pour programmer avec la bibliothèque .NET

- La bibliothèque .NET n'est pas spécifique à un langage donné (indépendante des langages).
- Elle est supportée par plusieurs langages (environ 27 langages d'après Microsoft : VB, C++, C#, Java, Jscript, Delphi, Eiffel, Perl, Python, COBOL, ...).
- Les trois langages les plus utilisées pour développer en .NET sont VB.NET, C#.NET, et C++.NET.
- Chaque langage qui supporte la bibliothèque .NET (conforme à la CLS) peut instancier les classes de cette bibliothèque et utiliser ses méthodes.

Exemple :

```
//// VB.NET ////
Dim i as Integer
i=5
Console.WriteLine(i)

//// C++.NET ////
int i;
i=5;
Console::WriteLine(i);

//// C#.NET ////
int i;
i=5;
Console.WriteLine(i);
```

Cet exemple montre que chacun des trois langages utilise sa propre syntaxe pour la déclaration des variables, mais les trois font appel à la même méthode .NET pour faire l'affichage bien sûr chacun avec ses propres règles d'écriture.

Organisation de la bibliothèque de classes .NET



La bibliothèque .NET est constituée de trois couches de classes offrant trois catégories de services :

Première couche : Base classes Library (BCL)

La BCL rassemble des classes permettant d'effectuer les opérations de base telles que la manipulation de chaînes de texte, la gestion des entrées/sorties, des communications réseaux, des threads, etc.

Deuxième couche : les classes de données et XML

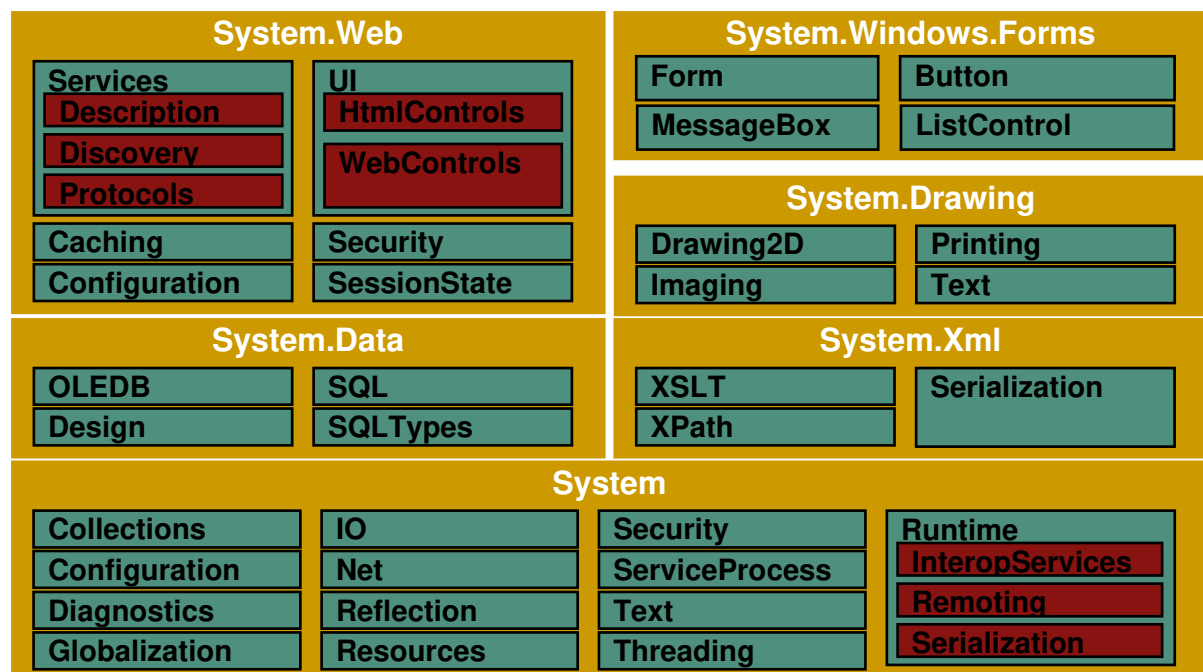
La deuxième couche est composée de deux bibliothèques de classes d'accès aux données :

- La bibliothèque ADO.NET, s'élevant sur les bases de l'ancien ADO (*ActiveX Data Objects*) et permettant l'accès sous format XML aux interfaces de bases de données SQL Server ODBC, OLEDB, ORACLE, et aux fichiers XML.
- Une bibliothèque de classes permettant de manipuler les données XML. On y trouve par exemple les classes XSLT permettant la transformation d'un document XML vers n'importe quel type d'autre document.

Troisième couche : les services Web, les applications Web et les applications Windows

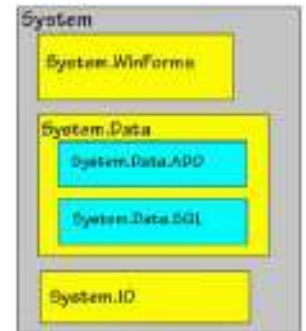
La dernière couche, la plus élevée, est utilisée pour la création des applications Web et des applications Windows et notamment la partie interface :

- Les applications Web peuvent se présenter sous formes de pages Web dynamiques et statiques ou sous forme de services Web. La technologie utilisée pour leur création est l'*ASP.NET*. Elle utilise pour la réalisation des interfaces de nouveaux composants appelés les *WebForms*.
- Les applications Windows et particulièrement leurs interfaces sont créées à l'aide des *WinForms*.



La bibliothèque de classes .NET

- Les classes de la bibliothèque .NET sont organisées sous forme d'espaces de noms hiérarchisés. Chaque espace de noms peut comporter un ensemble de classes et/ou un ensemble de sous-espaces de noms.
- L'accès à une classe s'effectue à l'aide son nom complet. Ce nom complet se compose de la liste hiérarchique des espaces de noms plus le nom de la classe en question. Ces noms étant reliés entre eux par des points.

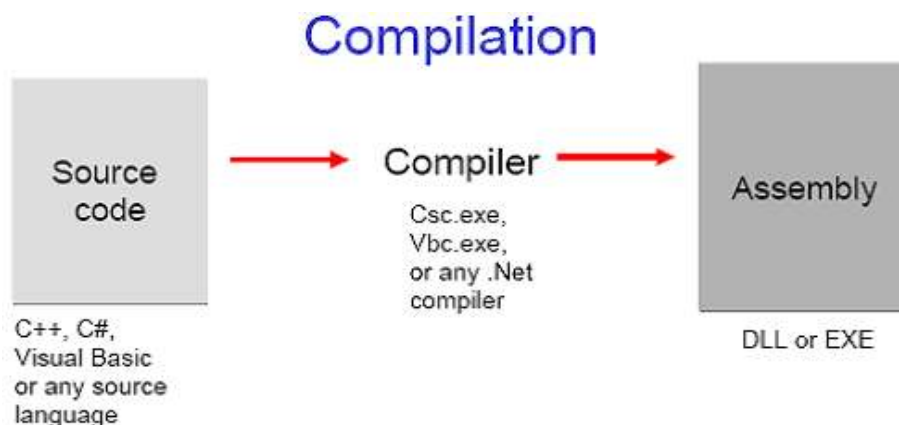


Exemple :

- La classe DataSet qui se trouve dans l'espace de noms "System.Data.ADO" se déclare comme "System.Data.ADO.Dataset".
- La classe Console qui se trouve dans l'espace de noms "System" se déclare comme "System.Console".

Le MSIL

- La compilation d'un programme écrit en .NET conduit vers la création d'un fichier exécutable (fichier .exe).
- Cet exécutable n'est pas écrit en code machine comme les exécutables classiques mais avec un langage intermédiaire appelé MSIL acronyme de **MicroSoft Intermediate Language**.
- L'exécution des fichiers compilés en MSIL ne peut pas être directement assurée par les services du système d'exploitation.



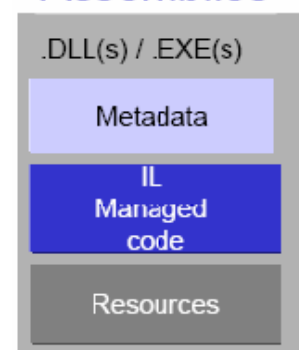
L'assemblage

- L'assemblage est le fichier *exe* ou *dll* produit par la compilation d'un code .NET.
- Un assemblage contient trois types de données :
 - Le code MSIL qui résulte de la compilation.
 - Les méta données.
 - Les ressources.

MSIL dans un assembly

Un code .NET peut comporter plusieurs classes. Au moment de la compilation, toutes ces classes sont compilées en MSIL dans un même assemblage. Les classes d'un même assemblage doivent avoir un seul point d'entrée (Une seule classe parmi celles d'un assemblage doit avoir une méthode qui joue le rôle de fonction principale). Il est à rappeler que le point d'entrée est défini par la méthode *Main* pour les applications de type console, par la méthode *WinMain* pour les applications Windows et par *DllMain* pour les dll (Dynamic Linked Library).

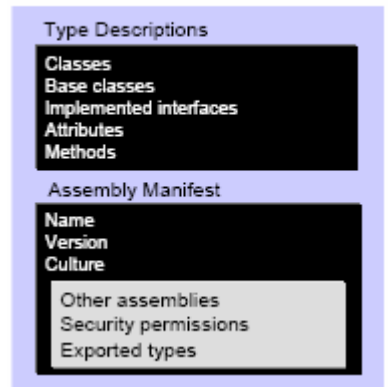
Assemblies



Les méta données

Les données qui accompagnent le code MSIL dans un assemblage sont de deux catégories :

- Des données de description des types : les types sont les classes compilées dans l'assemblage. La description concerne alors les spécifications de ces classes (noms, attributs, méthodes, droits d'accès, interfaces implémentées, ...). Ces informations sont généralement utilisées par les EDI pour la complétion de code.
- L'*assembly manifest* : Le manifeste contient des informations du genre :
 - *Nom* : le nom de l'assemblage. Il sert à la résolution de portée. Il est le même que celui du fichier de l'assemblage moins l'extension.
 - *Version* : la version de l'assemblage.
 - *Culture* : permet d'indiquer le type de ressource à utiliser en ce qui concerne la culture (langue, écriture de droite à gauche, type de calendrier ...).
 - *Autres assemblages* : les assemblages nécessaires à l'exécution de l'assemblage courant. Ces assemblages sont généralement des dlls.



Les ressources

Les éventuelles ressources utilisées par l'assemblage : icônes, bitmap, ...

La Common Language Runtime (CLR)

- La Common Language Runtime est un environnement qui assure l'exécution des programmes .NET. Elle joue le rôle de la machine virtuelle de Java mais pour les programmes écrits en .NET.
- Elle interprète les fichiers exécutables compilés en MSIL.
- Elle fournit des services tels que :
 - La gestion de la mémoire (à travers le Garbage Collector GC).
 - La gestion des exceptions.
 - La gestion des threads.
 - L'interopérabilité entre plusieurs langages.
 - Le chargement dynamique des modules à exécuter.
 - La compilation vers un code machine natif du MSIL et le contrôle de l'exécution des programmes.



- Un programme managé (Managed program) est un programme compilé en MSIL, son exécution est gérée par la CLR.
- Un programme non managé est un programme compilé en code natif. Son exécution est directement prise en charge par les services du système d'exploitation.
- VB.NET et C#.NET ne permettent de créer que des programmes managés. C++.NET permet de créer des programmes managés et non managés.