Vue d’ensemble de la plate-forme Microsoft .Net

Programmation .Net avec C#

Ce que nous faisons ici

- Votre Nom & Prénom.
- Votre parcours.
- Vos connaissances en développement.
- Vos connaissances en .Net.
- Ce que vous aimez / aimez moins.

Quelques petites règles de jeu

- Tout ceci dans le plaisir.
- Poser toutes les questions.
- Rallumer les téléphones aux pauses.
- Respect.
- Ecoute.
- Co-responsabilité.

Notre programme

- Vue d'ensemble de la plate-forme Microsoft .Net
- Visual Studio
- Le langage C#
- Programmation Orientée Objet avec C#
- Concepts avancés
- ADO.Net
- Web Services et WCF
- Windows Forms
- ASP.Net / MVC

Introduction

- Depuis la création de Windows, Microsoft a mis à disposition une API (Application Programming Interface), pour développer des logiciels exploitant les fonctionnalités du système.
  - Win 16 : Langage C sur 16 bits.
  - Win 32 : Evolution de la Win 16 sur les architectures 32 bits.
  - MFC : Encapsule la Win 16 puis la Win 32 dans un paradigme objet en utilisant le langage C++.
- Au fil des années, les APIs se sont multipliées.
- Pour répondre à cette complexification croissante, Microsoft a développé une technologie pour gouverner toutes les API :
  - La plateforme .Net

Présentation

- Le Framework .Net est le modèle de développement que propose Microsoft en code managé.
- Avec Visual Studio, il fournit un environnement cohérent et productif pour les développeurs.
- Il offre des possibilités de développement sur Windows XP, Vista, Seven, Windows Server, et également pour Microsoft Office System.

La plate-forme

- Multi Langages.
- Mono Plate-forme (Multi Plateformes Windows).
- Common Language Runtime (CLR).
- Common Intermediate Language (CIL).
- Framework Class Library (FCL).
- Déploiement.
- Sécurité
- Code managé.

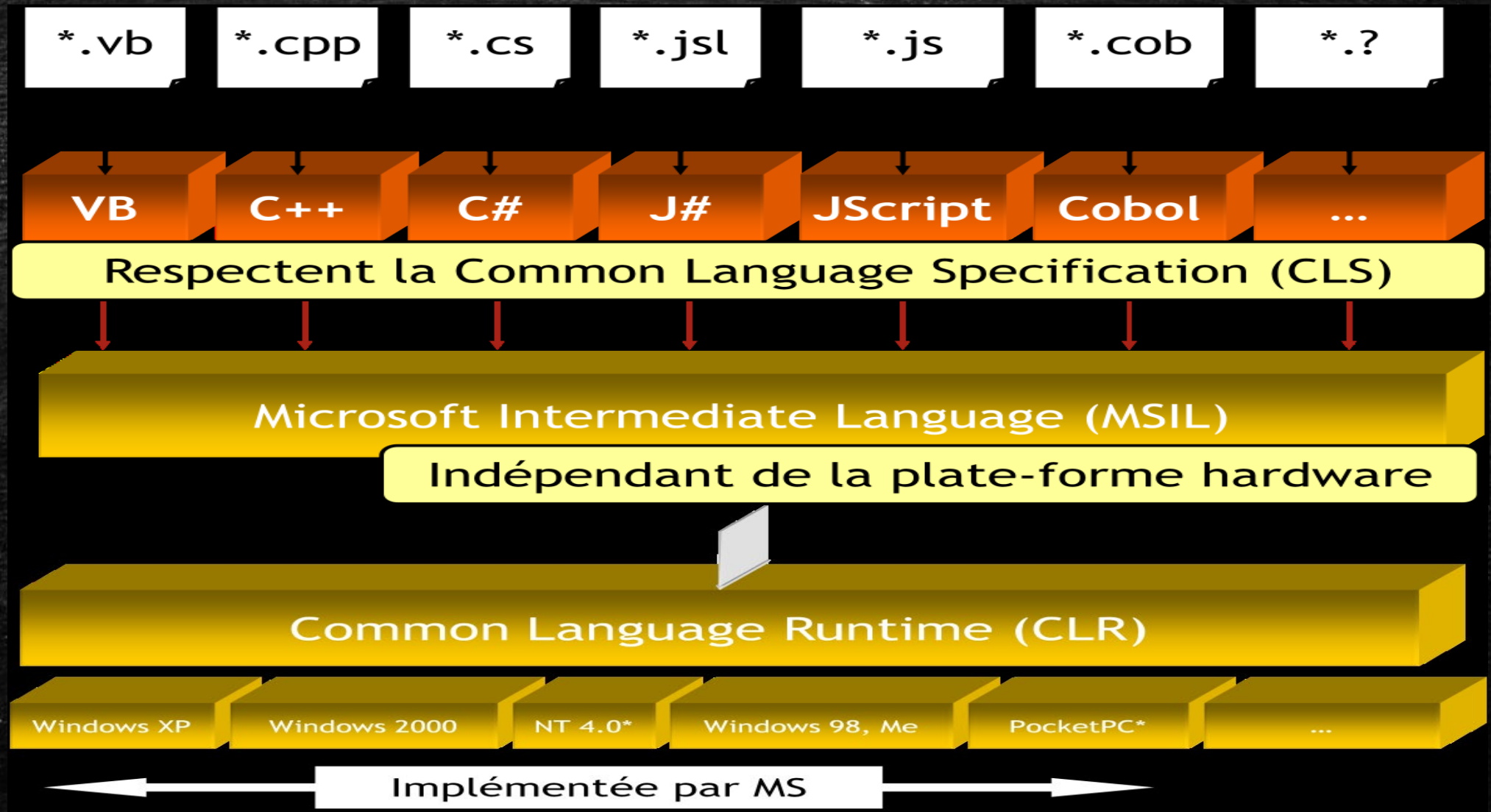
Managed Code Vs Unmanaged Code

- Code managé :
  - Code qui s'exécute sous le contrôle de la CLR.
  - Améliore la sûreté du fonctionnement.
  - Simplifie la vie au développeur.
  - Ne dépend pas d'un processeur donné.
- Code Non Managé :
  - Code qui s'exécute directement par le processeur.
  - Code machine spécifique à un processeur donné (natif).

Common Language Runtime

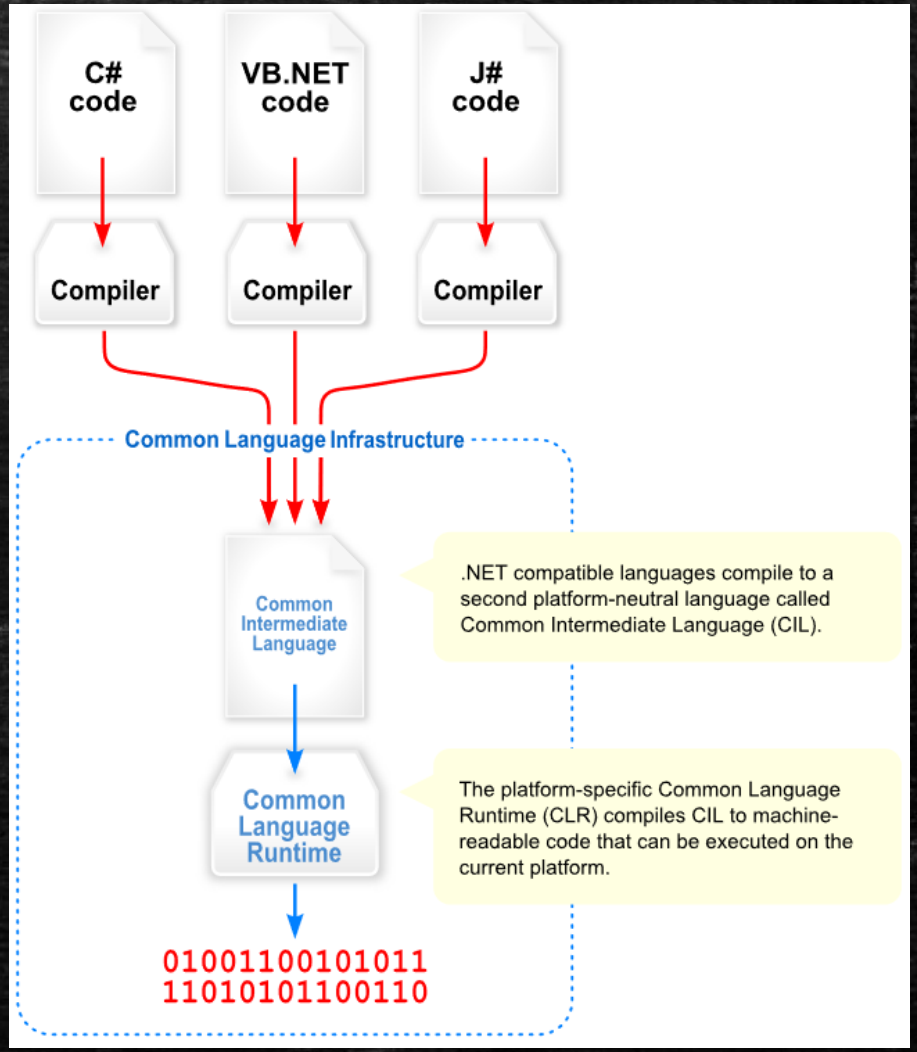
- Il s'agit de l'environnement dans lequel s'exécute tout code managé.
- Le CLR est souvent vu comme une machine virtuelle, dans la mesure où il permet l'exécution d'un code écrit pour une architecture différente.
- Il fournit de nombreux services :
  - La gestion de la mémoire (ramasse-miette).
  - La gestion des exceptions.
  - La gestion des tâches.
  - Les types de base (entier, chaîne de caractères…)

Multi Langages



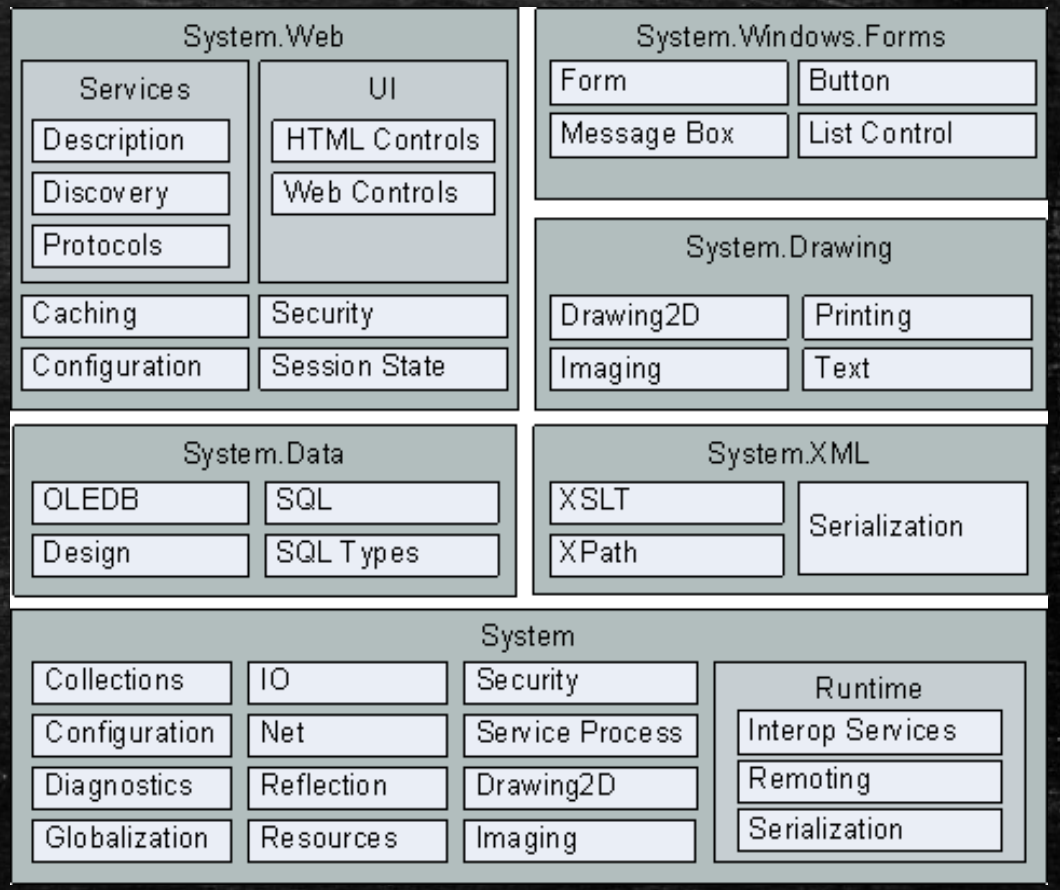
Compilation et Exécution

- Compilation du code source :
  - Syntaxe au choix (CLS).
  - Production d'un MSIL (CIL).
  - Insertion des méta données.
- Exécution par la CLR :
  - Chargement des classes (Class Loader).
  - Compilation en code natif à la volée.
  - Exécution du code natif avec application des règles de sécurité.



Framework Class Library

- Un ensemble de classes , d'interfaces, et de types partagés.
- Accessible depuis tous les langages.
- Inclut dans le kit de développement Microsoft .Net.
- Permet d'accéder au fonctionnalités du système.
- C'est le fondement des applications, composants, et contrôles du .Net Framework.



Les assemblies

- Une assembly est l'unité de déploiement minimale.
  - Un « exe » ou une « dll » sera appelée assembly en .Net
  - Les compilateurs .Net fabriquent ces assemblies.
- Une assembly est constituée de :
  - Code source en MSIL.
  - Métadonnées.
  - Une signature (RSA), ou un nom fort (N° Version).
- On distingue les assemblies :
  - Private : Utilisées par un programme.
  - Shared : Partagées au niveau du système (GAC).

Global Assembly Cache

- Emplacement physique : « C:\windows\Assembly ».
- Espace permettant de partager des assemblies « dll » qui seront accessibles par les programmes de la machine.
- Le GAC n'accepte que les assemblies signées.
- Le GAC permet la cohabitation de plusieurs versions d'un même assembly (d'où le besoin d'un nom fort).
- L'outil « gacutil.exe » gère l'inscription /suppression dans le GAC.

Garbage Collector

- Principe :
  - Le GC (ramasse-miettes) du .Net Framework manage l'allocation et la libération de la mémoire dans votre application.
- Utilisation :
  - Chaque fois que vous utiliser l'opérateur « new » pour créer un objet, le runtime alloue de la mémoire à l'objet.
  - Tant que de l'espace d'adressage est disponible dans le tas managé, le runtime continue à allouer de l'espace pour les nouveaux objets.
  - Toutefois, la mémoire n'est pas infinie. Le GC doit finir pour effectuer un garbage collection afin de libérer de la mémoire.

Garbage Collector

- Exécution du GC :
  - Le moteur d'optimisation du garbage collector détermine le meilleur moment pour effectuer un garbage collection en fonction des allocation en cours.
  - Lorsque le garbage collector effectue un garbage collection, il recherche les objets figurant dans la mémoire qui ne sont plus utilisés par l'application, et effectue les opérations nécessaires pour récupérer leurs mémoires.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 1.0 : Février 2002
  - Visual Studio 2002.
  - CLR V1.
  - C# 1.
  - Librairie de classe couvrant de nombreux besoins.
  - Prise en compte de la sécurité.
  - ASP « style » Windows Form.
  - Séparation du design et du code « Code Behind ».

Microsoft .Net : un peu d'histoire

- Microsoft Framework 1.1 : Avril 2003
  - Visual Studio 2003
  - CLR V1.1
  - Intégration des contrôles ASP .Net Mobile.
  - Intégration du support ODBC et Oracle.
  - Compact Framework
  - Support IPV6.
  - Évolution des librairies de classes.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 2.0 : Janvier 2006
  - Visual Studio 2005.
  - CLR V2
  - C# 2
  - Support complet 64 bits.
  - Ajout des génériques.
  - Amélioration importante de ASP.Net : Nouveaux contrôles, Thèmes, Skins, WebParts …
  - .Net Micro Framework dédié à la robotique, télécommandes, senseurs …
  - Évolution des librairies de classes : Multithreading, Memory Allocation, Assembly loading …

Microsoft .Net : un peu d'histoire

- Microsoft Framework 3.0 : Janvier 2007
  - Aucun environnement de développement associé !
  - Simple complément du Framework 2.0.
  - Windows Communication Foundation.
  - Intégré à Windows Vista.
  - 4 APIs : Interface graphique, Communication, Workflow, Sécurité.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 3.5 : Février 2008
  - Visual Studio 2008.
  - C# 3
  - Types et méthodes anonymes.
  - Requêtes intégrées au langages (LINQ).
  - Entity Framework.
  - Amélioration du Framework 3.0 : WCF.
  - Nouveaux ajouts aux langages C# et VB.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 4.0 : Avril 2010
  - Visual Studio 2010.
  - CLR V4.
  - C# 4.
  - Programmation dynamique.
  - Parallel computing (TPL, PLINQ, CDS, Threading).
  - ASP.Net 4.
  - AJAX 4.
  - Managed Extensibility Framework (MEF)
  - J#.

Microsoft .Net : un peu d’histoire

- Microsoft Framework 4.5 : Septembre 2012
  - Visual Studio 2012.
  - C# 5.
  - Supporte la génération des applications de Windows Store.
  - Bibliothèque de classes portables.
  - Collection des informations de diagnostic.
  - Améliorations générales des performances.
  - Les méthode Asynchrones.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 4.5.1 : Octobre 2013
  - Visual Studio 2013.
  - Améliorations générales des performances.
  - Compilation juste-à-temps (JIT) en arrière-plan.
  - Prise en charge de la console pour l'encodage Unicode.
  - Opérations asynchrones sur les fichiers.
  - ASP.Net 4.5.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 4.6 : Juillet 2015
  - Visual Studio 2015.
  - C# 6.
  - C# 7 (dans la version 4.6.2).
  - Chiffrement : prise en charge des certificats X509.
  - Amélioration des comportements de connexion dans ADO.Net.
  - Amélioration de la performance des applications WPF.

Microsoft .Net : un peu d'histoire

- Microsoft Framework 4.7 : Mai 2017
  - Visual Studio 2017.
  - Nouvelles fonctionnalités et améliorations de :
    - ASP.Net
    - WCF
    - Windows Forms
    - WPF