

L'infrastructure .NET se propose de rendre l'information disponible en tout lieu, à tout moment, et par le biais de n'importe quel terminal de consultation. Précurseur en ce domaine, Orsys a conçu des cursus de formation .NET permettant d'atteindre tous les niveaux de spécialisation et couvrant tous les domaines d'application : le Web et le Cloud, les applications Windows, les terminaux mobiles... Ils s'appuient sur les langages .NET phares, C# et VB.NET, l'outil de référence Visual Studio, et sur les méthodes les plus efficaces de conception, de tests et d'optimisation.

### Séminaires

[Technologies .NET, synthèse....](#) ( p3 )

### Stages Pratiques

[Applications .NET, solutions de développement....](#) ( p5 )

[Gestion de projets .NET....](#) ( p7 )

[Programmation Orientée Objet en .NET....](#) ( p9 )

[Visual Basic .NET, développer avec Visual Studio 2010....](#) ( p11 )

[C#, développer en .NET avec VS 2010....](#) ( p13 )

[F# et programmation fonctionnelle, prise en main....](#) ( p15 )

[ASP.NET 4.0/3.5, niveau 1 développement Web....](#) ( p16 )

[ASP.NET 4.0/3.5, niveau 2, développement Web....](#) ( p18 )

[Programmation Web MVC en .NET....](#) ( p20 )

[Silverlight 4/5, mise en œuvre....](#) ( p22 )

[Windows Azure, développer une appli. Cloud en .NET....](#) ( p24 )

[SharePoint 2010, développement d'appli. avec VS 2010....](#) ( p25 )

[Sharepoint 2007, développer en .NET....](#) ( p28 )

[Applications Windows avec Visual Studio 2010 et .NET 3.5....](#) ( p30 )

[Applications Windows, VS 2010 et .NET 3.5, perfect....](#) ( p32 )

[.NET 3.5, création d'applications WPF....](#) ( p33 )

[Windows Phone 7, développement d'applications....](#) ( p35 )

[MonoTouch, créer des appli. iPhone/iPad en C# et .NET....](#) ( p37 )

[C# 5.0, mise en œuvre des nouveautés avec VS 2011....](#) ( p39 )

[Visual Studio 2010/.NET 4.0, les nouveautés....](#) ( p41 )

[C# 3.0, développer en Framework 3.5....](#) ( p42 )

[.NET, accès aux données....](#) ( p44 )

[ADO.NET Entity Framework, maîtrise et optimisation....](#) ( p45 )

[.NET 3.5, communications entre applications....](#) ( p47 )

[Workflow Foundation, développement .NET d'applications....](#) ( p48 )

[Sécurité des applications en .NET....](#) ( p50 )

[Développement système en .NET....](#) ( p51 )

[Framework .NET 4.0, architecture des applications....](#) ( p53 )

[.NET, développement avec Design Patterns/frameworks...](#) ( p55 )

[TFS 2010, optimiser ses développements .NET....](#) ( p56 )

[Visual Studio/TSF 2010, mise en œuvre des tests...](#) ( p58 )

[Test Driven Development en .NET....](#) ( p60 )

[Optimisation d'applications en .NET....](#) ( p62 )

[C#, programmation multithread....](#) ( p64 )

[Programmer vos applications en Multicore....](#) ( p65 )

## Participants

Ce cours s'adresse aux responsables de projets et développeurs professionnels désirant comprendre le contenu de l'environnement de développement .NET et comment construire des applications basées sur .NET.

## Pré-requis

Connaissances de base en développement d'applications Internet. Expérience souhaitable en programmation objet.

Prix 2012 : 1775€ HT

Eligible DIF

## Dates des sessions

### Paris

4 juin 2012, 22 oct. 2012  
17 déc. 2012

# Technologies .NET, synthèse

## OBJECTIFS

Ce séminaire vous permettra d'appréhender les apports du framework .NET, notamment dans ses versions 3.5 et 4.0, ainsi que ses impacts sur la conception et le développement des applications. Il vous présentera les composants de l'architecture .NET et vous montrera comment développer des applications en utilisant ces composants. Tous les aspects de l'intégration de XML dans le .NET ainsi que la gestion de la sécurité de la plate-forme d'exécution seront analysés.

### [1\) Introduction à l'environnement .NET](#)

### [2\) Le framework .NET et les langages du .NET](#)

### [3\) Divers types d'applications et classes du Framework](#)

### [4\) Accès aux données dans le .NET : XML, ADO.NET, DataBinding](#)

### [5\) Web Services](#)

### [6\) Architecture des applications et sécurité](#)

### [7\) Evolutions et direction de .NET](#)

## 1) Introduction à l'environnement .NET

- Présentation du framework .NET : composant du framework. Les langages, le CLR/CLI. Classes du framework. Connexion avec les serveurs. Le framework version 3.5 et 4.0.
- Principes de la compilation Just In Time. Langage intermédiaire (CIL).
- Offre de développement Microsoft : Visual Studio.NET, Suite Visual Studio Team System.
- Intégration, de .NET dans l'offre produit de Microsoft : intégration dans SQL Server, Biztalk, Windows 2008/2003, Seven, etc.
- Présentation des principaux composants du Framework : ASP.NET, ADO.NET, les Web Services.
- Positionnement de .NET par rapport à l'Open Soucre : initiative Mono et dotGNU, approche de Novell.

### Travaux pratiques

Exemple de programmes développés en C# et VB.NET. Les mécanismes de compilation JIT et du CIL.

## 2) Le framework .NET et les langages du .NET

- Présentation des principaux langages et de leurs caractéristiques : C#, VB.NET. Indépendance par rapport au langage.
- Approche de développement du .NET : approche objet, homogénéité de l'utilisation des classes du Framework.
- Notions importantes dans les langages : notion de délégué, notion de typage global (Common Language Type), etc.
- Le Framework redistribuable : rôle du CLR dans l'exécution interlangage, environnement d'exécution en code natif (CLR/CLI). Intégration du CLR/CLI dans le système.
- Classes du Framework : éléments importants du Framework, homogénéité des classes, utilisation indépendante du langage.

### Travaux pratiques

Exemple de programme orienté objet en VB.NET, de programme multilingage en VB.NET, C# montrant les possibilités d'extension liées à l'approche objet et l'indépendance par rapport au langage.

## 3) Divers types d'applications et classes du Framework

- Application client riche avec les Windows.Forms. Composants de base et composants avancés. Intégration de la gestion des événements. Notion de délégué. Développement de composants personnalisés. Intégration dans l'environnement de développement Visual Studio.NET. Mise en place des Windows Forms dans le contexte client riche et client embarqué.
- Application client léger avec ASP.NET. Compréhension du cycle de vie de l'application. Compréhension des principaux composants. Contrôle HTML, contrôle Serveur. Développement de contrôles personnalisés et intégration dans Visual Studio.NET. Prise en compte de la compatibilité des navigateurs.
- Outils de mise au point dans le cadre du .NET : outils de trace et de mise au point dans Visual Studio.NET.

### Travaux pratiques

Exemple de programme mettant en oeuvre les Windows Forms, composants de base et avancés. Application ASP.NET. Application embarquée.

## 4) Accès aux données dans le .NET : XML, ADO.NET, DataBinding

- Vue générale sur XML et intégration dans le .NET.
- Sérialisation des données en XML : lecture des données XML, écriture des données XML. Définition du schéma XSD.
- Principe d'ADO.NET : classes d'accès aux données, utilisation des connexions bases de données.
- Intégration d'une connexion base de données, génération de requête. Création de listes liées à des bases de données.
- Outils de mise au point et de trace en SQL, intégration multilingage au travers de SQL.
- Modèle du DataBinding et utilisation dans .NET.

### **Travaux pratiques**

*Exemple d'application utilisant les bases de données et s'appuyant sur ADO.NET et mettant en oeuvre le DataBinding.*

## **5) Web Services**

- Principe de fonctionnement des Web Services. Architecture d'exécution (WSDL, SOAP, XML, UUDI, etc.).
- Développement des Web Services avec le .NET.
- Utilisation de Web Services développés dans le cadre du .NET. Utilisation de Web Services hors .NET : notion de classe proxy, invocation du Web Service et sérialisation XML.
- Objets à distance : .NET Remoting.
- Environnement d'exécution des Web Services sur le serveur (notion de serveurs d'applications et mise en oeuvre dans le .NET).

### **Travaux pratiques**

*Exemple de programme en VB.NET basé sur ASP.NET et mettant en oeuvre un service Web. Exemples de Web Services faisant des accès bases de données.*

## **6) Architecture des applications et sécurité**

- Gestion de la sécurité : authentification, autorisation, cryptage. Architecture de l'authentification en .NET.
- Architecture globale des applications en .NET, organisation autour des Web Services.
- Support des périphériques intelligents en .NET.

### **Travaux pratiques**

*Exemple de programme faisant un contrôle de l'identité des utilisateurs en .NET et gestion de la sécurité d'exécution.*

## **7) Evolutions et direction de .NET**

- Evolution vers le Framework 3.5 et 4.0 et intégration à Seven.
- Compréhension de Windows Foundation et mécanismes d'interaction avec .NET.

Stage pratique de 3 jour(s)  
Réf : ART

## Participants

Ce cours s'adresse aux responsables de projets et aux développeurs professionnels désirant comprendre le contenu de l'environnement de développement Visual Studio 2010/2008 afin de construire des applications basées sur le Framework .NET 4.0/3.5.

## Pré-requis

Connaissances de base en développement d'applications et des protocoles Internet. Expérience souhaitable en programmation objet.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

Paris

8 oct. 2012, 10 déc. 2012

# Applications .NET, solutions de développement

## OBJECTIFS

Cette formation vous permettra d'apprécier les apports du .NET 3.5 dans la conception des principaux types d'applications (clients lourds avec WinForms et WPF, clients légers en ASP .NET, clients Internet riches avec Silverlight, services Web WCF et services Windows). Elle vous présentera l'architecture du Framework .NET, avec ses composants d'accès aux données ADO .NET, et ses mécanismes de sécurité intégrés.

### 1) Introduction à l'environnement .NET

### 2) Gestion des accès aux données : ADO.NET et

### 3) XML

### 4) Développement d'applications fenêtrées

### 5) WinForms et WPF

### 6) Développement d'applications Web ASP .NET

### 7) et Silverlight

### 8) Méthodes de déploiement d'applications

### 9) avec ClickOnce

### 10) Services Web et services WCF

### 11) Sécurité des applications .NET

## 1) Introduction à l'environnement .NET

- Présentation de l'architecture du Framework .NET : CLR, hiérarchie des classes, espaces de noms, intégration des langages (CLS/CLI).
- Les principaux composants du framework : ASP.NET, ADO.NET, les Web Services.
- Le CLR, notions d'assembly et de domaine d'application.
- Assemblies signés, privés et publics (rôle et utilisation du GAC).
- Les outils du développeur : Structures et classes, génériques, délégués et événements, réflexion
- Les nouveautés de C# 3.0/4.0 et VB.9/10 dans le Framework 3.5 : inférence de type, types anonymes, méthodes d'extension, LINQ.

### Travaux pratiques

Exemple d'un programme de type Console en .NET. Illustration du MSIL. Illustration de l'approche orientée Objet dans le Framework.

## 2) Gestion des accès aux données : ADO.NET et

## 3) XML

- Principe d'ADO.NET : classes d'accès aux données, utilisation des connexions aux bases de données.
- Intégration d'une connexion base de données, génération de requête. Création de listes liées à des bases de données.
- Intégration et utilisation de XML dans .NET : sérialisation des données, définition des schémas.
- Principe du modèle en trois couches : couche présentation, couche métier et couche d'accès aux données.
- Démonstration des limites de l'accès direct aux données par les applications, couche d'accès typée aux données : TableAdapters, LINQ To SQL et EntityFramework.

### Travaux pratiques

Exemple d'application utilisant les bases de données et s'appuyant sur ADO.NET. Exemple de programmes manipulant XML.

## 4) Développement d'applications fenêtrées

## 5) WinForms et WPF

- Les différents types d'applications Windows : WinForms, WPF et Service Windows.
- La globalisation et la localisation d'applications Windows Forms.
- Déploiement de l'application avec ClickOnce.
- ADO .Net dans les Windows Forms : Simple Data Binding avec la BindingSource, Complex DataBinding avec la DataGridView.
- Windows Presentation Foundation (WPF) : méthode de rendu vectoriel, positionnement des éléments, contrôles et formes de base, styles, animations, modèle de contrôles, développement et déploiement d'applications autonomes et hébergées.
- XAML : présentation de la syntaxe (éléments de propriété, propriétés de contenu, propriétés attachées, extensions de balisage), des principaux outils de conception d'interfaces et de leur intégration avec Visual Studio.
- Modèle de programmation : rôle des propriétés de dépendance, événements routés, commandes, mécanismes de liaison de données avec formatage et validations des informations saisies.

### Travaux pratiques

Réalisation d'une application Windows Forms et WPF avec déploiement par ClickOnce

## 6) Développement d'applications Web ASP .NET

## 7) et Silverlight

- L'architecture d'ASP.Net, le cycle de vie d'une Page ASP.Net, gestion des événements d'interface utilisateur. Divers types de contrôles. Compatibilité des navigateurs.
- Survie des données applicatives avec Session.
- Conception d'un site : MasterPages, contrôles utilisateurs, fichiers de navigation SiteMap, gestion des thèmes graphiques.
- Amélioration de l'interface avec ASP.Net AJAX.
- La localisation d'une application Web ASP.Net.
- L'accès aux données en ASP.Net : Two Way Data Binding, SqlDataSource, LinqDataSource.
- Les objets métiers comme source de données : ObjectDataSource

#### **Travaux pratiques**

*Exemple de programme Web dynamique en VB.NET utilisant ASP.NET. Saisie d'informations et déclenchement de traitement. Validation des informations saisies.*

## 8) Méthodes de déploiement d'applications

### 9) avec ClickOnce

- Principe du client riche (Smart-Device) et des applications Silverlight.
- Développement d'applications pour mobiles.
- Développement et déploiement d'une application Silverlight.

#### **Travaux pratiques**

*Exemple d'application utilisant le Compact framework sur des assistants numériques.*

## 10) Services Web et services WCF

- Principe de fonctionnement du Remoting et des Services Web (WSDL, SOAP, XML, UUDI, etc.).
- Développement des Services Web avec Visual Studio 2010/2008.
- Utilisation d'un Service Web par une application cliente .NET en dehors de .NET : notion de classe proxy, invocation du Web Service et sérialisation XML.
- L'architecture SOA avec WCF : Interface de service, type de service, liaisons, End Points, binding et behavior.

#### **Travaux pratiques**

*Exemple de programme en .NET basé sur ASP.NET et mettant en oeuvre un service Web. Exemples de Web Services faisant des accès bases de données. Exemple de service WCF utilisant plusieurs liaisons.*

## 11) Sécurité des applications .NET

- Infrastructure des mécanismes de sécurité intégrés à la plateforme .NET : signature, code managé, authentification, autorisation, cryptage.
- Sécurité basée sur les rôles
- Sécurité basée sur le code, programmation et administration

#### **Travaux pratiques**

*Exemple de programme faisant un contrôle de l'identité des utilisateurs en .NET et gestion de la sécurité d'exécution.*

Stage pratique de 3 jour(s)  
Réf : GNE

## Participants

Chefs de projets,  
responsables d'applications,  
consultants techniques.

## Pré-requis

Connaissances de base de  
l'environnement .NET et des  
concepts de la gestion de  
projets informatiques.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

### Paris

4 juin 2012, 8 oct. 2012  
10 déc. 2012

# Gestion de projets .NET

## OBJECTIFS

Les stagiaires verront les meilleures pratiques pour conduire un projet en environnement Microsoft/.NET et gérer efficacement son avancement. Ils apprendront à maîtriser l'outillage disponible pour organiser et piloter leur projets de développement, ainsi qu'à intégrer dans leur démarche les pratiques issues des méthodes dites agiles.

- 1) [Cycles de vie des projets .NET](#)
- 2) [Visual Studio 2010 ALM](#)
- 3) [Spécification et démarrage d'un projet](#)
- 4) [Planification](#)

- 5) [Organisation des développements](#)
- 6) [Suivi des incréments](#)
- 7) [Gestion de la qualité dans les projets .NET](#)

## 1) Cycles de vie des projets .NET

- Rappels sur .NET : gestion de la présentation, de la couche métier et des données.
- Composants métiers, réutilisation et urbanisation du SI.
- Du cycle projet en cascade à la démarche incrémentale.
- Comparaisons des méthodes de développement MSF Agile 5.0, SCRUM et MSF for CMMI.
- Les différents acteurs d'un projet .NET.

## 2) Visual Studio 2010 ALM

- Les concepts de l'ALM - Application Lifecycle Management.
- Une approche globale basée sur les personnes, les processus et les technologies.
- Le cycle de développement et de tests. Les outils de modélisation.
- Le travail collaboratif.

## 3) Spécification et démarrage d'un projet

- L'alignement stratégique d'un projet.
- Etude des processus métiers et des exigences.
- Spécification des cas d'utilisation du système.
- Réaliser un découpage des exigences en tâches techniques avec Microsoft Project.

### Travaux pratiques

*Etude de cas. Analyse des processus métiers. Définition des spécifications et des cas d'utilisation.*

## 4) Planification

- Définir les itérations et les zones d'organisation du projet.
- Planifier des incréments et établir le plan projet.
- Conduites agiles des développements.

## 5) Organisation des développements

- Organisation de l'équipe.
- Coordonner l'équipe avec un portail SharePoint.
- Gestion de la documentation.
- Environnement de développement : Visual Studio 2010.
- Construction de l'application.
- Développement piloté par les tests (approche TDD).
- Organiser et piloter des campagnes de test avec Microsoft Test Pro.
- Récolter, qualifier, distribuer et suivre l'état d'avancement des bogues.
- Gérer des cycles de livraison et de déploiement automatique.

### Travaux pratiques

*Pratique des différents outils. Organisation de l'équipe et des activités.*

## 6) Suivi des incréments

- Les comités d'architecture technique et applicative.
- Evaluation/validation d'un incrément. Reporting.
- Actualisation des risques, mise à jour du plan projet.
- Créer les tableaux de bord et rapports du projet avec Microsoft Reporting Services et Microsoft Excel.

### Travaux pratiques

*Exemples de tableaux de bord. Choix des indicateurs.*

## 7) Gestion de la qualité dans les projets .NET

- Qualité de l'architecture et du code. Métriques. Lectures croisées. Inspections.

### Travaux pratiques

*Définition des métriques Qualité.*

Stage pratique de 3 jour(s)  
Réf : UNE

## Participants

Développeurs habitués à la programmation procédurale, qui souhaitent tirer parti de la POO pour développer des applications professionnelles basées sur le Framework .NET.

## Pré-requis

Bonnes connaissances en programmation procédurale. Expérience requise en développement logiciel.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

### Paris

4 juin 2012, 24 sep. 2012  
26 nov. 2012

# Programmation Orientée Objet .NET en C# ou VB.NET

## OBJECTIFS

*Cette formation vous permettra au moyen d'exemples simples et progressifs, de vous familiariser avec les concepts de la programmation orientée objet (POO) nécessaires au développement .NET. Vous apprendrez les bases de l'utilisation de Visual Studio ainsi qu'à structurer une application en couches logiques.*

### 1) Intérêts de la POO

#### 2) La notion d'architecture applicative

#### 3) La notion d'objet

#### 4) L'implémentation d'une classe

### 5) Les concepts de la POO

#### 6) Manipulations courantes d'objets en .NET

#### 7) Outils et méthodes

## 1) Intérêts de la POO

- Rappel des principes de la programmation procédurale.
- Avantages et inconvénients de cette approche.
- Exemple d'une application orientée objet.

### Travaux pratiques

*Comparaison d'un même programme simple réalisé avec un découpage fonctionnel et orienté objet.*

## 2) La notion d'architecture applicative

- Du Mainframe au client-serveur.
- Du client-serveur aux architectures multiniveaux.
- Architecture globale des applications en .NET : organisation en couches et en niveaux.
- Principe du modèle en trois couches : présentation, métier et d'accès aux données.
- Principe des applications distribuées et des architectures SOA.

### Travaux pratiques

*Analyse d'une application simple (starter kit) architecturée en trois couches logiques.*

## 3) La notion d'objet

- Démystifier la notion d'objet et terminologie associée.
- Conception et manipulations d'objets (classes, instances).
- Présentation et utilisation du diagramme de classes UML.

## 4) L'implémentation d'une classe

- Mise en oeuvre du mécanisme d'encapsulation.
- Implémenter une méthode avec surcharges.
- Ajouter des constructeurs. Créer un membre statique.
- Définir la visibilité d'un membre et contrôler son accès.

### Travaux pratiques

*Manipuler les différents types de membre d'une classe.*

## 5) Les concepts de la POO

- Principe et intérêt du mécanisme d'héritage.
- Redéfinir un membre dérivé.
- Rôle des classes abstraites et des interfaces.
- Faire du polymorphisme avec une interface.

## 6) Manipulations courantes d'objets en .NET

- Principe de la liaison précoce ou tardive et de la réflexion.
- Rôle et comportement des types valeur et référence.
- Regroupement d'objets en collections.
- Principe du DataBinding.
- Principe de LINQ.

### Travaux pratiques

*Se familiariser avec la manipulation des objets en .NET, avec exemple de LINQ, de LINQ To Object et To SQL.*

## 7) Outils et méthodes

- Aperçu des techniques d'analyse (RUP ; Agile).
- Tirer parti d'UML (principaux diagrammes, pièges à éviter...).
- Principe et intérêt des design patterns.

**Travaux pratiques**

*Réalisation du pattern Singleton et analyse des patterns exploités dans le Framework .NET.*

Stage pratique de 5 jour(s)  
Réf : VBN

## Participants

Pour les développeurs désirant acquérir les techniques et les réflexes pour développer tous types d'applications .NET avec Visual Studio 2008/2010.

## Pré-requis

Bonnes connaissances en programmation. Connaissances de base des concepts objet. Expérience requise en développement logiciel.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

18 18 juin 2012, 23 juil. 2012  
27 août. 2012, 24 sep. 2012  
22 oct. 2012, 19 nov. 2012  
17 déc. 2012

### Aix

25 juin 2012, 17 sep. 2012  
3 déc. 2012

### Bordeaux

25 juin 2012, 8 oct. 2012  
3 déc. 2012

### Bruxelles

8 oct. 2012, 3 déc. 2012

### Genève

8 oct. 2012, 3 déc. 2012

### Lille

8 oct. 2012, 3 déc. 2012

### Luxembourg

25 juin 2012, 8 oct. 2012  
3 déc. 2012

### Lyon

25 juin 2012, 17 sep. 2012  
3 déc. 2012

### Nantes

18 juin 2012, 10 sep. 2012  
26 nov. 2012

### Rennes

10 sep. 2012, 26 nov. 2012

### Sophia-antipolis

25 juin 2012, 17 sep. 2012  
3 déc. 2012

### Strasbourg

25 juin 2012, 8 oct. 2012  
3 déc. 2012

### Toulouse

25 juin 2012, 8 oct. 2012  
3 déc. 2012

# Visual Basic.NET

## applications avec Visual Studio 2008/2010

### OBJECTIFS

Cette formation vous apprendra tout d'abord à connaître l'architecture des Frameworks 2.0, 3.0, 3.5 et 4.0, puis à exploiter les classes de base communes aux différents types d'applications .NET (Windows, Web, DLL, etc.), avec le langage VB.NET. Vous apprendrez ensuite à créer vos classes/objets spécifiques et à tirer parti des mécanismes (encapsulation, héritage et polymorphisme) inhérents à la programmation orientée objet (POO) en .NET. Vous verrez également le principe des accès aux données avec ADO.Net et des applications Web : ASP.NET.

#### [1\) Introduction à la plate-forme .NET](#)

#### [2\) Syntaxe de base : données, expressions et instructions](#)

#### [3\) Gestion des exceptions](#)

#### [4\) Programmation Orientée Objet](#)

#### [5\) Classes et objets en VB.NET](#)

#### [6\) Objets et classes de base du framework .NET](#)

#### [7\) Exemples d'applications développées en .NET](#)

### Travaux pratiques

Les aspects syntaxiques du langage VB .NET et les classes de base sont illustrés par des exemples dans le cadre d'une introduction aux Windows Forms. S'ensuivent la conception d'objets spécifiques, puis la création d'une DLL réutilisable, avec son déploiement. Démonstrations sur les accès aux données (ADO .NET) et les applications de type Windows et Web (ASP .NET).

## 1) Introduction à la plate-forme .NET

- Principe et architecture de la plateforme .NET.
- Architecture du Framework .NET : CLR, BCL, et CLS (support multilingage).
- Aperçu des différents types d'applications dans un environnement multiculturel.
- Structure d'une application .NET : notion d'espace de noms.
- Outils et environnement de développement.
- Langage de MSIL : principe du langage intermédiaire, principe de la compilation JIT (Just In Time Compiler).
- Notion d'assembly, de métadonnées et de déploiement. Assembly privé et assembly partagé : signature, rôle du GAC (Global Assembly Cache), mise en place dans le GAC.

### Travaux pratiques

Exemple de programme VB.NET. Exécution en mode géré. Utilisation de l'environnement de développement Visual Studio.Net pour l'écriture du premier programme.

## 2) Syntaxe de base : données, expressions et instructions

- Variables et expressions : déclaration, constantes, opérateurs.
- Types de données : Common Type System, rôle de la classe de base System.Object et transtypage.
- Types annulables.
- Types valeur et types référence.
- Manipulation et gestion des tableaux.
- Instructions de contrôle de flux : boucles et tests.
- Nouveautés VB8 : Continue, IsNot, Using, visibilité des propriétés, support des types non signés, mot-clef Global, TryCast. Opérateurs IsTrue et IsFalse.

### Travaux pratiques

Écriture de programmes en VB.NET mettant en oeuvre des algorithmes classiques.

## 3) Gestion des exceptions

- Principe de la gestion des exceptions.
- Utilisation de l'instruction Throw.
- Utilisation des exceptions pour le traitement centralisé des erreurs.
- Structure de gestion des exceptions interlangages : déclenchement d'exception dans un langage, interception et traitement dans un autre. Exemple entre VB.NET et C#.

### Travaux pratiques

Exemples de gestion des exceptions en VB.NET.

## 4) Programmation Orientée Objet

- Classes et objets : modélisation du monde réel par objet.
- Notions de champ, méthode et propriété.
- Héritage.
- Polymorphisme.
- Implémentation d'interfaces multiples.
- Représentation du modèle objet.

## 5) Classes et objets en VB.NET

- Définition des classes. Définition des objets.
- Définition du contenu de la classe : méthodes et propriétés.
- Visibilité des membres d'une classe : propriétés et méthodes. Utilisation des espaces de noms.
- Cycle de vie des objets : constructeur, destructeur. Gestion de la mémoire avec le garbage collector.
- Le mécanisme de surcharge (constructeurs, méthodes et opérateurs).
- Implémenter un événement.
- Dérivation et héritage des classes : principe de dérivation, contrôle d'accès lors de la dérivation.
- Principe des interfaces (définition et implémentation).
- Principe du polymorphisme.
- Manipulation des attributs : principe des méta-données. Attributs de classe, de méthode, de champ.
- Régions de code et classes partielles.
- Classes génériques.
- Génération de documentation.

### Travaux pratiques

*Ecriture de classes de base. Manipulation de méthodes et propriétés dans les classes. Dérivation de classe. Exemple d'implémentation d'une interface et de mise en oeuvre du polymorphisme par héritage ou avec une interface.*

## 6) Objets et classes de base du framework .NET

- Qu'est-ce qu'un framework ? Principe. Hiérarchie des classes.
- Traitement des dates et des durées. Traitement des chaînes avec StringBuilder et les expressions régulières.
- Classes incontournables : manipulation du système de fichier, Math, Random, etc.
- Gestion des entrées/sorties : hiérarchie des classes. FileStream, StreamReader/StreamWriter.
- Principes des entrées/sorties asynchrones.
- Les différents types de collections, de dictionnaires et de tables de hachage.
- Les collections génériques.
- Principe et utilisations de LINQ.
- Se familiariser avec LINQ To Object.

### Travaux pratiques

*Utilisation des classes de base. Exemples de manipulation du système de fichiers et d'opérations de lecture/écriture dans un fichier texte.*

## 7) Exemples d'applications développées en .NET

- Principe et conception d'une bibliothèque de classes réutilisable.
- Principe et exemple des applications Web ASP .NET.
- Introduction aux accès aux données avec ADO .NET.

### Travaux pratiques

*Réalisation d'une bibliothèque de classes signée et installation dans le GAC. Réalisation d'une page ASPX simple, avec affichage de données.*

Stage pratique de 5 jour(s)  
Réf : PSH

## Participants

Ce cours s'adresse aux développeurs désirant acquérir les techniques et les réflexes pour développer tous types d'applications .NET avec Visual Studio 2008/2010.

## Pré-requis

Bonnes connaissances en programmation.  
Connaissances de base des concepts objet. Expérience requise en développement logiciel avec un langage de type C/C++ ou Java.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

21 mai 2012, 18 juin 2012  
9 juil. 2012, 20 août. 2012  
10 sep. 2012, 15 oct. 2012  
12 nov. 2012, 10 déc. 2012

### Aix

21 mai 2012, 24 sep. 2012  
26 nov. 2012

### Bordeaux

22 oct. 2012, 10 déc. 2012

### Bruxelles

11 juin 2012, 22 oct. 2012  
10 déc. 2012

### Genève

22 oct. 2012, 10 déc. 2012

### Lille

4 juin 2012, 22 oct. 2012  
10 déc. 2012

### Luxembourg

22 oct. 2012, 10 déc. 2012

### Lyon

21 mai 2012, 24 sep. 2012  
26 nov. 2012

### Nantes

21 mai 2012, 17 sep. 2012  
19 nov. 2012

### Rennes

17 sep. 2012, 19 nov. 2012

### Sophia-antipolis

24 sep. 2012, 26 nov. 2012

### Strasbourg

22 oct. 2012, 10 déc. 2012

### Toulouse

4 juin 2012, 22 oct. 2012  
10 déc. 2012

# C#, développer en .NET avec Visual Studio 2008/2010

## OBJECTIFS

Cette formation vous apprendra à connaître l'architecture des Frameworks 2.0, 3.0, 3.5 et 4.0, puis à exploiter les classes basiques communes aux différents types d'applications .NET (Windows, Web, DLL, etc.) avec le langage C#. Vous apprendrez à créer des classes métier et à tirer parti des mécanismes (encapsulation, héritage et polymorphisme) inhérents à la programmation orientée objet (POO) en .NET. Vous verrez également le principe des accès aux données avec ADO.NET et les différentes mises en oeuvre de LINQ ainsi que des applications et services Web : ASP.NET.

### 1) La plateforme .NET

### 2) Syntaxe de base : données, expressions et instructions

### 3) Gestion des exceptions

### 4) Programmation Orientée Objet

### 5) Classes et objets en C#

### 6) Objet et classes de base du framework .NET

### 7) Les différents types d'applications développées en .NET

## Travaux pratiques

Au tout début du stage, les aspects syntaxiques du langage C# et les classes de base sont illustrés avec des exemples simples. La plus grande partie du stage concerne la conception d'objets spécifiques, la création et le partage de bibliothèques métier réutilisables. Le stage se termine par une présentation des accès aux données (ADO .NET) et des applications de type Windows et Web (ASP .NET).

## 1) La plateforme .NET

- Principe et architecture de la plateforme .NET.
- Architecture du Framework .NET : CLR, BCL, et CLS (support multilingage).
- Aperçu des différents types d'applications dans un environnement multicible.
- Structure d'une application .NET : notion d'espace de noms.
- Outils et environnement de développement.
- Langage de CIL : principe du langage intermédiaire, principe de la compilation JIT (Just In Time Compiler).
- Notion d'assembly, de métadonnées et de déploiement. Assembly privé et assembly partagé : signature, rôle du GAC (Global Assembly Cache), mise en place dans le GAC. Les nouveautés du GAC 4.0.

## Travaux pratiques

Exemple de programme C# minimum. Exécution en mode géré. Utilisation de l'environnement de développement Visual Studio.Net pour l'écriture du premier programme.

## 2) Syntaxe de base : données, expressions et instructions

- Variables et expressions : déclaration, constantes, opérateurs, types anonymes (C# 3.0) et dynamiques (C# 4.0).
- Types de données : Common Type System, rôle de la classe de base System.Object et transtypage.
- Types valeur et types référence.
- Instructions de contrôle de flux : boucles et tests.
- Syntaxe de base : manipulation des tableaux, instructions de contrôle de flux, ...

## Travaux pratiques

Écriture de programmes en C# mettant en oeuvre des algorithmes classiques.

## 3) Gestion des exceptions

- Principe de la gestion des exceptions.
- Déclenchement d'erreurs personnalisées avec throw.
- Utilisation des exceptions pour le traitement centralisé des erreurs.

## Travaux pratiques

Gestion des erreurs de saisie en utilisant les exceptions.

## 4) Programmation Orientée Objet

- Classes et objets.
- Modélisation du monde réel par objet.
- Notions de champ, méthode et propriété.
- Héritage.
- Polymorphisme.
- Interfaces.

## 5) Classes et objets en C#

- Définition des classes. Définition des objets.
- Définition du contenu de la classe : méthodes et propriétés.

- Visibilité des membres d'une classe : propriétés et méthodes. Utilisation des espaces de noms.
- Cycle de vie des objets : constructeur, destructeur. Gestion de la mémoire avec le garbage collector.
- Le mécanisme de surcharge (constructeurs, méthodes et opérateurs).
- Dérivation et héritage des classes : principe de dérivation, contrôle d'accès lors de la dérivation.
- Principe du polymorphisme.
- Principe des interfaces (définition et implémentation).
- Manipulation des attributs : principe des méta-données. Attributs de classe, de méthode, de champ.
- Régions de code et classes partielles.
- Classes génériques.
- Délégués, covariance, contravariance et événements. Les variances appliquées aux interfaces (C# 4.0).
- Classes métier semblables à celles du framework .NET : réalisation de propriétés, d'indexeurs et d'énumérateurs.
- Génération de documentation.
- Les nouveautés de C# 3.0 et 4.0 : expression d'initialisation, types anonymes, types dynamiques, paramètres optionnels, propriétés automatiques, méthodes d'extension, méthodes partielles et lambda expressions.

#### **Travaux pratiques**

*Ecriture de classes de base. Manipulation de méthodes et propriétés dans les classes. Dérivation de classe. Exemple d'implémentation d'une interface et de mise en oeuvre du polymorphisme par héritage ou avec une interface.*

## 6) Objet et classes de base du framework .NET

- Qu'est-ce qu'un framework ? Principe. Hiérarchie des classes.
- Traitement des dates et des durées. Traitement des chaînes avec StringBuilder et les expressions régulières.
- Classes incontournables : manipulation du système de fichier, Math, Random, etc.
- Les différents types de collections, de dictionnaires et de tables de hachage.
- Les collections génériques et les bases de Linq avec LINQ To Object sur les collections standard.

#### **Travaux pratiques**

*Utilisation des expressions régulières et d'une table de hachage. Rendre une collection réalisée précédemment compatible avec LINQ et utiliser LINQ To Objects pour l'interroger.*

## 7) Les différents types d'applications développées en .NET

- Principe et conception d'une bibliothèque de classes réutilisables.
- Principe et exemple des applications Web ASP .NET.
- Principe et exemple des services Web ASP .NET.
- Introduction aux accès aux données avec ADO .NET et LINQ.
- Qu'est-ce que LINQ : exemple avec LINQ To Entities et LINQ to SQL.

#### **Travaux pratiques**

*Réalisation d'un formulaire Windows interrogeant une base de données. Démonstration d'une page ASPX simple, avec affichage de données. Démonstration d'un service Web simple.*

Stage pratique de 2 jour(s)  
Réf : FON

## Participants

Développeurs et architectes logiciels.

## Pré-requis

Bonnes connaissances du Framework .Net (ASP.Net, WPF, TPL, RX) et de Visual Studio.

Prix 2012 : 1285€ HT

Eligible DIF

## Dates des sessions

Paris

13 sep. 2012, 6 déc. 2012

# F# et programmation fonctionnelle, prise en main

## OBJECTIFS

La programmation fonctionnelle connaît un renouveau grâce à ses caractéristiques et points forts qui répondent à plusieurs problématiques actuelles de conception logicielle. Ce stage vous en présentera les paradigmes et vous permettra de les implémenter avec le nouveau langage F# de la plateforme .NET. Il vous montrera comment prendre en main ce langage et en tirer le meilleur parti pour créer des systèmes complexes.

[1\) Introduction à la programmation fonctionnelle avec F#](#)

[2\) Prise en main du langage F#](#)

[3\) Panorama des techniques de programmation avancées](#)

[4\) Construire des applications avec F#](#)

## Méthodes pédagogiques

Cours théorique accompagné d'exemples et de démonstrations. Ateliers de mise en oeuvre s'appuyant sur des cas réels et des problématiques rencontrées en entreprise.

## Travaux pratiques

Développement d'applications en F# sur cas concrets.

## 1) Introduction à la programmation fonctionnelle avec F#

- Les principes de la programmation fonctionnelle. Les paradigmes sous-jacents.
- La programmation fonctionnelle en .NET. L'environnement. Les possibilités offertes par la plateforme .NET.
- Vue générale du langage F# et de son écosystème.
- Pourquoi utiliser le langage F# ? Pour quoi faire ? Un petit historique.
- Les points forts du langage F#. Avantages et inconvénients.
- Impacts sur la façon de concevoir.

### Travaux pratiques

Démonstrations et tests d'applications.

## 2) Prise en main du langage F#

- Prise en main de l'environnement et des outils.
- Les bases du langage. Comment écrire un programme simple.
- Les types.
- Travailler avec les collections.
- Active Patterns, Quotations...
- Programmation impérative. Les principes de base avec F#.
- Interopérabilité avec .NET. L'intégration à l'écosystème .NET.
- Mode Interactif et Scripting.

### Exercice

Prise en main de l'environnement. Construction d'un programme de base (fonctions, collections).

## 3) Panorama des techniques de programmation avancées

- Programmation asynchrone avec les workflows asynchrones.
- Créer des interfaces utilisateurs : pour le Web, WPF et les applications Windows, les graphiques et l'interactivité.
- Algorithmique et Intelligence collective.
- Programmation scientifique et unités de mesure.
- Evénements F# et Rx (.NET Reactive Extensions).
- Programmation parallèle avec la TPL (.Net Task Parallel Library).
- Conception multi-agent avec le modèle Acteur.
- Language Oriented Programming et DSL (Domain Specific Languages).

### Exercice

Algorithmes et unités de mesure. Démonstration d'un automate cellulaire.

## 4) Construire des applications avec F#

- Les domaines d'application du langage F#.
- Gérer des projets en F# et les structurer.
- Construire une application e-Commerce ASP.NET MVC 3.
- Construire un moteur de recommandations pour un site e-Commerce.
- Construire un client Twitter WPF.
- Construire une application d'analyse des données d'un jeu de hasard.

### Travaux pratiques

Développement d'applications basées sur des cas concrets.

Stage pratique de 5 jour(s)  
Réf : AST

## Participants

Informaticiens souhaitant développer des applications Internet/Intranet en utilisant les technologies .NET (ASP.NET et ADO.NET) dans le cadre de l'environnement Visual Studio 2010/2008.

## Pré-requis

Bonnes connaissances de la programmation C# ou VB.NET. Connaissances de base des technologies Web côté client (HTML, JavaScript...).

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

11 18 juin 2012, 16 juil. 2012  
20 août. 2012, 17 sep. 2012  
15 oct. 2012, 12 nov. 2012  
10 déc. 2012

### Aix

11 juin 2012, 16 juil. 2012  
3 sep. 2012, 19 nov. 2012

### Bordeaux

11 juin 2012, 15 oct. 2012  
3 déc. 2012

### Bruxelles

15 oct. 2012, 3 déc. 2012

### Genève

15 oct. 2012, 3 déc. 2012

### Lille

15 oct. 2012, 3 déc. 2012

### Luxembourg

15 oct. 2012, 3 déc. 2012

### Lyon

16 juil. 2012, 3 sep. 2012  
19 nov. 2012

### Nantes

23 juil. 2012, 24 sep. 2012  
26 nov. 2012

### Rennes

23 juil. 2012, 24 sep. 2012  
26 nov. 2012

### Sophia-antipolis

16 juil. 2012, 3 sep. 2012  
19 nov. 2012

### Strasbourg

15 oct. 2012, 3 déc. 2012

### Toulouse

15 oct. 2012, 3 déc. 2012

# ASP.NET 4.0/3.5, niveau 1 développement Web avec Visual Studio 2010/2008

## OBJECTIFS

Ce cours vous apprendra à utiliser les techniques mises en oeuvre dans l'environnement .NET 4.0/3.5 (ou supérieur)/Visual Studio 2010-2008 pour la création d'applications Internet/Intranet. Il détaillera la technologie ASP.NET (Active Server Page.NET) pour la création de pages dynamiques, avec une présentation des possibilités offertes par Ajax. Vous verrez également comment lier vos applications avec des bases de données (ADO.NET) et intégrer des services Web utilisant les technologies XML dans une application Web ASP.NET.

- 1) [Rappels des concepts liés à l'Internet/Intranet et technologies .NET](#)
- 2) [Fonctionnement des pages ASP. NET](#)
- 3) [Contrôles serveur HTML](#)
- 4) [Contrôles serveur Web](#)
- 5) [Conception de la structure d'un site Web](#)
- 6) [Contrôle de sources de données](#)
- 7) [Gestion de la sécurité](#)
- 8) [Configuration et déploiement](#)
- 9) [Utilisation des services Web](#)

## Travaux pratiques

Les stagiaires apprendront à maîtriser l'infrastructure ASP.NET. Ils utiliseront l'environnement .NET/Visual Studio 2010/2008 pour mettre en oeuvre les techniques nécessaires aux applications Web professionnelles. Les exercices sont effectués en C# ou VB.Net suivant le choix des participants.

## 1) Rappels des concepts liés à l'Internet/Intranet et technologies .NET

- Serveur Web, sites statiques et dynamiques.
- Architecture d'une application Internet/Intranet.
- Technologies .NET. Application Web en .NET.
- Outils intégrés à Visual Studio.

### Travaux pratiques

Utilisation de Visual Studio pour la création d'un site, création d'un formulaire d'entrée de données.

## 2) Fonctionnement des pages ASP. NET

- Principe et problématique des pages dynamiques.
- Rappel des méthodes HTTP GET et POST.
- Principe des pages ASP .NET (ASPX).
- Utilisation des contrôles serveur WebForms. Intégration dans les pages ASPX avec Visual Studio.
- Gestion automatique de la persistance de l'état des contrôles serveur avec le ViewState.
- Gestion de la persistance des données applicatives : Session, Cache, Application, Cookies.
- Membres caractéristiques des classes page, HttpRequest, HttpResponse et HttpServerUtility.
- Utilisation et limitations du QueryString.
- Mise en oeuvre du CrossPagePostBack.
- Gestion des événements dans les pages ASP.NET : Load, UnLoad, PreRender...
- Gestion des événements de l'application : fichier global.asax, événements importants, utilisation.
- Structuration d'une application Web avec Visual Studio (dossiers spéciaux, ressources, etc.).

### Travaux pratiques

Écriture des premières pages ASP.NET basées sur des contrôleurs serveur standard (listes, boutons, etc.) et gérant des événements. Mise en oeuvre des mécanismes nécessaires aux applications professionnelles (session, cookie, etc.).

## 3) Contrôles serveur HTML

- Utilité des contrôles serveur HTML pour la création de pages actives côté client et serveur.
- Principaux contrôles. Utilisation.
- Intégration des contrôles serveur HTML avec les scripts clients JavaScript.

### Travaux pratiques

Utilisation des contrôles serveur HTML afin de rendre dynamique une page, du côté client et du côté serveur.

## 4) Contrôles serveur Web

- Principe des contrôles serveur.
- Caractéristiques des contrôles de base : TextBox, Label, Literal, Listes, Boutons, Table, TableCell, etc.
- Contrôles conteneur : Panel, MultiView, Wizard...
- Contrôles riches : Calendar, AdRotator, FileUpload...
- Contrôles de validation et groupe de validation.
- Réalisation d'un contrôle utilisateur.
- Principe et mise en oeuvre de l'Ajax.

### **Travaux pratiques**

*Mise en oeuvre des contrôles de validation côté client et côté serveur avec les contrôles Wizard. Exemple du contrôle FileUpload... Mise en oeuvre de l'Ajax avec UpdatePanel.*

## **5) Conception de la structure d'un site Web**

- Contrôles utilisateur pour la structuration des pages.
- Mise en forme d'un site à l'aide des MasterPages.
- Rôles des dossiers spéciaux (App\_Code, App\_Data, App\_Themes, etc.).
- Utilisation des feuilles de style CSS et des thèmes.
- Mise en place d'un système de navigation avec un fichier SiteMap et des contrôles SiteMapPath, TreeView et Menu.

### **Travaux pratiques**

*Exemple de réalisation d'une application de e-commerce...*

## **6) Contrôle de sources de données**

- Introduction à ADO.NET. Modèle objet ADO.NET.
- Mécanismes d'accès à une base de données : connexion, instruction SQL, procédure stockée, lecture de données. Utilisation d'un DataSet.
- Mécanisme de DataBinding.
- Utilisation des contrôles liés à une source de données : XmlDataSource, GridView (tri et pagination).
- Afficher une vue Maître/Détail entre un contrôle GridView et les contrôles DetailsView, FormView.
- Principes des modèles de lignes (Templates).

### **Travaux pratiques**

*Utilisation des contrôles SqlDataSource, GridView et DetailsView pour présenter les données d'entreprise (par exemple les articles d'un catalogue d'un site de e-commerce).*

## **7) Gestion de la sécurité**

- Les types d'attaques (SQL injection, spoofing, répudiation...) et les moyens de s'en prémunir.
- Contrôle applicatif des accès et des droits.
- Gestion de la sécurité basée sur Windows.

### **Travaux pratiques**

*Mise en oeuvre d'une authentification et d'un accès restreint, ainsi que d'une restriction d'accès pour un dossier Web.*

## **8) Configuration et déploiement**

- Gestion des exceptions.
- Rôle et format du fichier Web.config.
- Le Web Administration Tool intégré à Visual Studio.
- Sauvegarde des paramètres de l'application.
- Configuration de la gestion des sessions.
- Rôle du serveur Web : présentation de IIS.
- Le mode Release. Publication de l'application.

## **9) Utilisation des services Web**

- Principe. Architecture, communication XML/SOAP. Langage WSDL (Web Service Description Language).
- Appel d'un service Web à partir d'une requête HTTP et d'un proxy.

### **Travaux pratiques**

*Ecriture d'un Web Service simple (recherche dans une base de données) et interrogation via une ASP.NET et un proxy.*

Stage pratique de 5 jour(s)  
Réf : NEA

## Participants

Développeurs ASP.NET.

## Pré-requis

Bonnes connaissances de la programmation Web en .NET. Ou connaissances équivalentes à celles apportées par le stage "ASP.NET 4.0/3.5, niveau 1" (réf. AST). Expérience souhaitable.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

11 juin 2012, 8 oct. 2012  
17 déc. 2012

# ASP.NET 4.0/3.5, niveau 2 développement Web

## OBJECTIFS

Ce cours permettra aux participants d'appréhender les aspects avancés du développement Web avec ASP.NET 4.0 et 3.5, comme, entre autres, l'utilisation du cache de IIS et d'ASP.NET, la sécurité et les profils utilisateurs, l'internationalisation des applications, le tout dans un contexte d'architectures multiniveaux.

### [1\) Architecture des applications ASP.NET](#)

### [2\) Utilisation avancée de ADO.NET](#)

### [3\) Contrôles personnalisés](#)

### [4\) Optimisation des applications et utilisation du cache](#)

### [5\) Gestion des erreurs et journalisation](#)

### [6\) Sécurité et authentification](#)

### [7\) Représentation graphique des données](#)

### [8\) La réflexion](#)

### [9\) Introduction à ASP.NET Ajax](#)

### [10\) LINQ](#)

### [11\) Sites Dynamic Data \(ASP.Net 3.5 et 4.0\)](#)

## Travaux pratiques

Bonnes pratiques et techniques d'optimisation pour la création d'un site Web professionnel. Les exercices sont effectués en C# ou VB.Net suivant le choix des participants.

## 1) Architecture des applications ASP.NET

- Architecture des applications multitièrs.
- Composants d'une application ASP.NET.

## 2) Utilisation avancée de ADO.NET

- Dataset typé : importance et rôle.
- Utilisation des templates et des objets métiers.
- Personnalisation avec les événements des DataSources et les contrôles à base de templates.

### Travaux pratiques

Développement de la couche d'accès aux données pour optimiser la présentation de celle-ci.

## 3) Contrôles personnalisés

- Génération dynamique de contrôles dans une page.
- Les contrôles utilisateur de type ASCX.

### Travaux pratiques

Réalisation de contrôles utilisateur des trois types et interaction avec JavaScript.

## 4) Optimisation des applications et utilisation du cache

- Conseil dans le développement des applications.
- Le cache d'ASP.NET et les dépendances de cache.
- Interaction avec les bases de données et SQL Server.

## 5) Gestion des erreurs et journalisation

- Traitements centralisés des exceptions.
- Utilisation des journaux d'événements.

### Travaux pratiques

Mise en place d'une gestion centralisée des erreurs. Ecriture des événements.

## 6) Sécurité et authentification

- Principes : authentification, autorisation et personnalification.
- L'identité système avec WindowsAuthenticationModule.
- Usurpation de l'identité d'un client.

### Travaux pratiques

Réalisation d'une application Web de type forum gérant des rôles.

## 7) Représentation graphique des données

- Les images : manipulations, génération à la volée.
- Le contrôle ASP.Net 4.0 Chart pour la génération de graphes.

### Travaux pratiques

Protection d'un site contre les robots à l'aide d'images lisibles seulement par des utilisateurs "intelligents".

## 8) La réflexion

- La réflexion avec .NET. Les interfaces dynamiques.

### **Travaux pratiques**

*Saisie d'une valeur de type complexe inconnu à l'aide de l'API de réflexion de .NET.*

## 9) Introduction à ASP.NET Ajax

- Les extensions de Microsoft Ajax Library à JavaScript. Les templates MSAjax avec la classe JavaScript DataView.
- L'interaction entre les scripts client et les PageMethods ou Services Web ASP.NET.

### **Travaux pratiques**

*Programmation Ajax.*

## 10) LINQ

- LINQ To SQL (.Net 3.5 et 4.0). Attributs de mapping.
- LINQ To Entities (.Net 3.5 SP1 et 4.0).

## 11) Sites Dynamic Data (ASP.Net 3.5 et 4.0)

- Grands principes, gestion des routes, métadonnées impliquées, création d'un site Dynamic Data.

Stage pratique de 4 jour(s)  
Réf : MVC

## Participants

Développeurs .NET confirmés, désirant acquérir les techniques et les réflexes pour développer tous types d'applications Web avec ASP.NET MVC.

## Pré-requis

Bonnes connaissances de C#, HTML et JavaScript. Expérience requise en développement logiciel.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

Paris

4 sep. 2012, 20 nov. 2012

# Programmation Web MVC en .NET

## OBJECTIFS

Le Framework ASP.NET MVC est une évolution de la plateforme de développement ASP.NET. Il met l'accent sur l'utilisation de patterns d'architecture performants et la testabilité du code. Ce stage très pratique vous apprendra à le mettre en oeuvre pour construire et déployer des applications Web sécurisées et évolutives. Vous verrez comment concevoir des IHM riches avec Ajax et jQuery, et comment produire un code maintenable au moyen d'une architecture MVC (Modèle Vue Contrôleur).

### [1\) Découverte du Framework ASP.NET MVC](#)

### [2\) Architecture d'une application ASP.NET MVC](#)

### [3\) URL et routing](#)

### [4\) Extensibilité des contrôleurs](#)

### [5\) Vues avancées](#)

### [6\) Modèles approfondis](#)

### [7\) Techniques de construction des vues](#)

### [8\) jQuery et ASP.NET MVC](#)

### [9\) Intégration de services ASP.NET dans MVC](#)

### [10\) Déploiement d'une application ASP.NET MVC](#)

### [11\) Les bonnes pratiques ASP.NET MVC](#)

## 1) Découverte du Framework ASP.NET MVC

- Retour sur ASP.NET et limites de l'approche Webforms.
- Les solutions apportées par ASP.NET MVC.

## 2) Architecture d'une application ASP.NET MVC

- Rappels sur le pattern MVC.
- Routing.
- Contrôleurs et actions.
- Vues.
- Modèles.

### Travaux pratiques

Installation d'ASP.NET MVC. Création d'une première application ASP.NET MVC. Ecriture de tests unitaires.

## 3) URL et routing

- Configuration du routage.
- Contraintes et paramètres.
- Areas.
- Tester le routage.

## 4) Extensibilité des contrôleurs

- Les filtres.
- La factory de contrôleur.
- Les contrôleurs asynchrones.
- Tester les contrôleurs et les actions.

### Travaux pratiques

Mise en oeuvre de l'extensibilité des contrôleurs.

## 5) Vues avancées

- Remplacement du moteur de vue par défaut.
- Utilisation des méthodes utilitaires pour HTML.
- Différentes méthodes pour afficher du contenu dynamique.
- Les vues partielles.

### Travaux pratiques

Mise en oeuvre des principes de vues avancées.

## 6) Modèles approfondis

- Métadonnées et annotations.
- Binding.
- Validation.

### Travaux pratiques

Mise en oeuvre des annotations et du binding.

## 7) Techniques de construction des vues

- Utilisation de master pages.
- Règles CSS.
- Les outils Ajax.

## 8) jQuery et ASP.NET MVC

- Rappels sur jQuery.
- Transferts de données avec JSON.
- La bibliothèque jQuery UI.
- Animations et autres effets graphiques.

### **Travaux pratiques**

*Réalisation de vues avec jQuery, Ajax et CSS.*

## 9) Intégration de services ASP.NET dans MVC

- Authentification par formulaires.
- Membership et gestion des rôles et profils.
- Caching.
- Sitemap.
- Localisation.

### **Travaux pratiques**

*Exemple d'une application MVC sécurisée par authentification et gestion d'autorisations.*

## 10) Déploiement d'une application ASP.NET MVC

- Rappels sur IIS.
- ASP.NET MVC et IIS 7.
- ASP.NET MVC et IIS 5.1/6.

### **Travaux pratiques**

*Exécution d'un déploiement d'application MVC sous IIS6 et IIS7.*

## 11) Les bonnes pratiques ASP.NET MVC

- Les erreurs à éviter.
- Les pratiques recommandées (sécurité, SEO...).
- Comment migrer une application Webforms vers MVC ?
- Comment intégrer des contrôles ou des pages Webforms dans MVC ?
- Les outils complémentaires.

Stage pratique de 5 jour(s)  
Réf : SLV

## Participants

Ce cours s'adresse aux développeurs ASP .NET qui souhaitent découvrir la philosophie de Silverlight et les techniques associées afin de concevoir des pages web interactives, avec Visual Studio.

## Pré-requis

Bonne connaissance du Framework .NET 2.0 ou plus, XML, du Visual Studio 2008/2010 et pratique du langage C# ou VB .NET.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

21 mai 2012, 8 oct. 2012  
10 déc. 2012

# Silverlight 4/5, mise en oeuvre applications ASP.NET avec interfaces riches

## OBJECTIFS

*Silverlight sert à enrichir l'expérience utilisateur avec des pages Web constituées d'éléments visuels riches, redimensionnables et interactifs. Cette formation vous permettra de vous familiariser avec les différentes techniques de mise en oeuvre de Silverlight 4/5, au travers d'ateliers réalisés en C# ou en VB .NET.*

### 1) Introduction

### 2) Concepts fondamentaux et conception d'interfaces utilisateurs

### 3) Composants de l'application Silverlight

### 4) Manipulations de données

## Travaux pratiques

*Suite à une introduction aux possibilités offertes par Silverlight, une série d'ateliers allant d'une simple page à la conception d'applications manipulant des données sur un serveur sont réalisés de façon à découvrir comment utiliser les différents composants à mettre en oeuvre.*

## 1) Introduction

- Panorama des outils de développement (Visual Studio, SDK, MS Blend, MS Encoder, Deep Zoom et outils tiers).
- Structure d'une application Silverlight (XAML, code-behind, cycle d'exécution).
- Intégration et interaction avec une page Web avec HTML Bridge, déploiement et mises à jour.

### Travaux pratiques

*Créer une première application simple et exemples de HTML Bridge. Prise en main de MS BLEND (intégration avec Visual Studio, conception d'une interface, personnalisation des modèles de contrôles, réalisation d'animations).*

## 2) Concepts fondamentaux et conception d'interfaces utilisateurs

- Introduction au langage XAML avec équivalences dans le code.
- Hiérarchie des classes et familles d'éléments.
- Rôle des propriétés de dépendance.
- Gestion des événements routés et couplage avec le code-behind.
- Techniques de positionnement avec les éléments Panels, Canvas et Grid.
- Gestion des ressources (physiques et logiques) et des cultures.
- Système de transformations et d'animations.
- Personnalisation : styles et modèles de contrôles.
- Techniques d'intégration de médias (audio et vidéo).

### Travaux pratiques

*Créer, personnaliser et programmer les éléments courants (TextBox, TextBlock, listes, Menu, TabControl, Treeview). Réalisation d'une interface affichant une vidéo et paramétrage d'une application pour les cultures FR et EN.*

## 3) Composants de l'application Silverlight

- Système de navigation avec les contrôles Frame et Page.
- Utilisation et gestion du stockage isolé.
- Créer et déployer une application hors navigateur (OOB).
- Utilisation des Boîtes de dialogue OpenFileDialog, SaveFileDialog et PrintDocument.
- Exploitation des ressources accessibles par l'élévation des privilèges.
- Techniques d'optimisation (objets Dispatcher et BackgroundWorker).

### Travaux pratiques

*Création d'une application de type Business, avec exemples d'utilisation du stockage isolé et des boîtes de dialogue. Création d'une application OOB avec élévation des privilèges avec exemple d'automation COM. Exemple d'utilisation du BackgroundWorker.*

## 4) Manipulations de données

- Présentation et configuration du DataBinding : liaison entre éléments et sur des collections d'objets métiers.
- Techniques de conversions et de validations.
- Gestion des listes et des contrôles de données (ListBox, DataTemplate, DataForm, DataGrid, DataPager et CollectionViewSource).
- Faire des échanges de fichiers sur un serveur avec la classe WebClient.
- Manipuler des informations avec un service WCF ou un service de données.
- Interagir avec des données distantes via RIA Services.
- Principe de l'architecture MVVM avec utilisation de commandes.

### Travaux pratiques

*Création d'une interface d'édition d'une collection d'objets métiers. Récupération de fichiers avec la classe WebClient et traitement de fichiers XML. Exemples d'applications permettant de récupérer et de modifier des données sur un serveur avec un service WCF, un service de données et RIA Services.*

Stage pratique de 3 jour(s)  
Réf : WAZ

## Participants

Développeurs .NET (C# ou VB.NET). Architectes.

## Pré-requis

Bonnes connaissances de la plateforme .NET et d'un langage de développement .NET (C# ou VB.NET). Connaissances de base de la conception d'applications Web ASP.NET et de services WCF.

**Prix 2012 : 1730€ HT**

**Eligible DIF**

## Dates des sessions

**Paris**

1 oct. 2012, 17 déc. 2012

# Windows Azure, développer une application Cloud en .NET

## OBJECTIFS

*Cette formation vous fera découvrir la plateforme Cloud de Microsoft, Windows Azure, ses apports et son fonctionnement. Vous apprendrez à utiliser les outils et technologies disponibles pour développer des applications et les déployer sur cette plateforme.*

### [1\) Introduction : le Cloud Computing](#)

### [2\) Windows Azure Platform](#)

### [3\) Le stockage dans Windows Azure](#)

### [4\) SQL Azure](#)

### [5\) Windows Azure : API avancées](#)

### [6\) Windows Azure App Fabric](#)

## Travaux pratiques

*Les travaux pratiques permettront d'apprendre à développer des applications Web pour Windows Azure, d'utiliser le système de stockage prévu par cette plateforme ou encore la base de données relationnelle SQL Azure. Sera aussi abordée l'utilisation de Windows Azure Service Bus et du contrôle d'accès via Windows Azure App Fabric.*

## 1) Introduction : le Cloud Computing

- Définitions et généralités sur le Cloud Computing.
- Quels sont les acteurs du marché ?
- Notions fondamentales du Cloud Computing.
- La réponse de Microsoft : Windows Azure.

## 2) Windows Azure Platform

- Qu'est-ce que Windows Azure Platform ?
- Infrastructure et principes de fonctionnement.
- Les différents rôles d'exécution.
- Les technologies supportées et les outils.
- Le déploiement d'application.
- Déployer ma première application dans Windows Azure.

### Travaux pratiques

*Prise en main des outils pour Windows Azure.*

## 3) Le stockage dans Windows Azure

- Introduction et généralités.
- Les différents types de stockage : Blobs, Tables, Queues, Disques.
- Les API et leurs utilisations.

### Travaux pratiques

*Manipulation des différents types de stockage proposés par Windows Azure.*

## 4) SQL Azure

- Principes de fonctionnement d'une base de données relationnelle dans le Cloud.
- API et outils.
- Administration et configuration d'une base de données SQL Azure.
- Synchronisation d'une base de données SQL Azure avec une base de données locale.

### Travaux pratiques

*Utilisation de SQL Azure dans une application Windows Azure.*

## 5) Windows Azure : API avancées

- API de diagnostic.
- API de management.

### Travaux pratiques

*Manipulation des API de diagnostic et de management.*

## 6) Windows Azure App Fabric

- Windows Azure Service Bus.
- Le service de contrôle d'accès.
- Fédération d'identités avec Windows Identity Foundation.

### Travaux pratiques

*Service Bus et service de contrôle d'accès.*

Stage pratique de 5 jour(s)  
Réf : SHV

## Participants

Programmeurs ASP.NET.

## Pré-requis

Bonnes connaissances d'un langage de programmation .NET, VB.NET ou C#, et de l'ASP.NET. Expérience requise en développement .NET d'applications Web. Connaissances de base de l'administration et de la configuration de SharePoint 2010.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

21 mai 2012, 17 sep. 2012  
26 nov. 2012

### Strasbourg

21 mai 2012

# SharePoint 2010, développement d'applications avec Visual Studio 2010

## OBJECTIFS

Cette formation vous apprendra à développer, debugger et optimiser les performances d'applications SharePoint 2010 en VB.Net ou en C#. Vous verrez comment persister vos propres configurations, connecter des sources de données métiers externes, développer des Web Parts et des Workflows ou encore intégrer des applications Silverlight à votre environnement SharePoint.

### [1\) Intégration de SharePoint avec IIS 7.5 et ASP.NET](#)

### [2\) Topologie de l'API SharePoint](#)

### [3\) Applications In-Context et Out-of-Context](#)

### [4\) Architecture des outils de trace et de log dans SharePoint](#)

### [5\) Modèle de persistance des objets et des données de configuration](#)

### [6\) Best practices pour manipuler les objets et les collections](#)

### [7\) Architecture de déploiement par Solution SharePoint](#)

### [8\) Limitations du modèle de déploiement Sandbox](#)

### [9\) Architecture et modèle objet des Features](#)

### [10\) Cycle de vie des Features](#)

### [11\) Customisation de SharePoint à l'aide de Features](#)

### [12\) Développement de Web Part](#)

### [13\) Développement de Workflows](#)

### [14\) Business Connectivity Services](#)

### [15\) Développement d'applications Silverlight pour SharePoint](#)

## 1) Intégration de SharePoint avec IIS 7.5 et ASP.NET

- Principes d'exécution d'une application SharePoint au travers d'IIS 7.5 et ASP.NET 3.5.
- Différences fondamentales entre un site ASP.NET et un site SharePoint.
- Eléments d'architecture : fichier web.config d'une application SharePoint, bases de données de configuration et de contenu, modèle de provisionning, pages ASPX (pages de site et pages applicatives).
- Principes du mécanisme de customisation des pages.
- Architecture des Master Pages et des thèmes. Droits d'accès. Affichage basé sur les droits.

### Travaux pratiques

Premiers éléments de configuration.

## 2) Topologie de l'API SharePoint

- Architecture logique de SharePoint et hiérarchie associée du modèle objet.
- Topologie des Assemblies .Net de SharePoint 2010 côté serveur et côté client.
- Topologie des services WCF et REST exposés par SharePoint.

## 3) Applications In-Context et Out-of-Context

- Contexte d'utilisation du modèle objet.
- Développement " In-Context " d'applications SharePoint. Etude du pipeline http.
- Développement " Out-of-Context ".
- Développement d'applications utilisant l'API cliente de SharePoint.

### Travaux pratiques

Développement d'applications In-Context et Out-of-Context.

## 4) Architecture des outils de trace et de log dans SharePoint

- Architecture de trace de SharePoint. Outils d'analyse des logs.
- Développer Dashboard.
- Mise en oeuvre du " Patterns & Practices SharePoint Guidance " de Microsoft.

### Travaux pratiques

Mise en place d'une analyse de logs.

## 5) Modèle de persistance des objets et des données de configuration

- Modèle de persistance des objets dans la base de données de configuration et dans les bases de données de contenu.
- Property Bag.
- Scénarios d'utilisation.

### Travaux pratiques

Mise en oeuvre de la persistance des objets.

## 6) Best practices pour manipuler les objets et les collections

- Problématique d'utilisation des collections SharePoint dans le modèle objet.
- Classe SPBaseCollection. Méthode Dispose.
- Gestion des collections et mise en cache.
- Gestion du multithreading.
- SPDisposeCheck.
- Ressources et articles de référence.
- Symptômes évoquant le non-respect des best practices.

### Travaux pratiques

*Manipulation des collections.*

## 7) Architecture de déploiement par Solution SharePoint

- Cycle de vie d'un fichier WSP.
- Création de Pack de langues et de Pack de mise à jour.
- Modèles Full trust et Sandbox.
- Modèles d'exécution d'une application déployée par WS.

## 8) Limitations du modèle de déploiement Sandbox

- Déploiement hybride.
- Création, utilisation, modification et reprise d'une Solution SharePoint dans Visual Studio 2010.
- Comment auditer et analyser une Solution SharePoint.

## 9) Architecture et modèle objet des Features

- A quoi sert un Feature ? Ce que n'est pas un Feature.
- Méthodologie de développement et de création d'un Feature.
- Anatomie d'un Feature.
- Packaging de Features.
- Titre, description et gestion des ressources locales d'un Feature.
- Ressources globales d'un Feature.
- Fichier de manifeste associé à un Feature.
- Contraintes sur la création de Feature en fonction de la portée.
- Principes de mise à jour d'un Feature.
- Feature Stapling.

### Travaux pratiques

*Création et configuration de Features.*

## 10) Cycle de vie des Features

- Etape de création. Etape d'installation du Feature.
- Développer un Feature Receiver.
- Etape d'activation du Feature lorsque : Scope=Web, Scope=Site, Scope=WebApplication, Scope=Farm.
- Etape de désactivation du Feature lorsque : Scope=Web, Scope=Site, Scope=WebApplication, Scope=Farm.
- Etape de désinstallation du Feature.
- Etape de mise à jour d'un Feature.
- Feature Receiver.

### Travaux pratiques

*Activation, désactivation et désinstallation de Features.*

## 11) Customisation de SharePoint à l'aide de Features

- Création, modification de Colonnes de Site.
- Développement de type personnalisé de colonne.
- Création, modification de type de contenu.
- Création de modèles de listes. Création, modification de listes.
- Ajouts de fichiers et dossiers dans des bibliothèques.
- Ajouts de pages, de modèles de page.
- Customisation de l'interface graphique et du ruban office, création de fenêtres de dialogue.
- Customisation du menu ECB (Edit Control Block).
- Développement d'Event Receivers et de Timer Job.
- Delegate Controls.
- Customisation de la navigation TopNavBar, Quick Launch.
- Appliquer une nouvelle Master Page.
- Création par code de sites de niveaux d'autorisation et de groupes SharePoint.
- Elévation de privilèges avec RunWithElevatedPrivileges.

### Travaux pratiques

*Personnalisation de divers éléments via l'utilisation de Features.*

## 12) Développement de Web Part

- Architecture des Web Part.
- Déploiement et intégration d'une Web Part sur un serveur SharePoint.
- Modèle d'exécution d'une Web Part : Full Trust, Partial trust, Sandbox.
- Principe de développement d'une Web Part avec Visual Studio 2010.
- Visual Web part.
- Interconnexion de Web part.
- Shared View, Personal View, Maintenance View.
- Upgrade de Web part.
- Mise en oeuvre du Pattern MVP (Model View Presenter) dans une Web part.
- Gestion des erreurs avec la mise en oeuvre du Pattern Exception Shielding.
- Utilisation de contrôle d'affichage SharePoint au sein d'une Web Part.
- Développement d'un Editor Part, de filtres d'affichage.
- Création d'un Proxy Full trust pour étendre les fonctionnalités d'une Web part dans un environnement Sandbox.
- Accès aux données en utilisant SPQuery, SPSiteDataQuery, SPMetal, Linq to SharePoint.

## 13) Développement de Workflows

- Principes de développement et d'intégration d'un WorkFlow.
- Développement de formulaires d'initiation, d'association, de modification.
- Création de liste de tâches personnalisées contenant des données métiers.
- Enchaînements de Workflow sur un même document.
- Développer un Workflow avec SharePoint Designer 2010 ou avec Visual Studio 2010.
- Développer des Actions pour enrichir le développement de Workflow sous SharePoint Designer.
- Upgrade de Workflow.

## 14) Business Connectivity Services

- Liste externe.
- Type de contenu externe.
- Modèle objet du BCS.

## 15) Développement d'applications Silverlight pour SharePoint

- Création, déploiement et débogage d'une application Silverlight.

Stage pratique de 3 jour(s)  
Réf : SHD

## Participants

Développeurs .NET participant à des projets SharePoint Services 3.0 et/ou MOSS 2007.

## Pré-requis

Bonnes connaissances du langage C# ou VB.NET.

Prix 2012 : 1650€ HT

Eligible DIF

## Dates des sessions

### Paris

16 juil. 2012, 3 sep. 2012  
26 nov. 2012

### Aix

1 oct. 2012, 3 déc. 2012

### Lyon

1 oct. 2012, 3 déc. 2012

### Nantes

8 oct. 2012, 10 déc. 2012

### Rennes

8 oct. 2012, 10 déc. 2012

### Sophia-antipolis

1 oct. 2012, 3 déc. 2012

# Sharepoint 2007, développer en .NET

## OBJECTIFS

Ce cours a pour objectif l'apprentissage des fonctionnalités de SharePoint 2007 pour des développeurs .NET. Il concerne tous ceux qui doivent mettre en œuvre SharePoint Services et MOSS 2007 sur leurs projets.

### 1) Présentation de WSS v3 et Moss 2007

### 2) Nouvelles fonctionnalités WSS v3

### 3) Authentification WSS v3

### 4) Autorisation sous WSS v3

### 5) Features dans WSS v3

### 6) Définitions de sites WSS v3

### 7) Fondamentaux du modèle Objet WSS v3

### 8) Web Services dans WSS v3

### 9) Fondamentaux des Web Parts personnalisées

### 10) Techniques de Web Parts personnalisées

### 11) Techniques avancées Web Part

### 12) Les événements dans WSS v3

### 13) Les Workflows dans WSS v3

## 1) Présentation de WSS v3 et Moss 2007

- Présentation de l'architecture et des éléments clés dans WSS (collections de site, listes, web parts...).
- Introduction au namespace Microsoft.Sharepoint.

## 2) Nouvelles fonctionnalités WSS v3

- Introduction aux nouvelles fonctionnalités de WSS v3 : les galeries, les masters pages, les types de contenu, les colonnes de site, les Workflows.
- Introduction aux nouveautés de MOSS 2007.

## 3) Authentification WSS v3

- Comment choisir une stratégie d'authentification ?
- Comment implémenter de multiples zones d'authentification avec ASP. Net 2.0 ?

## 4) Autorisation sous WSS v3

- Configurer les permissions utilisateurs/groupes.
- Héritage des permissions.

## 5) Features dans WSS v3

- Features. Définition de sites WSS v3. Création d'un en-tête Feature dans un fichier XML. Ecrire un élément Feature dans un fichier XML. Installer, activer, désactiver et désinstaller une Feature.

## 6) Définitions de sites WSS v3

- Ecrire une définition de site personnalisée avec CAML (Collaborative Application Markup Language).
- Spécifier une feuille de style en cascade (CSS).

## 7) Fondamentaux du modèle Objet WSS v3

- Les namespaces du modèle objet WSS v3.
- Les tâches communes du modèle objet.

## 8) Web Services dans WSS v3

- Introduction, utilisation, avantages et limitations.

## 9) Fondamentaux des Web Parts personnalisées

- Définition et utilité d'une Web Part.
- Différence entre les Web Parts ASP 2.0 et WSS v3.
- Personnalisation d'une Web Part.

## 10) Techniques de Web Parts personnalisées

- Construire et déployer une Web Part.
- L'interface utilisateur d'une Web Part.
- Debugger et déployer une Web Part.

## 11) Techniques avancées Web Part

- Modifier le menu d'une Web Part.
- Utiliser le Web Part Page Services Component (WPSC) dans un environnement de développement de script client Web Part.

## 12) Les événements dans WSS v3

- Utilisation, installation, activation des Event Handlers.

### 13) Les Workflows dans WSS v3

- Créer des Workflows avec SharePoint Designer.
- Utiliser le starter kit Enterprise Content Management dans VS.NET.

Stage pratique de 5 jour(s)  
Réf : NET

## Participants

Ce cours s'adresse aux développeurs professionnels et expérimentés désirant apprendre le développement en utilisant le framework .NET et les Windows Forms.

## Pré-requis

Bonnes connaissances de la programmation en C# ou VB.NET.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

18 juin 2012, 3 sep. 2012  
26 nov. 2012

# Applications Windows avec Visual Studio 2010/2008 et .NET 3.5

## OBJECTIFS

Cette formation vous apprendra à utiliser les classes du Framework .NET avec l'un des langages de .NET pour développer des applications Windows Forms. Elle vous montrera également comment vous servir d'ADO.NET pour l'interaction aux bases de données, et vous proposera une première approche des Web Services.

### 1) Introduction

### 2) L'interface utilisateur : Windows Forms

### 3) Les contrôles IHM des Windows Forms

### 4) Création d'interfaces MDI

### 5) Globalisation et localisation des applications

### 6) Gestion des graphiques : GDI+

### 7) Les Windows Forms et le multitâche

### 8) Utilisation de XML dans le .NET

### 9) Accès aux données

### 10) Utilisation des services Web

## 1) Introduction

- Rappels sur .NET l'écriture d'une application.

### Travaux pratiques

*Écriture d'une application en mode console.*

## 2) L'interface utilisateur : Windows Forms

- Création d'interfaces utilisateurs.  
- Gestion des composants de l'interface utilisateur.  
- Gestion du drag and drop. Click Once Deployment.

### Travaux pratiques

*Création d'applications utilisant des contrôles IHM.*

## 3) Les contrôles IHM des Windows Forms

- Contrôles de base (textbox, checkbox, listbox, etc.) et plus complexes (calendar, DataGridView, WebBrowser...)  
- Développement de contrôles personnalisés.

### Travaux pratiques

*Personnalisation des contrôles. Manipulations avancées.*

## 4) Création d'interfaces MDI

- Principe. Interaction entre les fenêtres MDI.

### Travaux pratiques

*Création d'application utilisant l'interface MDI.*

## 5) Globalisation et localisation des applications

- Séparation du traitement et de l'IHM.  
- Gestion des cultures et des paramètres nationaux.  
- Gestion manuelle des chaînes. Gestion des ressources.

### Travaux pratiques

*Localisation d'une application. Interface multilingue.*

## 6) Gestion des graphiques : GDI+

- Interface graphique GDI+, création d'un dessin.  
- Gestion des images, du texte et des impressions.

### Travaux pratiques

*Création d'applications utilisant l'interface graphique GDI.*

## 7) Les Windows Forms et le multitâche

- Multithread et multiprocessus dans .NET.  
- Windows Forms faisant du multithreading.

### Travaux pratiques

*Écriture d'une application multithread/Windows Forms.*

## 8) Utilisation de XML dans le .NET

- Principe de XML. Données. Schéma XSD. Echanges.

**Travaux pratiques**

*Manipulation des classes de création XML (entrées/sorties).*

## 9) Accès aux données

- Principe d'ADO.NET.
- Lecture par flux. Utilisation des commandes.
- Utilisation en mode non connecté.
- Accès aux bases de données.
- Accès au schéma d'une base. Accès asynchrones.

**Travaux pratiques**

*Utilisation de ADO.NET pour l'accès aux données.*

## 10) Utilisation des services Web

- Introduction aux services Web. Description WSDL. SOAP.

**Travaux pratiques**

*Utilisation de Web Services pour l'accès aux données.*

Stage pratique de 4 jour(s)  
Réf : ASH

## Participants

Développeurs .NET Windows Forms.

## Pré-requis

Bonnes connaissances de la programmation Windows Forms avec C# ou VB.NET, ainsi que d'ADO.NET. Expérience requise en développement logiciel.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

Paris

16 oct. 2012, 11 déc. 2012

# Applications Windows, perfectionnement avec Visual Studio 2010/2008 et .NET 3.5

## OBJECTIFS

Cette formation fournit aux participants les connaissances nécessaires pour développer des applications Windows Forms avancées mettant en oeuvre ADO.NET et XML.

### 1) Architecture et conception des applications .NET/Windows Forms

### 2) Organisation en couches : couches de données et couche de présentation

### 3) Ecriture de contrôles de présentation

### 4) Outils de DataBinding

### 5) Utilisation des classes graphiques pour la génération d'images

### 6) Fiabilisation des applications : sécurité et paramétrage

### 7) Interopérabilité avec COM

## 1) Architecture et conception des applications .NET/Windows Forms

- Organisation multicouche : couche de présentation, couche d'accès aux données, couche métier.
- Structure et rôle des diverses couches.

## 2) Organisation en couches : couches de données et couche de présentation

- Composants d'accès aux données, utilisation des dataset typés.
- Utilisation de Web Services pour la couche métier.
- Couche de présentation.

## 3) Ecriture de contrôles de présentation

- Architecture des contrôles personnalisés et modèle objet associé.
- Approche de développement de l'affichage : extension des contrôles standard, utilisation des GDI.
- Evénements. DataBinding.

## 4) Outils de DataBinding

- Liaison de données sur des classes de collections.
- Personnalisation du DataGrid : utilisation avancée du DataBinding.
- Pagination du datagrid, mise en place.
- Vues Maître/Détail avec le DataGrid.

## 5) Utilisation des classes graphiques pour la génération d'images

- Namespace System. Drawing. Utilisation pour la définition d'images et le redessin de contrôles.

## 6) Fiabilisation des applications : sécurité et paramétrage

- Modèles de sécurité, credentials. Gestion des autorisations et rôles. Utilisation de la cryptographie.
- Gestion de la configuration des applications.
- Gestion du déploiement et de la mise à jour des applications. Approche de mise à jour automatique.

## 7) Interopérabilité avec COM

- Architecture de l'interopérabilité COM.
- Mise en place de l'appel aux objets COM, problématique de sécurité.

### Exemple

*Développement d'un module d'accès aux données. Intégration dans le projet et dans le concepteur. Intégration de Web Services et .NET Remoting pour le traitement. Gestion des communications et de la sécurité. Développement d'une application avec structuration en couche. Utilisation des Design Patterns de présentation. Développement de contrôles personnalisés. Utilisation des événements. Utilisation des GDI pour la présentation. Utilisation du DataBinding. Extension du DataGrid. Génération de graphique dans les applications Web. Mise en place de l'authentification. Utilisation du chiffrement. Utilisation de composants COM externes. Paramétrage externe. Lecture du fichier de paramétrage.*

Stage pratique de 4 jour(s)  
Réf : NPF

## Participants

Ce cours s'adresse aux développeurs .NET qui souhaitent découvrir les possibilités offertes par WPF et comprendre les outils permettant sa mise en oeuvre.

## Pré-requis

Connaissance du langage XML et de la plate-forme .NET (Framework 3.5 ou supérieur, Visual Studio et C# ou VB .NET).

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

12 juin 2012, 2 oct. 2012  
4 déc. 2012

# WPF, développement .NET d'applications Windows Presentation Foundation et VS 2010/2008

## OBJECTIFS

Cette formation vous permettra de vous familiariser avec la conception des différents types d'applications WPF (.exe, applications de navigateur et Silverlight), avec des ateliers pratiques réalisés avec Visual Studio en C# ou en VB .NET et MS Blend.

### 1) Introduction

#### 2) Conception des interfaces

#### 3) Prise en main de MS Blend

### 4) Le DataBinding

#### 5) Principe des applications de navigateur et introduction à Silverlight

## Travaux pratiques

Mise en oeuvre des différentes techniques (grammaire XAML, personnalisation des contrôles et des éléments visuels, modèle de programmation et déploiement) au sein d'une application WPF.

## 1) Introduction

- Comprendre les objectifs, l'architecture de WPF et les différents types de projets disponibles.
- Le langage XAML : syntaxe, rôle des différents types d'éléments et des extensions de balisage.
- Les outils Visual Studio, MS Blend et du SDK.
- Structure d'une application WPF : fichier XAML et code-behind, hiérarchie des classes, compilation.
- Modèle de programmation : propriétés de dépendance, cycle d'exécution, événements routés et des commandes.

### Travaux pratiques

Applications types. Créer une interface simple en XAML et ajouter un comportement avec Visual Studio.

## 2) Conception des interfaces

- Conception d'une fenêtre et méthodes de positionnement des éléments.
- Familles d'éléments (contrôles, Formes 2D et 3D, transformations et animations).
- Personnalisation : styles et modèles de contrôles.
- Utilisation et gestion des ressources.
- Intégration d'éléments multimédias.
- Gestion des documents (flux et fixes).

### Travaux pratiques

Créer, personnaliser et programmer les éléments courants, avec exemples de transformations et d'animations. Réaliser un contrôle personnalisé. Créer un document par programmation. Gérer l'affichage de RTF et XPS existants.

## 3) Prise en main de MS Blend

- Organisation de l'interface et intégration avec Visual Studio.
- Gestion des ressources et des assets.
- Conception d'une interface.
- Réalisation d'animations et de templates.

### Travaux pratiques

Réalisations d'une interface de saisie avec MS Blend.

## 4) Le DataBinding

- Rappel du fonctionnement.
- Editeur d'expressions d'extension de balisage.
- Choix et configuration du mode de liaison avec les extensions de balisage ou par code.
- Gestion des différents types de sources.
- Modèle de présentation des données (DataTemplate).
- Opérations de conversion et de validation.
- Contrôle DataGridView.

### Travaux pratiques

Réaliser des liaisons de données en lecture/écriture sur une collection d'objets métiers, avec validation et conversion.

## 5) Principe des applications de navigateur et introduction à Silverlight

- Présentation des usages types avec les solutions .NET.
- Créer et déployer une application de navigateur.
- Présentation de Silverlight (modèle de développement et de déploiement, affichage en plein écran...).

### Travaux pratiques

*Créer une application de navigateur avec déploiement par ClickOnce. Réaliser une application Silverlight.*

Stage pratique de 4 jour(s)  
Réf : OWS

## Participants

Ce cours s'adresse aux développeurs qui souhaitent exploiter .NET et les techniques associées telles que XAML pour créer des applications Windows Phone.

## Pré-requis

Bonnes connaissances de C# (ou VB .NET) et de XML.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

12 juin 2012, 25 sep. 2012  
27 nov. 2012

# Windows Phone 7, développement d'applications en .NET avec Silverlight et XNA

## OBJECTIFS

*Cette formation détaille par la pratique comment utiliser les frameworks .NET, Silverlight et XNA afin de développer des applications pour Windows Phone 7. Vous apprendrez les règles d'ergonomie liées à l'interface Metro, à concevoir des interfaces MultiTouch, à utiliser efficacement le langage XAML et à mettre en oeuvre la gestion des événements routés et l'exploitation des capteurs (GPS, accéléromètre).*

### [1\) Introduction](#)

### [2\) Concepts fondamentaux](#)

### [3\) Conception d'interfaces utilisateurs](#)

### [4\) Interactions avec le téléphone](#)

### [5\) Gestion des données](#)

### [6\) XNA](#)

### [7\) Ressources complémentaires](#)

## Travaux pratiques

*Après avoir pris connaissance des caractéristiques matérielles et logicielles des appareils ciblés, les ateliers focaliseront sur les principaux usages liés aux applications pour mobiles (conception d'interfaces, navigation, interactions avec l'interface MultiTouch, GPS, gestion des données, communications avec un serveur et jeux). Développement sous Visual Studio 2010 pour Windows Phone.*

## 1) Introduction

- Présentation de la plateforme Windows Phone 7.
- Organisation des interfaces Metro.
- Configuration de l'environnement de développement.
- Utilisation de l'émulateur.
- Architecture d'une application Silverlight.
- Cycle de vie d'une application et gestion Tombstoning.
- Présentation d'exemples et des ateliers proposés.

### Travaux pratiques

*Configurer l'environnement de développement Visual Studio 2010 pour réaliser une première application Silverlight avec l'émulateur sur Visual Studio pour Windows Phone.*

## 2) Concepts fondamentaux

- Gestion de l'affichage (tailles standard et orientation).
- Introduction au langage XAML : syntaxe, propriétés attachées, extensions de balisage.
- Hiérarchie des classes et familles d'éléments (positionnement, formes graphiques de base, contrôles).
- Gestion des événements routés.

### Travaux pratiques

*Mise en oeuvre des événements routés sur des contrôles simples (boutons, images et listes).*

## 3) Conception d'interfaces utilisateurs

- Techniques de positionnement avec les éléments StackPanel, Canvas, Grid et ScrollView.
- Présentation des contrôles standard (TextBlock, TextBox, CheckBox, listes, boutons, images, etc.) et des formes 2D.
- Utilisation des styles et gestion des ressources.
- Système de transformations et d'animations.
- Système de navigation entre pages.
- Introduction à l'utilisation de Blend pour Windows Phone.

### Travaux pratiques

*Créer, personnaliser et programmer les éléments courants, avec exemples de transformations et d'animations. Réaliser une application multipages, avec échanges de données entre pages.*

## 4) Interactions avec le téléphone

- Utilisation des capteurs (accéléromètre, GPS), des Lanchers et des Choosers (appareil photo, téléphone, email, etc.).
- Interactions avec l'interface tactile MultiTouch.
- Présentation des opérations avec un ApplicationBar.
- Mise en oeuvre des contrôles Pivot et Panorama (Hub).
- Gestion des médias (images, photos, son et vidéo).
- Utilisation du service Bing Map.

### Travaux pratiques

*Conception d'une interface avec les éléments Pivot et Panorama. Réalisation d'une interface d'application avec BLEND pour Windows Phone. Utilisation des capteurs, de l'appareil photo, gestion de fichiers audio, vidéo et du service BING Maps.*

## 5) Gestion des données

- Intégration de LINQ.
- Rappel sur le DataBinding et mise en oeuvre en Silverlight.
- Utilisation de l'Isolated Storage.
- Se connecter à un serveur avec la classe WebClient ou par un service WCF.

### **Travaux pratiques**

*Exemples de manipulations des classes utiles en Silverlight. Réalisation d'une application permettant de récupérer des fichiers (XML et binaires) sur un serveur. Gestion du Tombstoning.*

## 6) XNA

- Architecture d'une application XNA.
- Concepts fondamentaux : vecteurs et gestion des mouvements.
- Sprites et textures.
- Gestion de l'interactivité.

### **Travaux pratiques**

*Exemples de conception d'interface et réalisation d'un jeu de labyrinthe.*

## 7) Ressources complémentaires

- Présentation du MarketPlace.
- Présentation de la plateforme Azure.

Stage pratique de 4 jour(s)  
Réf : MTP

## Participants

Développeurs .NET ayant à créer des applications pour iPhone et iPad.

## Pré-requis

Connaissance impérative du langage C# 2.0 ou Java.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

12 juin 2012, 2 oct. 2012  
11 déc. 2012

# MonoTouch, créer des applications iPhone/iPad en C# et .NET

## OBJECTIFS

Ce stage s'adresse à tous ceux qui souhaitent développer des applications natives pour iPhone et iPad en utilisant le langage C# et les technologies .NET au sein de la plateforme MonoTouch. Ils apprendront à utiliser l'environnement MonoDevelop, et à programmer les IHM, l'accès aux données, les éléments multimédias, les interactions hardware...

[1\) Introduction au développement d'applications pour l'iOS d'Apple](#)

[2\) Plateforme de développement MonoDevelop](#)

[3\) Navigations et contrôles de données](#)

[4\) Utilisation des outils de cartographie et de localisation](#)

[5\) Interaction avec la partie hardware de l'iPhone](#)

[6\) Programmation des éléments multimédias](#)

[7\) Intégration avec des applications externes](#)

## 1) Introduction au développement d'applications pour l'iOS d'Apple

- Historique du développement d'applications pour iPhone.
- La plateforme de développement et les technologies Apple.
- Brève introduction à Objective-C et Cocoa.
- Positionnement des Frameworks Mono et MonoTouch pour le système d'exploitation iOS.
- Installation des composants.
- Publication des applications iPhone vers le magasin Apple Store.

### Démonstration

*Exemples d'applications. Démonstration des possibilités.*

## 2) Plateforme de développement MonoDevelop

- Introduction à l'atelier de développement MonoDevelop.
- Les différents types de projets.
- Revue des principaux contrôles utilisables.
- Création d'une première application pour iPhone.
- Le linker et les techniques de débogage.
- Revue des bonnes pratiques conseillées par Apple lors du développement des interfaces pour iPhone.
- Types d'interfaces disponibles : radio, commande, navigation, modal, etc.
- Exploration de la notion de contrôleur.

### Travaux pratiques

*Prise en main de la plateforme de développement. Création d'un projet et d'une première application.*

## 3) Navigations et contrôles de données

- Comprendre le fonctionnement des fenêtres, des vues, des contrôleurs.
- Afficher des données à l'utilisateur.
- Utiliser la barre de menus et la navigation.
- Travailler avec des données locales (SQLite) et distantes via SOAP et REST.

### Travaux pratiques

*Ajout de contrôles utilisateurs et d'éléments de navigation à l'application. Programmation de l'accès aux données et affichage des données dans l'application.*

## 4) Utilisation des outils de cartographie et de localisation

- Comprendre les outils CoreLocation et Mapkit.
- Ajouter des services de géolocalisation.
- Utilisation des services web de Google.

### Travaux pratiques

*Programmation d'un service de localisation.*

## 5) Interaction avec la partie hardware de l'iPhone

- Interactions avec les accélérations.
- Gérer l'orientation du terminal.
- Gérer et interagir avec la batterie.
- Gérer le réseau.
- Gérer les informations systèmes.

### Travaux pratiques

*Mise en oeuvre des interactions hardware.*

## 6) Programmation des éléments multimédias

- Téléchargement d'images.
- Intégration et gestion des images.
- Jouer et enregistrer des éléments audio.
- Visionner et enregistrer des éléments vidéo.
- Développer des animations.

### **Travaux pratiques**

*Utilisation et gestion d'images au sein de l'application.*

## 7) Intégration avec des applications externes

- Utilisation de la méthode OpenURL.
- Intégration avec iTunes.
- Intégration avec Google Earth.
- Intégration avec Skype.
- Intégration avec le carnet d'adresses.

### **Travaux pratiques**

*Mise en oeuvre de l'interaction avec quelques applications externes.*

Stage pratique de 5 jour(s)  
Réf : CVS

## Participants

Ce stage s'adresse aussi bien aux chefs de projets, qu'aux développeurs qui souhaitent maîtriser le langage C#5.0 et Visual Studio 2011.

## Pré-requis

Connaissance du langage C# 2.0 ou très bonne connaissance des langages Java ou C++. Les notions de programmation orientée objet doivent être connues et maîtrisées.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

24 sep. 2012, 3 déc. 2012

# C# 5.0, mise en oeuvre des nouveautés avec Visual Studio 2011

## OBJECTIFS

Ce stage focalise sur la programmation d'applications .NET en C# 5.0. Les stagiaires découvriront les nouveautés du langage et de Visual Studio 2011. Ils apprendront à mettre en oeuvre les services WCF, l'Entity Framework pour la gestion des données et le Framework WPF pour la création d'IHM performantes et modernes.

### 1) Evolutions du langage C# : V1.0 à V5.0

### 2) Architectures orientées services via Windows Communication Foundation

### 3) Connexion aux bases de données via LINQ et l'Entity Framework

### 4) Développement d'IHM via Windows Presentation Framework

### 5) Déploiement des applications .NET

### 6) Applications spécifiques

## 1) Evolutions du langage C# : V1.0 à V5.0

- C# 1.0 : bases d'un langage orienté objet.
- C# 2.0 : génériques et méthodes anonymes.
- C# 3.0 : les méthodes d'extension pour l'infrastructure LINQ.
- C# 4.0 : interopérabilité dynamique et parallélisme.
- C# 5.0 : évolution de la programmation asynchrone.
- Bonnes pratiques en matière de développement .NET.
- Nouveautés de l'atelier Visual Studio 2011.

### Travaux pratiques

Analyse d'une application bancaire développée en C# 4.0 utilisant un client WinForm et dont le service métier est exposé via un Service Web. Divers exemples concrets illustrant l'évolution de C#.

## 2) Architectures orientées services via Windows Communication Foundation

- Introduction à l'architecture WCF.
- Etudes de cas d'utilisation.
- Exposer un service métier.
- Utilisation d'un hôte .NET avec points de terminaison netTcp, MSMQ, wsHttp et basicHttp.
- Gestion des identités et mise en place d'une sécurité basée sur les rôles.
- Implémentation d'un service transactionnel de type DTC et WS-AtomicTransaction.
- Implémentation du pattern d'abonnement via Callback.

### Travaux pratiques

Réalisation d'un service WCF avec illustration des implémentations.

## 3) Connexion aux bases de données via LINQ et l'Entity Framework

- Présentation des fonctionnalités des classes de base de l'Entity Framework :ObjectContext et EntityObject.
- Génération du code LINQ pour les opérations de requêtage CRUD sur le modèle de données.
- Utilisation des procédures stockées mappées aux opérations CUD.
- Utilisation directe des procédures stockées et exploitation des données retournées.
- Personnalisation du modèle via associations, projections, héritage.
- Générations de classes POCO dans le cadre d'une utilisation via WCF.

### Travaux pratiques

Réalisations d'applications mettant en oeuvre les opérations CRUD et utilisation de classes POCO via WCF.

## 4) Développement d'IHM via Windows Presentation Framework

- Architecture des applications WPF et fondamentaux du langage XAML.
- Principaux contrôles conteneurs et environnement graphique.
- Ressources, Styles, Transformations, Animations, Modèle de contrôle, Médias et contrôles du Toolkit WPF.
- Infrastructure de liaison de données, modèle de données avec ListBox, ListView et DataGrid.
- Positionnement de Silverlight 5.0 par rapport à WPF et illustration du modèle WCF RIA Services.

### Travaux pratiques

Développement d'une application WPF de gestion de stages (affichage, planning, inscription, envoi mail, transaction), avec utilisation d'un service WCF et de l'Entity Framework pour les données.

## 5) Déploiement des applications .NET

- Récapitulatif des architectures généralement implémentées et leur mode de déploiement.
- Les différents modèles de projets de déploiement exposés par Visual Studio 2011.
- Utilisation de Visual Studio pour le déploiement d'applications ASP.Net 3.5 et WCF vers la plateforme Windows Azure.

**Travaux pratiques**

*Déploiement d'une application WCF vers Windows Azur.*

## 6) Applications spécifiques

- Introduction aux développements d'applications Métro.
- Introduction aux applications pour Windows Phone 7.
- Introduction aux applications C# ciblant iPhone et iPad via le Framework MonoTouch.

Stage pratique de 3 jour(s)  
Réf : TUD

## Participants

Ce cours s'adresse aussi bien à des décideurs, qu'aux développeurs qui souhaitent découvrir les fonctionnalités du Framework .NET 4.0 et de Visual Studio 2010 au moyen d'exemples simples et concrets.

## Pré-requis

Bonnes connaissances de C# ou VB .NET.

**Prix 2012 : 1730€ HT**

**Eligible DIF**

## Dates des sessions

**Paris**

8 oct. 2012, 17 déc. 2012

# Visual Studio 2010/.Net 4.0, les nouveautés

## OBJECTIFS

*Cette formation vous fera découvrir par la pratique les nouveautés des langages C#4 et VB 10, mais aussi de l'environnement Visual Studio 2010, en vue de développer les principaux types d'applications .NET avec le Framework 4.0.*

### [1\) Evolutions des langages et du Framework](#)

### [2\) Applications WPF](#)

### [3\) Applications ASP .NET](#)

### [4\) Entity Framework](#)

### [5\) Plateforme Azure](#)

## 1) Evolutions des langages et du Framework

- Aperçu des améliorations du runtime et de la BCL (sécurité, instrumentation, runtime dynamique, etc.)
- Gestion de la compatibilité (framework targeting et types obsolètes).
- Typage dynamique, arguments nommés et optionnels, variance et covariance.
- Programmation parallèle.

### Travaux pratiques

*Exemples adaptés de la documentation de VS 2010, avec compilation dynamique et programmation parallèle.*

## 2) Applications WPF

- Améliorations du concepteur et nouveaux contrôles (DataGrid, Calendar et DatePicker).
- Support des interfaces tactiles.
- Applications de navigateur (gestion du déploiement et de la sécurité).
- Support de la barre des tâches de Windows 7.

### Travaux pratiques

*Réalisation d'une application simple et présentation d'exemples.*

## 3) Applications ASP .NET

- Améliorations des Web Forms (accessibilité, contrôle des ID générés, nouveau rendu du contrôle Menu, contrôle Chart, gestion des Méta-tags, gestion des navigateurs récents, optimisation du viewstate...).
- Améliorations de l'application (nouveaux modèles d'applications, intégration de JQuery, gestion des redirections permanentes et des réécritures d'url, gestion du cache).
- Nouvelles bibliothèques AJAX (cliente et intégrée aux Web Forms).
- Principe des ASP.NET Dynamic Data pour créer des applications Web dites " data-driven ".
- Principe des applications ASP .NET MVC.
- Amélioration du déploiement avec un package Web.
- Principe et conception d'une application Silverlight.

### Travaux pratiques

*Réalisation d'une application simple sur le modèle standard (PostBack) et sur le modèle MVC. Réalisation d'une application Silverlight affichant des données stockées dans une base SQL avec un service WCF.*

## 4) Entity Framework

- Présentation du Framework : architecture, étendue des possibilités, modèle conceptuel et mapping.
- Opérations de requêtage et de mise à jour de données.
- Gestion des exceptions et des accès concurrentiels.
- Présentation du service de données ADO .NET (principe de l'architecture REST, avec opérations de requêtage et de mise à jour).

### Travaux pratiques

*Exemples de configuration du modèle d'entités, avec manipulation de données. Création et consommation d'un service WCF basé sur ADO .NET Data Services.*

## 5) Plateforme Azure

- Présentation de la plateforme : architecture et services.

Stage pratique de 5 jour(s)

Réf : PSB

## Pré-requis

Bonnes connaissances du langage C# 2.0.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

18 juin 2012, 9 juil. 2012  
20 août. 2012, 10 sep. 2012  
15 oct. 2012, 12 nov. 2012  
17 déc. 2012

### Aix

11 juin 2012, 17 sep. 2012  
19 nov. 2012

### Bordeaux

1 oct. 2012, 10 déc. 2012

### Bruxelles

1 oct. 2012, 10 déc. 2012

### Genève

1 oct. 2012, 10 déc. 2012

### Lille

1 oct. 2012, 10 déc. 2012

### Luxembourg

18 juin 2012, 1 oct. 2012  
10 déc. 2012

### Lyon

11 juin 2012, 17 sep. 2012  
19 nov. 2012

### Nantes

24 sep. 2012, 26 nov. 2012

### Rennes

24 sep. 2012, 26 nov. 2012

### Sophia-antipolis

11 juin 2012, 17 sep. 2012  
19 nov. 2012

### Strasbourg

1 oct. 2012, 10 déc. 2012

### Toulouse

1 oct. 2012, 10 déc. 2012

# C# 3.0, développer en Framework 3.5 avec Visual Studio 2010

## OBJECTIFS

Ce stage présente les nouveautés apportées par le Framework .Net version 3.5, et la version 3 de C#, notamment les extensions LINQ pour l'accès aux données. Vous verrez également comment créer des contrôles pour les nouvelles IHM Windows, construire des workflows et faire les bons choix en matière d'applications distribuées.

### 1) Introduction

### 2) Le langage Microsoft IntegratedQuery (LINQ)

### 3) XML avec le framework 3.5

### 4) Evolutions de ADO.Net et LINQ To DataSet

### 5) LINQ To SQL (Entities)

### 6) Développement de services WCF

### 7) Intégration d'applications Workflow

### 8) Introduction à Windows Presentation Foundation

## Travaux pratiques

Les exercices sont réalisés en langages C# 3.0. Par analogie, les programmeurs VB.NET tireront également parti de ce cours, les librairies et méthodes étant communes aux deux langages.

## 1) Introduction

- Introduction au Framework 3.5.
- Nouveautés de C# 3.0 et VB 9.0.
- L'atelier Visual Studio 2010.

## 2) Le langage Microsoft IntegratedQuery (LINQ)

- Introduction au langage LINQ.
- Architecture et principales classes.
- Construire des requêtes à l'aide du langage LINQ.
- LINQ to Object : C# 3.0 pour LINQ.

## 3) XML avec le framework 3.5

- Traitement des flux XML.
- Sérialisation SOAP et XML.
- DataSet et XML.
- Nouveaux types XML pour LINQ.
- LINQ et espaces de nom.
- Axes LINQ to XML.
- Transformations : LINQ versus XSLT.

## 4) Evolutions de ADO.Net et LINQ To DataSet

- L'architecture ADO.Net.
- DataSets typés et TableAdapters.
- LINQ to DataSet : typage des requêtes LINQ to DataSet.

## 5) LINQ To SQL (Entities)

- Mapping relationnel -> Objet.
- Accès en écritures avec LINQ to SQL.
- Mapping de l'héritage avec LINQ to SQL.
- Sources de données ASP.Net.
- LINQ et ASP.Net : LinkDataSource.

## 6) Développement de services WCF

- Introduction à Windows Communication Foundation.
- Comparer un service Web et WCF.
- Hébergement et sécurité d'un service WCF.
- Routage de messages.
- Opérations asynchrones et callback.
- Identité avec Windows Card Foundation.

## 7) Intégration d'applications Workflow

- Notion de business process, les différentes activités.
- Utilisation d'un service WCF et d'un service Web.
- Gestion des erreurs.
- Communication entre le workflow et l'hôte à l'aide d'une classe de service.

- Workflow hébergé par un service web ou une application ASP.Net 3.5.
- Le trio gagnant : WF + WCF + service Windows.
- Le principe des machines à état.
- Workflows persistants : déshydratation avec SqlWorkflowPersistenceService.

## 8) Introduction à Windows Presentation Foundation

- Langage XAML et techniques de développement code et XAML.
- Interaction avec les Windows Forms existants.
- Applications Windows et Web avec WPF.

Stage pratique de 4 jour(s)  
Réf : LKI

## Participants

Pour les développeurs qui souhaitent découvrir la philosophie des ADO .NET et les techniques associées telles que LINQ, avec des exemples concrets permettant de développer des applications professionnelles basées sur le Framework .NET 4.0/3.5 avec Visual Studio 2010/2008.

## Pré-requis

Bonnes connaissances de C# ou VB.NET.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

22 mai 2012, 4 sep. 2012  
6 nov. 2012

# .NET, accès aux données

## ADO.NET, LINQ

### OBJECTIFS

Cette formation vous permettra de vous familiariser avec les différentes techniques d'accès aux données disponibles en .NET. Les scénarios standard seront présentés au travers d'ateliers pratiques, réalisés en C# ou en VB .NET, dans le contexte d'applications de type Windows Forms, WPF et Web.

#### 1) Introduction

#### 2) Programmer avec les objets ADO .NET

#### 3) Introduction à LINQ

#### 4) LINQ To SQL

#### 5) Entity Framework

## 1) Introduction

- Philosophie et architecture des ADO .NET. Principe des fournisseurs d'accès aux données managées.
- Organisation des classes et des espaces de noms.
- Modèle d'architecture applicative en trois couches.
- Principe du DataBinding en Windows Forms, WPF et en Web.
- Les différents usages de XML en ADO .NET.

### Travaux pratiques

Faire une liaison de données d'une propriété d'un contrôle visuel sur un objet en Windows et en Web.

## 2) Programmer avec les objets ADO .NET

- Gestion d'une connexion (paramétrage, pooling,...).
- Configuration et exécution d'une commande.
- Lecture de données avec un DataReader.
- DataSet non typé/typé : structure, opérations d'édits locales, techniques de synchronisation.
- Techniques d'accès aux données génériques.

### Travaux pratiques

Programmer avec ADO .NET (objets Connection, Command et DataReader) et réaliser une classe d'accès aux données générique. Utiliser un objet DataTable et un DataSet typé dans une application Windows et Web.

## 3) Introduction à LINQ

- Philosophie et architecture de LINQ. Le MRO (Mapping Relationnel Objet).
- Techniques de programmation : types anonymes, méthodes d'extension et expressions lambda.
- LINQ to Objects (opérateurs de requêtage).
- Manipuler du XML (création et requêtage) avec LINQ To XML.

### Travaux pratiques

Requêter une collection d'objets et manipuler un fichier XML.

## 4) LINQ To SQL

- Présentation des avantages et des limitations.
- Utilisation du concepteur Relationnel Objet, avec configuration de procédures stockées.
- Opérations Insert, Update, Delete.
- Résolution des conflits liés aux accès concurrentiels.

### Travaux pratiques

Manipulations de données sur une base SQL Server.

## 5) Entity Framework

- Présentation : architecture, fonctionnalités, mapping.
- Conception, configuration et schéma du modèle de données.
- Approche "Domain Driven Development" avec génération de la base à partir du modèle.
- Opérations de requêtage (LINQ to Entities et Entity SQL) et de mise à jour de données.
- Utilisation des fonctions et des procédures stockées.
- Support des objets POCO (code only) et des types complexes.
- Personnaliser le modèle : associations, projections, héritage de types, manipulations de graphes d'objets.
- Gestion des exceptions et des accès concurrentiels.
- Présentation de l'architecture REST et mise en oeuvre des Services de données ADO .NET.

### Travaux pratiques

Exemples de configuration du modèle d'entités, avec manipulation de données. Création et consommation d'un service WCF basé sur ADO .NET Data Services dans une application WPF.

Stage pratique de 4 jour(s)  
Réf : DOF

## Participants

Développeur en charge de l'accès aux données dans les applications .NET. Chef de projets, architecte logiciel désirant avoir une vue complète des possibilités de la solution Entity Framework.

## Pré-requis

Bonnes connaissances du langage C# et de l'accès aux données avec le Framework .NET. Connaissance de base d'un modèle d'interface utilisateur (WPF, WinForms). Expérience requise.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

19 juin 2012, 9 oct. 2012  
11 déc. 2012

# ADO.NET Entity Framework, maîtrise et optimisation

## OBJECTIFS

*Durant ce stage, vous allez acquérir la compréhension de l'ensemble des services de l'Entity Framework, et verrez comment les utiliser pour simplifier l'accès aux données dans vos applications .NET. Vous apprendrez à perfectionner vos requêtes et à obtenir des temps optimaux d'exécution. Vous découvrirez également comment organiser la maintenance des modèles en regard des bases de données.*

### 1) Introduction

### 2) L'accès aux données avec LINQ

### 3) L'Entity Data Model et les services du Framework

### 4) Le mappage des données, du schéma physique au schéma conceptuel

### 5) L'Entity SQL

### 6) Object Services et LINQ to Entities

### 7) Mettre à jour les entités, la maintenance du modèle

### 8) Le Data Binding et les sources de données

## Travaux pratiques

*Création d'une application fil rouge, suivie et progressivement complétée en fonction des différents sujets traités pendant le cours.*

## 1) Introduction

- L'accès aux données avec l'Entity Framework. Les principes.
- Découverte des outils du Framework.
- Les services de l'Entity Framework.

### Travaux pratiques

*Découverte des outils de Visual Studio et de l'application fil rouge. Création d'un premier modèle de données.*

## 2) L'accès aux données avec LINQ

- Principes et fonctionnement.
- Expressions lambda et méthodes d'extensions.
- Exécution des requêtes.
- Expressions et opérateurs de LINQ.

### Travaux pratiques

*Création de requêtes LINQ à partir du modèle généré sur l'application suivie.*

## 3) L'Entity Data Model et les services du Framework

- Introduction au modèle Objet/Relationnel dans Visual Studio.
- Les services associés au designer dans Visual Studio.
- Historique et comparaison avec d'autres outils.

### Travaux pratiques

*Manipulation du modèle dans le designer.*

## 4) Le mappage des données, du schéma physique au schéma conceptuel

- Principes clés. Le mappage des données. Les schémas utilisés dans les différentes phases.
- La personnalisation du modèle de données.
- Explorer et personnaliser le fichier EDMX.

### Travaux pratiques

*Ajouter des entités au modèle. Modifier le mappage de données. Ajouter des classes partielles au modèle de l'application suivie.*

## 5) L'Entity SQL

- Les différences entre e-SQL et Transact SQL.
- Le e-SQL. Structure et éléments syntaxiques.
- Développer des requêtes e-SQL.
- Exécuter des requêtes e-SQL. Récupérer et exploiter le résultat des requêtes.

### Travaux pratiques

*Comparer les exécutions des requêtes de l'application suivie entre LINQ natif, e-SQL et des procédures stockées dans la base.*

## 6) Object Services et LINQ to Entities

- Explorer les classes disponibles.
- Manipuler l'ordre et le timing de chargement des données à l'exécution.

### Travaux pratiques

*Optimiser les requêtes de l'application en contrôlant l'ordre et le chargement des données.*

## 7) Mettre à jour les entités, la maintenance du modèle

- Insertion, suppression, modification.
- Valider les données, les accès concurrents.
- Mettre à jour le modèle.

### **Travaux pratiques**

*Ajouter la mise à jour des données à l'application suivie.*

## 8) Le Data Binding et les sources de données

- La liaison aux données dans les ASP.NET, avec AJAX.
- La liaison aux données WPF, Silverlight.

### **Travaux pratiques**

*Intégration du modèle de données dans un design pattern MVVM.*

Stage pratique de 4 jour(s)  
Réf : NCA

## Participants

Développeurs .NET voulant appréhender tous les types de communication proposés par le framework.

## Pré-requis

Très bonnes connaissances en programmation C#3 ou VB.NET avec Visual Studio 2005 ou 2008.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

29 mai 2012, 25 sep. 2012  
27 nov. 2012

# .NET 3.5, communications entre applications Web Services, .NET remoting, WCF

## OBJECTIFS

Ce cours vous permettra d'appréhender les possibilités de communication du framework .Net 3.5 et de choisir l'API la plus adaptée. Il traite des API bas niveau comme les Sockets jusqu'aux API haut niveau des services Web en passant par .Net Remoting et les WCF (Windows Communication Foundations).

### 1) Introduction

### 2) Les sockets

### 3) Les services Web ASP.NET

### 4) .NET Remoting

### 5) WCF

## Travaux pratiques

Les exercices sont effectués en C# ou VB.Net suivant le choix des participants.

## 1) Introduction

- Présentation des différentes API.
- La réflexion pour les communications : rôle et présentation de l'API.
- La sérialisation pour les communications.

## 2) Les sockets

- Le protocole TCP/IP, communications mode stream et mode datagram.
- Traitement multithread à l'aide de Select et multithread à l'aide du pool de threads.
- Les NetworkStreams.

### Travaux pratiques

Réalisation d'un serveur pouvant gérer la connexion simultanée de multiples clients.

## 3) Les services Web ASP.NET

- Notion de contrat WSDL et génération de proxies avec WSDL.exe et Visual Studio.
- Services Web avec état à l'aide de la Session.
- En-têtes SOAP.
- Authentification ASP.Net et application aux en-têtes SOAP.
- Les communications asynchrones et le cas particulier des pages ASP.NET.

### Travaux pratiques

Réalisation d'un service Web accessible uniquement aux clients authentifiés. Gestion d'un contexte de client par un en-tête SOAP personnalisé. Réalisation d'une interrogation parallèle de multiples services Web de manière asynchrone.

## 4) .NET Remoting

- Objets sérialisés ou accessibles à distance. Notions de stub et de proxy. Types de canaux.
- Configuration par programmation / par fichier.
- Gestion de la durée de vie des objets par sponsors.
- Sécurité par cryptage et authentification des clients.
- Rôle des CallContexts.
- Problèmes de synchronisation liés aux Windows Forms.

### Travaux pratiques

Réalisation d'un client et d'un serveur simple. Réalisation d'un serveur chat et de clients de type Windows Forms (gestion de problèmes de synchronisation).

## 5) WCF

- Principe et problématiques des architectures orientées services (SOA).
- Comprendre l'architecture WCF en vue d'unifier les techniques de communications.
- Créer et implémenter un contrat de service.
- Programmation et configuration des points de terminaison.
- Configuration et hébergement du service.
- Utilisation de sessions.
- Echanges synchrones et asynchrones.
- Mécanismes de sécurité et de transaction.

### Travaux pratiques

Réalisation d'un client Windows et d'un serveur simple, avec différentes combinaisons de configuration des échanges côté client et côté serveur.

Stage pratique de 3 jour(s)  
Réf : WWF

## Participants

Développeurs et chefs de projets devant mettre en oeuvre des applications faisant appel à des Workflows.

## Pré-requis

Bonnes connaissances du Framework .NET et, soit du C#, soit du VB.NET. Connaissances de base du Web.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

Paris

8 oct. 2012, 17 déc. 2012

# Workflow Foundation, développement .NET d'applications

## OBJECTIFS

*Workflow Foundation permet de développer en .NET des applications mettant en oeuvre une logique métier sous forme de workflows. Ce stage vous montrera comment identifier les cas d'applications, concevoir vos propres workflows, les mettre en oeuvre et les intégrer à vos systèmes.*

### [1\) Introduction à Workflow 4.0](#)

### [2\) Création et exécution d'un workflow simple](#)

### [3\) Utilisation de XAML 2009](#)

### [4\) Hosting de workflows](#)

### [5\) Workflows en tant que services](#)

### [6\) Utilisation et création d'activités](#)

### [7\) Compléments](#)

## 1) Introduction à Workflow 4.0

- Qu'est-ce qu'un workflow ?
- Les workflows dans le framework .NET.
- Présentation du designer et de la création de workflows avec Visual Studio.
- Présentation des différents types de workflows.

### Travaux pratiques

*Illustrations de différents workflow. Cas d'usage.*

## 2) Création et exécution d'un workflow simple

- La structure et le cycle de vie d'un workflow.
- Le concept d'activité. Les principales activités.
- Echange d'informations entre les activités.
- Les Templates. Le Designer. Le débogage.

### Travaux pratiques

*Création d'un workflow simple. Gestion des échanges d'informations avec la console.*

## 3) Utilisation de XAML 2009

- Présentation de la syntaxe XAML.
- Définir des workflows, des applications et des services avec XAML.

### Travaux pratiques

*Exécution d'un workflow pour XAML.*

## 4) Hosting de workflows

- Invoquer un workflow avec WorkflowInvoker.
- Utiliser un modèle d'exécution WorkflowApplication.
- WorkflowService. Créer, configurer et accéder aux propriétés d'un Workflow Service.
- La persistance. Exécuter des workflows sur une longue durée et sauvegarder leurs états.

### Travaux pratiques

*Hosting de workflows longs.*

## 5) Workflows en tant que services

- Implémentation de services avec des workflows.
- Recevoir et consommer des messages.
- Exposition de WorkflowServices.
- Corrélation via en-tête SOAP. Corrélation métier.
- Transactions.

### Travaux pratiques

*Création d'une application contrôlée par un Workflow distant.*

## 6) Utilisation et création d'activités

- La nouvelle bibliothèque d'activités de .NET 4.
- Présentation des différents types d'activités : les collections, les contrôles de flux, les transactions...
- Créer de nouvelles activités : par composition des standards, par dérivation de CodeActivity ou NativeActivity.
- Les activités asynchrones.
- Les bookmarks et la persistance.

### Travaux pratiques

*Création d'activités métier.*

## 7) Compléments

- Management des workflows. Contrôler l'état des workflows. Introduction à AppFabric.

Stage pratique de 3 jour(s)  
Réf : SET

## Participants

Développeurs et chefs de projets en charge de développements .NET.

## Pré-requis

Bonnes connaissances de la programmation en C# ou VB.NET. Connaissance de base en développement .NET d'applications distribuées.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

Paris

3 sep. 2012, 26 nov. 2012

# Sécurité des applications en .NET

## OBJECTIFS

A l'issue de ce cours, vous maîtriserez les mécanismes de sécurité de la plate-forme .NET et serez en mesure de les mettre en oeuvre dans le cadre de développements applicatifs.

### 1) Problématique de sécurité

### 2) Mécanismes de protection dans le CLR

### 3) Cryptage, certificats et signature

### 4) Gestion de l'authentification et des habilitations en .NET

### 5) Mécanismes de sécurité dans le cadre des standards de Web Services

## 1) Problématique de sécurité

- Qu'est-ce que la sécurité ? Positionnement de l'authentification, de la protection, du cryptage.
- Outils de sécurité dans .NET.
- Sécurité d'exécution, authentification, protection des données (cryptage), protection des accès (habilitation).

## 2) Mécanismes de protection dans le CLR

- Protection du contenu des assembly au travers de la signature. Vérification du code.
- Protection de l'exécution des programmes : modèles de protection, configuration du CLR au travers de la configuration XML.
- Déploiement d'une stratégie de sécurité du CLR. Outils de déploiement.
- Stratégie de sécurité et déploiement des applications. Principe d'utilisation des " preuves ". Règles d'exécution selon la provenance des applications. Définition des règles dans les stratégies de sécurité.

## 3) Cryptage, certificats et signature

- Modèles de cryptographie, moteur de cryptographie.
- Classes disponibles en .NET pour la gestion de la cryptographie et des certificats.
- Mécanismes de signatures en .NET, les certificats.
- Mécanismes de cryptage, utilisation en .NET.
- Protection de la communication : exploitation de SSL et HTTPS, gestion du protocole.

## 4) Gestion de l'authentification et des habilitations en .NET

- Principes d'authentification. Accès aux informations d'identité en .NET, les mécanismes d'authentification.
- Gestion de la sécurité à partir des rôles.
- Mécanismes d'impersonnification, accès au jeton d'accès. Transmission du jeton d'accès et impersonnification dans le cas d'architectures client-serveur.
- Gestion des habilitations : mise en place d'un modèle de protection, classes sous-jacentes à la gestion des habilitations. Gestion déclarative ou impérative des mécanismes de sécurité.
- Restriction de l'exécution du programme et vérification des droits d'exécution par le programme.
- Utilisation des environnements de stockage isolés.

## 5) Mécanismes de sécurité dans le cadre des standards de Web Services

- Authentification du client : validation de l'identité, transmission de l'identité entre Web Services.
- Gestion de la confidentialité et de l'intégrité des informations circulant entre le client et le serveur.
- Mécanismes définis dans le cadre de WSS-I. Mise en oeuvre dans .NET.

Stage pratique de 5 jour(s)  
Réf : DSN

## Participants

Développeurs ayant à réaliser des applications utilisant des interfaces système en .NET.

## Pré-requis

Bonnes connaissances de C# ou VB.NET. Connaissances de base des mécanismes de fonctionnement .NET, des réseaux et de l'architecture Windows.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

18 juin 2012, 24 sep. 2012  
26 nov. 2012

# Développement système en .NET

## OBJECTIFS

*Cette formation vous apportera toutes les connaissances nécessaires pour développer des applications de type système dans le cadre du framework .NET.*

### [1\) Le multitâche en .NET](#)

### [2\) Communication réseau avec les sockets](#)

### [3\) Introduction à .NET Remoting](#)

### [4\) Développement de services Windows](#)

### [5\) Manipulation de la base de registre](#)

### [6\) Journal des événements](#)

### [7\) Analyseur de performance](#)

### [8\) Interopérabilité de .NET avec les DLL et COM](#)

### [9\) Mise en œuvre de la sécurité](#)

### [10\) Attributs et réflexion](#)

## 1) Le multitâche en .NET

- Mise en œuvre des threads : cycle de vie.
- Gestion des priorités. Synchronisation entre threads.
- Communication entre threads. Pool de threads.

### Travaux pratiques

*Création de programmes mettant en œuvre les threads et les classes de synchronisation.*

## 2) Communication réseau avec les sockets

- Utilisation directe des sockets TCP et UDP.
- Client/serveur TCP et UDP. Classe UdpClient.
- Les classes WebClient, WebRequest, WebResponse. Gestion synchrone et asynchrone. Proxy.

### Travaux pratiques

*Réalisation d'un client et d'un serveur multithreads. Réalisation d'un client http.*

## 3) Introduction à .NET Remoting

- Architecture : principe et fonctionnement.
- Objets à distance : manipulation des types, écriture du serveur et du client.

### Travaux pratiques

*Réaliser un serveur de "chat" et de clients Windows Forms.*

## 4) Développement de services Windows

- Création de services en .NET. Test et mise au point.
- Gestion de la sécurité. Installation.

### Travaux pratiques

*Utilisation d'un service Windows pour héberger un serveur .Net Remoting.*

## 5) Manipulation de la base de registre

- Fonctionnement de la base de registre.
- Classes liées à la base de registre. Types de données.

### Travaux pratiques

*Utilisation de la base de registre pour sauvegarder les options de fonctionnement du service Windows.*

## 6) Journal des événements

- Utilisation dans le Framework .NET.
- Gestion des événements et surveillance du journal.

## 7) Analyseur de performance

- Principe : compteur, collecte, mise à jour.
- Les compteurs dans les applications. Intérêt.

### Travaux pratiques

*Création de compteurs pour espionner les performances d'un programme multithread.*

## 8) Interopérabilité de .NET avec les DLL et COM

- Rappel sur COM et les DLL natives. Interopérabilité.
- .NET et COM : rôle client et rôle serveur.

### Travaux pratiques

*Appels de composants COM et de DLL depuis .NET.*

## 9) Mise en œuvre de la sécurité

- Les principes de la sécurité sous Windows 2000.
- Les fonctions de gestion de la sécurité.

## 10) Attributs et réflexion

- Métadonnées dans le CLR. Utilisation des attributs.

Stage pratique de 5 jour(s)  
Réf : FRA

## Participants

Architecte logiciels, chef de projets, développeur.

## Pré-requis

Bonne connaissance du langage C#, expérience requise. Connaissances de base sur WPF et Silverlight.

Prix 2012 : 2315€ HT

Eligible DIF

## Dates des sessions

### Paris

18 juin 2012, 8 oct. 2012  
3 déc. 2012

# Framework .NET 4.0, architecture des applications

## OBJECTIFS

*L'architecture a un impact important sur la réutilisabilité d'une application et sur sa capacité à évoluer. Cette formation vous permettra de vous approprier les outils conceptuels et logiciels les plus utiles dans l'environnement du Framework .NET 4.0, pour concevoir efficacement une architecture applicative à la fois robuste et flexible.*

### 1) Introduction

### 2) Les évolutions du langage C#, l'interface utilisateur avec WPF

### 3) Le pattern MVVM

### 4) Prism et le développement d'applications modulaires

### 5) La composition de l'interface utilisateur

### 6) Conclusion

## Travaux pratiques

*Création d'une application fil-rouge pendant les cinq jours, enrichie par les sujets abordés en cours et les démonstrations.*

## 1) Introduction

- Présentation de l'architecture des applications.
- Prism.
- Le pattern MVVM (Model-View-ViewModel).
- Le pattern MVC3 (Model-View-Controller).
- MEF (Managed Extensibility Framework).

### Travaux pratiques

*Découverte de l'intégration des outils dans Visual Studio. Exploration des bibliothèques et de leur installation.*

## 2) Les évolutions du langage C#, l'interface utilisateur avec WPF

- Rappel des évolutions du langage.
- Rappel des expressions Lambda.
- Rappel des méthodes d'extensions.
- Rappel des méthodes partielles.
- LINQ, XAML, WPF et Silverlight.
- La gestion des ressources.
- Les commandes.
- Les contrôles utilisateurs.
- La liaison aux données aux collections observables.

### Travaux pratiques

*Réalisation des couches de base de l'application et l'interface utilisateur.*

## 3) Le pattern MVVM

- Les caractéristiques et les responsabilités.
- Les interactions de classes et les relations avec les autres patterns.
- Tester les applications MVVM.

### Travaux pratiques

*Réalisation de la description du modèle de classes de l'application, ajout d'un projet de test à la solution de l'application.*

## 4) Prism et le développement d'applications modulaires

- Les bibliothèques de Prism.
- Les extensions de Prism.
- L'intégration de MEF comme alternative.
- La navigation entre composants.
- La communication entre composants.

### Travaux pratiques

*Exploration des QuickStarts de Prism et re-modélisation de l'application fil-rouge.*

## 5) La composition de l'interface utilisateur

- Les concepts de conception.
- Les scénarios de conception.
- Les règles de conception.
- Le partage des éléments entre WPF et Silverlight
- Les applications multicibles.
- La découverte de chargement.
- Les dépendances de chargement.

- Les options de chargement.

### **Travaux pratiques**

*Mise en oeuvre de la nouvelle modélisation de l'application fil-rouge.*

## **6) Conclusion**

- Le déploiement des applications modulaires.
- La maintenance des applications modulaires.

Stage pratique de 4 jour(s)  
Réf : TQN

## Participants

Ce cours s'adresse aux développeurs.

## Pré-requis

Bonnes connaissances en programmation C# et du Framework .NET. Expérience requise en développement d'applications .NET.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

5 juin 2012, 16 oct. 2012  
4 déc. 2012

# .NET, développement avec des Design Patterns et des frameworks

## OBJECTIFS

Vous apprendrez à concevoir et développer des applications d'entreprise autour du framework .NET 3.5 et des designs patterns les plus courants, chacun des patterns sera implémenté avec un framework phare de l'écosystème .NET.

### 1) Introduction

### 2) Les tests unitaires

### 3) Le couplage faible dans une architecture n-tiers

### 4) La persistance

### 5) Exposition d'un composant métier dans une architecture distribuée

## 1) Introduction

- Présentation et comparaison des architectures 2-tiers, 3-tiers, n-tiers.
- Rappel sur les delegate et méthodes anonymes, les event, les types génériques, les types nullables.
- Les nouveautés de la version 3 du langage C# : LINQ, lambda expressions, méthodes d'extension, types anonymes.

### Travaux pratiques

Prise en main de l'environnement de développement (Visual Studio 2008).

## 2) Les tests unitaires

- Organisation d'un développement en C#, notion de '#test project'.
- Ecriture de tests unitaires.
- Utilisation des mocks.

## 3) Le couplage faible dans une architecture n-tiers

- Définition d'une architecture à couplage faible grâce aux patterns factory et strategy, gestion du cycle de vie des instances avec les patterns singleton et prototype.
- Mise en place de cette architecture avec le conteneur léger Spring.NET
- Introduction aux autres solutions disponibles : Castle, Unity (framework IOC en développement chez Microsoft)

### Travaux pratiques

Gestion de la création et du cycle de vie des instances dans un contexte Spring.

## 4) La persistance

- Présentation du pattern dao et du gain apporté par les Frameworks de mapping objet/relationnel
- Présentation du projet Hibernate comme framework de mapping objet/relationnel
- Création d'un mapping avec nHibernate
- Manipulation de données persistantes, application du pattern proxy pour isoler notre application de nHibernate.
- Rappel des erreurs à éviter.
- Introduction aux autres solutions disponibles, notamment Microsoft Entity Framework

### Travaux pratiques

Réalisation d'un mapping, écriture de requête HQL.

## 5) Exposition d'un composant métier dans une architecture distribuée

- présentation des enjeux d'une architecture distribuée et de Windows Communication Foundation.
- présentation des avantages du pattern command, mise en place du pattern facade pour réceptionner les commandes.
- mise en oeuvre des patterns chain of responsibility et observer pour orchestrer le traitement des commandes.
- Travaux pratiques : création d'un endpoint WCF pour exposer la facade et la rendre disponible à d'autres applications.

### Travaux pratiques

Création d'un endpoint WCF pour exposer la facade et la rendre disponible à d'autres applications.

Stage pratique de 3 jour(s)  
Réf : IDE

## Participants

Chefs de projets, architectes, développeurs, équipes de déploiement.

## Pré-requis

Bonnes connaissances d'un langage .NET, du C# ou du Visual Basic.NET, de Visual Studio et du Framework .NET. Expérience requise sur des projets de développement .NET.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

### Paris

11 juin 2012, 15 oct. 2012  
17 déc. 2012

# Team Foundation Server 2010 optimiser ses développements .NET

## OBJECTIFS

*Cette formation vous apportera toutes les connaissances pratiques en vue d'industrialiser vos développements d'applications .NET avec TFS 2010. Vous apprendrez à définir l'architecture de vos applications, gérer avec efficacité le code source, piloter les tests, mettre en place une démarche d'intégration continue, et fonctionner en mode projet.*

### 1) Introduction à Visual Studio 2010

### 2) Création et configuration d'un Team Project

### 3) Conception d'architecture

### 4) Définition des tâches avec les Work Items

### 5) Mise en place du planning avec MS Project

### 6) Gestion du code source

### 7) Tests techniques et optimisation

### 8) Campagne de tests fonctionnels

### 9) Intégration continue

### 10) Suivi de projet

## 1) Introduction à Visual Studio 2010

- L'industrialisation des applications.
- La solution Team System.
- L'architecture.

## 2) Création et configuration d'un Team Project

- Collection de projets. Ajouter une nouvelle collection.
- Team Project et les processus.
- Créer un projet avec le Team Explorer. Paramétrage.
- Définition des équipes et configuration de la sécurité.

### Travaux pratiques

*Collection et Team Project : création et paramétrage.*

## 3) Conception d'architecture

- Les outils d'architectures de Visual Studio 2010.
- Exploiter les diagrammes UML.
- Définir et valider l'architecture en couche d'une application.
- Utilisation de l'Architecture Explorer.

### Travaux pratiques

*Créer des diagrammes UML. Exploiter l'architecture générée.*

## 4) Définition des tâches avec les Work Items

- Les principaux concepts.
- Liens typés. Historisation.
- Gestion et personnalisation des Work Items.

### Travaux pratiques

*Création de Work Items et de requêtes personnalisées.*

## 5) Mise en place du planning avec MS Project

- Liaison des Work Items aux données Microsoft Project.
- Mise en place et suivi du planning et des ressources.

## 6) Gestion du code source

- Contrôle, archivage et extraction.
- Gestion des évolutions d'un projet avec les labels et les branches.
- Comparaison de fichiers et résolution de conflits.

### Travaux pratiques

*Mise en place du contrôle de sources. Gestion des conflits.*

## 7) Tests techniques et optimisation

- Tests unitaires, tests web, tests de montée en charge.
- Automatisation des saisies avec une base de données.
- Détection et résolution de problèmes.

### Travaux pratiques

*Mise en place de différents types de tests.*

## 8) Campagne de tests fonctionnels

- Planifier des campagnes de tests avec l'outil Test and Lab Manager.
- Définir des scénarios fonctionnels de tests.
- Exécuter les tests, communiquer avec l'équipe, gérer les anomalies, suivre les évolutions, reporter.

### **Travaux pratiques**

*Définition d'une campagne de tests et exécution.*

## 9) Intégration continue

- Team Foundation Build. Workflow de build.
- Génération automatisée d'un cadre de validation.

## 10) Suivi de projet

- Utilisation et personnalisation du site d'équipe.
- Mise en place d'un tableau de bord. Reporting.

Stage pratique de 4 jour(s)  
Réf : TSF

## Participants

Chefs de projets. Testeurs.  
Développeurs ou n'importe quel membre d'une équipe de développement qui souhaite améliorer la qualité de son travail au quotidien.

## Pré-requis

Connaissances de base du développement logiciel et des processus de tests.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

11 sep. 2012, 4 déc. 2012

# Visual Studio 2010 TSF 2010, mise en oeuvre des tests

## OBJECTIFS

*Cette formation vous apprendra à mettre en place une stratégie de tests logiciels au moyen des solutions Visual Studio et Team Foundation Server 2010 de Microsoft. Elle vous permettra également de découvrir et de réaliser des tests unitaires, des tests Web et des tests de charge, et de prendre en main les outils dédiés aux tests fonctionnels.*

### 1) Introduction à Visual Studio 2010 ALM

#### 2) La gestion de projets

#### 3) Les tests fonctionnels

#### 4) Les tests unitaires

### 5) Les tests Web et les tests de performance

#### 6) Les tests automatisés d'interfaces graphiques

#### 7) La gestion des environnements de tests

#### 8) Suivre l'avancement de ses tests

## 1) Introduction à Visual Studio 2010 ALM

- Introduction à Team Foundation Server 2010 et Visual Studio 2010.
- L'ALM (Application Lifecycle Management) chez Microsoft. Les principes.
- La gestion du processus de développement de la conception au déploiement.
- La gamme de produits.

## 2) La gestion de projets

- Les grands principes.
- La création d'un projet d'équipe.
- La gestion des exigences avec les différents outils.

### Travaux pratiques

*Mise en place d'un projet au démarrage.*

## 3) Les tests fonctionnels

- Les principes. L'organisation.
- Créer un plan de test.
- Créer un cas de test.
- Configurer l'exécution d'un plan de test.
- Exécuter manuellement une campagne de test.
- Reporter une anomalie aux développeurs.

### Travaux pratiques

*Mise en place d'un plan de test sur un projet et son exécution.*

## 4) Les tests unitaires

- Comprendre les tests unitaires. Principes, usage et fonctionnement.
- Créer un test unitaire. Comment faire ? Que tester ?
- La liaison à une source de données.
- Les tests impactés et la couverture de code.

### Travaux pratiques

*Mise en place de tests unitaires pilotés par les données.*

## 5) Les tests Web et les tests de performance

- Le cycle d'un test de charge. Les différentes étapes.
- Créer un test Web. Les éléments clés.
- Créer un test de charge. Que tester ? Que mesurer ?
- L'exécution d'un test de charge. Comment suivre cette exécution.

### Travaux pratiques

*Mise en place d'un test de charge.*

## 6) Les tests automatisés d'interfaces graphiques

- Les principaux concepts.
- Créer un test d'interface graphique basique.
- Valider l'interface graphique.
- Extensibilité et code personnalisé.

### Travaux pratiques

*Mise en place de tests automatisés d'interfaces graphiques.*

## 7) La gestion des environnements de tests

- Les principaux concepts. Les bonnes pratiques.
- Installer et configurer un environnement de tests.

- Utiliser un environnement de tests.
- Découvrir le lab management.

#### **Travaux pratiques**

*Mise en place d'un agent de tests.*

### 8) Suivre l'avancement de ses tests

- Concepts de reporting.
- Utiliser les rapports de base.
- Créer et faire évoluer les rapports Excel.

#### **Travaux pratiques**

*Création des rapports de différents types. Mise en forme de base.*

Stage pratique de 3 jour(s)  
Réf : TDN

## Participants

Ce stage s'adresse à des ingénieurs ou chefs de projets en développement logiciel.

## Pré-requis

Connaissances de base de la programmation objet avec .NET.

Prix 2012 : 1730€ HT

Eligible DIF

## Dates des sessions

### Paris

11 juin 2012, 10 sep. 2012  
26 nov. 2012

# Test Driven Development en .NET

## OBJECTIFS

Ce cours pratique vous apportera toutes les connaissances nécessaires pour développer vos applications .NET autour des meilleurs outils et pratiques de tests unitaires et de recettes. Vous apprendrez à intégrer le test dans votre cycle de développement, le refactoring, la conception émergente, les pratiques agiles, à implémenter des Mock et à utiliser le Test Driven Requirement ou le Behaviour Driven Development.

### [1\) Introduction](#)

### [2\) Définition et principes du TDD](#)

### [3\) Les meilleures pratiques de développement agiles](#)

### [4\) Techniques avancées avec le TDD](#)

### [5\) Les objets Mock et Stub](#)

### [6\) Techniques d'écriture de tests](#)

### [7\) Couverture des tests](#)

### [8\) Test de code hérité](#)

### [9\) Les outils](#)

### [10\) Les qualités du TDD](#)

## 1) Introduction

- Origine du TDD.
- L'agilité et les tests.
- Les limitations du débogueur.
- Etat de l'art.
- Industrialiser, c'est tester !
- Le besoin de tester : les types de tests.

## 2) Définition et principes du TDD

- Théorie.
- La bibliothèque NUnit.
- Les 3A.
- Gestion des Exceptions.
- Refactoring.
- Conception émergente.
- Test Driven Design.
- Gestion des scénarii.

### Travaux pratiques

Conception et intégration de tests dans le cycle de développement d'un projet.

## 3) Les meilleures pratiques de développement agiles

- Gestion des conditions " limites ".
- Framework de tests.
- TDD et gestion des données SGBDR.
- TDD et gestion des données de type Fichier et Repository.
- TDD et gestion des interfaces graphiques.

### Travaux pratiques

Mise en oeuvre des différentes pratiques.

## 4) Techniques avancées avec le TDD

- Corriger des anomalies.
- Gérer la montée en charge et la sécurité des produits NTIC.
- Gestion de la sécurité logicielle.
- Gestion de la performance.
- Stress tests.

### Travaux pratiques

Gestion des anomalies. Tests de performance.

## 5) Les objets Mock et Stub

- La théorie.
- Application de la théorie sans utiliser de bibliothèque.
- Découverte des bibliothèques du marché.

### Travaux pratiques

Utilisation des objets Mock.

## 6) Techniques d'écriture de tests

- Fixtures.
- Qualités d'un code de test.
- Tests basés sur la responsabilité.

- Tests basés sur l'implémentation.
- Styles de TDD.

#### **Travaux pratiques**

*Amélioration de la qualité des tests écrits.*

### 7) Couverture des tests

- La bibliothèque MSTest et ses principales fonctionnalités.
- Les axiomes sur la couverture des tests.
- Types de couverture.
- Combien de tests faut-il écrire ?
- Tests en présence d'héritage.
- Tests de régression.
- Outils de couverture.

#### **Travaux pratiques**

*Définition d'une politique de couverture de tests. Application.*

### 8) Test de code hérité

- Qu'est-ce que du code hérité ?
- Cycle d'évolution du code hérité.
- Tests fonctionnels avec Fit et FitNesse.
- Tests fonctionnels et TDD.
- Écriture de tests fonctionnels exécutables avec FitNesse.

### 9) Les outils

- Les outils Open Source et commerciaux.
- Architecture matérielle de tests.
- L'intégration continue (SVN, CVS, Sourcesafe).
- Intégrateur continu (parmi lesquels, CruiseControl).
- Couverture de test.
- Etude d'un outil de gestion des tests et de communication entre MOA et MOE : Fitnessse.

#### **Travaux pratiques**

*Mise en oeuvre de plusieurs outils.*

### 10) Les qualités du TDD

- Gains du TDD.
- Description des cas d'utilisation.
- Description des différents scénarii.
- Intérêt de l'industrie, des NTIC et des autres R&D.
- Les nouveaux enjeux industriels.

Stage pratique de 4 jour(s)  
Réf : OPA

## Participants

Développeurs, ingénieurs,  
architectes, chefs de projet.

## Pré-requis

Bonnes connaissances du  
langage C#.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

19 juin 2012, 9 oct. 2012  
4 déc. 2012

# Optimisation d'applications en .NET

## OBJECTIFS

*Ce cours vous propose une méthodologie pour améliorer l'efficacité de vos applications .NET. Il vous apprendra à utiliser les principaux outils d'analyse de performances et vous permettra de maîtriser les différentes techniques d'optimisation en matière d'architecture et de codage.*

### 1) Introduction

### 2) Fabrication des codes IL et machine

### 3) Les constructions simples du langage

### 4) Le moniteur de performances

### 5) Le ramasse-miettes GC

### 6) Optimisation et bibliothèques externes

### 7) Communications interdomaines

### 8) La gestion des threads

### 9) Paralléliser les traitements

### 10) Optimisation de la phase de démarrage

### 11) Conception et architecture des programmes

## 1) Introduction

- Optimiser : quoi, comment, quand ? Objectifs.

## 2) Fabrication des codes IL et machine

- Efficacité intrinsèque des différents langages.
- Utiliser C++ (managé ?) pour les parties sensibles ?
- Etapes de la fabrication : compilateur et CLR.

### Exercice

*Instrumentation du code C#, C++ et C++ managé.*

## 3) Les constructions simples du langage

- Traitement des chaînes de caractères.
- Performance des classes et des structures.
- Exceptions. Delegates et lambdas. Interfaces et fonctions virtuelles.

### Exercice

*Exercice et démonstration sur les éléments de codage.*

## 4) Le moniteur de performances

- Fonctionnement. Anciens compteurs toujours utiles, nouveautés .NET, compteurs personnalisés.

## 5) Le ramasse-miettes GC

- La gestion de la mémoire. Les "finaliseurs", le pattern Dispose. Utilisation d'un profileur et de SOS.DLL.

### Exercice

*Détection et correction des fuites de mémoire.*

## 6) Optimisation et bibliothèques externes

- Portage depuis COM ou Win32, coût de la réutilisation. Accès aux DLL système ou utilisateur : Interop et P/Invoke. Le marshaling et son coût.

### Exercice

*Comparaison C# et P/Invoke versus C++ mixte.*

## 7) Communications interdomaines

- Domaine d'application et .NET Remoting : rappels.
- Communication entre domaines : d'un même processus, de processus différents.
- Mesure de performances. Les règles à respecter.

## 8) La gestion des threads

- Optimiser les traitements concurrents et leur synchronisation. Interblocages. Utilisation du pool de threads ou création de threads. Les appels asynchrones.

## 9) Paralléliser les traitements

- La bibliothèque de parallélisation de Microsoft. Nouvelles classes de System.Threading. Synchronisation.

## 10) Optimisation de la phase de démarrage

- Agir à ce niveau ? Chargement des assemblages, initialisations, précompilation avec NGEN.

## 11) Conception et architecture des programmes

- Une synthèse et quelques conseils.
- Bien concevoir ses classes.
- Gestion de cache pour les données.
- Utilisation de composants distants : le pattern "Facade".
- Processus différents ou Application Domains ?

Stage pratique de 4 jour(s)  
Réf : PMU

## Participants

Développeurs, ingénieurs,  
architectes, chefs de projet.

## Pré-requis

Bonnes connaissances en  
programmation C#.

Prix 2012 : 2150€ HT

Eligible DIF

## Dates des sessions

### Paris

12 juin 2012, 2 oct. 2012  
4 déc. 2012

# C#, programmation multithread

## OBJECTIFS

*Cette formation vous montrera les avantages du traitement multithread ainsi que la manière de le mettre en oeuvre de manière efficace en C#. Vous verrez comment gérer le partage de la mémoire, les problèmes de synchronisation, les pools de threads, ainsi que les meilleures pratiques de développement.*

### [1\) Introduction](#)

### [2\) Les outils de base du framework](#)

### [3\) Les classes de synchronisation du framework](#)

### [4\) Le pool de threads](#)

### [5\) L'utilisation d'exécutions asynchrones](#)

### [6\) L'instrumentation et le profilage du code](#)

### [7\) Conception et architecture des programmes](#)

## 1) Introduction

- Plusieurs threads : pourquoi ? Les objectifs.
- Deux modèles de multithreading : concurrent/parallèle.
- Difficultés et défis de la programmation multithread.

## 2) Les outils de base du framework

- Qu'est-ce qu'un thread ? Composants d'un thread.
- Le partage de la mémoire par les threads.
- La classe Thread du framework (version 2 et sup).
- Le besoin de synchronisation et l'instruction lock.

### Travaux pratiques

*Création de thread. Synchronisation. Influence du nombre de processeurs.*

## 3) Les classes de synchronisation du framework

- Les objets de synchronisation historiques de Windows : Event, Mutex, Semaphore.
- Les classes équivalentes du framework.
- Les sections critiques de Windows et la classe Monitor du framework.
- Le danger d'étreinte fatale et sa détection.
- Les variables conditionnelles et la classe Monitor.
- La classe ReadWriteLock La classe Interlocked.

### Travaux pratiques

*Synchronisation de deux threads, producteur et consommateur de données. Utilisation des classes.*

## 4) Le pool de threads

- La ThreadPool et ses deux catégories de threads.
- Le paramétrage du nombre de threads.
- Utilisation explicite et implicite de threads de travail.

### Travaux pratiques

*Usage explicite de threads du pool.*

## 5) L'utilisation d'exécutions asynchrones

- La mécanique Windows sous-jacente.
- Le design pattern Begin/End et les classes Stream ou autres concernées. Le design pattern Async du framework 3.x pour les sockets.
- L'asynchrone dans les ASP.NET.

### Travaux pratiques

*Ecriture d'un programme utilisant des appels asynchrones.*

## 6) L'instrumentation et le profilage du code

- La compétition pour les données, l'oubli de synchronisation et sa granularité, les lectures/écritures mémoire cassées, le réordonnement des instructions.
- L'utilisation d'outils de diagnostics pour juger de la pertinence de l'utilisation de threads.

### Travaux pratiques

*Mise en évidence de l'impact de certains choix de design d'application.*

## 7) Conception et architecture des programmes

- Penser/concevoir multithread et parallèle.
- Threads et interface graphique.
- Thread Vs AppDomain.

Stage pratique de 3 jour(s)  
Réf : MUC

### Pré-requis

Bonnes connaissances d'un des 3 langages objet utilisés dans le cours (Java, C#, C++). Connaissances de base des concepts liés aux applications multicore.

Prix 2012 : 1650€ HT

Eligible DIF

### Dates des sessions

#### Paris

18 juin 2012, 8 oct. 2012  
17 déc. 2012

# Programmer vos applications en Multicore

## OBJECTIFS

*Ce cours vous permettra de comprendre les architectures Multicore et leur programmation : les techniques de mise en œuvre d'une approche multithread ou multiprocessus et les langages dédiés à la programmation parallèle. Vous verrez également les contraintes en matière de synchronisation d'accès aux données et les précautions à prendre.*

### 1) Introduction

### 2) Modélisation des applications

### 3) Threads

### 4) Processus

### 5) La programmation parallèle

### 6) Synthèse et conclusion

## Travaux pratiques

*L'outil Open source BOUML sera utilisé pour la modélisation UML, Visual Studio express(tm) pour la manipulation des langages C# et C++, NetBeans(tm) 6.x pour Java. Les exemples seront démontrés grâce aux langages UML, Java, C++ et Dot Net.*

## 1) Introduction

- Les enjeux de la programmation Multicore dans les années à venir.
- Tableau général des technologies utilisables : processus, threads et parallélisme.
- Description et rappel du fonctionnement d'un processeur.
- Les architectures en " Hyperthreading ".
- Les architectures des processeurs INTEL™ et AMD™.
- Les architectures NVidia™ et API™.
- Les aspects synchronisation à prévoir : cas généraux.

## 2) Modélisation des applications

- Importance des aspects modélisation.
- Parallélisation des traitements (ex : calcul).
- Utilisation des mécanismes asynchrones : processus, threads...
- Développer une nouvelle application : précautions et modélisation.
- Eviter les " singletons ".
- Modifier une application existante en Multicore : problèmes rencontrés.
- Choix d'architecture : un compromis synchronisation et performance.
- Choix multiprocessus/multithreads.

## 3) Threads

- Organisation d'un système à base de threads.
- Les threads dans les systèmes et les langages (exemple : Java, .NET et C++).
- Apport des threads dans une application industrielle.
- Ordonnement des threads dans les systèmes : technique du " round robin ".
- Gestion des stacks et " call stack " dans les threads.
- Les débogueurs multithreads : ex Visual Studio et NetBeans™.
- Gestion des objets de synchronisation : sections critiques, Mutex et Sémaphores.
- Développer " Thread safe ".
- Règles pour développer en approche multithread.
- Les API de threads avec Windows, Java et DOT Net.
- Les API de POSIX.

### Travaux pratiques

*Threads et synchronisation en DOT Net, Java et C++.*

## 4) Processus

- Espaces d'adressage des processus, organisation.
- Critères de choix d'une approche multiprocessus.
- Ensemble des techniques de communication interprocessus (IPC).
- Techniques de debugging multiprocessus, nouveaux outils (ex : Visual Studio 2005).
- Avantage et inconvénients des techniques multiprocessus.
- Particularité : les " Domain " DOT Net.

### Travaux pratiques

*Gestion de traitements asynchrones avec l'API Windows.*

## 5) La programmation parallèle

- L'apport et les objectifs de la programmation parallèle.
- " Parallel FX " la bibliothèque DOT Net pour paralléliser.
- Architecture PFX et philosophie.

- Composants TPL et PLINQ.
- La librairie " OpenMp " C++.
- Directives " OpenMP " disponibles.
- Utiliser les GPU des cartes graphiques pour le calcul.
- Les kits de NVidia (CUDA) et ATI.
- Exemple d'applications utilisant les bibliothèques.
- Exemple de code et commentaires.

#### **Travaux pratiques**

*Paralléliser des algorithmes avec PFX C#. Paralléliser des algorithmes avec " OpenMP " en C++.*

### **6) Synthèse et conclusion**

- Conclusion des techniques étudiées.
- L'avenir de C++ avec le multicore.
- Synthèse des approches de Design, threads, multiprocessus et programmation parallèle.