Hors-Série .Net

Programmez !

www.programmez.com

Décembre 2007 Janvier 2008 - Hors-Série

LE MAGAZINE DU DÉVELOPPEMENT

Plongée au cœur de .Net 3.5

Allez plus loin avec Expression

Silverlight Streaming par la pratique, développement WPF dans Blend, découvrir Expression Design et Expression Web

Métier Intégrateur : nouveau métier du web ?

Outils L'usine à logiciels selon .Net

Office Excel Services : la nouvelle face d'Excel !

Linux Mono : le .Net open source

M 02104 - 9 H - F: 4,95 € - RD

Printed in France - Imprimé en France - BELGIQUE 5,45 € - SUISSE 10 FS - LUXEMBOURG 5,45 € - Canada 7,45 $ CAN - DOM Surf 5,90 € - TOM 780 XPF - MAROC 50 DH

COMSOFT direct

Bechtle's Software Specialist

SOS DEVELOPERS

Comsoft-SOS Developers votre revendeur conseillé en outils de développement vous présente sa sélection de logiciels de développement de décembre.

Soyez détendu !

Comsoft-SOS Developers recherche pour vous les logiciels qui simplifient vos développements.

Microsoft® Visual Studio® 2008

Déjà disponible !

dans votre abonnement Visual Studio avec MSDN

PROMOTION !

15% de remise supplémentaire

*jusqu'au 28 décembre 2007**

Visual Studio avec MSDN Pro : 520 €HT par an et par développeur

Visual Studio avec MSDN Premium : 1185 €HT par an et par développeur

*** NOUS CONSULTER POUR LES DÉTAILS DE LA PROMOTION**

Flare 3.0

Créez vos aisément vos aides

- Outils XML-based pour vos aides et documentations.

Mise à jour depuis RoboHelp : 409 €HT

ReSharper 3.0

Plug-in Visual Studio

- Analyse de code, test unitaire...
- Supporte C#, VB, .NET, XML, XAML, ASP.NET

Resharper for C# : 185 €HT

Incredibuild

Compilation accélérée de votre code C/C++

- Fait tourner le compilateur de Visual Studio selon un processus de distribution contrôlé

Incredibuild VS .NET/2005/6 : 345 €HT

Intel Compiler Suite 10.1

Compilateur C++ et Fortran

- La nouvelle suite contient le compilateur Intel C++ et Intel Fortran
- Contient Performance Libraries

Compiler Suite for Windows : 685 €HT

JGear Performance for Eclipse

Débogage et profiling pour Eclipse

- Découvrez instantanément les problèmes de performances
- Vue temps réel

JGear Performance : 239 €HT

Teamprise

Faites collaborer .NET et Eclipse

- Accédez à Microsoft Visual Studio Team Foundation server depuis un IDE qui n'est pas Visual Studio.

Teamprise 01 utilisateur : 435 €HT

Comsoft - SOS Developers : Tel : 0825 07 06 07

infos@sosdevelopers.com - www.sosdevelopers.com

SOMMAIRE HORS-SÉRIE

> Actus
4

> Cas Client
Le surf se met à Silverlight 8

> Evénements
TechEd 2007 9

> Découverte
Plongez au cœur de .Net 3.5 12

> Dossier : Expression
Allez plus loin avec Expression 18
Retour d'expérience sur Expression Design et Blend 19
Construire une application WPF avec Expression Blend et VS 2008 20
Expression Design : un nouvel outil de création graphique 22
Expression Blend : orchestrez vos applications 25
Silverlight avec Blend 2 et Visual Studio 2008 28
Silverlight Streaming par la pratique 31
Expression Web au banc d'essai 33

> C#
Utiliser Diagrammer for.Net 35

> Office 2007
Présentation des Excel Services 38

> Architecture
Vous avez dit Softwares Factories ? 41

> Développement Web
IIS 7 : le nouveau serveur web de Microsoft 36

> Carrière
Intégrateur WPF : un nouveau métier 46

> Linux
.Net sur Linux avec Mono 47

> Sécurité
La cryptographie en .Net 49

Édito

Le futur est déjà là !

Microsoft a dévoilé il y a quelques semaines une partie importante des nouveautés pour 2008. Et elles promettent beaucoup, que ce soit dans la synchronisation de données, la base de données, le travail en équipe, le développement web et mobile, les services en ligne. Dès à présent, d'importantes fondations sont disponibles : Visual Studio 2008 et .Net 3.5.

Cette nouvelle version du framework consolide les fondations apparues dans .Net 2.0 et fait évoluer les librairies propres à .Net 3.0. Avec la v3.5, nous arrivons désormais à une réelle maturité avec des évolutions dans les langages et surtout avec l'introduction d'un nouveau langage de requêtage, Linq, ouvrant la voie à une nouvelle manière de manipuler et d'interroger les données d'une base de données, directement dans le code de son application. Nous allons plonger au cœur de ce framework pour mieux le connaître.

Dans ce nouveau numéro de notre hors-série .Net, nous vous proposons un dossier approfondi sur la gamme Expression, les nouveaux outils de création et de développement web, disponibles depuis cet été. Nous allons explorer quelques unes des nombreuses possibilités de ces environnements avec Windows Presentation Foundation et Silverlight, pour finir sur un service qui ne manquera pas de faire parler de lui : Silverlight Streaming !

Ce numéro vous étonnera par la richesse du monde .Net. Outre l'actualité très abondante, nous allons explorer les Excel Services ou comment transformer son bon vieux tableur en de multiples services accessibles dans ses applications. Nous verrons aussi comment et pourquoi .Net permet de faire de véritables usines à logiciels… Nous aborderons aussi IIS 7, une introduction à Mono ou encore les mécanismes du cryptage.

Bon codage et rendez-vous aux TechDays 2008 !

François Tonic
Rédacteur en chef

Programmez!
LE MAGAZINE DU DÉVELOPPEMENT

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - abonnements.programmez@groupe-gli.com Tél. : 01 55 56 70 55 - Fax : 01 55 56 70 20 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30.

Tarifs abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 45 € - Etudiant : 39 € - CEE et Suisse : 51,83 € Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € Tom : 79,61 € - Dom : 62,84 € Autres pays : nous consulter. **PDF : 30 € (Monde Entier) souscription en ligne.**

Rédaction : redaction@programmez.com
Directeur de la Rédaction : Jean Kaminsky
Rédacteur en Chef : François Tonic
Ont collaboré : M. Perfetti, E. Groise, A. Leblond, G. André, J. Joubert, S. Picamelot, A. Pagnier, P. Sentenac, S. Souane, X. Vanneste, M. Mézil, T. Lebrun, C. Durand, M. Benlmostapha.
Photo de couverture : ©Tim De Waele/Corbis
Maquette : AJE Conseils
Publicité : Régie publicitaire, K-Now sarl - Pour la publicité uniquement : Tél. : 01 41 77 16 03 - coordination@programmez.com
Editeur : Go-02 sarl, 6 rue Bezout - 75014 Paris Coordination@programmez.com - Dépôt légal : à parution - Commission paritaire : 0707K78366 - ISSN : 1627-0908 - Imprimeur : ETC - 76198 Yvetot
Directeur de la publication : Jean-Claude Vaudecrane

Prochain numéro : 8 février 2008
✓ Gamme 2008 : SQL Server, Windows Server, Visual Studio
✓ Spécial TechDays

WEB

Silverlight 2.0 remplace Silverlight 1.1 !

Microsoft vient de lancer officiellement Silverlight 2.0. Quid de la v1.1 ? Elle passe à la trappe. Il faut avouer qu'elle semait le trouble depuis quelques mois dans les esprits. En passant directement à la 2.0, Silverlight gagne en visibilité. La première version bêta doit apparaître durant le premier trimestre 2008 (pour MacOS X et Windows). Cette version doit combler les vides fonctionnels de la 1.0. On notera notamment :

- WPF UI Framework : doit étendre considérablement l'interface graphique grâce à ce nouveau framework.
- Support réseau : Silverlight 2.0 aura le support de REST, RSS, WS*communication, POX.
- Support de librairies .net plus large : LINQ to XML, HTML DOM API, collections, génériques, etc.
- Sans oublier le support .Net, et la nouvelle DLR.

Bref, du très lourd pour le développeur web. D'autre part, Expression 2.0 devrait pouvoir tirer parti partiellement, ou totalement de Silverlight 2.0.

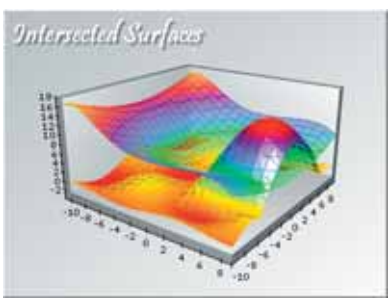
PROJET

Acropolis passe dans le giron .net

Le projet Acropolis est un toolkit destiné à développer et construire des applications composites. Des pré-versions étaient disponibles depuis plusieurs mois. Désormais, le projet ne sera plus accessible en indépendant. Il a rejoint fin novembre le .Net Framework. Acropolis sera entièrement ou partiellement inclus dans les prochaines versions du framework et aussi dans Silverlight. Si les CTP restent accessibles, elles ne seront plus mises à jour, mais vous pourrez toujours apporter du feedback aux équipes. Celle d'Acropolis rejoint l'équipe Patterns & Practices. Pour en savoir plus sur les applications composites dans .Net : http://blogs.msdn.com/gblock/archive/2007/10/26/wpf-composite-client-guidance-it-s-coming.aspx

COMPOSANT

L'éditeur Nevron peaufine votre interface !



Vous connaissez sans doute ChartFX, ComponentOne ou encore Infragistics, mais vous ne connaissez peut être pas Nevron, un éditeur de composants graphiques. Nevron propose pas moins de 6 packages de composants, principalement pour le monde .Net. L'une des offres les plus intéressantes est Nevron User Interface for .Net. Il s'agit de composants professionnels et extensibles pour vos applications .Net. L'interface est facilement intégrable dans son code. On dispose de docking panel, de barres d'outils, de boîtes à outils personnalisables, etc. Fonctionne pour le moment uniquement avec Visual Studio 2005. On dispose aussi de packages pour les graphiques et diagrammes. L'offre composants est vraiment diversifiée et surtout de qualité. À découvrir d'urgence !

Site : nevron.com

OUTIL

PowerBuilder aime aussi .net

L'éditeur Sybase continue à implémenter .Net au cœur de sa solution de développement, PowerBuilder. La version 11 favorise encore un peu plus le support de .Net : support de Vista, compilation incrémentale pour le framework .Net, Datawindow .net amélioré. Sur Datawindow, on peut maintenant définir comme source de données un web service et on bénéficie d'une intégration plus fine avec Visual Studio 2005.

ARCHITECTURE

Scort intègre .Net et Biztalk

L'éditeur Scort propose depuis peu SOA Fasttrack pour .Net. Un environnement pour réutiliser la logique métier mainframe dans des applications front office basées sur .Net et les services Biztalk. Microsoft BizTalk Server 2006 R2 s'appuie désormais pleinement sur les WCF pour intégrer tout type de processus et de service web. Entièrement développé en C++, SCORT SOA Fastrack pour.NET exploite ces nouvelles capacités pour fournir un accès direct à la logique métier des services Mainframe, sans faire appel aux couches de présentation 3270, sans modifier le code du Mainframe, ni exiger l'intégration d'un composant Java intermédiaire.

CLUB

Le Club Visual Studio Team System vous sonde !

La gestion de projet est souvent mal comprise ou mal utilisée, beaucoup de sociétés utilisent des programmes pour gérer leurs développements, des tableaux de bord, ou encore des rapports. Le tout fonctionne très souvent autour d'une ou plusieurs méthodologies. Le club VSTS a mis en ligne un premier sondage pour permettre de faire un bilan sur ces outils et méthodologies.

Lien : http://www.prosondage.com/survey.aspx?GUID=36a6cc74-159e-4b54-9952-3fc2d47e8f58

Trois personnes seront tirées au sort et se verront remettre un livre : **Writing Secure code, Asp.net WebMatrix, Programming ADO.net 2.0 Applications.** Le sondage est en ligne jusqu'au 31 Décembre 2007.

PROJET

Le projet Dagba pour Team Foundation Server (TFS)

Le projet Dagda a été imaginé pour permettre l'administration de projets TFS depuis une interface web comme le fait déjà Team System Web Access, mais Dagda ne s'arrête pas là. Il sera possible d'ajouter des fonctionnalités de Dagda directement dans vos portails SharePoint, mais aussi des interfaces basées sur Silverlight 2.0 ou sur des Pocket PC (elles sont en cours de réflexion). Dagda est un portail implémentant son propre système de Web Part : il permet aux développeurs de créer leurs propres modules et de les ajouter au portail, de configurer un système de permissions basé sur les groupes de TFS.

Site : http://www.codeplex.com/dagda/Thread/List.aspx

Le rendez-vous incontournable des développeurs et professionnels de l'informatique !

Microsoft® TechDays2008

L'innovation avance avec vous

Retrouvez toutes les innovations informatiques présentes et à venir réunies dans un événement unique et plus que jamais incontournable.

Édition Spéciale Lancement

Windows Server® 2008

Microsoft® SQL Server™ 2008

Microsoft® Visual Studio® 2008

11, 12, 13 février 2008

au Palais des Congrès de Paris

Circuit recrutement Microsoft et partenaires !

À vous de sélectionner votre parcours :

22 parcours techniques

220 et plus sessions sur tous les produits, toutes les solutions Microsoft

... Datacenter I Sécurité I Développement web I Décisionnel I Windows Vista I SharePoint...

msdn Microsoft TechNet

Inscrivez-vous, c'est gratuit !

www.microsoft.com/france/mstd08

Microsoft France - RCS Evry B327 733 184 - WUNDERMAN

LANGAGE

XAML arrive sur Java !

Qui a dit que XAML était réservé au monde Windows, .Net ? La société Soyatec propose une approche originale et surprenante : rendre disponible XAML aux développeurs, aux applications Java ! Voilà une idée qui mérite votre regard ! eFace est un framework de présentation se voulant neutre et indépendant. Il utilise une grammaire XML et toute la logique est compatible XAML. Le coeur de la plate-forme se nomme Universal Presentation Framework, et a l'ambition d'être compatible avec WPF de .Net 3. Et surtout, il est entièrement Java. Il possède des composants afin de favoriser la réutilisation. A partir d'un seul code, on peut cibler plusieurs technologies de présentation : SWT, Swing, etc. Et il est possible, selon l'éditeur, de partager un même fichier XAML pour l'interface entre son application Java et .Net ! On peut coder via Eclipse ou NetBeans. Actuellement, seul le rendu avec SWT est disponible dans la première version et elle supporte Vista, XP, MacOS X et Linux.

GÉNÉRATION

CodeFluent en version 1.5

CodeFluent 1.5 est désormais disponible depuis fin octobre 2007. Cette nouvelle version apporte diverses améliorations et intègre un producteur natif 'Windows Communication Foundation' pour la fabrication d'applications de type client intelligent. Par ailleurs, SoftFluent a lancé le développement de la version 2.0 intégrant un concepteur graphique de modèles, avec une cible de lancement à mai 2008. Une version pré-commerciale sera présentée lors des TechDays 2008, l'événement technologique de Microsoft en février.

ARCHITECTURE

Symposium DNG 2008 est lancé !

Bonne nouvelle. Après plusieurs années de report, la dernière édition remontant à 2005, le Symposium Dot Net Guru se déroulera en 2008 durant les TechDays qui auront lieu en février prochain à Paris. L'architecture sera une fois de plus à l'honneur, même si rien n'est encore totalement arrêté sur le contenu et les sessions, entre autres : le Domain Driven Design avec .Net, le projet Volta, Linq, le ROA. L'agenda devrait être disponible courant janvier.

MOBILITÉ

Windows CE 6r2 disponible

Si Vista et Windows Server 2008 occupent les esprits, Microsoft travaille activement sur la mobilité et l'embarqué. Cette release 2 offre des nouveautés intéressantes comme le WSD (Web Services on Devices), un sous-ensemble de WS-*. WSD doit faciliter les connexions entre appareils. Vista le supportait déjà. Les appareils sont vus comme des services et ils peuvent être soit client, soit serveur. On fait en quelque sorte du Machine to Machine. Pour la découverte " réseau ", on dispose de DPWS et de WS Discovery. Les scénarios d'utilisation sont nombreux. On peut imaginer des capteurs équipés de WSD, qu'un appareil sous Windows CE pourrait voir et utiliser. La VoIP a été améliorée dans son support (audio en triple voix, vidéo téléphonie). La r2 se présente sous la forme d'un add-on à CE 6. Actuellement, le kit de développement est disponible uniquement pour Visual Studio 2005.

Introduction à WCF

Difficulté : • • • •
Editeur : *O'Reilly*
Auteur : *Michele Leroux Bustamante*
Prix : *45 €*

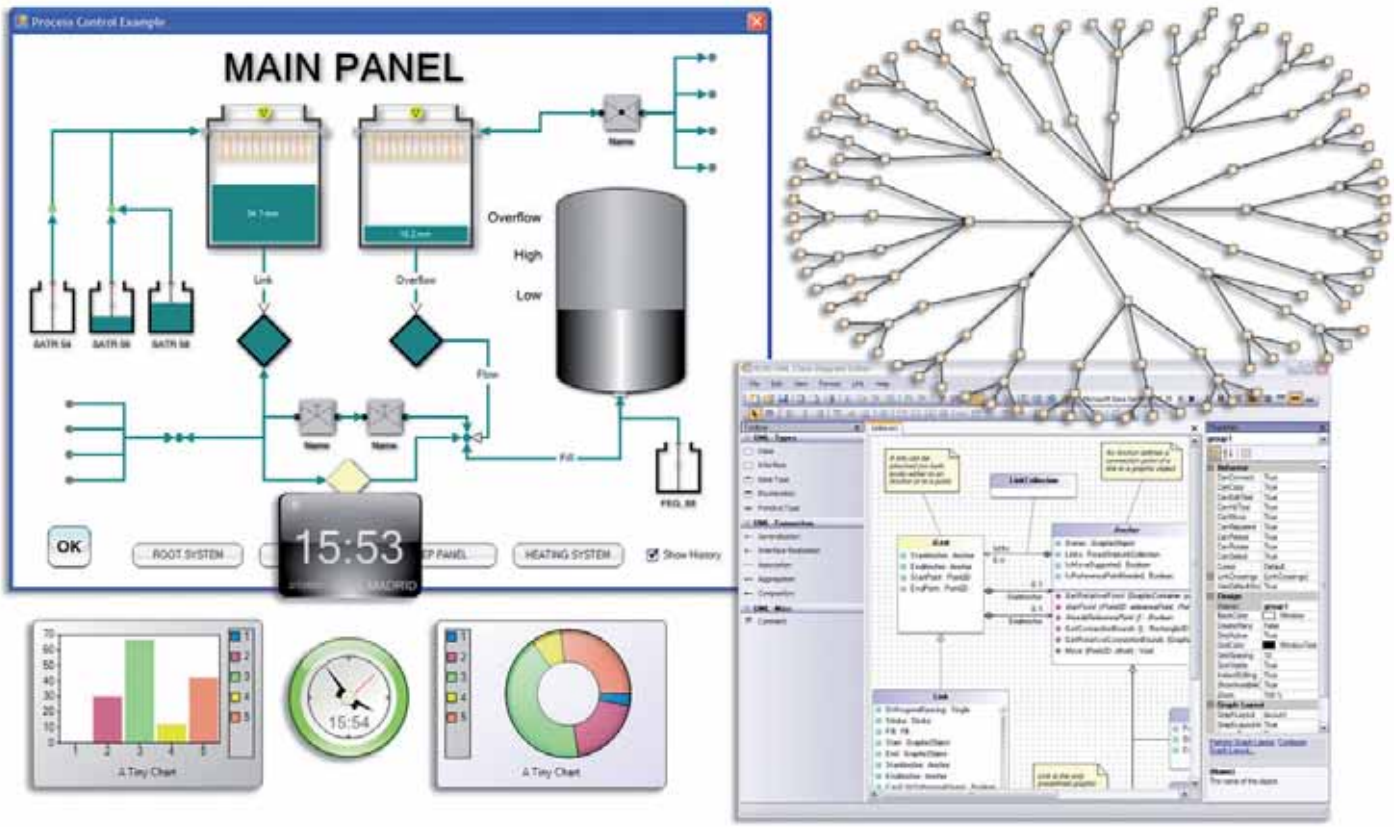
Les premières références pour les nouvelles briques .Net 3.0 arrivent enfin en français O'Reilly propose avec " introduction à WCF " un ouvrage phare pour .Net 3.0 et 3.5. Avant tout technique, l'ouvrage s'adresse aux développeurs .Net et aux architectes, chefs de projet travaillant sur la SOA. L'auteur a misé sur une centaine d'exemples parsemant chaque chapitre. L'ouvrage se segmente simplement : contrats, liaisons, hébergement, instanciation et concurrence, fiabilité, sécurité et erreurs et exceptions. On débute par un rappel sur l'origine de WCF avec le projet Indigo et sur la définition de la SOA qui constitue un axe central du livre. Incontournable !

The security development lifecycle

Difficulté : • • • •
Editeur : *Microsoft Press*
Auteur : *collectif*
Prix : *34,99 $ (USA)*

Après le codage sécurisé, Microsoft Press propose un autre ouvrage sur la sécurité dans le développement en expliquant la démarche SDL (Security Development Lifecycle ou cycle de vie du développement sécurisé). SDL se compose de 13 étapes, décrites par les auteurs. Ce n'est pas une théorisation de plus, mais de bonnes pratiques réelles, applicables sur le terrain. SDL a l'ambition de réduire les vulnérabilités, les failles dans les projets. Dès les premières pages, les auteurs sensibilisent les lecteurs sur l'impossibilité, avec les modèles et outils actuels, de fournir du code, des applications, sécurisés et au fait qu'il ne faut pas confondre les méthodes de type CMMi et méthodes agiles avec le SDL, elles n'ont pas pour but de fournir une sécurisation du code. La réussite du SDL dépend du soutien des responsables, de la compétence des personnes la mettant en œuvre et de la bonne éducation des développeurs et des intervenants aux projets. Il y a une structure à mettre en place pour garantir la réussite du SDL dans les équipes et les projets. Passionnant et instructif. De nombreuses références et sources complètent le texte. Livré avec un CD. En Anglais.

NEW ILOG Diagrammer for .NET



- All kinds of diagrams for all applications
- Automatic graph layout algorithms
- Ready-to-use visual editors
- Extends Microsoft Visual Studio 2005
- Deploy to desktop or Web with Ajax
- Appealing real-time dashboards

http://diagrammerfornet.ilog.com

Microsoft CERTIFIED Partner | ISV Software Solutions

Optimized for Microsoft Visual Studio.net

ILOG®

Changing the rules of business™

Le surf se met à Silverlight

Si Silverlight est aujourd'hui une technologie phare pour Microsoft, encore fallait-il dévoiler un projet ambitieux et innovant. Dans le même temps, Quiksilver, marque emblématique du monde du surf, cherchait à redynamiser son site et sa boutique. Décision fut prise de mettre en œuvre Silverlight et de proposer de la vidéo à la volée. Récit d'une expérience originale.

Leader mondial dans l'outdoor et particulièrement dans les sports de la glisse, Quiksilver produit et distribue une large gamme de vêtements, matériels sous différentes marques. La société est présente dans plus de 90 pays. Elle dispose d'une boutique en ligne et est très active dans le surf et les compétitions.

Une vaste réflexion a eu lieu en interne à partir du premier trimestre 2007. La seconde étape fut le lancement d'un appel d'offres sur la partie ecommerce. Dans ce cadre, l'outil Commerce Server de Microsoft fut retenu. Le premier travail concernait la refonte de la boutique en ligne et de tout le back office lié à la boutique. Mais c'est tout le site Europe que la société voulait changer en profondeur.

Mick Fanning (AUS). Photo: AFFIF - Quiksilver

Un investissement

C'est ainsi que José Cledera, DSI de Quiksilver et ses équipes, eurent droit à une démonstration de la technologie Silverlight et particulièrement sur la gestion vidéo et vidéo haute définition. *" L'équipe Microsoft (France) a été très à l'écoute. "* confie José Cledera. Car les délais étaient très courts, le choix d'utiliser Silverlight dans le frontal du site Europe et Commerce Server fut réellement décidé en juin et le développement devait se réaliser intensivement durant l'été pour une disponibilité fin septembre 2007 ! Et il faut dire que les développeurs internes avaient plus l'habitude de Flash que de Silverlight… *" On a été séduit par Silverlight et particulièrement par son player vidéo, on pouvait avoir du direct live "*.

Comme en interne, il n'y avait pas de compétences sur .Net et Silverlight, Quiksilver a travaillé avec un prestataire extérieur et Microsoft France s'est fortement investi dans le projet pour fournir les compétences techniques. Il est prévu que pour la fin 2007, les développeurs internes puissent acquérir une technicité autour des environnements Microsoft.

Finalement, le projet fut un équilibre entre *" Quiksilver en qualité de maître d'œuvre avec notre prestataire, et les aspects techniques furent assurés par Microsoft. "*. Le site web, une vitrine du monde du surf visait les sportifs et les amateurs. Ce n'est pas un hasard si le site fut lancé durant le Quiksilver Pro France, championnat du Monde de Surf Masculin. *" Les besoins ont été définis par les athlètes "* précise José Cledera. L'objectif était un site simple d'utilisation, même s'ils sont habitués à utiliser le web et notamment MySpace. Mais il fallait pouvoir les fédérer sur un site commun afin de partager les informations, mettre les vidéos, des forums, et offrir du direct. Au-delà du site, il y avait aussi le back office à remanier avec Commerce Server, proposer une nouvelle boutique et mieux gérer le catalogue avec des technologies comme WPF.

Multi-plate-forme et vidéo

" On voulait un site multi-plate-forme, on a un parc Mac non négligeable. Après notre engagement avec Microsoft, nous n'avons pas eu beaucoup de craintes. Le projet a été bien géré ", analyse José Cledera. Et pour Microsoft, il s'agissait de montrer une vitrine technologique à la pointe de Silverlight avec un player dynamique, une interface RIA, de la vidéo en direct, des gros plans, de la haute définition. Bref, tout ce que peut apporter Silverlight. Et le résultat est particulièrement réussi, notamment sur l'interface, l'ergonomie et le fonctionnement même du site. Les vidéos sont encodées en haute et basse définition pour fournir les deux canaux du site. Parmi les outils mis en œuvre, on retrouve Expression, Visual Studio et le serveur web IIS.

Les tests n'ont pas créé de réels soucis. Dans le futur, il s'agit de déployer la technologie sur l'ensemble de la marque et même de l'utiliser dans les autres marques maison. Si pour le budget du projet, rien n'a été dévoilé, José Cledera assure que cela n'a pas coûté plus cher qu'avec les technologies " anciennes ". Et le succès a été au rendez-vous car plus de 500 000 vidéos ont été vues durant les premières semaines suivant le déploiement !

François Tonic

Les points clés à retenir

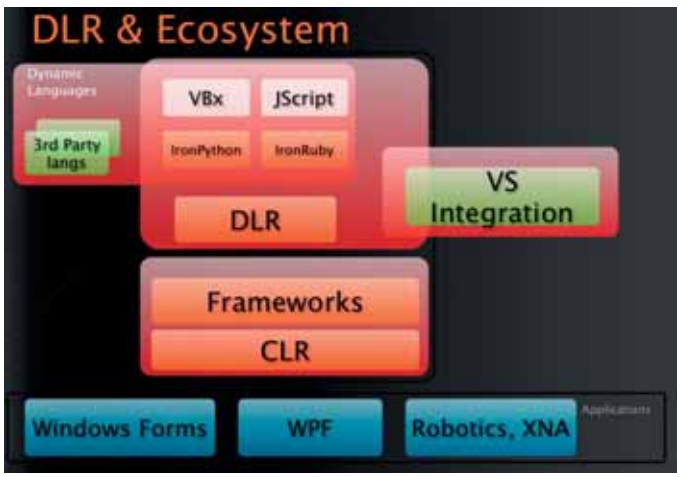
- Un site image centré sur les champions Quiksilver ; ces derniers participeront activement à la vie du site en devenant les principaux contributeurs.
- A l'occasion du Quiksilver Pro France, Quiksilver propose aux internautes de vivre une nouvelle expérience interactive. Ils pourront suivre l'ensemble des compétitions Quiksilver en direct, au travers d'un player multimédia premium
- Une nouvelle plate-forme eCommerce basée sur Microsoft Commerce Server 2007, pour lequel un back-office innovant a été développé en WPF. S'intègre de façon transparente dans les services communautaires afin de permettre des actes d'achats fluides.

TechEd 2007 : *Visual Studio 2008 en cadeau !*

Début novembre s'est déroulée la traditionnelle conférence développeur Microsoft européenne, à Barcelone, la TechEd. Plus de 4000 développeurs se sont pressés dans les salles durant une semaine pour apprendre, écouter, toucher les nouveaux produits. Plus intéressante que l'édition précédente, TechEd 2007 a créé une véritable surprise avec l'annonce de la sortie de Visual Studio 2008 et du framework .Net 3.5, ainsi que la possibilité de découvrir de nombreux projets disponibles courant 2008…

Durant la session inaugurale, l'éditeur a rappelé l'importance des outils de développeurs : Visual Studio, Visual Studio Team System, la gamme express et Popfly. Quelques chiffres : un million de développeurs professionnels sous Visual Studio 2005, 17 millions de téléchargement des versions Express, 25% des développeurs utilisent Team System, et pas moins de 100 000 utilisateurs de la version bêta de Popfly. Le but de Microsoft est d'adresser l'ensemble des développeurs, de l'amateur averti aux professionnels. La sortie de .Net 3.5 et du futur Sync Framework va dans le sens d'une intégration toujours poussée entre les applications, les technologies et les outils. Cela passe aussi par la documentation dans MSDN : avec les galeries de code, les wiki. Le Software + Services est un autre axe de la stratégie de l'éditeur avec les disponibilités de documents sous forme de Blue Print, des bonnes pratiques ainsi que d'un plug-in pour Visual Studio.

L'écosystème constitue plus que jamais un élément vital pour Microsoft et la vitalité du marché du développement. Ainsi, il y a plus de 200 partenaires produisant plus de 2 000 produits ! La nouveauté passe par une modification de licencing, ce changement concernera Visual Studio 2008 et son SDK. Il n'y aura plus de limites pour les partenaires pour créer sur Windows ou Visual Studio des solutions… À cela, se rajoute un accès au code source de Visual Studio pour les partenaires " premier " (dans le programme Visual Studio Industrie Partner). Cette volonté passera notamment par l'utilisation de Visual Studio Shell. Il s'agit en quelque sorte d'une coquille vide fournissant l'infrastructure Visual Studio dans laquelle on ajoute ses modules, ses fonctions, ses applications. Très souple, il est personnalisable. Surtout, on peut déployer sa solution avec Visual Studio Shell. Un SDK est disponible. Popfly n'a pas été oublié dans cette ouverture. La nouveauté est la disponibilité de Popfly Explorer (uniquement sous Windows). Il s'agit d'un plug-in destiné à Visual Studio. On peut y créer des projets Popfly ou encore accéder à des projets de n'importe où via Popfly. Surtout, Il intègre désormais les gadgets silverlight.

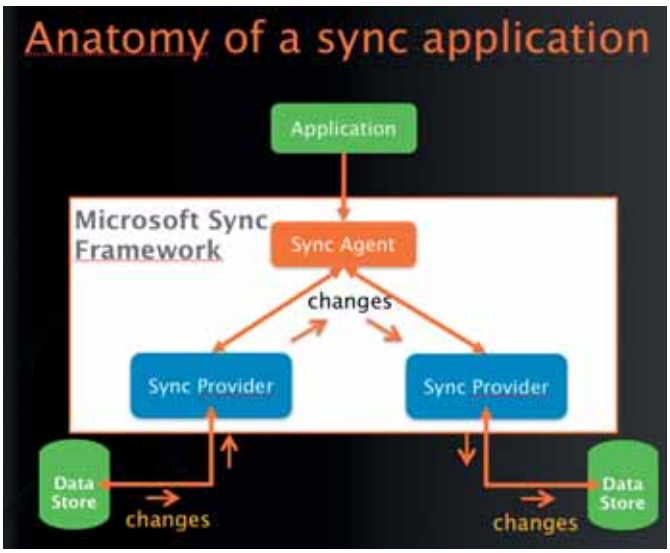


Côté agenda, 2008 sera chargé avec en particulier : SQL Server, Windows Server, Biztalk v6, IIS 7, Expression Studio 2, Team System (alias Rosario), Popfly Explorer, le framework Sync, sans oublier Live 2.0. Mais surtout, pour les deux ans à venir, il faut s'attendre aux nouveautés suivantes : Visual Studio 10, Silverlight vNext (version 2.0 ?), ainsi que le framework .Net 4.0.

D'autre part, avec l'arrivée du framework 3.5, on aura droit à d'autres sorties plus ou moins rapides. L'une d'elles concernera ASP.Net Futures. Ce futur ASP.Net inclura : le projet Astoria (voir plus bas), un meilleur support d'Ajax et celui de Silverlight, l'apparition d'un framework MVC et de contrôle de données dynamiques. La CTP est prévue courant décembre.

SQL Server 2008 : l'autre vedette

La donnée a été une des vedettes de cette TechEd, avec de nombreuses sessions techniques sur le sujet (ainsi que sur Linq ou le futur framework Sync), qui furent aussi très populaires. François Ajenstat, directeur de SQL Server, a durant plus d'une heure dévoilé l'ensemble des nouveautés du SGBD maison qui tiennent en 4 piliers : plate-forme de données d'entreprise, au-delà du relationnel (notamment avec le XML et les nouveaux types de données, le développement dynamique et enfin l'intégration des données dans son environnement. Le déve-



loppeur bénéficiera de nombreuses nouveautés comme Linq, un designer graphique pour Entity, de nouveaux modèles de données ou encore comment Sync va interagir avec SQL Server pour synchroniser les données déconnectées (ou non), l'environnement collaboratif et le serveur. Une nouvelle CTP arrivera très prochainement pour une disponibilité finale en février prochain. La version suivante devrait apparaître vers 2010-2011 sous le nom de SQL Server vNext++.

Une exposition réduite mais de qualité

Contrairement à l'an dernier, la partie exposition était plus réduite mais les exposants avaient des produits intéressants à montrer. Nous avons été particulièrement impressionnés par la qualité des nouveaux composants (Infragistics, Gaïa, Software FX), avec une mention spéciale à IdentityMine et des composants WPF haut de gamme, blendables, que l'on peut utiliser directement sous Blend ! Une série d'une dizaine de composants exploitant au mieux les possibilités graphiques de WPF. Un autre stand proposait une solution de développement dédié aux technologies vocales (à base de web services) : Web21C SDK. Le Micro Framework tenait aussi un petit stand. Si aujourd'hui, ce mini framework se destine avant tout aux spécialistes de l'embarqué, de l'enfoui, Microsoft travaille à la prochaine version et surtout à des outils de développement qui pourraient apparaître dans les 12 à 18 mois.

Du Web et encore du Web

Le développement web n'est pas oublié, quelques dizaines de sessions étaient spécifiquement dédiées au Web, aux outils, aux technologies. La gamme Expression et la collaboration développeur – designer occupèrent quelques sessions mais sans plus. Par contre, Silverlight, Live, ASP.Net et même Popfly eurent droit à une audience élargie. Ainsi, si IronPython est disponible depuis un an en v1, la v2 apparaîtra début 2008 et la v3 début 2009. Occasion pour Microsoft de faire connaître les nouvelles API et fonctions de Windows Live ou de Virtual Earth. Les langages dynamiques furent aussi mis à l'honneur et notamment IronPython et tout ce qui tourne autour de la DLR (CLR mais dédié au monde dynamique comme Ruby, Python, etc.).

Et après ?

Microsoft a profité de Barcelone pour commencer à parler du futur. Ainsi, nous avons pu voir quelques idées autour de VB.NexT. Un des avenirs de VB passera par le langage dynamique et son intégration dans la DLR. Pour VB Next, les axes de réflexions concernent la simplification de la syntaxe, la concurrence, le modèle de données, le workflow.

Toujours dans les langages, Microsoft travaille activement sur les langages dynamiques et surtout fonctionnels (avec l'intégration à court terme de F#). La programmation concurrente constitue un axe important, notamment avec le langage C Omega. Occasion aussi de voir de plus près la version 2.0 de PowerShell, disponible en CTP. On notera parmi les nombreuses nouveautés : le remoting, la possibilité de créer des espaces limités d'exécution, des améliorations dans le langage ou encore le support de WMI. Toujours parmi les projets dévoilés ou montrés, on citera encore le projet Astoria. Il s'agit d'une interface de données REST qui doit simplifier l'accès aux données et les affichages, leurs manipulations. Actuellement en CTP, la version finale est attendue mi-2008.

La prochaine édition se tiendra à Barcelone en novembre 2008.

■ **François Tonic**

Projet Oslo (2009) : la prochaine plate-forme applicative

Oslo est un projet ambitieux car il s'agit de construire une nouvelle plate-forme entièrement intégrée incluant les outils, les frameworks, les parties serveurs et données. Oslo n'est donc pas un outil mais avant tout une plate-forme. L'objectif est de pouvoir y construire des applications composites, faire de la SOA, du SaaS. Oslo s'articulera autour de 5 zones :

- le framework .Net 4.0 avec d'importantes évolutions des couches workflow et communication
- le middleware Biztalk : importante refonte pour le bâtir sur WCF et WF, avec une forte intégration SOA et surtout du BPM.
- Les services : avec notamment la disponibilité de l'Internet Service Bus et des Biztalk Services.
- Les outils : disponibilité de nouveaux modeleurs. Team System " Rosario " constitue la première étape. Ces nouveaux outils seront disponibles avec Visual Studio 10.
- Disponibilité d'un référentiel. Pour la première fois, Microsoft proposera un référentiel commun à l'ensemble des outils. Pour le moment, il n'a pas de nom de code ou de nom. Il sera embarqué dans les outils.

Les premières pré-versions (ou CTP) seront disponibles à la fin de l'année et durant l'année 2008. La disponibilité générale n'est pas attendue avant fin 2008, ou plus sûrement courant 2009. Côté migration, sur le .Net framework, le travail sera sans doute limité et sur Biztalk, tout sera fait pour réduire les problèmes. Pour le moment, nous ne savons pas comment sera intégrée la mobilité qui demeure un axe important pour l'éditeur. Sur Live, les services en ligne permettront de consommer les applications composites et services créés et hébergés par Oslo. Pareillement pour Office, qui devient un conteneur naturel des applications Oslo. Enfin, sur la possibilité d'utiliser ces applications en dehors de l'environnement Microsoft, là encore, cela reste à préciser.

La Voie Express Vers .Net !

Si l'on associe souvent .NET à VisualStudio, n'est-ce pas parce que Microsoft est propriétaire des deux ?

Nous allons vous montrer comment Sybase PowerBuilder 11, le leader mondial du développement L4G, constitue une alternative simple et rentable pour déployer sur la plate-forme de votre choix, et ce qui nous intéresse particulièrement ici, .NET.

Grâce à PowerBuilder 11, .NET n'a jamais été aussi simple !

.NET ne se réduit pas à un système d'exploitation. La plate-forme .NET permet d'accéder à une vaste palette de services et de classes qui enrichissent vos développements.

Voilà pourquoi Sybase a intégré l'ensemble de ces nouvelles ressources dans la dernière version de son logiciel PowerBuilder 11.

A partir de votre code source, vous pouvez déployer facilement vos applications en WinForms, ClickOnce Smart Client, Web forms ASP .NET, et vos NVO PB en .NET assemblies et en Services Web .NET.

PowerBuilder est un RAD 4GL fiable, intuitif et convivial, en particulier grâce à la DataWindow. Développer en PowerBuilder est facile: moins de code, moins d'erreurs, meilleure productivité

Pérennité et ouverture

Sybase PowerBuilder, c'est 17 ans d'innovation et d'expérience dans le domaine du développement d'applications puissantes, à interface riche reposant sur une connectivité ouverte sur toutes les données de l'entreprise en natif ou standard (ODBC, ADO.net, JDBC, XML, Services Web,...).

A partir d'un code source unique, vos applications sont déployées partout: J2EE, Win 32, Web, terminaux mobiles, et maintenant .NET.

Si PowerBuilder poursuit son ouverture, il reste néanmoins fidèle à ses fonctionnalités fondamentales. A travers le monde, dans les banques, compagnies d'assurances, sociétés financières, médias, organismes locaux, départementaux, régionaux, gouvernementaux, des dizaines de milliers d'applications critiques sont gérées par PowerBuilder.

PowerBuilder 11 supporte Microsoft SQL server SNC, Oracle 10g RAC et Informix 10. Il est certifié avec Vista.

Sybase a par ailleurs dévoilé ses plans pour le support de AJAX, WCF/WPF,...

Découvrez PB 11 et .NET en 15 minutes : http://www.sybase.com/products/development/powerbuilder/videos

Evaluez PowerBuilder aujourd'hui : http://response.sybase.com/forms/WW_PB11_Eval

Contactez l'équipe PowerBuilder dès aujourd'hui:
Tel : +33 (0)1 41 91 96 80
Email : europePB@sybase.com

SYBASE®

Plongez au cœur de .Net 3.5

Nous y sommes. Les versions RTM de Visual Studio et du framework .Net 3.5 ont été livrées fin novembre, comme annoncé durant la dernière TechEd. Nous vous avions déjà proposé une première plongée dans ce nouveau framework, aujourd'hui, nous continuons notre exploration.

Si .Net 3.5 intègre des nouveautés dans les langages, il est très important de comprendre que cette évolution n'a pas entraîné de modification au niveau du CLR. En réalité, les nouveautés apportées aux langages l'ont été uniquement au niveau du compilateur.

Parmi ces nouveautés, on trouve :

- Les " automatic properties "
- Les " object initializers "
- Les " collection initializers "
- Les " local variables type inference "
- Les " implicitly typed arrays "
- Les " anonymous types "
- Les " extension methods "
- Les " lambda expressions "
- Les " partial methods "
- Les " expression trees "
- " XML Litteral "
- " Language INtegrated Query "

La plupart de ces points ont déjà été évoqués en détail dans le hors série spécial .NET d'octobre. Pour résumer, elles permettent essentiellement de réduire le code, de gagner en lisibilité, et donc en maintenance. Plusieurs nouvelles librairies sont venues se greffer sur le framework .NET 3.0 :

- **System.AddIn.dll et System.AddIn.Contract.dll**

Ces deux librairies implémentent un nouveau modèle d'add-in (présenté plus loin dans cet article).

- **System.Core.dll**

Comme son nom l'indique, cette librairie est utilisée par plusieurs autres. Elle introduit un grand nombre de nouvelles classes, ainsi que beaucoup d'extension methods (utilisées notamment pour écrire des requêtes LINQ To Object).

Parmi les nouvelles classes introduites, on trouve notamment :

- HashSet<T> : cette classe est une collection non ordonnée qui assure l'unicité de ses éléments
- TimeZoneInfo : cette classe gère le décalage horaire (elle est présentée plus en détail à la fin de cet article),
- ReaderWriteLockSlim : cette classe améliore la classe existante : ReaderWriterLock définie dans mscorlib (classes permettant de synchroniser les accès aux données). Elle est bien plus performante et évite des cas de blocage de ReaderWriterLock. Elle supporte trois modes : Read, Write et UpgradeableRead.
- System.IO.Pipes.* : Ce namespace est utilisé pour l'Inter-Process Communication. Il fournit un certain nombre de nouvelles classes facilitant les échanges entre processus d'une même machine, ou de machines appartenant à un même réseau.
- System.Security.* : Le namespace System.Security n'est pas nouveau. La librairie System.Core.dll rajoute plusieurs algorithmes cryptographiques basés sur les courbes elliptiques, utilisés avec Vista. Parmi ces derniers, on trouve des algorithmes de hashage :

  - MD5Cng
  - SHA1Cng
  - SHA256Cng
  - SHA384Cng
  - SHA512Cng

  ainsi que des algorithmes asymétriques :

  - ECDSACng
  - ECDiffieHellmanCng

- System.Diagnostics.Eventing.* : Ce namespace définit un nouveau Trace Listener : EventSchemaTraceListener. Ce nouveau Trace Listener permet de définir un schéma que partageront tous les évènements tracés, de la même façon que ce qui a été introduit dans l'Event Viewer de Windows Vista (technologie ETW, " Event Tracing for Windows ").
- **System.Diagnostics.PerformanceData.* :** Ce namespace permet d'accéder aux nouveaux compteurs de performances introduits avec Windows Vista.
- **System.DirectoryServices.AccountManagement.dll**

Cette librairie implémente un wrapper extensible sur l'Active Directory. Elle est présentée plus en détail ci-après.

- **System.Data.DataSetExtensions.dll**

Avec .NET 3.5, le designer de DataSet ne génère plus des classes héritant de DataTable (pour les tables) mais de TypedTableBase<T> avec T, le type de DataRow. Cette classe implémente l'interface IEnumerable<T>. Grâce à cela, il devient possible d'exécuter des requêtes LINQ sur les tables d'un DataSet. TypedTableBase est définie dans la librairie System.Data.DataSetExtension.dll.

On trouve également dans cette dll, des extension methods, qui vont permettre de récupérer un EnumerableRowCollection<DataRow> (classe définie dans la librairie System.Data.DataSetExtension.dll) à partir d'un DataTable. Cette classe implémente l'interface IEnumerable<T>. Pour pouvoir requêter un IEnumerable<DataRow>, il faudra utiliser une autre extension method : Field (également définie dans System.Data.DataSetExtension), afin de récupérer la valeur d'une colonne dans le DataRow. L'extension method Field est générique. Il va falloir lui spécifier le type de la colonne. On pourra en profiter pour tricher et lui passer un type nullable. LINQ To DataSet s'appuie sur ces deux extension methods.

- **System.Data.Entity.dl**

Parmi les nouveautés de l'ADO 3.5, on trouve l'Entity Data Model. Cette nouveauté ne devrait apparaître en version définitive qu'au deuxième semestre 2008. EDM permet de définir un mapping objet relationnel très puissant. En effet, EDM est capable de faire du mapping objet relationnel sans correspondance entre les classes et les tables, ce qui permet, par exemple, de faire disparaître la table de lien d'une relation n-n. Une même entité devient ainsi capable de manipuler des données issues de plusieurs tables, et inversement plusieurs entités pourront manipuler des données issues d'une même table. EDM s'appuie sur les

classes définies par la librairie System.Data.Entity.dll. Parmi celles-ci, notons System.Data.Objects.DataClasses.EntityObject qui est la classe de base des entités.

• **System.Data.Linq.dll**

LINQ To SQL s'appuie sur les classes définies dans la librairie System.Data.Linq.dll. Tout comme EDM, LINQ To SQL est une technologie de mapping objet relationnel. Par rapport à EDM, elle a l'avantage de ne pas contraindre les entités (pas de classe de base imposée). En revanche, LINQ To SQL est un mapping objet relationnel très simple en relation 1-1 avec la base mais bien suffisant dans beaucoup de cas.

• **System.Management.Instrumentation.dll**

Cette librairie implémente un provider WMI 2.0. Pour rappel, WMI (Windows Management Instrumentation) est un système de gestion interne de Windows qui prend en charge la surveillance et le contrôle de ressource système via un ensemble d'interfaces (source : wikipedia). La librairie de System.Core.dll définit, dans l'espace de nom System.Management.Instrumentation, plusieurs attributs et exceptions utilisés par la librairie System.Management.Instrumentation.dll.

• **System.Net.dll**

Cette librairie propose une suite d'APIs qui simplifie l'écriture d'applications Peer-To-Peer (P2P). Elle facilite les 3 principales étapes nécessaires à la mise en place de tels systèmes : la découverte des différents participants, la connexion et la communication.

• **System.Web.Extensions.dll**

Cette librairie implémente, entre autres, la technologie ASP .NET AJAX. Elle rajoute également de nouveaux contrôles ASP .NET 3.5, tels que le LinqDataSource, le ListView et le DataPager. Elle permet aussi, pour les applications web ou win, de simplifier l'authentification des utilisateurs, la gestion des rôles et des profiles.

• **System.Windows.Presentation.dll**

Cette librairie implémente le support de WPF pour la librairie System.AddIn (Voir ci-dessous).

• **System.WorkflowServices.dll et System.ServiceModel.Web.dll**

Ces deux librairies améliorent les technologies WF et WCF en permettant par exemple

- A un workflow de consommer un service WCF
- D'exposer un Workflow comme un web service
- Avec WCF, de renvoyer un JavaScript Object Notation (JSON)
- De gérer le support des flux RSS, ATOM

• **System.Xml.Linq.dll**

LINQ To XML s'appuie sur la librairie System.Xml.Linq.dll. On trouve dans cette librairie tout un ensemble de classes décrivant un flux xml : XElement, XAttribute, etc. Ces classes vont également simplifier l'écriture de flux XML, surtout en VB, avec " XML Litteral ". XML Litteral permet de créer un XElement uniquement sous forme de balises XML :

```
Dim clientsXml As XElement = <Clients>
  <Client Nom="Dupond" Prenom="René"/>
</Clients>
```

Regardons de plus près quelques-unes de ces dll.

System.DirectoryServices.AccountManagement.dll

Cette librairie permet de simplifier considérablement la gestion de l'Active Directory.

Prenons quelques cas nominaux courants :

• **Validation du login / mot de passe**

Pour cela, il faut commencer par récupérer une instance de System.DirectoryServices.AccountManagement.PrincipalContext. Lors de l'instanciation de cette classe, on peut choisir entre un contexte de type Machine, Domaine ou AD. Une fois le contexte récupéré, un appel à la méthode ValidateCredentials indique si un compte (identifié par un login et un password) est valide.

• **Création d'un nouvel utilisateur**

Il suffit d'instancier un System.DirectoryServices.AccountManagement.UserPrincipal. Pour cela, il faudra, là aussi, une instance de System.DirectoryServices.AccountManagement.PrincipalContext. Une fois l'utilisateur instancié et ses propriétés renseignées, il suffit d'appeler la méthode Save.

En général, la création d'un compte est faite par un administrateur qui affecte un mot de passe temporaire (l'utilisateur devra le redéfinir lors de sa première connexion). Pour cela, on se contente d'appeler la méthode ExpirePasswordNow sur l'instance de UserPrincipal.

Il est également possible de créer un utilisateur temporaire en renseignant la propriété AccountExpirationDate sur l'instance de l'utilisateur.

• **Ajout / Suppression d'un utilisateur dans un groupe**

Cette opération est une nouvelle fois très simple. On commence par récupérer l'instance de l'utilisateur (généralement en passant par la méthode static UserPrincipal.FindByIdentity) et l'instance du groupe (classe System.DirectoryServices.AccountManagement.GroupPrincipal, généralement obtenue via la méthode static System.DirectoryServices.AccountManagement), puis il suffit de rajouter l'utilisateur à sa collection de membres (group.Members.Add(user)) et d'appeler la méthode Save sur l'instance du groupe.

La suppression d'un utilisateur d'un groupe se fait exactement de la même façon (group.Members.Remove(user))

• **Désactivation d'un compte utilisateur**

Rien de plus simple, il suffit d'affecter la propriété Enabled à false sur l'instance de l'utilisateur puis d'appeler la méthode Save.

• **Récupération des groupes auxquels appartient un utilisateur**

Là aussi, c'est enfantin puisqu'il suffit d'appeler la méthode GetGroups sur l'instance de l'utilisateur.

Grâce à cette librairie, les opérations sur l'AD, très fastidieuses dans les versions précédentes, sont devenues extrêmement simples. Bien entendu, il faut toujours avoir les droits adéquats pour chacune de ces opérations.

System.AddIn.dll et System.AddIn.Contract.dll

Le fait de développer une application autorisant les add-in présente plusieurs intérêts :

- Permettre à d'autres développeurs d'étendre les fonctionnalités de l'application initiale
- Permettre à un éditeur d'intégrer entre eux les différents modules achetés par un client, même si l'achat n'est pas simultané, en une seule application (ex : une gestion commerciale avec une compta avec un module de CRM, etc.).
- Eventuellement permettre à un client, dans le cadre précédent, de ne racheter la nouvelle version que d'un seul module et garder une cohabitation avec les autres modules (plus anciens).

Pour ce genre de développement, l'application hôte ne connaît pas les add-in éventuels. Afin de pouvoir communiquer avec ceux-ci, l'application définit un contrat que les add-in doivent respecter. En .NET, ce

contrat se traduit par une interface. L'application hôte doit, à son lancement trouver ses add-in. Pour cela, il y a deux possibilités : soit c'est l'utilisateur qui les a spécifiés à l'application et celle-ci a enregistré leurs emplacements, soit l'application va rechercher les librairies définies à un emplacement prédéfini, puis rechercher si une des classes de ces librairies implémente le contrat. Avec .NET 3.5, ce dernier cas est devenu extrêmement simple. En effet, il suffit d'appeler la méthode statique AddInStore.FindAddIns (typeof(Contrat), addInRoot). Cette méthode retourne une Collection<AddInToken>.

Une fois les add-in trouvés, l'application hôte va les intégrer. Pour cela, l'application hôte a besoin d'une instance de la classe des add-in implémentant le contrat. Le framework .NET 3.5 évite d'instancier, par réflexion, la classe. Il suffit maintenant d'appeler la méthode Activate sur l'instance de l'AddInToken, tout le reste est transparent pour le développeur. L'application hôte définit un SDK qui va permettre aux plug-in de faire réaliser des opérations par l'application hôte. Cela permet également à l'application hôte de garder un contrôle sur l'exécution de ces opérations.

Au niveau sécurité, il est possible d'isoler le contexte d'exécution des add-in avec une application domain différente de celle de l'application et / ou en appliquant une isolation sur le processus. Isoler les add-in, apporte plusieurs avantages :

- L'application hôte peut récupérer une exception de l'un des add-in et l'arrêter, ou le redémarrer, sans impacter l'application hôte ou les autres add-in
- Il est possible d'appliquer des droits restreints à l'add-in (ce qui permet par exemple de ne pas compromettre la sécurité d'une application critique en y faisant fonctionner un add-in externe).

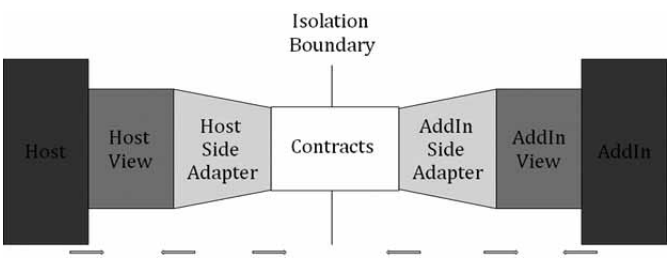
Avec .NET 3.5, il y a quatre niveaux d'exécution par défaut :

- FullTrust
- Host
- Internet
- Intranet

Cependant, il est également possible de définir des droits spécifiques.

Pour arrêter un add-in, il faut appeler la méthode statique AddInController.GetAddInController(contractInstance).Shutdown().

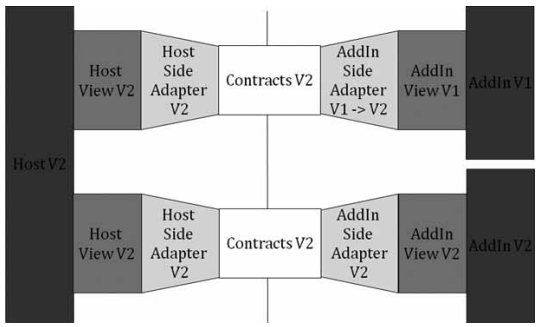
Le schéma suivant représente le schéma de communication entre l'application hôte et les plug-in :



Il y a quelques contraintes à respecter :

- Le contrat doit être une interface et hériter de IContract.
- Les signatures des méthodes ou propriétés définies dans le contrat doivent utiliser des types sérialisables, définis dans mscorlib (Int32, String, DateTime, etc), dans d'autres contrats ou dans un enum défini dans l'assembly du contrat.
- Les vues contiennent des interfaces ou des classes abstraites qui représentent les méthodes et propriétés appelables par l'hôte (pour la couche Host View) et les opérations méthodes et propriétés définies dans l'add-in (pour la couche AddIn View). La vue n'est pas dépendante du contrat. Le diagramme illustre la dépendance de l'hôte avec la vue et l'indépendance de celle-ci. La vue de l'add-in doit posséder l'attribut System.AddIn.Pipeline.AddInBase.
- L'Adapter va permettre de faire le lien entre le contrat et la vue.
  Dans le cas de l'Adapter de l'hôte, la classe hérite ou implémente la vue, possède une instance du contrat et possède l'attribut System.AddIn.Pipeline.HostAdapter.
  Dans le cas de l'Adapter de l'add-in, la classe hérite de System.AddIn.Pipeline.ContractBase, implémente le contrat et possède l'attribut System.AddIn.Pipeline.AddInAdapter
- L'add-in hérite de sa vue et possède l'attribut System.AddIn.AddIn.

Ce pipeline en couche présente un énorme intérêt : réduire l'impact d'une mise à jour du contrat. Ainsi, il suffira de développer un Adapter capable de passer d'une version à l'autre pour pouvoir utiliser les add-in définis avec une version antérieure du contrat.



Les deux librairies System.AddIn.dll et System.AddIn.Contract.dll permettent de simplifier le développement d'applications autorisant les add-in. L'architecture de pipeline proposée permet une plus grande flexibilité dans le versionning des add-in et de l'hôte.

Workflow Foundation et Windows Communication Foundation

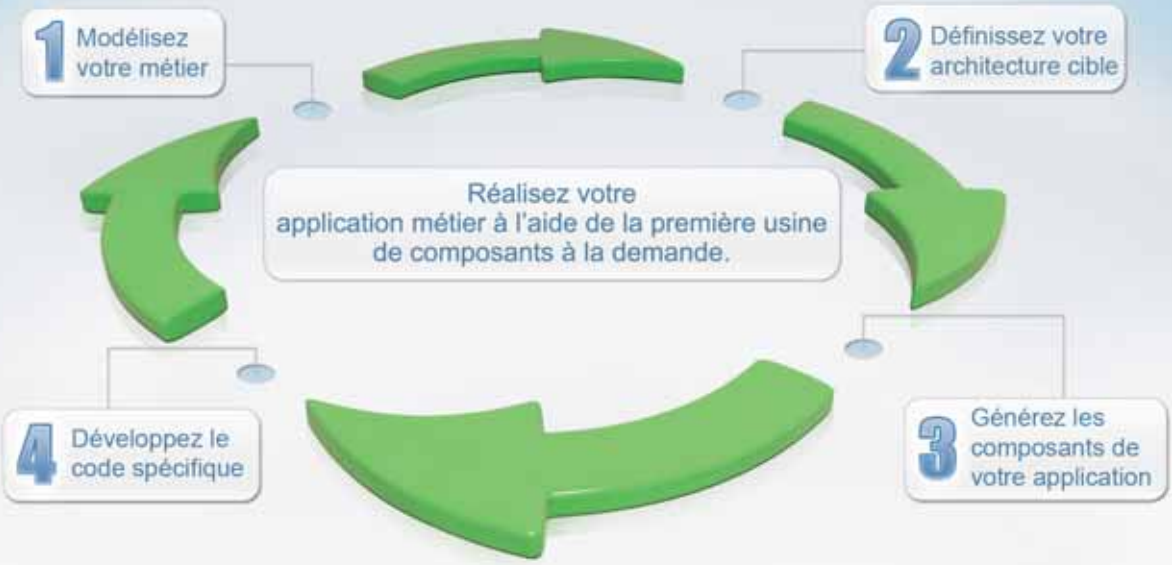
Workflow Foundation (WF) et Windows Communication Foundation (WCF) ne sont pas deux nouvelles technologies, elles sont déjà présentes dans le Framework .Net depuis la version 3.0. Par contre, faire communiquer un workflow grâce à WCF est une nouveauté du Framework 3.5. L'API de WF s'enrichit de deux nouvelles activités :

- SendActivity
- ReceiveActivity

SendActivity permet à un workflow de communiquer avec un service WCF. A partir d'un contrat WCF et du Endpoint associé, l'activité permet de relier les paramètres et le résultat de l'appel à des champs du Workflow. Pour illustrer cette manière de communiquer, prenons cette définition de service :

```
namespace WcfServiceLibrary
{
    [ServiceContract]
    public interface IMyService
    {
        [OperationContract]
        string BigMethod(int a, int b);
    }
}
```

Accélérez vos développements .NET



CodeFluent.com

La fabrique en ligne est totalement gratuite pour les modèles métier comprenant jusque 30 entités.

nouveau !

Inscrivez-vous dès aujourd'hui sur : http://www.codefluent.com

0€

CodeFluent

Version complète Développeur donnant accès aux codes sources des composants générés.

Prix public : ~~2490€ht~~

Prix spécial lecteurs de "Programmez" : **1990€ht**

CodeFluent est une usine logicielle qui automatise la création des composants de votre application métier .NET selon les meilleures pratiques des experts de la plate-forme Microsoft.

SoftFluent est une société spécialisée dans le développement sous plate-forme Microsoft. Elle compte déjà de nombreux clients prestigieux tels qu'ILOG, VCS Timeless, TF1 ou Microsoft France et Europe. Contactez-nous sur info@softfluent.com

Softfluent recrute !

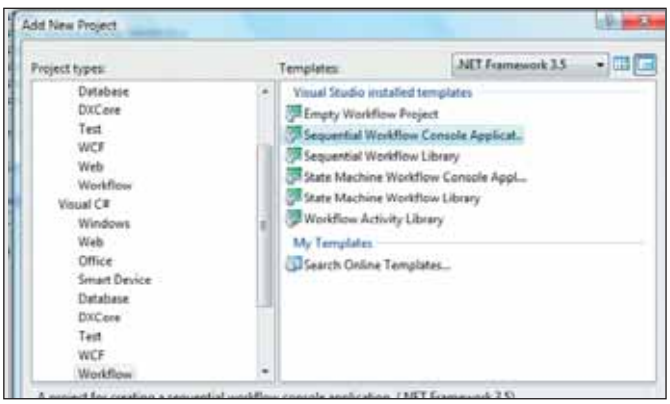
Venez assouvir votre passion du développement logiciel au sein d'une équipe dynamique !

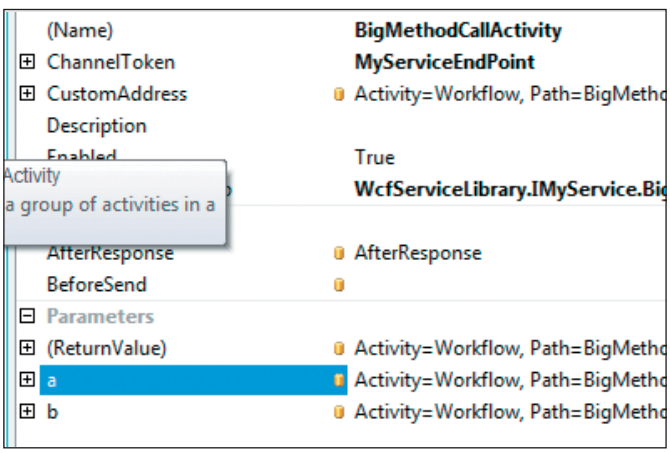
SoftFluent 46, rue Auguste Blanqui 94250 GENTILLY - 01 46 16 05 02

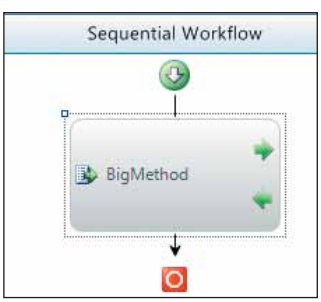
Microsoft | PROGRAMME IDÉES

Il est implémenté par un service hébergé dans un projet web :

```
namespace WCFService
{
    public class MyService : WcfServiceLibrary.IMyService
    {
        public string BigMethod(int a, int b)
        {
            return string.Format("{0} - {1}", a, b);
        }
    }
}
```







La première étape est la création du Workflow. Un simple projet de console de Workflow suffit.

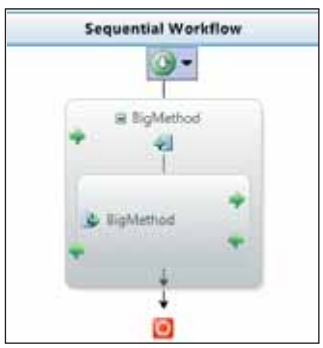
Ajoutons l'activité SendActivity. Elle est localisée dans la boîte à outils dans un onglet particulier " Windows Workflow 3.5 ". Cet onglet contient aussi l'activité ReceiveActivity.

Le projet de workflow doit référencer le contrat pour pouvoir configurer l'activité, et le " endpoint " vers le service doit aussi être renseigné. Une fois ces informations entrées, la dernière étape concerne la liaison entre les paramètres du workflow et le résultat du workflow : trois champs du workflow sont créés, deux pour les paramètres et un pour le résultat de l'appel.

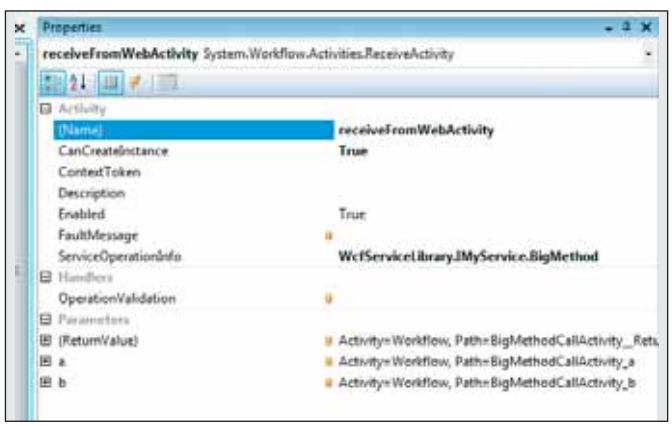
En l'état, le workflow est opérationnel, mais risque de ne pas fonctionner car les valeurs des paramètres de l'appel du service WCF ne sont pas initialisées. Pour cela, il faut soit utiliser d'autres activités pour initialiser ces valeurs, ou les initialiser à la création du workflow :

```
Dictionary<string, object> properties = new Dictionary<string, object>();
properties.Add("BigMethodCallActivity_a", 1);
properties.Add("BigMethodCallActivity_b", 2);
WorkflowInstance instance = workflowRuntime.CreateWorkflow(typeof
(WorkflowLibrary.Workflow), properties);
```

Passons à l'activité ReceiveActivity. Comme pour la SendActivity elle a besoin d'un " endpoint " ainsi que d'un contrat. Lorsque nous plaçons l'activité dans un workflow, nous remarquons qu'elle nous permet de placer d'autres activités en son sein. Ces activités seront lancées entre le moment ou la requête arrive au workflow et le moment ou le workflow répond à l'émetteur. Pour l'exemple, nous allons utiliser le même contrat que celui utilisé plus haut. Le workflow jouera donc le rôle d'intermédiaire entre le service métier, et les clients (ci-contre).



Nous parlions plus haut de l'initialisation des valeurs des paramètres de l'activité SendActivity. Dans notre cas, elles seront initialisées grâce à l'activité ReceiveActivity :



La dernière étape consiste à créer le programme qui communiquera avec le service WCF hébergé par le workflow.

La gestion des fuseaux horaires.

Depuis les premières versions du Framework .Net la classe " DateTime " est la classe qui encapsule une date et une heure. Pour une date et une heure donnée, en fonction de la configuration locale de la machine, plus précisément en fonction du fuseau horaire de l'utilisateur courant, nous avons la possibilité de convertir une date locale en date en temps universel et inversement. Cette conversion est essentielle car elle permet aux applications de se synchroniser sur un temps unique quel que soit l'endroit où sont situés les utilisateurs. Prenons un exemple, avec une application de prise de rendez-vous : 12h à Paris, ne correspond pas à 12h à San Francisco ou à Pékin. De plus, en fonction du moment de

l'année, 12h à Paris correspond à soit 10h ou 11h en temps universel à cause du décalage dû au passage de l'heure d'été à l'heure d'hiver et inversement. Encore une chose qui rend ce traitement difficile : les passages entre les heures d'hiver et d'été, pour les pays qui les pratiquent encore, ne sont pas forcement au même moment : il y a par exemple 1 semaine de décalage entre les Etats-Unis et la France.
Toutes ces informations forment le paramétrage des différents fuseaux horaires disponibles pour l'utilisateur :
Ces informations sont donc disponibles directement depuis l'API native de Windows. Maintenant, avec le Framework 3.5 elles sont directement disponibles en managé. La classe principale de cette gestion est la classe " TimeZoneInfo " : elle contient l'ensemble des informations qui permet de décrire un fuseau horaire. En particulier :

- Son nom
- S'il possède des changements d'heure
- Un identifiant universel
- Son décalage avec UTC par défaut

Le décalage par défaut ne correspond pas toujours au décalage courant à cause du système heure d'été/hiver. Pour connaître le décalage courant, la classe TimeZoneInfo possède la méthode GetUtcOffset. La propriété statique Local renvoie le fuseau courant :

```
TimeZoneInfo currentTZI = TimeZoneInfo.Local;
DateTime now = DateTime.Now;
TimeSpan offset = currentTZI.GetUtcOffset(now);
Console.WriteLine(@"Zone courante:
Nom: {0}
Ecart UTC courant: {1} heure(s);
", currentTZI.DisplayName, offset.TotalHours.ToString("f"));
```

La classe TimeZoneInfo nous fournit encore deux services:

- La liste des fuseaux horaires du système
- Un système de conversion entre fuseaux

La liste des fuseaux correspond directement à celle que le système nous propose :

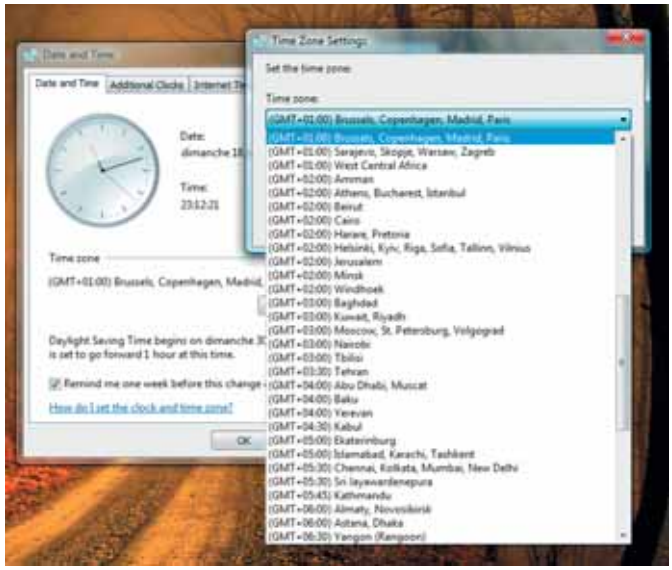
```
Console.WriteLine("Zones du système:");
foreach (TimeZoneInfo info in TimeZoneInfo.GetSystemTimeZones())
{
   Console.WriteLine("{0} (Ecart UTC:  {1})", info.StandardName, info.
BaseUtcOffset.TotalHours.ToString("f"));
}
```

Le résultat est le suivant :

```
Zones du système :
Greenwich Standard Time (Ecart UTC:  0,00
GMT Standard Time (Ecart UTC:  0,00)
W. Europe Standard Time (Ecart UTC:  1,00)
Central Europe Standard Time (Ecart UTC:  1,00)
Romance Standard Time (Ecart UTC:  1,00)
Central European Standard Time (Ecart UTC:  1,00)
…
```

Concernant les conversions, deux méthodes sont disponibles :

- Directement entre un fuseau et UTC
- Entre deux fuseaux horaires



La première méthode est utilisable pour convertir en temps local des dates venant d'une base de données par exemple. Utilisons la deuxième pour convertir la date et l'heure courante en date et heure courante à San Francisco et à Pékin. Pour retrouver leurs fuseaux horaires, nous utilisons la méthode *FindSystemTimeZoneById.* L' " Id " correspond au nom du fuseau horaire.

```
TimeZoneInfo PST = TimeZoneInfo.FindSystemTimeZoneById("Pacific Standard
Time");
TimeZoneInfo China = TimeZoneInfo.FindSystemTimeZoneById("China
Standard Time");

DateTime SanFrancisco = TimeZoneInfo.ConvertTime(now, TimeZoneInfo.
Local, PST);
DateTime Pekin = TimeZoneInfo.ConvertTime(now, TimeZoneInfo.Local, China);
Console.WriteLine(@"
Paris:        {0} {1}
San Francisco: {2} {3}
Pékin:        {4} {5}",
   now.ToShortDateString(), now.ToShortTimeString(),
   SanFrancisco.ToShortDateString(),SanFrancisco.ToShortTimeString(),
   Pekin.ToShortDateString(), Pekin.ToShortTimeString());
```

Cela donne le résultat suivant :

```
Paris :          17/11/2007 23:29
San Francisco :  17/11/2007 14:29
Pékin :          18/11/2007 06:29
```

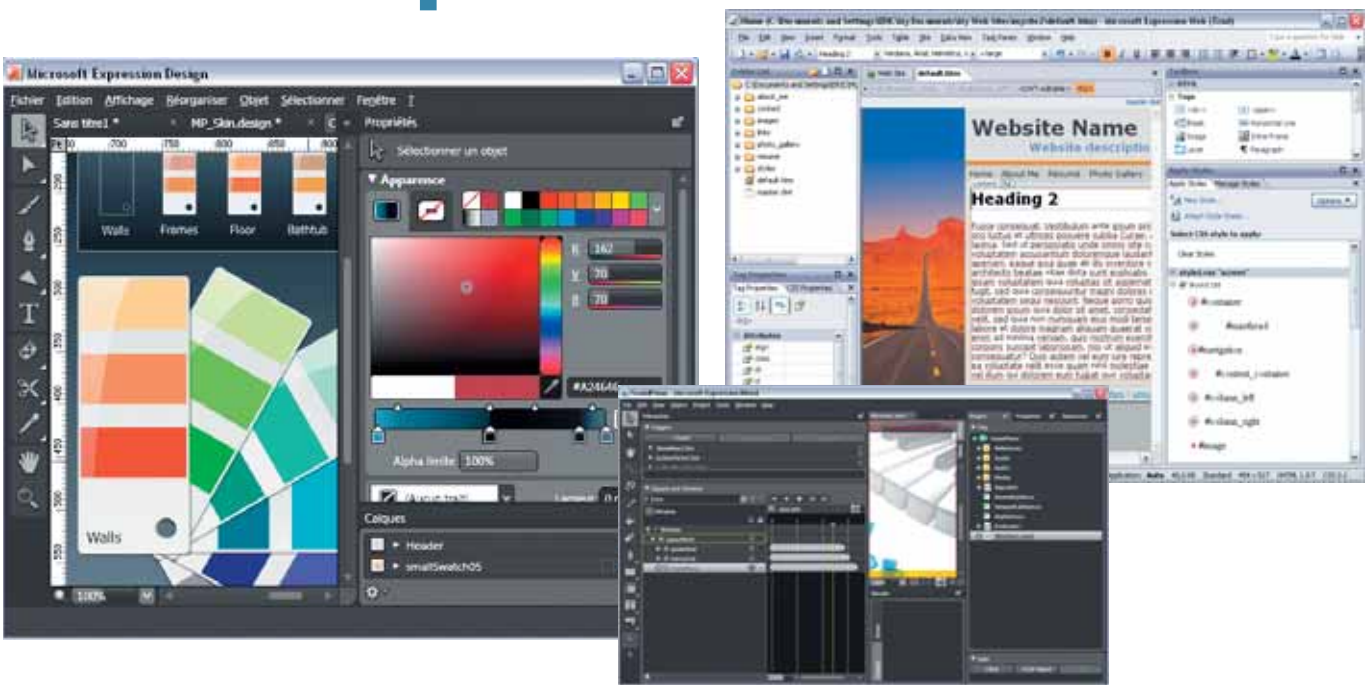
Conclusion

Ceci est seulement un aperçu de la richesse des nouvelles API du Framework .Net qui ne sont qu'une consolidation d'un socle d'API déjà riche. Si l'on pouvait dire qu'un développement avec le Framework 3.0 était soit du WF, soit du WCF, soit du WPF, nous pouvons maintenant affirmer que le développement avec le Framework 3.5 est réellement WF+WCF+WPF.

Matthieu Mézil
Consultant – Winwise

Michel Perfetti
Consultant – Winwise – MVP C#

Allez plus loin avec Expression



Avant de s'intéresser aux outils, il n'est pas inutile de revenir un instant sur ce qui compose la gamme Expression : Design, Web, Blend et Media.

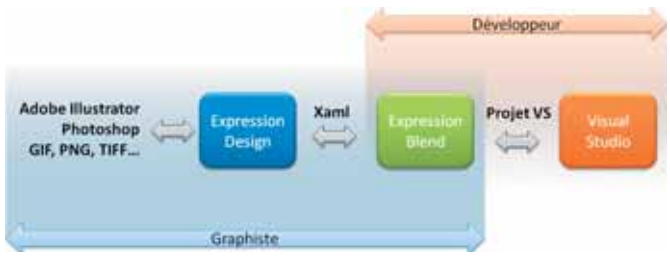
Expression Design est le logiciel de création graphique et vectoriel de Microsoft. C'est généralement par ici que débute l'aventure XAML de votre projet. Il est destiné aux graphistes, mais il a bien du mal à égaler son aîné : Adobe Illustrator. Heureusement qu'Expression Design sait importer les fichiers AI !

Si vous souhaitez développer un site Web, c'est **Expression Web** qu'il vous faut. Qu'il s'agisse d'un site HTML à l'ancienne ou en ASP.Net 2.0, vous disposerez d'un outil nettement plus évolué que Visual Studio, du niveau d'un Dreamweaver. C'est une excellente surprise.

Si vous souhaitez vous lancer dans le développement d'applications WPF (et je vous le souhaite vivement !),vous utiliserez **Expression Blend.** Cet outil a plutôt vocation à satisfaire votre âme de designer, mais en pratique les développeurs devront également mettre la main à la pâte afin d'aider les designers à résoudre des problèmes nécessitant une connaissance technique (comme le data binding sur les objets par exemple).

Enfin, **Expression Media** permet de manipuler toutes sortes de vidéos. On peut les organiser, écrire des scripts pour automatiser des conversions, publier les vidéos sur un site Web ou sur un CD. Les possibilités sont abondantes, et c'est pour moi la deuxième bonne surprise de la gamme Expression. A noter qu'Expression Media est également disponible pour Max OS X. Pour les utilisateurs Windows il s'accompagne d'Expression Encoder : le moteur d'encodage à utiliser derrière votre site Silverligth. Je ne l'ai pas encore essayé, mais les premiers retours glanés sur le net sont plus que positifs sur les possibilités offertes.

Voici le contexte typique des outils Expression dans un projet .NET à forte composante graphique.



On dispose maintenant d'une gamme d'outils allant du 100% graphiste au 100% développeur. Les va-et-vient entre les outils se déroulent plutôt bien, et pour une fois on peut vraiment exploiter efficacement les compétences de chacun. On va enfin voir apparaître de belles applications .NET sur nos PC. Par contre, les limitations d'Expression Design poussent à l'utiliser comme simple outil de conversion vers XAML et il a du mal à tenir son rôle de " centre de la créativité graphique ", ce qui est bien dommage.

■ **Eric GROISE**
Senior Offering Leader à Octo Technology www.octo.com

Retour d'expérience sur Expression Design et Blend

Expression Design peut être considéré comme une version (très) simplifiée d'Adobe Illustrator. L'interface, qui ressemble à celle d'un logiciel de graphisme vectoriel standard, est sobre et plutôt séduisante. Design risque pourtant d'être mis au placard assez vite.

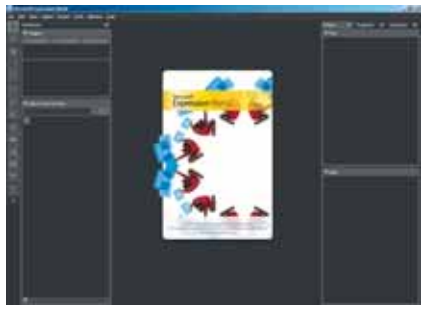
En effet, dans le cas de formes simples, Blend proposera une palette d'outils suffisante. Si jamais elle ne vous satisfait pas, il y a fort à parier que celle de Design n'y parviendra pas non plus. En situation d'urgence, vous pouvez toujours vous rabattre sur Illustrator. Créez vos objets, copiez-les et collez-les dans Design, oui, ça fonctionne ! Et pour une cohabitation complète entre Illustrator et les logiciels de la gamme Expression, essayez le plug-in développé Mike Swanson "Adobe Illustrator to XAML" http://www.mikeswanson.com/xamlexport/

Expression Blend, l'occasion de mettre le nez dans le code...

Si l'on fait rapidement le lien entre Design et Illustrator, le rapprochement entre Blend et Flash est encore plus évident. Sans revenir sur les fonctionnalités de Blend, sachez qu'on y trouve tout ce qu'il faut pour créer des applications de A à Z et, sans avoir à écrire une seule ligne de code. Dans la pratique, ceux qui désirent réaliser des choses plus complexes se rendront vite compte qu'on ne peut pas tout faire à partir de l'interface de Blend et que, de temps en temps, on n'aura d'autre choix que de travailler directement dans le code. Voilà pourquoi une bonne connaissance de XAML s'avère indispensable.

Blend est tout d'abord, l'outil de dégradés de couleurs (qu'on retrouve dans Design), qui est vraiment très complet et pratique. Dans Expression Blend 2, de nouvelles fonctionnalités ont fait leur apparition concernant les polices de caractère.

En effet, les perfectionnistes ont maintenant la possibilité d'inclure leurs propres fontes à leur application, sans avoir à redouter qu'elles soient remplacées par des fontes système, si l'utilisateur ne les possède pas. Mieux encore, en vue d'optimiser le poids de l'application, on peut choisir de n'inclure que les caractères nécessaires.



Comment aborder WPF/Silverlight : suivre une formation...

Quelques organismes proposent des formations WPF/Silverlight, de l'initiation à la réalisation d'applications complètes, en voici un rapide tour d'horizon.

Winwise http://www.winwise.fr
- Développer des applications Silverlight (2 jours)
- développer des applications WPF (3 jours)
- développement d'interfaces riches avec WPF (5 jours)

Regart.net http://www.regart.net
- Expression Blend, création d'applications riches bureautiques connectées (5 jours)
- Expression Blend/Design QuickStart, présentation pédagogique du logiciel Expression Blend (2 jours)

Demos http://www.demos.fr
- Microsoft Expression Design, initiation (3 jours)
- Microsoft Expression Blend, initiation (3 jours)

...ou bien se former soi-même.

Bien que n'ayant pas trouvé de vrai bon didacticiel sur le web, il existe une multitude de sites remplis d'informations. En cas de blocage, il y a presque toujours la réponse sur internet, mais pas forcément tout au même endroit (et les premiers ouvrages arrivent, ndlr).

La collaboration graphistes/développeurs

Dans le cas d'un projet conséquent, l'équipe "idéale" sera constituée de développeurs d'un côté (sous Visual Studio), de graphistes de l'autre (sous Blend), et d'un intégrateur. Cet intégrateur sera chargé de faire la passerelle entre les graphistes et les développeurs; il devra donc posséder de solides bases dans les différents domaines, avoir en tête les contraintes de chacun, et saura communiquer avec les uns et les autres de manière à ce que tout le monde puisse se comprendre. Très souvent, cet intégrateur n'existera pas. Développeurs et graphistes devront donc collaborer étroitement et se mettre d'accord d'emblée sur le point sensible, à savoir l'intégration de la partie "interface graphique" dans la partie "code" (ou l'inverse). Et c'est là que les graphistes devront faire preuve de discipline : une nomination claire des objets sera indispensable. Pour une relation durable, connaissance de XAML et notions de programmation seront les bienvenues. Sur des projets pas forcément très lourds, Blend risque de "quitter inopinément" assez souvent et des ralentissements sont à prévoir... Pour conclure, l'utilisation de ces logiciels en production est tout à fait possible, même si les utilisateurs devront rester indulgents vis-à-vis de ces outils qui n'en sont qu'à leurs débuts. Pour ma part, je pense que l'enthousiasme pour cette nouvelle technologie l'a nettement emporté sur les petits désagréments causés par ces premières versions.

Antoine Leblond
Designer graphique indépendant
http://www.tichat.com
antoine.leblond@gmail.com

Construire une application WPF avec Expression Blend et VS 2008

Nous vous proposons aujourd'hui de créer ensemble une application WPF reprenant le concept du Post-It que l'on met souvent sur nos bureaux. Pour ce faire, nous allons utiliser Expression Blend et Visual Studio 2008.

Vous trouverez les sources utilisés pour cet article ici : www.luminal.com/SimplePostIt.

Notre application va comporter deux types de fenêtres : le block de notes qui permettra de créer de nouvelle notes, et les notes elles-mêmes.

Designer les écrans dans Blend

Fig.1

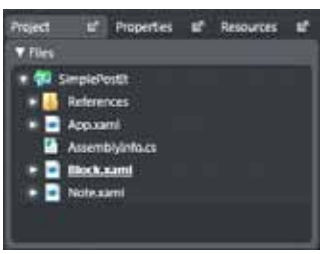


Fig.2

Commençons donc par lancer Expression Blend afin de créer un projet WPF (Fig.1).

Par défaut, il va créer un premier écran Window1.xaml. Renommons-le en Block.xaml, puis ajoutons un second écran nommé Note.xaml (Fig.2).

Pour le block, nous allons utiliser une image qui en représente un. Pensez à retirer les décorations de la fenêtre (WindowStyle : None) ainsi qu'à la rendre transparente (AllowsTransparency). On va également ajouter deux boutons : un pour fermer l'application, et un second pour créer une nouvelle note. Mes boutons ne sont pas beaux, mais comme la majorité des développeurs, mes compétences de graphistes sont limitées ;-) (Fig.3).

Pour la Note, on va utiliser un rectangle jaune dans lequel on ajoute une RichTextBox qui permettra à l'utilisateur de taper son texte. Ici aussi on va ajouter ce disgracieux bouton rouge qui permettra de fermer la note (Fig.4). Afin de permettre à l'utilisateur de modifier la taille de la note, faisons apparaître un " Resize Grip " (ResizeMode : CanResizeWithGrip).On a terminé la partie design des " Window ".

Ajouter une animation dans Blend

Nous allons maintenant ajouter une animation qui permettra à la note d'apparaître avec un effet de zoom. Tout en restant sur la note nous allons créer une nouvelle entrée dans " Objects and Timeline " (Fig.5).

La ligne verticale jaune indique votre position actuelle dans votre animation. Tout en laissant la barre sur le zéro, modifiez la taille de votre note (propriétés Width et Height) pour la rendre toute petite. Ensuite Déplacez cette ligne sur le 1 (1 seconde plus tard), puis modifiez de nouveau la taille de la note pour lui redonner une taille normale. Les petits ovales qui apparaissent indiquent les points d'actions de votre animation.

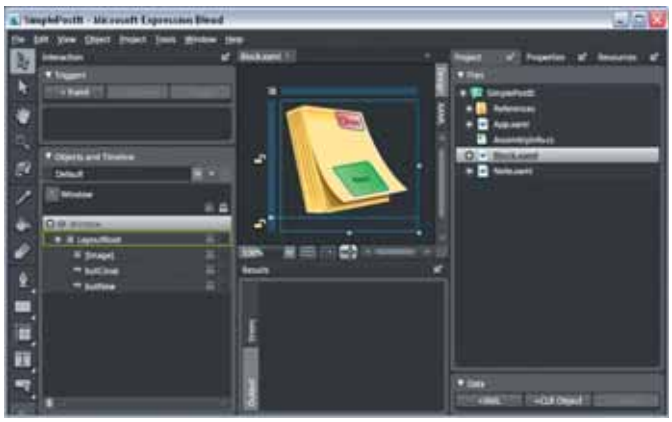


Fig.3

Fig.4

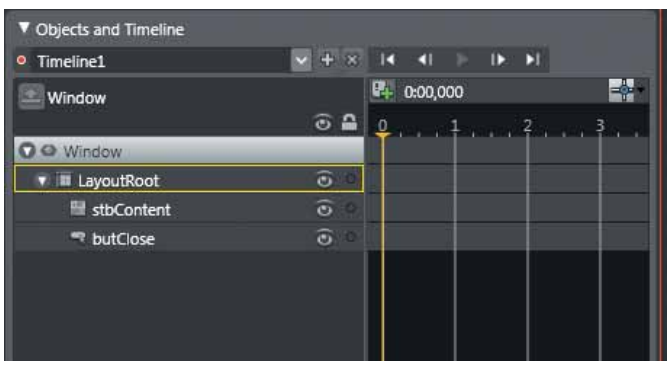


Fig.5

On va faire de même avec la position de la note afin de la faire déplacer vers 400,400. On va également faire apparaître le bouton close à la fin de l'animation en jouant sur la propriété Visibility. Vous pouvez contrôler à tout moment le résultat de votre animation avec les boutons play, next, previous…. (Fig.6). Normalement, Blend a automatiquement défini le déclenchement de votre animation sur l'évènement Loaded de votre Note. Donc, à

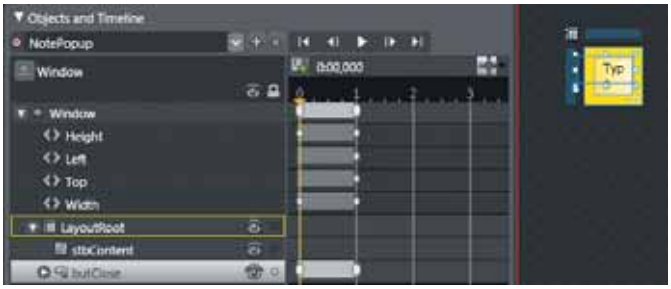
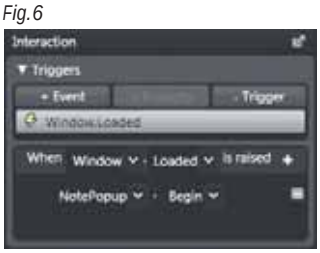


Fig.6



chaque fois que l'on créera une note, on aura droit à cette animation (Fig.7).

Ajoutons le code dans Visual Studio

Passons maintenant aux choses sérieuses : le code. Nous fermons notre projet depuis Blend pour aussitôt le ré-ouvrir avec Visual Studio. Pour le block, j'ai nommé ma classe BlockWindow, et pour la note, NoteWindow. Nous allons ajouter à notre BlockWindow un attribut qui contiendra la liste des notes créées.

```
static List<NoteWindow> _notes = new List<NoteWindow>();
```

Sur le click du bouton New, nous allons créer une note et la stocker dans notre liste. Ajoutons dans le XAML la gestion du click…

…puis ajoutons le code suivant :

```
private void butNew_Click(object sender, RoutedEventArgs e)
{
    NoteWindow newNote = new NoteWindow(this);
    _notes.Add(newNote);
    newNote.Show();
}
```

Pour le bouton close, ce sera :

```
this.Close();
```

Et on n'oublie pas de fermer toutes les notes lorsque le block se ferme :

```
private void Window_Closing(object sender, CancelEventArgs e)
{
    foreach (NoteWindow note in _notes)
    {
        note.Close();
    }
}
```

Nous allons aussi rajouter une méthode qui permettra aux notes de signaler au block qu'elles ont été fermées.

```
public void RemoveNote(NoteWindow pNoteWindow)
{
    _notes.Remove(pNoteWindow);
}
```

Et pour en finir avec le block, il nous reste à appeler la méthode magique DragMove sur le MouseDown qui permettra à l'utilisateur de déplacer le block à l'écran. Pour la note, nous allons aussi appeler Close() sur le click du bouton Close, et également appeler DragMove pour permettre le déplacement de la note. Et voila ! Vous pouvez lancer le build et admirer le résultat. Ce n'est pas encore Versailles, mais c'est un début.

Perfectionnons l'animation avec un peu de binding.

Le principal défaut de notre animation est que les notes apparaissent depuis le coin en haut à gauche de notre écran (0,0), alors qu'il serait plus logique qu'elles apparaissent depuis notre block. Une solution possible est d'utiliser une classe intermédiaire qui stockera la position du block afin de permettre à l'animation d'utiliser cette position grâce au data binding. Voici cette classe :

```
public class BlockPosition
{
    public static double _Top;
    public static double _Left;

    public double Top { get { return _Top;}}
    public double Left { get { return _Left;}}

    public static void SetPosition(double pTop, double pLeft)
    {
        _Top = pTop;
        _Left = pLeft;
    }
}
```

Nous allons appeler SetPosition sur le click du bouton New du bloc.

```
BlockPosition.SetPosition(Top, Left);
```

Maintenant, retournons dans Blend. Ouvrons notre Note et ajoutons un data binding sur les propriétés Left et Top de notre Window. Pour ce faire, cliquez sur le petit carré à droite de la propriété Left, sélectionnez Data Binding et allons chercher notre classe BlockPosition (bouton +CLR Object). Sélectionnez bien la bonne propriété Left et répétez l'opération cette fois ci pour la propriété Top. Ainsi, lorsque l'on crée une nouvelle note, sa position initiale sera celle du block.

Pour aller plus loin

La prochaine étape consistera à sauvegarder dans un fichier XML le contenu de nos notes afin qu'on les retrouve une fois l'application relancée. La seule chose à savoir est : comment on récupère le contenu (en XAML) d'une RichTextBox :

```
String contenuXaml = XamlWriter.Save(richTextBox.Document);
```

Et pour recharger le texte dans la RichTextBox :

```
MemoryStream mstream = new MemoryStream(System.Text.Encoding.
UTF8.GetBytes(contenuXaml));
richTextBox.Document = (FlowDocument)XamlReader.Load(mstream);
```

■ **Eric GROISE**
egroise@octo.com - Senior Offering Leader à Octo Technology www.octo.com

Expression Design : un nouvel outil de création graphique

Expression Design est un outil destiné aux professionnels du monde du graphisme et de l'illustration. Ce logiciel permet de créer, illustrer et construire tout type de document graphique. Au cours de cet article nous allons étudier les points forts d'Expression Design via une brève présentation et nous aborderons la façon dont ce logiciel va pouvoir venir s'intégrer dans un fil de production d'entreprise.

Si vous souhaitez tester ce logiciel, nous vous invitons à télécharger sa version d'essai disponible à cette adresse : http://www.microsoft.com/expression/products/download.aspx?key=design. Le SP1 de cet outil est aussi disponible et permet d'améliorer les options d'export de fichier XAML.

Une approche vectorielle

Expression Design permet de créer des illustrations sophistiquées pour quasiment tout type de support. Ce logiciel facile à la prise en main regroupe des outils standard d'illustration tels que la plume, les dégradés vectoriels, les " brush ", l'outil texte vectoriel, les outils géométriques et bien plus encore. Ce logiciel propose de plus, des outils innovants, en voici un bref aperçu : !

- " Polyline ", outil permettant de créer des arcs de cercle. Voici un exemple de tracé que l'on peut obtenir. @
- " B-Spline " permet de créer des formes dynamiques sans perdre des heures à travailler un tracé avec l'outil plume. Voici un exemple de forme que l'on peut obtenir #
- " Polygone ", enfin un outil pour réaliser des polygone à X côté(s) ! Il suffit de renseigner le nombre de côtés souhaité et tracer le polygone. Dans notre premier exemple nous avons tracé un triangle. C'est typiquement cet outil qui permet de réaliser des " stickers " de type web2.0 que l'on rencontre un peu partout sur Internet. Après avoir travaillé un polygone voici ce qu'on peut très rapidement obtenir. $

Toujours dans l'objectif du " tout vectoriel " nous pouvons retrouver, parmi les fonctionnalités les plus attrayantes de Expression Design, la possibilité de convertir des bitmaps classiques directement en vectoriel. %

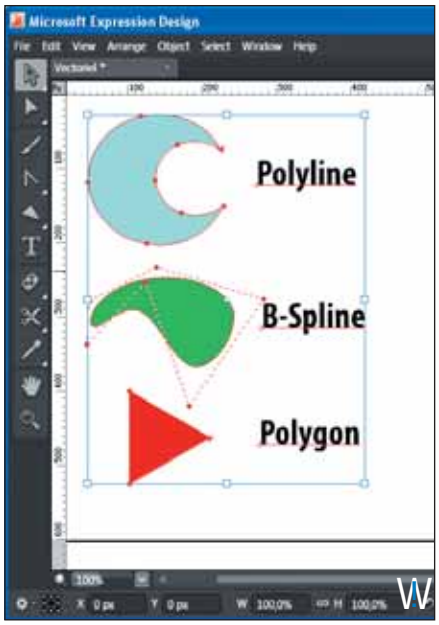
Ainsi, vous allez pouvoir exploiter ces bitmaps au format vectoriel directement via Expression Design comme suit : ^. Comme vous pouvez le voir tout les tracés sont éditables et votre bitmap a subi un véritable régime en terme de taille de fichier !

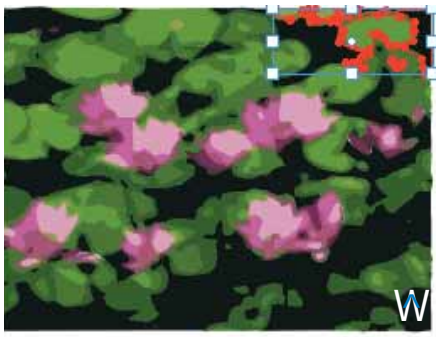
Un environnement de travail innovant

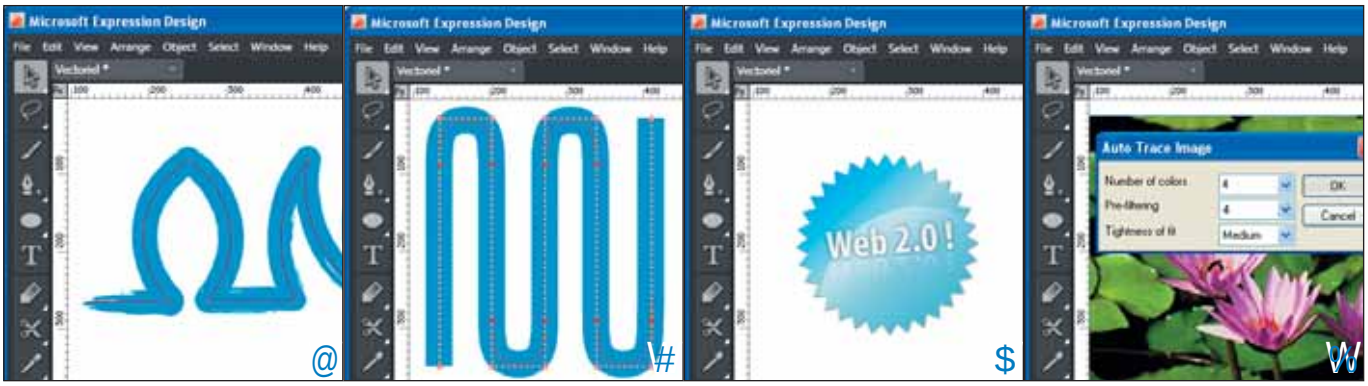
Lorsqu'on ouvre Expression Design pour la première fois nous tombons nez à nez avec une interface très caractéristique des outils de la gamme Expression Studio. &

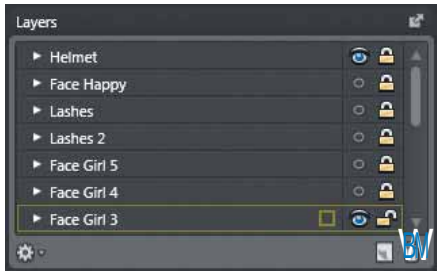
Comme le montre l'aperçu, l'interface est très sobre et très agréable à utiliser au quotidien. En effet, le gris/noir est plutôt reposant pour la vue, les icônes sont représentatives des actions illustrées et nous pouvons retrouver tous les éléments d'un logiciel de graphisme classique :

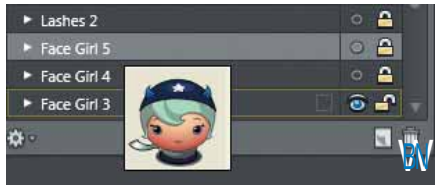
- Un menu classique situé en haut de la fenêtre regroupant toutes les actions pos-

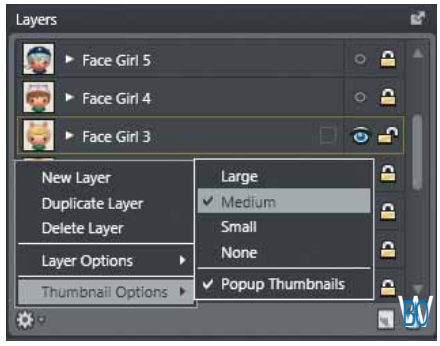


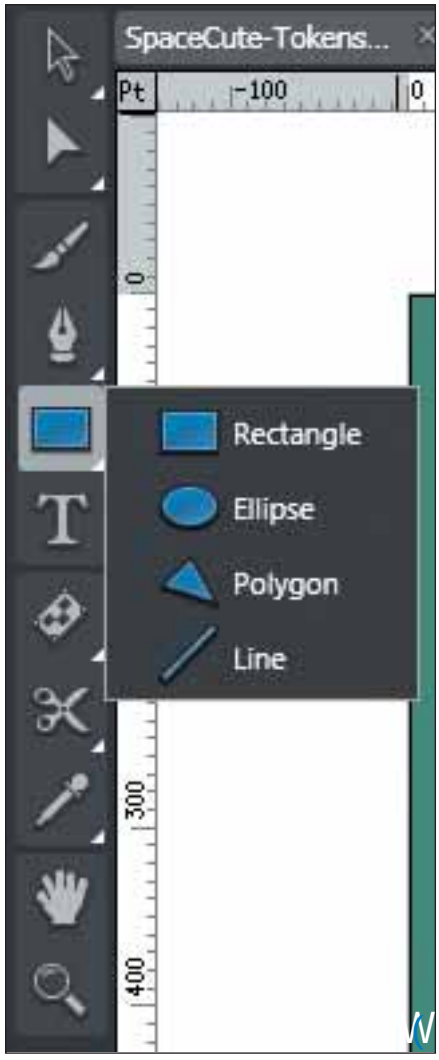


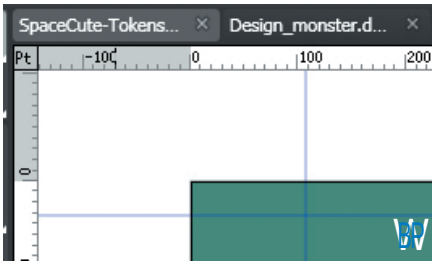






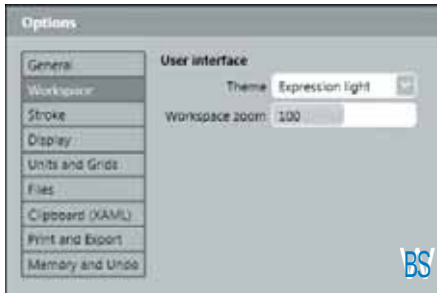






BQ

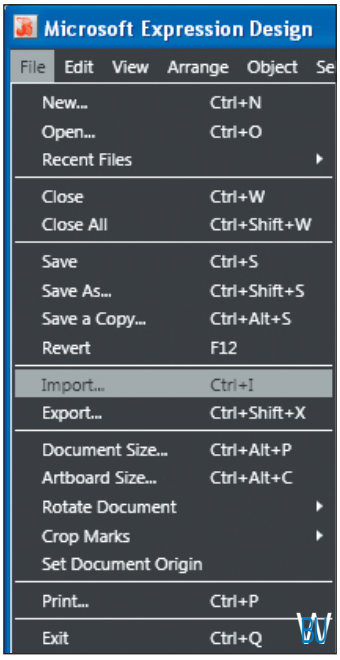
BR

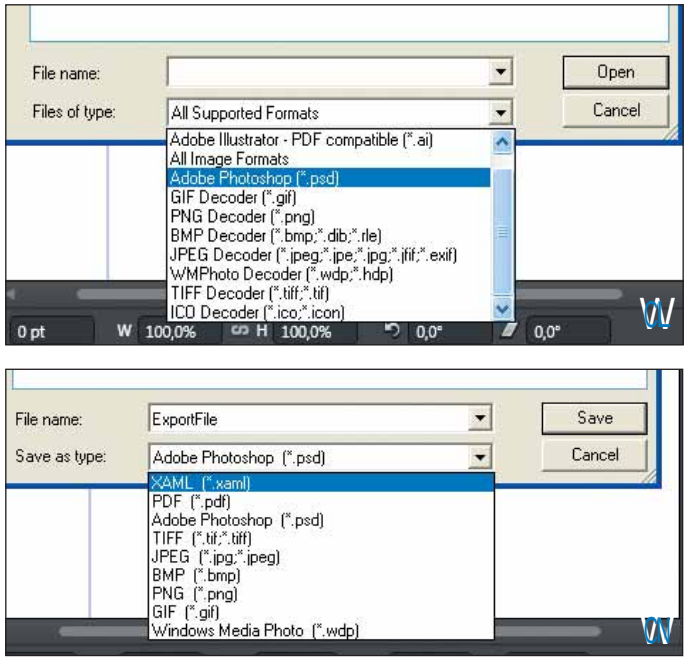


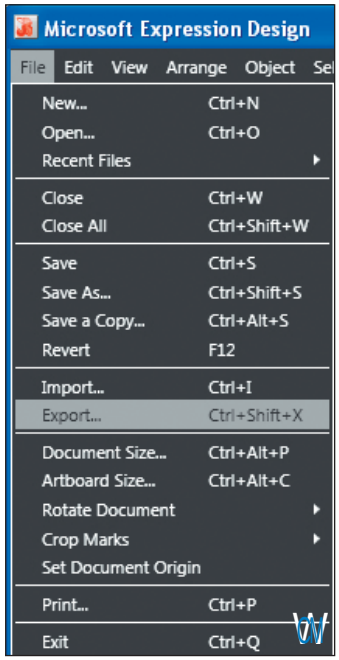
sibles et réalisables au travers du logiciel. *

- La barre d'outils alignée sur une colonne, non déplaçable, mais bien située (sur la gauche de l'écran). (
- La fenêtre des Propriétés, située à droite de l'écran et exposant toutes les propriétés de l'élément sélectionné (Touche V) dans l'espace de travail. Cette fenêtre comporte deux sous rubriques : *Apparences* dans laquelle on retrouve tout ce qui est de l'ordre de l'apparence de la sélection (couleur, taille, opacité, etc.) et *Effets* dans laquelle nous retrouvons les effets. Ceux-ci sont comparables à des filtres. (blur, glow, shadow, etc.) BL
- La fenêtre des Calques, située aussi à droite de l'écran qui permet d'administrer les calques au sein de votre document (ajout, suppression, duplication, verrouillage, masquage) BM. En passant la souris sur les différents calques, nous obtenons un aperçu de ce qu'ils contiennent. BN Il est possible, via les options, d'appliquer des couleurs aux calques afin de leur associer un type. Enfin, un aperçu dans le listing des calques est envisageable. BO
- L'espace de travail : celui-ci propose la possibilité de zoomer BP, appliquer des repères via les règles horizontales et verticales et enfin, il est possible de gérer de multiples documents via des onglets BQ. Malheureusement il n'est pas encore possible de mettre côte à côte deux documents distincts. Nous pouvons donc espérer que cette fonctionnalité sera intégrée dans la version 2.0.
- La barre d'action, permet d'accéder très rapidement aux actions que l'on peut effectuer sur une sélection (Touche V). BR Cette barre d'action permet un gain de productivité évident. Avec celle-ci, inutile d'aller rechercher dans les menus les actions correspondantes, vous avez tout, directement sous les yeux, et à portée de clic.

Enfin, il est possible de modifier totalement le rendu de l'espace de travail via la fenêtre *Options* accessible via le menu *Edit*. Grâce à cette option vous accédez à un style beaucoup







plus gris clair BS et avez la possibilité de " zoomer ", mais cette fois-ci sur toute l'interface du logiciel. Ceci peut s'avérer très pratique pour les personnes ayant de petites ou grandes résolutions d'écran, chose très courante dans le monde de la création graphique. Expression Design amène donc, en plus des fonctionnalités classiques d'un logiciel de création graphique, des fonctionnalités très intéressantes en termes de confort et de productivité.

Des effets de haute qualité

Comme souligné dans la partie précédente il existe une partie dédiée aux effets dans la fenêtre *Propriétés*. Ces effets ne sont rien d'autre que des filtres. Ils permettent de styler et modifier le rendu de vos objets en haute qualité. Expression Design travaillant essentiellement sur des bases vectorielles, les effets se devaient d'être fidèles à cette qualité de tracé. Pour créer un effet il suffit de cliquer sur l'icône FX. BT Comme vous pouvez vous en rendre compte il y a vraiment un choix conséquent de filtres. Surfaces, couleurs, teintes, ombrages… sont au rendez-vous. Cela apporte une réelle valeur ajoutée pour toutes sortes de créations, qu'elles soient graphiques ou illustratives.

Des formats ré-exploitables

Faire un nouveau logiciel de création graphique qui ne soit pas compatible avec les formats de fichier les plus utilisés du marché n'est pas concevable pour une entreprise qui souhaite percer dans ce domaine. Cela, Microsoft l'a bien compris, en donnant la possibilité à Expression Design d'importer et exporter différents types de formats.

L'import

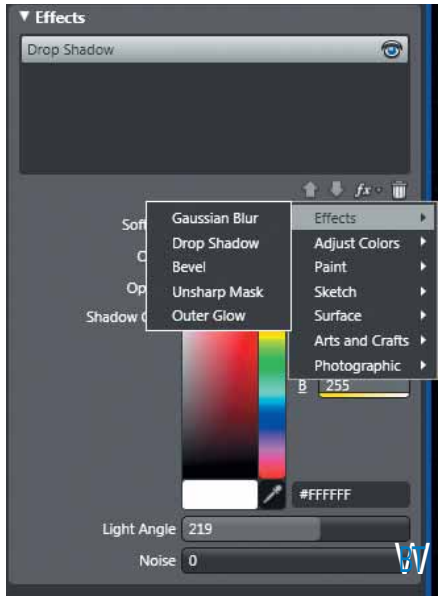
Il est possible d'importer des formats tels que .AI, .PSD, .PNG, .JPG, .GIF, .TIFF, .ICO, etc. Vous êtes donc capables d'exploiter vos anciennes créations réalisées avec des logiciels tiers directement dans Expression Design.BU CL

L'export

En ce qui concerne l'export, Expression Design donne la possibilité d'exporter au format .XAML, .PDF, .PSD, .TIFF, .JPG, .GIF. Pour information, les extensions XAML sont des fichiers de type XML, donc déclaratif, qui permettent de créer des interfaces utilisateurs avancées pour les applications de type Silverlight et Windows Presentation Foundation. En ce qui concerne les .PSD et les .PDF, les calques sont supportés lors de l'export. Vous pouvez donc directement exploiter vos créations et vos calques d'Expression Design dans d'autres logiciels supportant le format .PSD. CM CN

Conclusion

Pour un premier cru, Expression Design est un produit abouti. Tellement même, qu'il peut venir s'adapter et s'incorporer dans votre chaîne de création graphique. Facile à prendre en main, même les moins habitués aux logiciels de création graphique sauront rapidement obtenir des résultats pointus. Pour les professionnels, vous retrouverez toutes vos habitudes acquises avec d'autres logiciels de



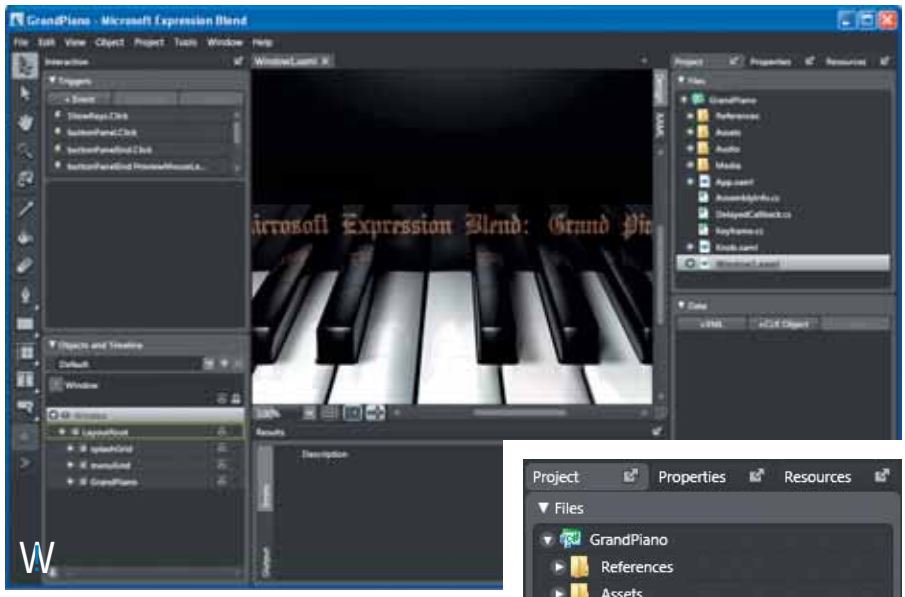
production graphique très rapidement et sans chercher ! Compatible avec la plupart des formats du marché, ce logiciel n'est pas un frein à votre production mais bien au contraire une valeur ajoutée quant aux nouveautés apportées, tant au niveau de la productivité qu'au niveau du confort de travail. Expression Design vient donc se positionner en concurrent de taille à tous les logiciels destinés aux professionnels du monde du graphisme et de l'illustration.

Guillaume André
Intégrateur .NET & Silverlight
http://www.wygwam.com/
http://blogs.codes-sources.com/guillaume

Expression Blend : Orchestrez vos applications

Expression Blend est destiné aux graphistes et aux intégrateurs. Ce logiciel leur permet de créer des interfaces riches en XAML. Au cours de cet article nous allons découvrir et aborder les principales nouveautés et caractéristiques de cet outil. Dans un premier temps voici comment se les procurer.

Expression Blend 1.0 :
http://www.microsoft.com/expression/products/download.aspx?key=blend
Expression Blend 2 preview :
http://www.microsoft.com/expression/products/download.aspx?key=blend2preview
La fonctionnalité fondamentale qui dissocie la version 1 et la version 2 d'Expression Blend est le support de projet de type Silverlight 1.0. Silverlight est une technologie cross-plateforme et cross-browser basée sur XAML, qui améliore les présentations à base de contenus riches (2D, animation, dessin vectoriel, vidéo et audio).

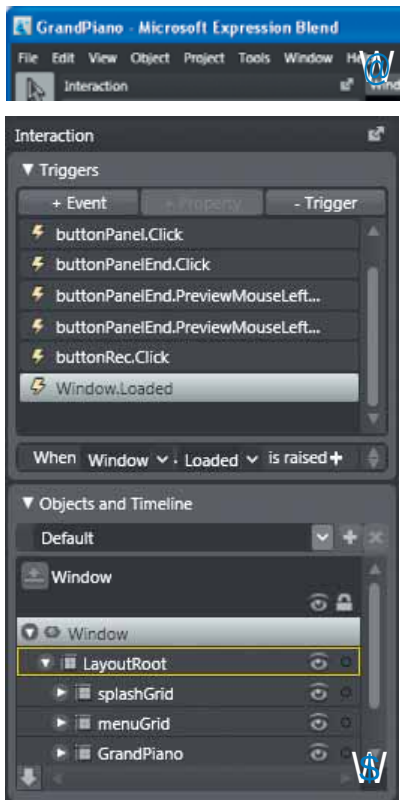


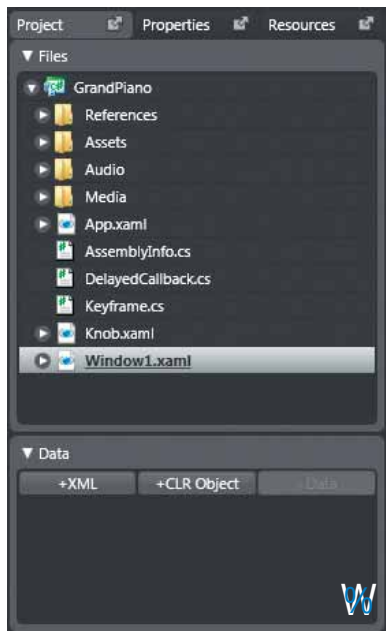
Un outil collaboratif

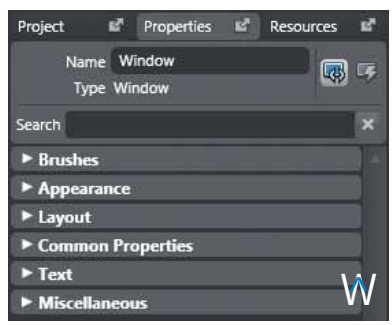
L'interface

Comme son confrère Expression Design, le logiciel Expression Blend bénéficie du même style graphique caractéristique des logiciels de la gamme Expression. !

- Sur la partie haute de l'écran, on retrouve un menu classique. Celui-ci permet d'accéder à toutes les fonctions proposées par le logiciel. @
- Sur la zone gauche de l'écran se trouve la barre d'outils, qui permet d'accéder aux outils classiques tels que la plume, la loupe, l'outil de dégradé, etc. #
- La fenêtre " interaction " : C'est ici que vous allez pouvoir créer vos propres animations en associant, pourquoi pas, certaines actions de l'utilisateur à des animations spécifiques. $
- La fenêtre " Project " où l'on retrouve toute l'arborescence du projet WPF, ou Silverlight sur lequel vous êtes en train de travailler. %
- La fenêtre " Properties ". C'est ici que se passe toute la partie *rendu.* Vous avez directement accès aux propriétés de l'objet sélectionné. Vous pouvez modifier les propriétés, appliquer des dégradés, des brushs, etc. ^
- La fenêtre " Resource ", permet d'ajouter des ressources au projet. On peut entendre par







ressource l'aspect " styling " d'un bouton par exemple.

- La fenêtre " Results " retournera les erreurs éventuelles de XAML et de code, lors de la compilation de votre application.

Son interface repose sur la technologie Windows Presentation Foundation et est de ce fait personnalisable. L'interface est vectorielle, il est donc possible d'ajuster la taille de la zone de travail mais aussi la taille de zoom de l'interface globale via les options : " Tools > Options... " > &.

Pour les graphistes curieux, vous pourrez visionner le code XAML que vous générez automatiquement via les onglets " Design " / " Xaml " : *

Collaboration avec les développeurs

Via la fenêtre " Project ", vous accédez à votre projet. Il en existe 2 sortes. Les projets VB : " VBPROJ " et les projets C# " CSPROJ ". (

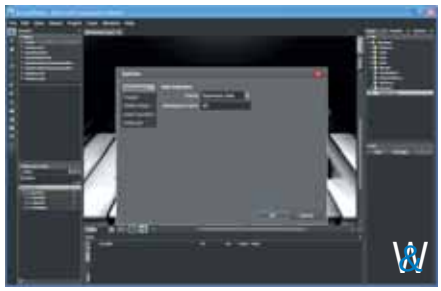
Ceci permet aux développeurs de choisir le langage de programmation qu'ils vont utiliser. mais n'a strictement aucun impact pour le graphiste. Les projets générés par Blend sont compatibles avec Visual Studio. Cependant, la gestion d'outil tiers qui permettent le partage de fichier entre membres d'une équipe tel que Visual Source Safes n'est pas encore supportée.

Créer les nouvelles interfaces

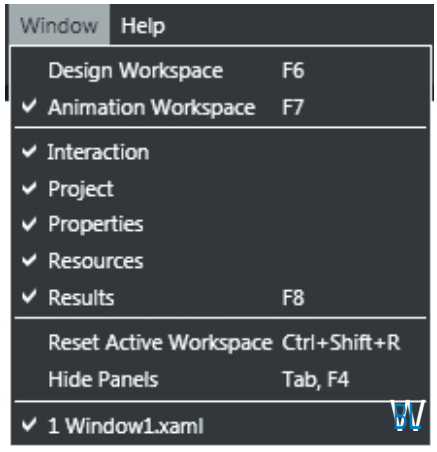
Avec Blend il est possible de créer des animations. Pour cela, il est préconisé de passer en mode " Animation Workspace " comme le montre l'illustration (touche F7). BL Nous allons à présent créer une animation très basique pour bien comprendre le fonctionnement de cet outil.

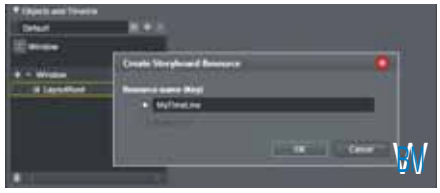
Commencez par créer un projet. " File > New Project ". Dans la fenêtre " Interaction " ajoutez une Timeline via le bouton " + ". BM Dans l'espace de travail, tracez un rectangle. BN Enfin, déplacez la barre de lecture (jaune) dans la timeline. Une fois cette opération effectuée déplacez le rectangle dans l'espace de travail BO. Appuyez sur lecture pour visualiser votre animation.

Comme vous pouvez vous en rendre compte il est très simple de créer soi même sa propre animation. Vous pouvez aussi associer des animations à des évènements. Par défaut, toute animation créée se lance sur le chargement de la page. Pour modifier, ajouter, supprimer des évènements, aussi appelés déclencheurs, dirigez vous dans la fenêtre intitulée " Triggers ". Cliquez sur " + Events " pour ajouter un évènement, choisissez l'élément déclencheur, dans la case " When " et choisissez l'évènement désiré. BP Dans notre exemple l'animation se lancera lors du " MouseDown " du rectangle.



```
Window1.xaml
1 <Window
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xml:lang="en-US"
5     xmlns:d="http://schemas.microsoft.com/expression/blend/2006" xmlns:mc="http://schema
6     xmlns:GrandPiano="clr-namespace:GrandPiano"
7     x:Class="GrandPiano.Window1"
```





Créer des interactions sans la moindre ligne de code

Toujours dans le but de faciliter la vie du graphiste, Expression Blend permet de créer des interactions entre les éléments de vos applications sans pour autant devoir écrire du code. Le " Data binding " en est sans doute l'exemple le plus approprié. Le " Data binding " va vous permettre de lier la propriété d'un objet à la propriété d'un autre objet. Typiquement, vous allez pouvoir associer la taille d'un rectangle à la valeur d'un slider, ceci permettant à l'utilisateur de redimensionner son rectangle. Le " Data binding " peut s'effectuer côté XAML et ne nécessite donc pas nécessai-

rement de code. Afin de comprendre le fonctionnement de " Data binding " nous allons réaliser un exemple concret. Le but de l'exercice va donc d'être de lier la taille d'un texte à la valeur d'un slider. Cela aura donc pour effet de faire une sorte de " zoom " sur le texte lorsqu'on animera le slider. Commençons cet exercice en incluant un élément " Slider " et un élément " TextBlock ". BQ. Sélectionnez l'élément " TextBlock " et dirigez-vous dans la fenêtre " Properties " onglet " Text ". Cliquez sur le petit carré situé à droite de la taille de la police BR et validez " Data Binding… ". BS La fenêtre de Data Binding apparaît. Dans l'onglet " Element property " choisissez l'élément " Slider " qui se trouve dans la colonne " Scene Elements ". Ensuite, dans la colonne de droite choisissez la propriété " Value ". Cliquez " Finish ", un liseré orangé apparaît autour de la textbox de la propriété bindé. BT Le " Data Binding… " est donc terminé, vous pouvez avoir un aperçu de votre travail dans l'onglet XAML. Vous pouvez aussi compiler votre application et visualiser le résultat à l'aide de la touche " F 5 ". En bougeant le Slider, le texte diminue ou augmente en taille. Vous pouvez bien entendu modifier les tailles minimales et maximales via les propriétés du Slider (Minimum et Maximum).

Conclusion

Expression Blend est un outil dédié aux graphistes et intégrateurs désireux de créer des applications riches avec des interfaces riches. Il s'agit de l'outil le plus poussé du marché en ce qui concerne la création d'interfaces riches Windows Presentation Foundation. Pensé essentiellement pour une utilisation WYSIWYG ce logiciel est très facile d'accès et permet d'arriver à des résultats très intéressants à en juger la qualité des exemples d'applications livrés avec le logiciel.

Guillaume André
Intégrateur .NET & Silverlight
http://www.wygwam.com/
http://blogs.codes-sources.com/guillaume

L'information permanente

Rubrique .Net :
actu, téléchargement etc.

- Les archives
- Newsletter hebdo
- Le téléchargement
- Les forums

www.programmez.com

Silverlight avec Blend 2 et Visual Studio 2008

La version 2.0 de Silverlight permet de développer en .Net. Pour créer l'interface, on utilisera Blend. Pour la partie code, on se servira de Visual Studio 2008. Ajoutez à cela les extensions Silverlight pour Visual studio 2008, ainsi que le SDK que je conseille vivement, notamment pour les contrôles inclus dedans et qu'on utilisera dans le projet.

L'application qui va être développée.

L'application est un " juke-box vidéo ", cela permet d'avoir une liste de vidéos qui se trouve sur le côté et, lorsqu'on clique sur l'une de celles-ci, l'application lance la lecture et l'affiche au milieu de l'application. Cette dernière sera composée d'une page principale, et d'un contrôle utilisateur qui sera répété dans la liste par rapport au nombre de vidéos que donne un fichier XML. Voici les données :

```
<?xml version="1.0" encoding="utf-8"?>
<Media>
  <item
    title="Panel s'ouvrant"
    description="Video d'un panel qui s'ouvre"
    image="panelOpening.jpg"
    video="panelOpening.wmv"
  />
  <item
    title="Video Sample"
    description="Video d'exemple de Blend"
    image="SampleVideo.jpg"
    video="SampleVideo.wmv"
  />
  <item
    title="Panel se fermant"
    description="Video d'un panel qui se ferme"
    image="panelClosing.jpg"
    video="panelClosing.wmv"
  />
</Media>
```

Ce fichier XML donnera le titre de la vidéo, une description, une image et l'url de la vidéo. Il sera lu par le code de la page qui remplira une listbox. Celle de l'exemple fait partie du Silverlight SDK qui contient plusieurs contrôles, relativement intéressants à utiliser.

Le contrôle utilisateur

Pour le contrôle utilisateur, on va créer un projet Silverlight 1.1 dans Blend 2 (dans la CTP de septembre, Silverlight 2.0 s'appelle encore Silverlight 1.1) (Fig.1). Dans le design, on va mettre deux textblock et une image pour former le squelette de notre contrôle utilisateur (Fig.2).
Il est important de nommer les contrôles qui sont utilisés car on accèdera aux contrôles par leur nom dans Visual Studio 2008. Pour cela, on sélectionne le contrôle qu'on veut manipuler et dans la fenêtre des propriétés on donne un nom. On peut aussi manipuler les couleurs ou propriétés (Fig.3). Le code XAML (généré automatiquement) est destiné aux graphistes, le programmeur n'aura donc pas à y toucher. Blend a aussi créé un ensemble de fichiers pour le projet (Fig.4). Ces fichiers sont identiques au projet et fichier solution, généré par Visual Studio 2008.

Le code avec Visual studio 2008

Une fois le design réalisé, il faut mettre le code qui permettra de créer une interface de propriétés vers les contrôles *textblock, images et canvas* internes. On va donc mettre des champs internes correspondant aux différents contrôles et les propriétés qui correspondent.

```
using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;

namespace MyProject
{
    public class ListImage : Control
    {
        Image _image;
        TextBlock _title;
        TextBlock _description;
        Canvas _rootCanvas;
        string _videoUrl;
         public ListImage()
         {
              Stream s = this.GetType().Assembly.GetManifestResourceStream("My
Project.ListImage.xaml");

              _rootCanvas = this.InitializeFromXaml(new System.IO.Stream
Reader(s).ReadToEnd()) as Canvas;
            _image = _rootCanvas.FindName("Thumbnail") as Image;
            _title = _rootCanvas.FindName("Title") as TextBlock;
          _description = _rootCanvas.FindName("Description") as TextBlock;
            this.Loaded += new EventHandler(ListImage_Loaded);

         }
        void ListImage_Loaded(object sender, EventArgs e)
        {
            UpdateLayout();
        }
        public new double Height
        {
            get { return ((FrameworkElement)this).Height; }
            set
            {
                ((FrameworkElement)this).Height = value;
                UpdateLayout();
            }
        }
        private void UpdateLayout()
```

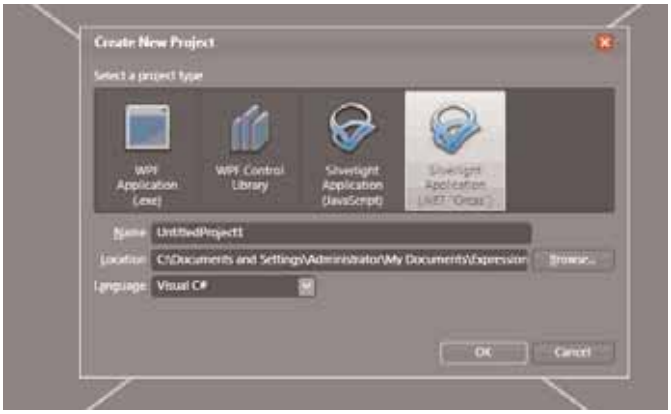


Fig.1

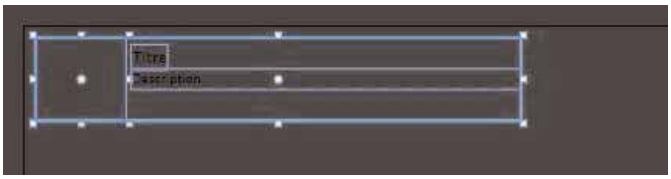


Fig.2

Fig.3

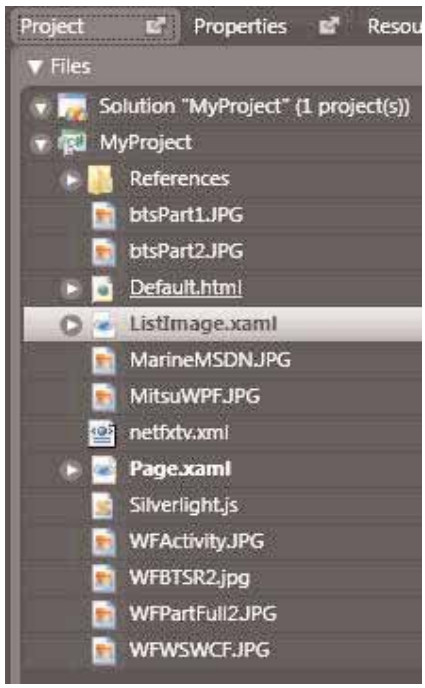


Fig.4

```
        {
            _rootCanvas.Height = Height;
            _rootCanvas.Width = Width;
        }

        public new double Width
        {
            get { return ((FrameworkElement)this).Width; }
            set
            {
                ((FrameworkElement)this).Width = value;
                UpdateLayout();
            }
        }

        public string Title {
            get {
                return _title.Text;
            }
            set {
                _title.Text = value;
            }
        }
        public string Description {
            get {
                return _description.Text;
            }
            set {
                _description.Text = value;
            } }
        public Uri ImageUrl
        {
            get
            {
                return _image.Source;
            }
            set
            {
                _image.Source = value;
            }
        }
        public string VideoUrl {
            get {
            return _videoUrl;
            }
            set {
                _videoUrl = value;
            }
        }
    }
}
```

L'initialisation du contrôle nécessite de s'y attarder. Prenons les deux premières lignes d'initialisation des champs internet :

```
_rootCanvas = this.InitializeFromXaml(new System.IO.StreamReader(s)
.ReadToEnd()) as Canvas;
_image = _rootCanvas.FindName("Thumbnail") as Image;
```

La première ligne lit le code XAML, le canvas étant le root du XAML, on récupère celui-ci pour le mettre dans _rootCanvas. La ligne suivante recherche le contrôle Thumbnail qui est dans le XAML. Ce contrôle étant une image, on le met donc dans _image.

La page

Le rôle de la page va être d'héberger le contrôle *ListBox* et de le remplir avec les contrôles *ListImage.* La page va aussi héberger le *media Player* pour le lancer. Pour le design, on va aller chercher le contrôle dans les *Custom contrôles* (Fig.5). On va positionner notre *listbox* sur la droite de la fenêtre, puis un *ScrollViewer,* dans le cas ou notre *listbox* serait trop longue. On va aussi ajouter un *media element* pour afficher la vidéo (Fig.6).

Le code avec Visual Studio 2008

Le code est relativement simple, on ouvre le fichier XML (pour le projet il est compilé en ressource embarquée, mais il pourrait être lu à partir du web). Comme il n'y a pas de datasource en Silverlight, on lit le fichier " à l'ancienne ", c'est-à-dire qu'on va boucler dessus et pour chaque item on construira un objet *ListImage* et on remplira les propriétés de l'objet. Ce dernier sera ajouté à la *listbox.*

```
using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using Silverlight.Samples.Controls;
using System.Xml;

namespace MyProject
{
    public partial class Page : Canvas
    {
        public Page()
        {
            this.Loaded += this.Page_Loaded;
        }

        private void Page_Loaded(object sender, EventArgs args)
        {
            // Required to initialize variables.  Needs to be done from loaded
 event so FindName works properly.

            InitializeComponent();

            System.IO.Stream s = this.GetType().Assembly.GetManifest
ResourceStream("MyProject.netfxtv.xml");
            ListBox listbox = this.FindName("ListBox1") as ListBox;

            Canvas standinCanvas = this.FindName("standinCanvas") as
Canvas;
            ((Canvas)(standinCanvas.Parent)).Children.Remove(standin
Canvas);
            scrollViewer.Content = standinCanvas;

            XmlReader xr = XmlReader.Create(s);
            while (xr.Read())
            {

                if (xr.IsStartElement("item"))
                {
                    string title = xr.GetAttribute("title");
                    string description = xr.GetAttribute("description");
                    string image = xr.GetAttribute("image");
                    ListImage li = new ListImage();
                    Uri u = new Uri(image, UriKind.Relative);
                    li.ImageUrl = u;
                    li.Title = title;
                    li.Description = description;
```

Fig.5

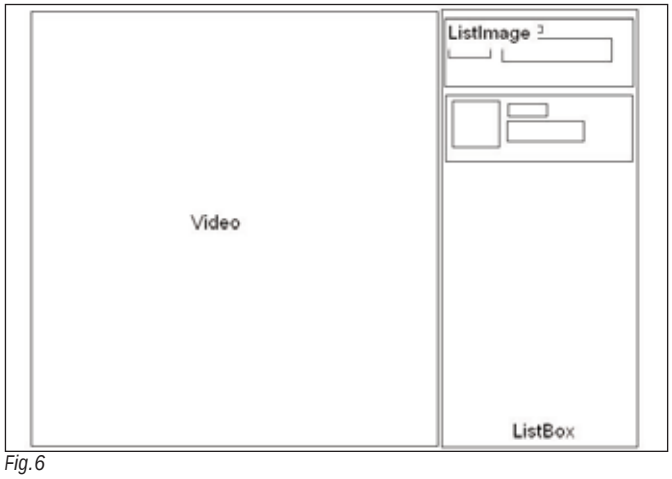


Fig.6

```
                    li.Height = 110;
                    li.Width = 500;
                    li.VideoUrl = xr.GetAttribute("video");
                    listbox.Items.Add(li);
                }
            }
            listbox.UpdateItems();
            listbox.SelectionChanged += new EventHandler(listbox_Selection
Changed);

            // Insert code required on object creation below this point.
        }

        void listbox_SelectionChanged(object sender, EventArgs e)
        {

            mPlayer.Source = new Uri(((ListImage)ListBox1.SelectedItem).Video
Url, UriKind.Relative);
            mPlayer.Play();
        }
    }
}
```

Pour finir, on intercepte la sélection d'un des items de la listbox. Lorsqu'un élément de celle-ci est sélectionné, on change la source du *media élément* et on lance l'application.

■ **Xavier Vanneste** - *Xavier.VANNESTE@ict7.com* - *ICT7 luxembourg*

Silverlight Streaming par la pratique

Silverlight Streaming est un service de Microsoft qui permet aux développeurs et aux designers de diffuser leurs applications Silverlight sur Internet et de rendre ainsi leurs contenus multimédias et applications riches, accessibles à tous.

Silverlight streaming propose un modèle gratuit jusqu'à 4Go de stockage et 1 000 000 de minutes de vidéo/mois en streaming. Ce service nécessite un compte LiveID. A terme il sera possible d'outrepasser ces limitations de stockage via des options payantes et financées par de la publicité.

Pourquoi profiter du service Silverlight Streaming ?

Silverlight Streaming est une solution gratuite d'hébergement d'application Silverlight. Ce service s'adapte à la demande des utilisateurs pour fournir un service performant. Suivant le lieu où vous vous trouvez, Microsoft sélectionne le serveur adéquat. C'est le système CDN (Content Delivery Network) qui permet une mise en cache similaire aux services Akamai. Ainsi, fini les temps d'attente interminables !

Comment profiter du service Microsoft Silverlight Streaming ?

Pour profiter du service Microsoft Silverlight Streaming il suffit de vous inscrire sur le site http://streaming.live.com/ avec votre compte LiveID. Une fois inscrit sur le service, un identifiant ainsi qu'une clé vous sont attribués. Ces identifiants vont vous permettre d'accéder au service, via l'interface web proposée sur le site ou bien via une API. !

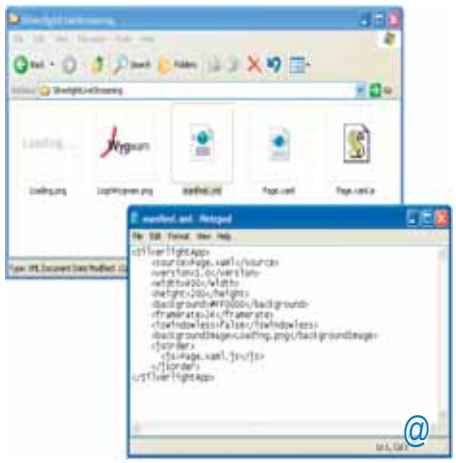
Héberger votre application Silverlight

Comme énoncé ci-dessus, Silverlight Streaming donne la possibilité d'héberger n'importe quelle application Silverlight. Cependant, vous allez devoir faire quelques adaptations avant d'exposer votre application sur la plate-forme Live.

1 Au sein de votre application il va falloir utiliser un fichier manifeste.

Ce fichier va permettre à la plate-forme Live de générer la page recevant votre application correctement. Voici un exemple de fichier manifeste : @

```
<SilverlightApp>
    <source>Page.xaml</source>
    <version>1.0</version>
    <width>400</width>
    <height>200</height>
    <background>#FFFFFF</background>
    <framerate>24</framerate>
    <isWindowless>false</isWindowless>
      <backgroundImage>Loading.png</back
groundImage>
    <jsOrder>
      <js>Page.xaml.js</js>
    </jsOrder>
</SilverlightApp>
```

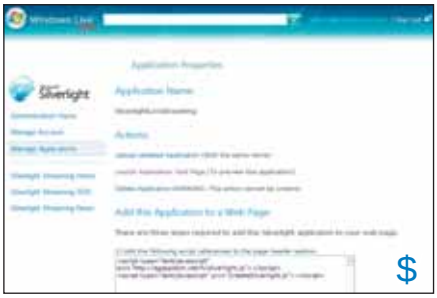


2 Vous allez aussi devoir vous assurer que tous les formats de fichier présents dans votre application sont autorisés par le service Silverlight Streaming. Le tableau ci-dessous vous donne la liste de ces formats.

Extension	MIME Type
Ttf	application/octet-stream
Js	Application/x-javascript
Dll	application/x-msdownload
Xaml	application/xaml+xml
Xml	application/xml
Zip	application/x-zip-compressed
Wma	audio/x-ms-wma
Wmv	audio/x-ms-wmv
mp3	audio/mpeg
Gif	image/gif
Jpeg	image/jpeg
Png	image/png
Txt	text/plain
Sdx	text/plain

A noter que si l'upload du fichier dure plus de 15 minutes, une erreur survient. Pour information, la MSDN nous précise qu'une vidéo de 22 Mo mettra 10 minutes à être uploadée avec une connexion de 300 Kbps.

3 Une fois ces points traités, vous pouvez zipper votre application et l'uploader via l'interface web si le ZIP pèse moins de 30 Mo. # Dès lors, une page vous explique comment insérer votre application au sein d'une page web via le service Live Streaming. Cependant, vous remarquerez que la solution proposée nécessite d'avoir accès à la partie <head> de votre page HTML $. Or, les bloggeurs le



savent bien, cette partie n'est pas souvent accessible lorsqu'on n'est pas propriétaire de son web/blog. Je vous propose donc en attendant une solution plus adéquate, de passer par une IFRAME.

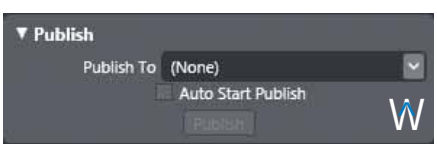
```
<iframe src="http://silverlight.services.live.com/invoke/<Account LiveStreaming>/<Nom de l'application>/iframe.html" scrolling="no" frameborder="0" width="400" height="200"></iframe>
```

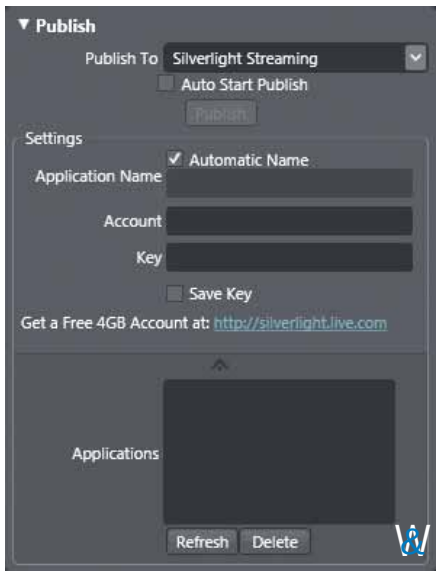
Héberger votre player Silverlight sur la plate-forme Streaming Live via Expression Encoder

Par défaut, Expression Encoder ne permet pas de publier une application Silverlight directement sur le service Live Streaming. Pour pallier ce manque il existe un plug-in mis à disposition par Microsoft, permettant d'ajouter cette fonctionnalité au logiciel. Ce plug-in, Silverlight Streaming Publishing Plug-In for Expression Encoder, est téléchargeable à l'adresse suivante :

http://www.microsoft.com/downloads/details.aspx?FamilyID=382a3306-b212-4df3-af86-5d48be550b94&displaylang=en

Il s'agit d'un fichier MSI qui va ajouter une DLL dans le répertoire " C:\Program Files\Microsoft Expression\Encoder 1.0\Plugins " de votre machine. Cette manipulation va donc





vous permettre de retrouver des nouveaux menus au sein de l'interface du logiciel *Expression Encoder* notamment en ce qui concerne la partie de publication. %

Une nouvelle partie intitulée " Publish " apparaît dans l'onglet " Output " et permet donc de renseigner la façon dont le document sera publié ^. Le plug-in installé permet ainsi de publier vers Silverlight Stream via une API. Il ne reste plus qu'à renseigner vos identifiants Windows Live ID et d'encoder la vidéo &. Surtout, n'oubliez pas de renseigner un template pour votre player ! * Une fois l'encodage et donc la publication terminée Expression Encoder génère le HTML à copier-coller dans vote site web et vous permet d'avoir un mini aperçu de votre application . (

En résumé :

- Installation du plug-in "Silverlight Streaming Publishing Plug-In for Expression Encoder"
- Création d'un projet Expression Encoder classique, insertion de la vidéo, mise en place des markers, etc.
- Renseignement des identifiants Live Streaming
- Choix d'un template pour le player Silverlight
- Encodage et récupération du code HTML permettant d'insérer l'application dans votre page web.

Vous pouvez télécharger le projet de test avec le fichier manifeste à l'url suivante : http://www.wygwam.com/guillaume/programmez/SilverlightLiveStreaming.zip

Quelques liens à retenir

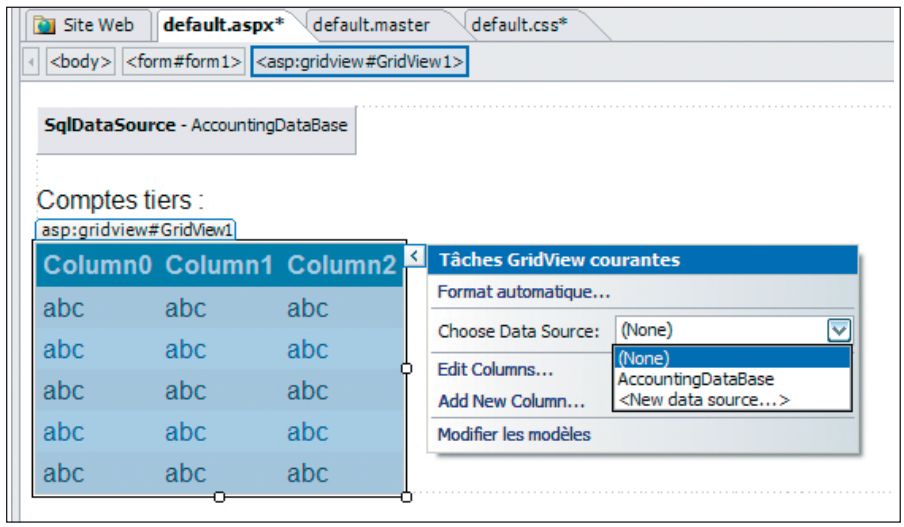
- Silverlight Streaming http://streaming.live.com/
- Silverlight Streaming pour les développeurs http://dev.live.com/silverlight/
- Microsoft Silverlight Streaming SDK http://msdn2.microsoft.com/en-us/library/bb851621.aspx
- Silverlight Streaming REST API http://msdn2.microsoft.com/en-us/library/bb851616.aspx

Guillaume André
Intégrateur .NET & Silverlight
http://www.wygwam.com/
http://blogs.codes-sources.com/guillaume

Expression Web au banc d'essai

La gamme Expression est le fer de lance de Microsoft dans sa conquête du marché des web designers. Détrôner les produits d'Adobe tels que Dreamweaver ou Flash n'est pas une mince affaire. Et à l'heure des RIA, il convient d'offrir, à ceux qui préféreront Silverligth à Flash, des outils de conception et de création aussi puissants qu'ergonomiques. Les produits Microsoft Expression proposent donc l'ensemble des outils nécessaires à la réalisation d'applications ou de sites web avec les dernières technologies : le " must have " du développement web… selon Microsoft.

Expression Web, dont il est question dans cet article, est l'équivalent de Blend pour des projets basés sur ASP.NET. Rien de révolutionnaire donc, si ce n'est le fait de mettre entre les mains des développeurs web un outil plus simple et mieux dimensionné, correspondant mieux à leurs attentes que Visual Studio.



Premier contact

Au premier lancement, on sera surpris de ne pas retrouver dans Expression Web l'interface "nouvelle génération", bien que parfois déconcertante, de Blend. Au contraire, l'éditeur a pris le parti de revenir à des valeurs sûres, en misant sur un look & feel classique, ce que l'on peut regretter au regard des prétentions novatrices du logiciel. Quand bien même l'interface d'Expression Web serait intuitive, l'exploration de quelques tutoriaux permet de repérer l'essentiel des fonctionnalités utiles. Beaucoup de ressources (webcast, exemples de code, etc.) sont disponibles sur la page de l'Expression Web Knowledge Center hébergé sur le site officiel consacré à Expression. Ce dernier promet " des outils innovants pour une nouvelle ère ". Concrètement, rien n'a révolutionné le développement web ces dernières années, hormis AJAX dont nous traiterons plus loin. Au contraire, l'utilisation rigoureuse de méthodes de travail et surtout des standards bien établis est la clé du succès dans un projet web un peu ambitieux : cette prétention de réaliser " des sites web basés sur des standards " est le deuxième slogan prometteur clamé sur la page web commerciale consacrée au produit.

Les outils

Expression Web propose donc l'ensemble des outils pour réaliser l'essentiel des fondements d'une application web. Tout d'abord un éditeur (X)HTML, toujours basé sur le double principe de conception visuelle et de réalisation par le code. Les avantages de ce système sont nombreux. Il permet de dessiner un canevas graphique, par le biais des master pages par exemple, sans qu'il soit nécessaire de maîtriser HTML. Expression Web évite l'écueil d'un code généré peu pertinent et peu lisible. La qualité du balisage et de l'indentation du code est plutôt bien préservée, même si l'on pousse l'outil dans ses derniers retranchements. Le codage manuel est facilité par IntelliSense qui permet l'autocomplétion des balises et de leurs attributs. On regrette cependant l'absence d'aide contextuelle précisant le rôle ou la signification d'un tag.

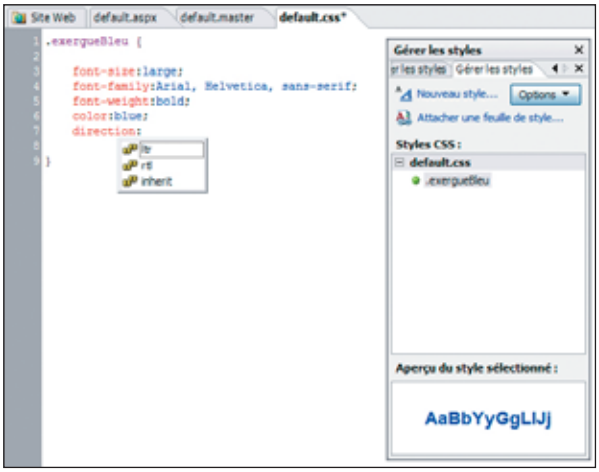
Pour chaque balise, la boîte de dialogue des propriétés de l'objet en cours d'édition, propose une liste exhaustive des attributs supportés et de leurs valeurs. Voilà un moyen pratique pour garder à l'esprit des attributs trop souvent sous-utilisés et pourtant indispensables !

IntelliSense ne propose que les ressources disponibles pour la version de (X)HTML choisie à la création du document. L'éditeur de code alerte l'utilisateur sur d'éventuelles erreurs ou incohérences, il signale également l'absence d'attributs qui nuirait à la validité du document.

L'outil de conception CSS est également assez intuitif, le codage des feuilles de style est également simplifié par l'autocomplétion qui propose à la définition d'un style les différentes propriétés disponibles, ainsi que les valeurs qu'elles peuvent prendre. Une fois encore, IntelliSense ne propose que les ressources disponibles pour la version de CSS choisie à la création du document. On peut regretter une fois de plus une aide contextuelle explicitant le rôle des propriétés. Comme pour les balises HTML, une fenêtre " propriétés " offre la liste des attributs d'une classe CSS et affiche également un aperçu rapide mais fort utile du rendu de cette même classe.

Coder en Javascript

Le développement JavaScript est aujourd'hui une étape indispensable dans un projet web digne du web 2.0 et d'Ajax. Expression web offre un support intéressant du code JavaScript. Il propose les fonctionnalités IntelliSense pour les fonctions basiques de JavaScript :



autocompletion, liste des arguments attendus par une méthode. Les ressources JavaScript extérieures telles que des bibliothèques comme Prototype, JQuery, Microsoft Ajax Library ou encore vos propres fonctions et classes, sont aussi prises en compte par IntelliSense dès qu'elles ont été référencées convenablement.

Côté serveur

Expression Web permet de structurer son projet selon les standards disponibles depuis ASP.NET 2.0. L'utilisation de pages maîtres (master pages) permet ainsi de conserver une unité visuelle dans l'organisation des pages. Il est possible d'utiliser la vaste bibliothèque de contrôles utilisateurs fournis en standard par ASP.NET. Les éléments les plus basiques comme les plus complexes peuvent être conçus et organisés au sein d'un page de façon visuelle et rapide par simple glisser-déposer. L'accès aux données XML, à celles de bases SQL, se réalise et se configure de cette même façon visuelle et intuitive. L'outil vient une fois de plus à la rescousse des développeurs les moins expérimentés et soulage les plus lassés par des tâches aussi fastidieuses. Cependant, dès qu'il s'agit de réaliser du code pour exploiter ses composants, les choses se compliquent. IntelliSense semble absent lorsqu'il s'agit de taper du code C# ou VB.NET… L'utilisateur, alléché par les possibilités d'ASP.NET, se trouvera donc fort dépourvu devant la pénibilité à l'exploiter dans Expression Web.

De plus, la logique de conception ASP.NET 2.0 n'est pas poussée au maximum La séparation du code et de la présentation est possible par le mécanisme standard de " code behind " qui permet d'associer une page avec un fichier de ressources en langages .NET qui sera compilé à l'exécution. L'interface d'Expression Web ne permet cependant pas de travailler intuitivement et confortablement de cette manière. Le développeur est donc plus ou moins incité à écrire le code serveur directement dans le document aspx… ce qui semble aller à l'encontre des bonnes pratiques en matière de développement ASP.NET. Les répertoires " spéciaux ", permettant de structurer encore mieux les ressources de code : classes personnalisées, entités métiers, couche d'accès aux données, etc. ne sont également pas explicitement disponibles dans la structure du projet web.

De toutes façons, il semble que l'objectif d'Expression Web est de cacher au maximum la complexité de réalisation et d'implémentation du code serveur. On réalise donc des choses assez intéressantes sans peine et très rapidement, mais sans pouvoir aller au-delà des " figures imposées " par l'outil pour, par exemple, afficher des données provenant d'une base, authentifier un utilisateur, etc.

Pour finir.

Si Expression Web a parfois l'allure de Visual Studio, il semble tout à fait incompatible avec les extensions de ce dernier. L'évolutivité du produit est donc très limitée. Il est ainsi frustrant de ne pas pouvoir profiter dans Expression Web des templates de projets web actuels et futurs comme ASP.NET MVC framework dont on parle déjà et qui pourrait devenir incontournable lorsqu'il s'agira de développer selon les règles de l'art…

Pas de modèles non plus pour la création de webservices ASP.NET. On pourra trouver cette limitation assez regrettable à l'heure du " vil mashup " et du " noble SOA ".

En conclusion, le public visé est donc celui des designers web, assez peu enclins à mettre les mains trop en avant dans le code. Beaucoup de fonctionnalités d'Expression Web sont là pour les aider sur leur chemin vers ce mirage qu'est de réaliser un site web dynamique sans effort… ASP.NET Developpement Server permet, par exemple, d'exécuter le site web au cours de son développement sur un PC familial puisqu'on peut ainsi se passer de IIS. Mais l'absence de débugger pas à pas rend le travail de réalisation ASP.NET vraiment délicat, quand on sait ce qu'apporte en qualité et en productivité un outil réellement taillé pour… tel que Visual Studio.

Le sentiment qui demeure après ce test est qu'Expression Web est un outil RAD assez intéressant, bien pensé pour l'intégration web standard et efficace, grâce à son éditeur HTML, CSS, et… JavaScript ! La partie ASP.NET peut également séduire ceux qui voudront rapidement mettre un peu code serveur dans leur site web, mais il faut garder à l'esprit les limitations peut-être rebutantes pour ceux qui savent aller plus loin dans la programmation .NET

■ **Arnaud Pagnier**
Eutech SSII

Tous les jours : l'actu et le téléchargement
www.programmez.com

Utiliser Diagrammer for .NET

Visual Studio et les services du Framework .NET permettent le développement d'interfaces utilisateur très variées, basées sur des bibliothèques standard de contrôles et sur des API graphiques. Dès que l'information nécessite un traitement vraiment convivial et interactif, l'effort de développement devient important et requiert une expertise particulière et un investissement rarement compatible avec le calendrier prévu.

Pour combler l'écart entre les potentialités de l'environnement .NET et les besoins réels des développeurs en matière de création d'interfaces graphiques, ILOG a récemment lancé Diagrammer for .NET, un ensemble d'outils, de composants et de bibliothèques pour créer des applications de visualisation, d'édition graphique ou de supervision. Les différents services de Diagrammer for .NET utilisés dans le cycle de développement, permettent de réaliser des interfaces riches, ou bien leur équivalent Web/Ajax.

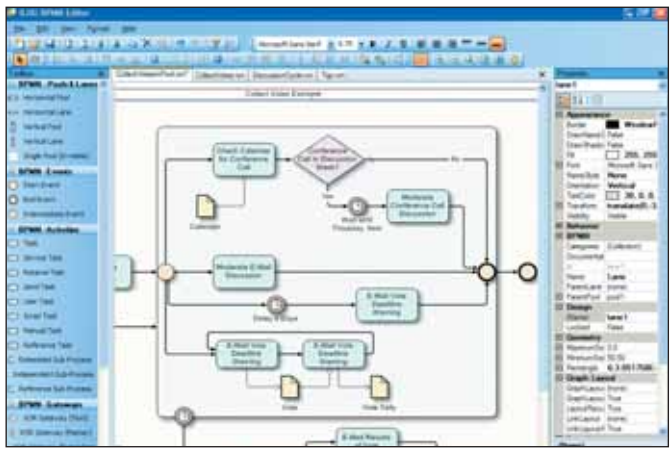
Diagrammes, Tableaux de bord et Editeurs

Diagrammer for .NET permet la création de diagrammes et de tableaux de bord, qui correspondent à des manières habituelles de présenter l'information et d'interagir avec des données. Les diagrammes montrent les interconnexions entre des objets représentant des entités réelles ou conceptuelles (comme dans les réseaux de communications, des diagrammes de PERT ou des diagrammes de processus). Quant aux tableaux de bord et aux synoptiques, également importants, ils sont souvent utilisés dans les entreprises pour suivre les activités métier (Business Activity Monitoring ou BAM), ou dans l'industrie pour contrôler les processus ou surveiller des installations (Supervisory Control And Data Acquisition ou SCADA). Exemples d'applications et d'éditeurs créés avec Diagrammer for .NET (Figures ci-contre).

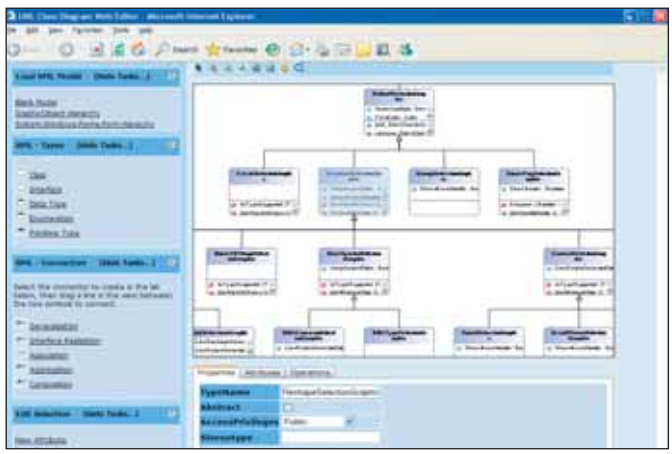
Une bibliothèque graphique

Au départ, Diagrammer. NET est constitué d'une bibliothèque d'objets graphiques vectoriels légers qui peuvent être manipulés à l'instar d'autres contrôles au travers de C# ou VB.NET. Cette bibliothèque définit des dizaines d'objets graphiques, des conteneurs, des liens graphiques et des panneaux, qui peuvent être assemblés pour créer des vues et des applications dédiées. Grâce à différents contrôles, ces objets peuvent être affichés et manipulés au travers de Windows Forms ou de Web Forms. Les objets graphiques de la bibliothèque offrent les principaux services suivants :

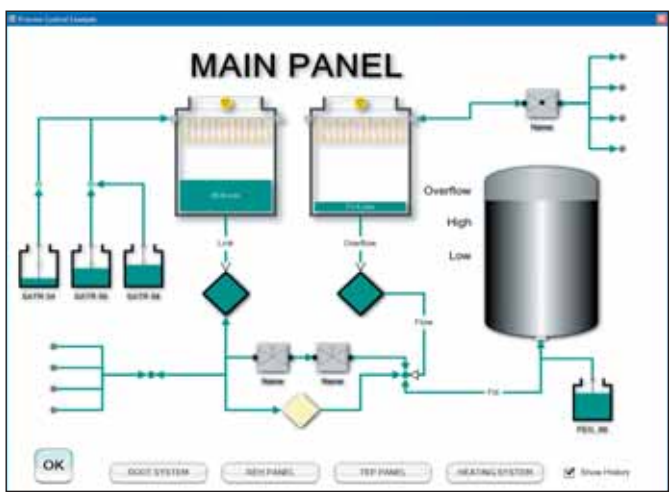
- Des classes graphiques de base, telles que des formes, du texte, des images, des polygones, des liens entre objets, etc.
- Une longue liste de propriétés associées aux objets, liées à leur apparence, telles que les couleurs, les types de remplissage, les bordures, les textures, l'opacité, les transformations, la rotation, les effets, etc.
- L'implémentation de fonctions plus avancées, comme des poignées, des ancres utilisées pour accrocher les liens entre objets graphiques ou des éditeurs de propriétés.
- La notification de toutes sortes d'événements, comme lorsque leur apparence ou leurs propriétés ont changé.
- Le traitement d'autres types d'événements, en particulier ceux liés aux interactions de l'utilisateur.
- Des conteneurs pour gérer leurs enfants grâce à des contraintes de placement et de règles d'affichage.



Un éditeur de processus, basé sur le standard de notation BPMN.



Un éditeur interactif Web/Ajax permettant la création de schémas UML.



Un synoptique animé pour de la supervision industrielle.

Fig.1 : Environnement de Design sous Visual Studio

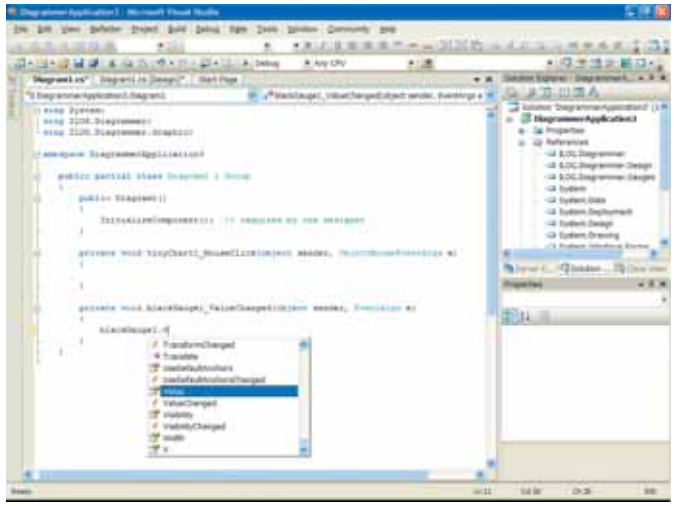


Fig.2 : Codage des événements sur un symbole graphique

L'outil supporte également les services généraux suivants :

- Gestion des interactions de l'utilisateur pour les éditeurs graphiques. Cela comprend la sélection, le redimensionnement, les transformations, la rotation, le glisser-déplacer, les palettes, etc.
- Services d'importation, tels que le SVG (Scalable Vector Graphics) du W3C, et d'autres formats d'images.
- Un moteur d'animation, qui permet de définir et de contrôler des animations temporelles.
- Des palettes de symboles prédéfinis, pour représenter des diagrammes, des processus, des schémas UML, et des manomètres.
- La persistance XML de tous les objets d'un diagramme ou d'un conteneur.
- Des services d'impression.
- Des algorithmes d'arrangement automatique de graphiques et de diagrammes.

Voici un exemple typique de code C# :

```
public class TestForm : Form
{
    public TestForm() : base()
    {
        // Create a view control
        DiagramView view = new DiagramView();
        view.Dock = System.Windows.Forms.DockStyle.Fill;
        // Create and configure graphics objects in a container
        Group container = new Group();
        Ellipse ellipse = new Ellipse(0, 0, 100, 100);
        ellipse.Stroke = new Stroke(Color.Blue, 3);
        ellipse.Fill = new LinearGradientFill(……);
        container.Objects.Add(ellipse);
        // Put all together
        view.Content = container;
        Controls.Add(view);
        Size = new Size(200, 200);
    }
}
```

L'API peut être utilisée pour créer n'importe quel type de représentation graphique, basée sur l'assemblage et la configuration des formes de base. Mais le plus souvent, elle est utilisée pour ajouter un comportement ou pour peaufiner une interface utilisateur qui n'a pas été créée par du code. En effet, il existe différentes manières de créer du graphisme, soit par génération de code au sein de Visual Studio, soit par le biais d'un éditeur graphique.

Développement d'applications dans Visual Studio

Diagrammer for .NET est totalement intégré à Visual Studio, avec un processus de développement très similaire à la conception habituelle des interfaces utilisateur. Diagrammes et objets graphiques peuvent être créés directement à partir de la boîte à outils. La génération de code est effectuée à la volée, et la gestion d'événements est définie comme pour les contrôles traditionnels. Des éditeurs et des wizards permettent aux développeurs de configurer les différentes propriétés des objets, et en particulier les paramètres d'arrangement automatique de graphes.

Cet environnement permet de créer des applications riches (en Windows Forms) ou bien Web/Ajax grâce à l'add-on ASP.NET Ajax 1.0 de Microsoft. Une liste de contrôles ASP.NET permet la création de vues graphiques et d'éditeurs Web.

Les services suivants sont disponibles directement à partir de Visual Studio :

- **Assistants pour la création d'applications :** des modèles d'applications sont disponibles au travers des assistants de création de projet. Il est possible de choisir entre des applications Windows Forms ou Web/Ajax. On peut également créer de nouvelles bibliothèques de symboles, qui seront plus tard utilisées dans des applications, sous forme de classes prêtes à l'emploi.
- **Mode Design :** lors du chargement d'un schéma ou d'un symbole complexe, un éditeur dédié apparaît dans Visual Studio. Il permet la conception de nouveaux symboles, de diagrammes complets ou de tableaux de bord prêts à être animés. Le dessin est ensuite traduit en code et en fichiers de ressources correspondants. Un "Smart Tag" permet aussi d'accéder aux options les plus importantes des objets sans passer par les éditeurs de propriétés.
- **Boîtes à Outils :** de nouvelles boîtes à outils sont disponibles dans Visual Studio et contiennent différents types d'objets graphiques tels que des formes de base, des objets de diagramme, des connecteurs, des jauges et des conteneurs. On peut également y référencer les

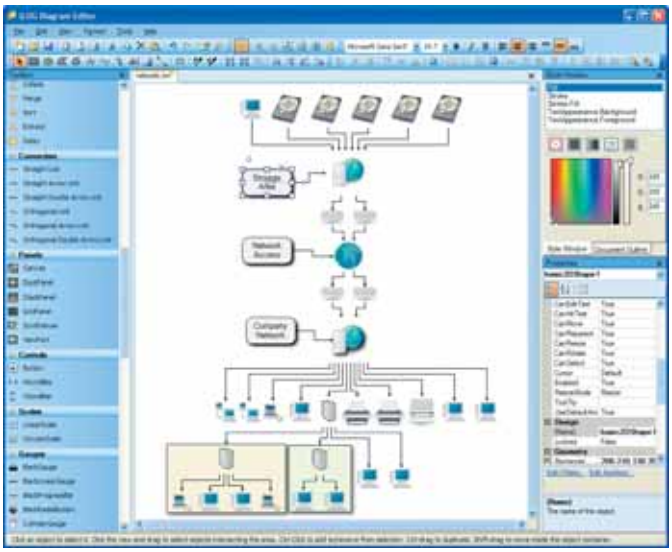


Fig.3 : Exemple d'éditeur utilisant un placement hiérarchique

nouveaux symboles que l'on a créés.

- **Gestion d'événements :** les événements et les changements de propriété peuvent être directement configurés et codés.
- **Assistants :** des assistants sont disponibles pour les tâches plus complexes, comme le réglage de certains effets visuels avancés ou des paramètres de placement de graphes.
- **Aide en ligne :** il est possible d'invoquer la fonction d'aide contextuelle intégrée (en appuyant sur F1).

Les illustrations suivantes montrent la manière dont Visual Studio est utilisé pour créer des applications graphiques. (Fig.1 et 2)

Agencement automatique de graphes

Il faut bien admettre que la représentation de schémas demande plus que des fonctions d'affichage. Il faut pouvoir assurer la création automatique d'un graphe ou d'un schéma à partir d'une source de données ne contenant qu'une description logique des entités. De même, pour un éditeur graphique, il est nécessaire d'avoir accès à des fonctions de placement automatique plus puissantes que les simples techniques habituelles d'alignement. Pour remplir ce besoin, Diagrammer for .NET propose une collection d'algorithmes de placement, de nœuds et de routage de liens adaptés à différents types de graphes. Ces techniques sont accessibles soit par code, soit sous forme d'assistant interactif à utiliser dans des éditeurs graphiques personnalisés (Fig.3).

Le code suivant illustre la création d'un algorithme TreeLayout appliqué à un groupe d'objets graphiques :

```
TreeLayout treeLayout = new TreeLayout();
treeLayout.FlowDirection = TreeLayoutFlowDirection.Bottom;
treeLayout.GlobalLinkStyle = TreeLayoutGlobalLinkStyle.OrthogonalStyle;
Group group; // Conteneur d'objets graphiques
…
group.GraphLayout = treeLayout;
```

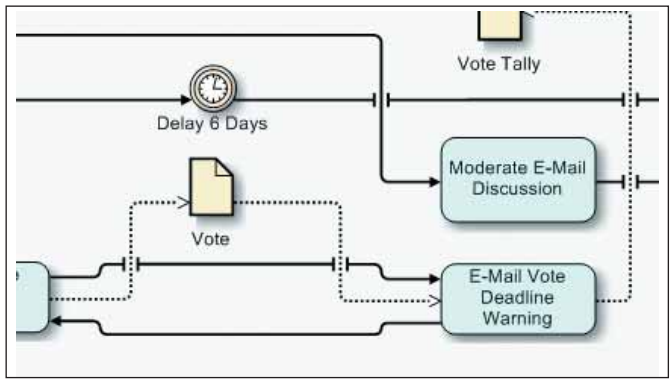
Le conteneur peut être soit un dessin complet, soit simplement un certain nombre d'objets regroupés ensemble. L'exécution de l'algorithme de placement peut se faire automatiquement ou sur requête en invoquant la méthode correspondante :

```
group.PerformGraphLayout();
```

Lorsque les objets sont déjà placés, il est utile de pouvoir router les liens, afin d'optimiser au mieux l'affichage et éviter les croisements. Voici un exemple de code permettant la configuration d'un des algorithmes de routage automatique :

```
Group group = new Group();
… // Remplir le groupe avec des objets et des connexions
ShortLinkLayout layout = new ShortLinkLayout();
group.LinkLayout = layout; // Affecter le layout
group.PerformGraphLayout(); // Invoquer le layout global noeuds+liens
```

Finalement, certaines options de présentation vont encore améliorer la qualité de l'affichage de schémas, par exemple en configurant l'aspect des éventuels croisements de liens, comme illustré dans la figure ci-dessous :



Amélioration de l'affichage des croisements

En conclusion, Diagrammer for .NET est une solution qui complète Visual Studio pour la création d'applications graphiques et interactives sous Windows Forms, Web Forms et Ajax. Il fonctionne sur plate-forme XP ou Vista grâce à la version 2.0 du Framework .NET. Le produit est fourni avec un grand nombre d'exemples, dont plusieurs éditeurs complets pouvant être utilisés comme des points de départ pour de nouvelles applications. Il contient également un outil de localisation pour que les interfaces créées soient accessibles en plusieurs langages. La documentation offre plus de 500 pages, sans compter la documentation des API, intégrée à Visual Studio. Au final, Diagrammer for .NET est un complément indispensable aux développeurs .NET qui veulent rapidement apporter la dimension visuelle de qualité nécessaire à leurs applications.

■ **Jerôme Joubert**

PROCHAIN NUMÉRO : 8 février 2008

Programmez! Hors-série .Net

Spécial TechDays

Présentation des Excel Services

Excel services est une nouvelle technologie fournie avec Microsoft Office SharePoint Server (MOSS) Entreprise.

Elle se divise en trois composants principaux :

- Une WebPart d'accès web (EWA) qui permet d'afficher des feuilles et classeurs Excel 2007 dans un environnement sécurisé sans avoir besoin d'Excel sur la machine client.
- Les services de calcul Excel (ECS) qui effectuent l'ensemble des opérations nécessaires à l'affichage des documents Excel.
- Les web services Excel (EWS) pour interagir à distance avec le moteur de calcul.

Architecture

La WebPart EWA (Excel Web Access) affiche les feuilles de données avec du DHTML et du JavaScript permettant ainsi de garder pratiquement le même rendu sur de nombreuses plates-formes et navigateurs. Une fois que vous avec correctement configuré Excel Services, vous pouvez déposer cette WebPart sur n'importe quelle page pour visualiser vos feuilles Excel, la connecter à d'autres WebParts et même créer des tableaux de bord. Les ECS (Excel Calculation Services) s'occupent des calculs effectués côté serveur, de mettre en cache les données, et de la gestion des sessions utilisateurs liées aux documents. Il est tout à fait possible dans le cas où la charge sur le serveur deviendrait trop lourde, de mettre en place un système d'équilibrage de charge afin d'améliorer les performances. Les EWS (Excel Web Services) fournissent un ensemble d'interfaces de programmation (API) pour Excel Services. En utilisant ces web services vous pouvez développer vos propres applications qui utiliseront le moteur de calcul d'Excel Services. Cela permettra à vos applications de se décharger du traitement du calcul, coûteux en termes de ressources.

Administration et Configuration

La configuration et l'administration d'Excel Services sont relativement basiques et l'ensemble des éléments d'administration est accessible via les SSP (Shared Services Providers) dans le site d'administration SharePoint. Même si les Excel Services fonctionnent parfaitement avec la configuration par défaut, il sera néanmoins nécessaire de configurer au moins une zone de confiance (trusted file location) avant de pouvoir l'utiliser. Cette zone de confiance peut être un site Internet, un dossier partagé sur un réseau d'entreprise, voire une liste ou un site SharePoint. Seuls les documents Excel faisant partie d'une zone de confiance seront traités par les Excel Services. Une précision cependant : les Excel Services permettent de partager des feuilles de calcul Excel, mais ils ne sont pas pour autant un tableur en ligne : vous ne pourrez pas créer de documents Excel avec les Excel Services. Excel 2007 reste nécessaire pour la création et l'édition des feuilles de calcul.

Fonctionnalités

Le classeur Excel à publier se comporte comme un classeur classique. Il peut contenir des formules de calcul, utiliser des connexions aux bases de données et à Analysis Services, ….

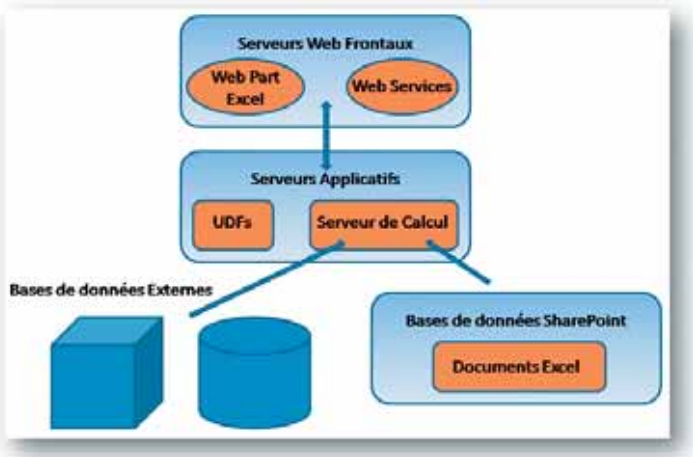


Figure 1 : Architecture des Excel Services.

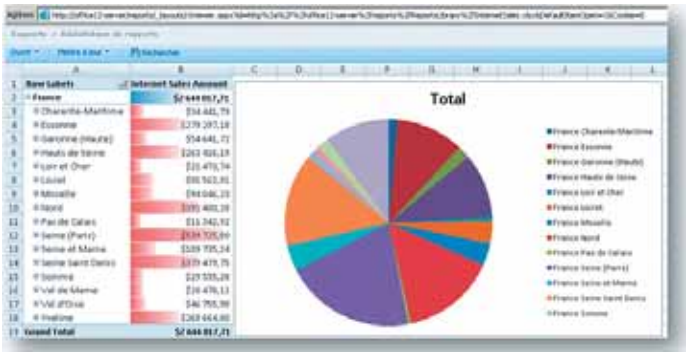


Figure 2 : Vue dans un navigateur d'un classeur Excel utilisant Analysis Services

Une fois publié, le document s'ouvre automatiquement dans le navigateur. De plus, si vous en avez les droits vous aurez toujours la possibilité de l'ouvrir dans Excel 2007 ou d'accéder à un cliché (SnapShot). Le " SnapShot " sauvegarde uniquement les données ainsi que le format du document, et cache les formules et les connexions aux données. Cela peut s'avérer très utile lorsque vous souhaitez protéger votre logique métier ou sauvegarder des données pour en faire des archives.
Mais les Excel Services ne permettent pas uniquement de publier et d'afficher des documents Excel, car la possibilité d'utiliser des paramètres apporte un certain niveau d'interactivité avec les documents.
Les paramètres permettent de spécifier les cellules à l'intérieur des feuilles que l'utilisateur pourra utiliser pour saisir certaines valeurs. Toutes les modifications que l'utilisateur réalise sur la page (via les filtres, paramètres, etc.) sont liées à sa session et n'impactent donc pas les autres utilisateurs. En revanche, si un utilisateur ayant suffisamment de permissions édite et publie à nouveau le classeur Excel, le document accessible depuis le navigateur est mis à jour.
Enfin, les services de calcul permettent aux applications tierces de bénéficier de centaines de formules natives dans Excel ou publiées dans les Excel Services. Il peut être très intéressant de se baser sur ces formules et sur le serveur de calcul pour développer une application interne capable, par exemple, d'utiliser la formule de jours ouvrés, jours ouvrables fournie en standard dans Excel, tout en se libérant de la charge que le calcul des opérations peut représenter pour l'application.

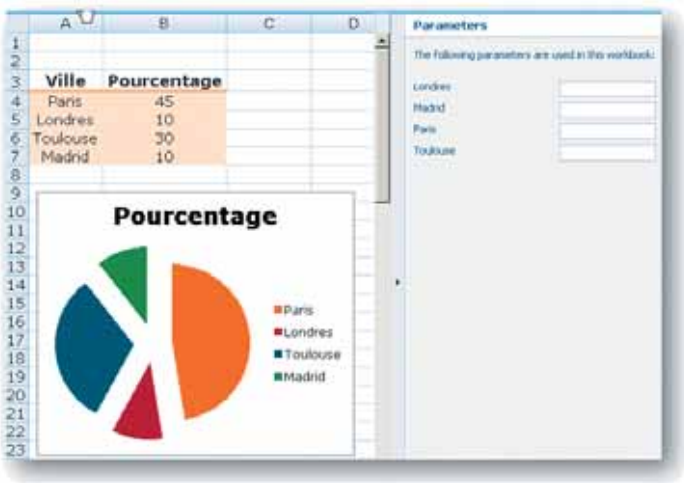


Figure 3 : Utilisation de paramètres avec les Excel Services

Excel Web Services

Une fois les feuilles de calcul créées et publiées, via Excel Services, elles deviennent accessibles via les web services, tout comme les formules et les cellules qui les composent. Ces web services sont accessibles depuis le serveur où résident les SSP.

Une fois ces web services référencés, la méthode OpenWorkbook offre un accès aux documents publiés. Cette méthode retourne un identifiant de session qui permet de manipuler les fichiers avec précision (accès en lecture et écriture sur les cellules, déclenchement de calculs, etc.)

Reprenons l'exemple des jours ouvrés du chapitre précédent en nous appuyant sur un simple classeur Excel comportant deux cellules vides (destinées à recevoir les valeurs d'entrée) et une cellule avec formule de jours ouvrés :

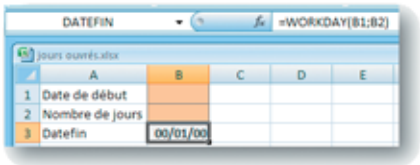


Figure 4 : Calcul des jours ouvrés depuis le classeur Excel

Une publication dans un site MOSS 2007 permet de rendre ce classeur accessible via les Excel Services et les Excel Web Services. Il est alors possible de le manipuler directement au sein du site, ou depuis une application externe. Voyons ce que cela peut donner concrètement sous Visual Studio :

1re étape : ajout d'une WebReference :

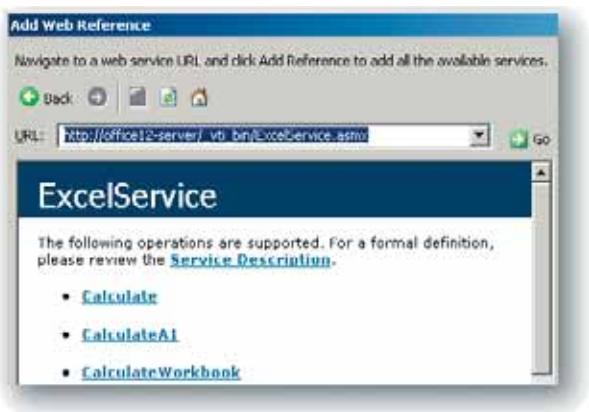


Figure 5 : Ajout de la WebReference aux Excel Web Services

L'adresse du WebService correspond à celle du site hébergeant le classeur suivi de " _vti_bin " (comme pour les autres WebServices SharePoint) et du nom des Excel Web Services : ExcelService.asmx.

2ème étape : appel au WebService au sein de l'application :

```
ES.ExcelService service = new Congés.ES.ExcelService();
service.Credentials = CredentialCache.DefaultCredentials;
ES.Status[] status = null;
string sessionID = service.OpenWorkbook("http://office12-server/Reports
/ReportsLibrary/jours%20ouvrés.xlsx", string.Empty, string.Empty, out status);
status = service.SetCell(sessionID, "Sheet1", 0, 1, monthCalendar1.
SelectionStart);
status = service.SetCell(sessionID, "Sheet1", 1, 1, System.Convert.ToInt
32(textBox1.Text));
status = service.CalculateA1(sessionID, "Sheet1", "DATEFIN");
object result = service.GetCell(sessionID, "Sheet1", 2, 1, true, out status);
textBox2.Text = System.Convert.ToDateTime(result).Date.ToString();
```

Figure 6 : Une application utilisant les formules définies dans le classeur Excel

Seulement 2 lignes de code sont nécessaires pour fixer les valeurs d'entrée, deux autres pour effectuer le calcul et récupérer la date du dernier jour de congés. Intégrer à vos applications des éléments jusqu'alors hébergés dans Excel devient un jeu d'enfant.

User Defined Functions

Il est possible de rajouter des fonctionnalités aux Excel Services via des librairies nommées " User Defined Functions " (UDF). Elles peuvent être exécutées depuis une feuille de calcul via les Excel Services comme toute autre fonction Excel standard.

Comme le développement des UDF s'appuie sur le framework .Net, les possibilités d'extension sont pratiquement illimitées. Côté technique, l'écriture d'UDF ne se distingue des développements classiques que par un attribut venant préfixer les méthodes exposées.

Les UDF représentent, entres autres, un moyen simple de centraliser des formules jusqu'alors disponibles sous forme de macros, sans en dévoiler les règles métier. Le code d'une UDF ressemble beaucoup à celui d'un WebService, comme l'illustre le bloc de code ci-dessous :

```
using System;
using System.Collections.Generic;
using System.Text;
using Microsoft.Office.Excel.Server.Udf;

namespace UDFSample
{
    [UdfClass]
    public class MyUDFClass
```

```
    {
        [UdfMethod]
        public int GetFailuresSummary(DateTime targetedDate)
        {
            return FailureMgr.CountFailuresByDate(targetedDate);
        }
    }
}
```

Notez la présence de l'attribut [UdfMethod] : les méthodes préfixées par cet attribut sont accessibles par les Excel Services.
Le code s'intègre dans une bibliothèque de classes qu'il vous faudra déployer sur le serveur et déclarer au niveau des SSP SharePoint. Une fois cette étape réalisée, la méthode GetFailuresSummary peut être utilisée depuis les clients Excel :

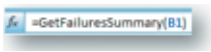


Figure 7 : Utilisation des UDF depuis le client Excel

Notez que pour pouvoir être chargée par les Excel Services, une UDF doit être une classe publique non abstraite avec un constructeur sans paramètre. Le nom de la classe n'a en soi aucune importance, car seule la signature de la méthode précédée de l'attribut UdfMethod permet d'identifier l'UDF en question. Si par ailleurs, lors du chargement des UDF, les Excel Services rencontraient la même signature pour plusieurs méthodes, seule la première serait chargée en mémoire, empêchant ainsi les possibilités de surcharge de méthode.
Il existe toutefois une autre alternative pour passer des paramètres optionnels à une UDF : l'utilisation d'un tableau de paramètres. En réalité, le dernier paramètre d'une UDF est un tableau. Le compilateur s'occupe de récupérer les paramètres entrés en dernières positions et les rassemble en un tableau passé en dernier paramètre de la méthode.

```
[UdfMethod]
public string myStringSplit(string phrase, params string[] args)
{
   string splitter = " ";
   if (args.Length > 0)
   {
      splitter = args[0];
      if (splitter.Length > 1)
      {
         throw new Exception();
      }
   }
   …
}
```

Ainsi les appels myStringSplit(''Hello, World'') et myStringSplit(''Hello, World'', ',') seront tout à fait corrects.
Par défaut, les Excel Services optimisent les échanges, via notamment une gestion du cache : plusieurs appels à une méthode UDF avec les mêmes arguments retourneront le résultat déjà mis en cache. Il est cependant possible de désactiver l'utilisation du cache en modifiant l'attribut comme suit : [UdfMethod(IsVolatile = true)]. De même, il est possible de désactiver le partage de données entre plusieurs sessions utilisateurs pour une même UDF en spécifiant l'attribut comme suit : [UdfMethod(ReturnsPersonalData = true)].

Les limitations

Bien que l'objectif des Excel Services soit, à terme, d'intégrer toutes les fonctionnalités d'Excel, certaines, ne sont pas encore opérationnelles dans la première version :

- **VBA :** Visual Basic for Applications ne sera vraisemblablement jamais supporté par les Excel Services, l'alternative proposée étant l'utilisation des UDF.
- **Ancien format de fichier :** Seuls les fichiers .xslx et .xslb (format Excel 2007) fonctionnent avec les Excel Services. Il existe néanmoins des convertisseurs qui vous permettront d'exploiter vos anciens documents.
- **Classeur externes :** Excel a la possibilité de référencer des cellules provenant d'autres classeurs. Cette fonctionnalité pose des problèmes lors de publication qui peuvent être facilement contournés grâce à un développement en interne.
- **Données en temps réel :** Excel supporte une fonctionnalité nommée RTD (Real Time Data) qui se connecte à la source des données et met à jour la feuille de calcul si nécessaire. Cette fonctionnalité n'est pas présente par défaut dans Excel Services, même s'il est possible avec certaines limitations de mettre en place ce principe de fonctionnement.
- **Pas de support client :** Excel ne gère pas l'utilisation d'UDF et affichera #name dans la cellule où vous faites appel à votre formule. Néanmoins, il est possible de modifier les UDF pour leur ajouter le support COM.
- **Autres fonctionnalités :** Entre autres, les Clip Art et Word Art, les Contrôles ActiveX, et l'IRM ne sont pas intégrés à cette version.

Pourquoi utiliser Excel Services

Au final, en quoi la possibilité de mettre à disposition un classeur Excel sur le Web via SharePoint est elle utile ? En quoi est-ce différent de le rendre accessible via un partage réseau classique ?
Tout d'abord, Excel a été conçu pour être une application cliente. Les problématiques de montée en charge ou de robustesse de l'environnement, lorsqu'un grand nombre d'utilisateurs accèdent à un même document, n'ont jamais été envisagées. C'est au contraire dans ce but que les Excel Services ont été pensés.
En plus de la possibilité de filtrer les formules présentes dans vos documents avec l'utilisation de " SnapShots ", vous avez la possibilité de décider quelles parties de votre classeur (feuille, table, graphique) seront visibles pour l'utilisateur final.
Le fait d'avoir votre classeur accessible sur votre navigateur via SharePoint vous permet aussi d'éviter certains écueils du travail en collaboration autour d'un fichier en partage réseau en bénéficiant des logiques de sécurité et de versioning inhérentes à l'outil : une seule version d'un document, la possibilité de stocker et d'afficher les versions antérieures, être assuré qu'uniquement les personnes agréées pourront modifier un fichier, plus de risque de suppression de document, etc.
Finalement, les Excel Services permettent de créer des applications robustes éditées par les utilisateurs finaux qui connaissent les règles métier. Les développeurs peuvent alors utiliser cette logique dans leurs nouvelles applications grâce à l'utilisation de la WebPart Excel et des web services.

■ **Sébastien Picamelot**
■ **Philippe Sentenac**
Consultants/formateurs -- Winwise

Vous avez dit softwares factories ?

Nous avons coutume de dire que l'informatique est encore à l'ère préhistorique ou artisanale. Si nous parlons des technologies Microsoft, il faut savoir que des progrès ont été enregistrés ces dernières années dans la volonté de mettre en application l'industrialisation des développements. Des technologies ayant chacune leur histoire, leur objectif, concourent toutes dans l'accompagnement du développeur et l'amélioration de la qualité de la production logicielle. Nous allons ensemble explorer ces outils pour amorcer dans votre entreprise une démarche d'industrialisation.

Les architectes Jack Greenfield et Keith Short de l'équipe Visual Studio Team System chez Microsoft Corporation tirent la sonnette d'alarme dans l'article " Moving to Software Factories " au sujet de la demande en développement logiciel qui croît régulièrement. La solution ne peut se cantonner à augmenter le nombre de développeurs, il s'agit dès lors d'améliorer la productivité de ces derniers en mettant à disposition des outils et des méthodes adéquats. De plus, les coûts de réalisation ne sont pas maîtrisés et les coûts de maintenance sont importants.

Sur la base de ce constat, la fabrique logicielle (ou Software Factory) apporte une réponse adéquate. Elle permet d'accélérer le processus de création de logiciels à l'aide de modèles et pratiques recommandés en générant un socle structurant. Le développeur pourra alors dans ce cadre assurer une implémentation de qualité.

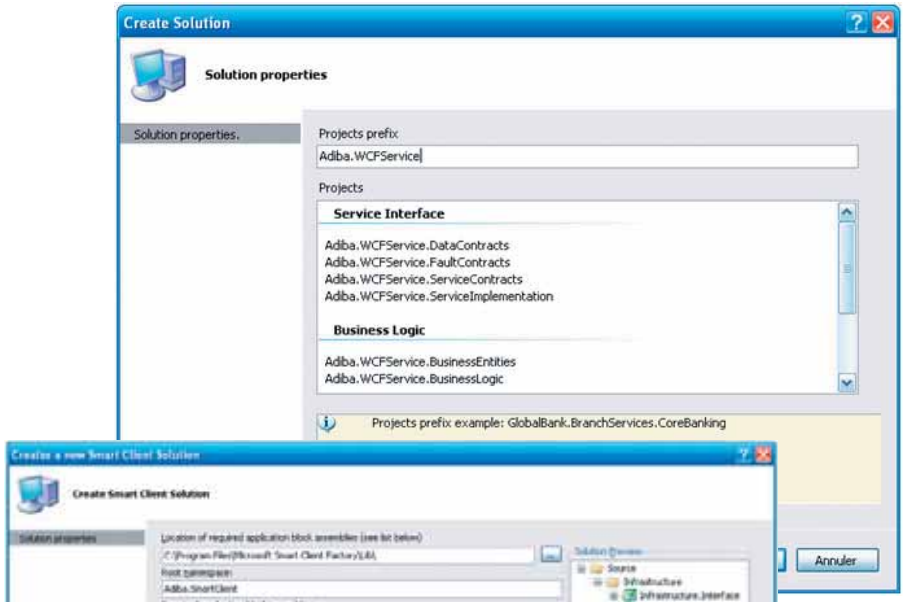


Fig.1

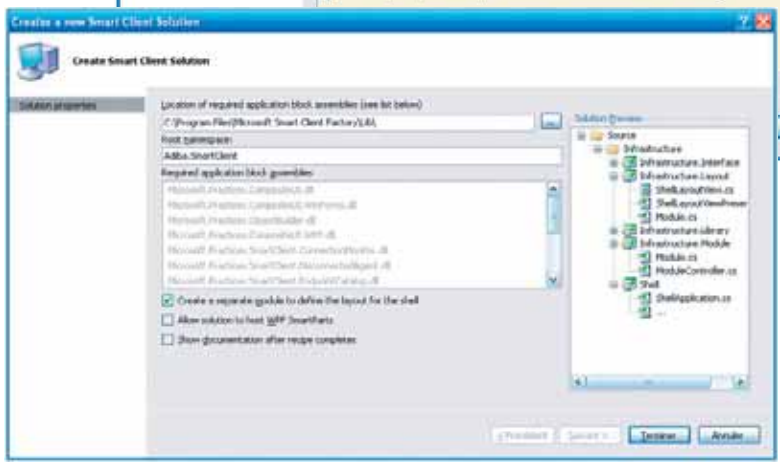


Fig.2

Software factories

Microsoft, via l'équipe Patterns and Practices, fournit un ensemble de recommandations et " meilleures pratiques " automatisées au travers de fabriques logicielles.

Ces outils de développement autorisent la génération de code à partir d'une collection de composants techniques (frameworks, socles techniques, bibliothèques de composants) et proposent de la documentation mais aussi des implémentations de référence.

- Web Service Software Factory facilite le développement de services web asmx ou WCF (Windows Communication Foundation). Elle propose un assistant permettant la génération de composants basés sur les messages habituellement utilisés pour assurer de l'interopérabilité.

Vous pouvez alors générer différents composants tels que la couche d'accès aux données, couche métier et couche de service WCF. En déclarant votre base de données, vous pouvez générer des procédures stockées (CRUD) et la couche d'accès aux données associée. Ensuite, vous pouvez générer une liste d'entités métiers et implémenter votre logique métier. Enfin, vous pouvez exposer votre métier sous la forme de services WCF par le biais d'artefacts qui ont chacun leur rôle (Data Contracts, Fault Contracts, Service Contracts, Message Contracts et Service Implementation). WSSF se charge de normaliser le code WCF que vous générez (Fig.1).

- Smart Client Software Factory facilite la construction d'applications composites Windows reposant sur le framework Composite UI Application Block (CAB). Le CAB est un framework destiné à construire des applications clients riches composites sur la base de modules indépendants qui peuvent interagir. Ce framework est utilisé pour les applications qui possèdent de nombreux écrans et qui nécessitent une souplesse dans leur développement et leur déploiement, notamment en termes de personnalisation. SCSF propose un assistant permettant la génération du squelette de code de l'application qui chargera dynamiquement les modules développés. En automatisant ces tâches, cette fabrique masque la complexité du framework CAB. En effet, vous pouvez simplement ajou-

ter un nouveau module, créer une nouvelle vue de présentation, implémenter vos événements basés sur le mécanisme de l'Event Broker (design pattern " observateur " ou publication / abonnement qui permet la communication inter-modules à base de messages), etc. L'applicatif, l'affichage et l'accès aux données sont séparés dans chaque module sur la base du design pattern dérivé de MVC (Model-View-Controller) : MVP (Model-View-Presenter) " (Fig.2).

- Web Client Software Factory facilite la construction d'applications composites Web reposant sur le framework Composite Web Application Block (CWAB) et le Page Flow Application Block (PFAB). CWAB est un framework destiné à construire des applications Web modulaires avec ASP.NET. PFAB est un framework destiné à intégrer Windows Workflow Foundation dans une application Web pour gérer la navigation inter pages dans le site. WCSF propose ainsi de découpler votre application Web en plusieurs modules compilables de façon indépendante et rattachés à une racine unique de la même manière que la fabrique logicielle SCSF que nous avons vue précédemment. Une classe " Controller " est chargée de contrôler la navigation parmi les pages. Elle implémente le modèle Application Controller. Toujours comme dans les autres fabriques logicielles, des assistants visuels sont proposés pour vous faciliter la conception et la génération de code pour votre application.
- Mobile Client Software Factory facilite la construction de clients mobiles reposant sur les frameworks Composite UI Application Block (CAB) et Data Access Application Block portés pour le .NET Compact Framework. De la même manière que les autres fabriques logicielles de clients Web ou Windows, MCSF fournit de la documentation d'architecture, une implémentation de référence et des assistants visuels automatisant le développement avec l'environnement Visual Studio. Le découpage proposé contient 3 couches : couche de présentation, couche métier et couche d'accès aux données.

GAT / GAX (Fig.3)

Guidance Automation Toolkit est une extension à Visual Studio permettant d'élaborer des modèles dans le but d'automatiser la génération de code. C'est le successeur des " Entreprise Template Projects " (ETP) avec Visual Studio 2003. Le moteur pour écrire un modèle s'appelle T4.

Fig.4

On y décrit les éléments suivants :

- recette : nom de la tâche que l'on souhaite automatiser, composée d'un ensemble d'actions séquencées.
- wizard : l'assistant visuel qui permet à l'utilisateur de renseigner et paramétrer les actions.

Lors du déploiement de votre package " GAT ", vous devez installer les Guidance Automation Extensions (GAX) pour permettre à Visual Studio d'utiliser vos fabriques GAT. GAT/GAX constitue des briques d'extensibilité pour Visual Studio et doit être utilisé dans la phase de développement de vos projets.

Pour faire le lien entre votre projet et le guidance package que vous avez installé sur votre poste, vous pouvez utiliser le Guidance Package Manager, le fichier gpstate, ou créer votre projet sur la base du template proposé dans Visual Studio.

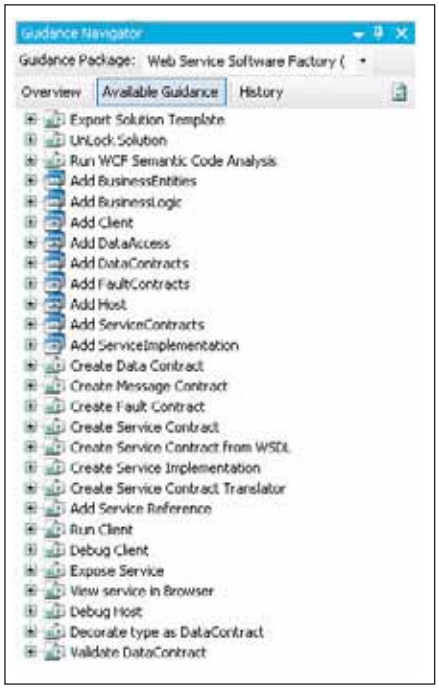


Fig.3

Dsl, le designer de designer

(Fig.4, 5 et 6)

Les Domain Specific Language Tools vous permettent de créer vos propres " designer " afin de concevoir dans Visual Studio vos modèles métier. L'objectif ici est d'exprimer une problématique métier dans Visual Studio dans un langage qui sera compris par vos clients et vous-même, dans l'optique de générer le squelette de vos applications. En effet, après avoir élaboré et déployé votre DSL dans Visual Studio, vous pourrez modéliser votre application sur la base des éléments que vous avez préparés. Le modèle est sérialisé en XML dans un fichier avec l'extension DSL lors de son élaboration. Vous pourrez choisir une extension " maison " qui sera utilisée sur les postes où vous déploierez votre DSL. On peut y ajouter des modèles GAT de génération de code qui peuvent être appelés depuis votre modèle DSL. L'investissement dans la construction d'un DSL est profitable s'il est intégré dans une démarche globale d'industrialisation des développements. Si le métier est connu et spécialisé, il est intéressant d'investir dans un langage " maison " vous permettant de modé-

liser vos applications et d'y associer de la génération de code. Au-delà du contexte métier, vous pouvez utiliser les DSL également pour des modélisations orientées technique (tels les diagrammes Architectes proposés dans Visual Studio Team Suite). Les DSL Tools sont intégrés au SDK de Visual Studio 2005.

Macros / Add-in / VSIP

La macro est le moyen le plus simple pour un développeur d'automatiser des tâches répétitives. Il peut enregistrer des séquences de son travail pour les rappeler ensuite en appuyant sur la combinaison de touches ALT + F11. Les limites sont que le seul langage supporté pour écrire une macro est le VB.NET et que le déploiement d'une bibliothèque de macros est limité au partage du projet qui contient le code source.

Plus avancé, un Add-in Visual Studio permet, à la manière d'une macro, d'automatiser certaines tâches et possède l'avantage de pouvoir être compilé et ainsi moins permissible, donc plus sécurisé.

Visual Studio Integration Package (VSPackages ou VSIP Interfaces) vous permet d'atteindre directement le modèle objet de Visual Studio. Bon à savoir, si vous souhaitez déployer votre package, vous devez demander une PLK (Package Load Key) à partir du site Microsoft Visual Studio Industry Partner Program (http://www.vsipmembers.com/).

Retours d'expérience

Les architectures devenant plus complexes, il est impératif d'expliquer et de former à l'utilisation pertinente de ces outils afin de ne pas polluer le développeur avec l'apprentissage d'un énième framework.

De même que le monde ne s'est pas fait en un jour, il est difficile d'élaborer sa fabrique logicielle en une seule étape. Ainsi, en observant l'utilisation de votre package (vous pourrez partir d'une fabrique existante), vous pourrez en déduire les nouveaux besoins en automatisation de code et enrichir votre package avec de nouvelles " recettes " (voir GAT). Le redéploiement n'impactera que le moteur de génération sous-jacent à votre solution et à vos projets. Un simple projet de setup suffit à générer un MSI que vous déploierez sur les postes de développement et qui enregistrera votre package à la liste des templates de Visual Studio. Développer une industrialisation logicielle nécessite une connaissance des outils à disposition ainsi qu'une analyse du contexte. L'architecte logiciel est la personne la plus à même à accomplir cette tâche en collaboration avec les développeurs.

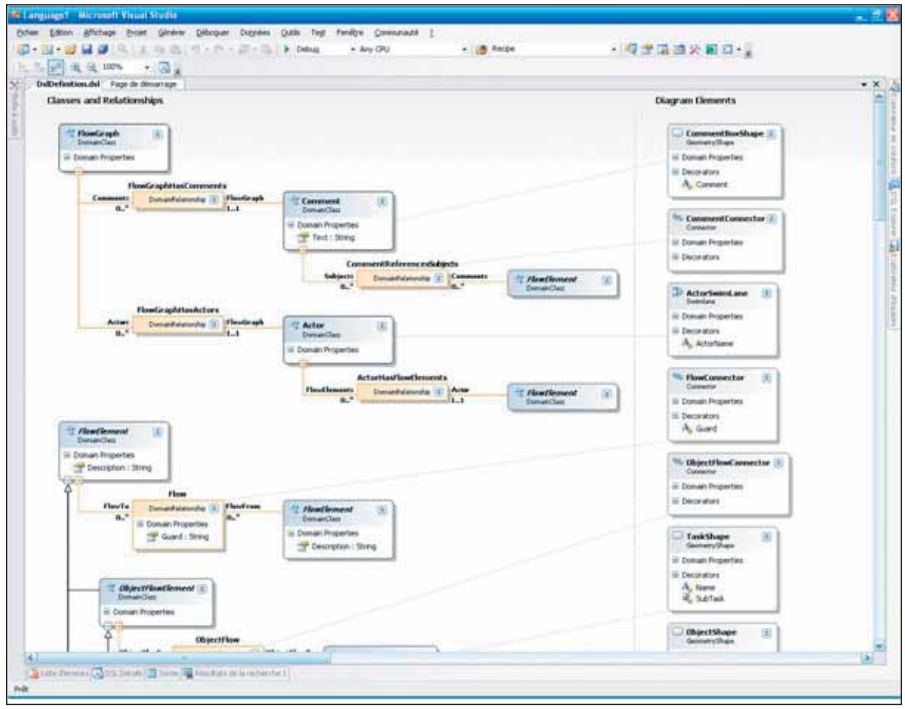


Fig.5

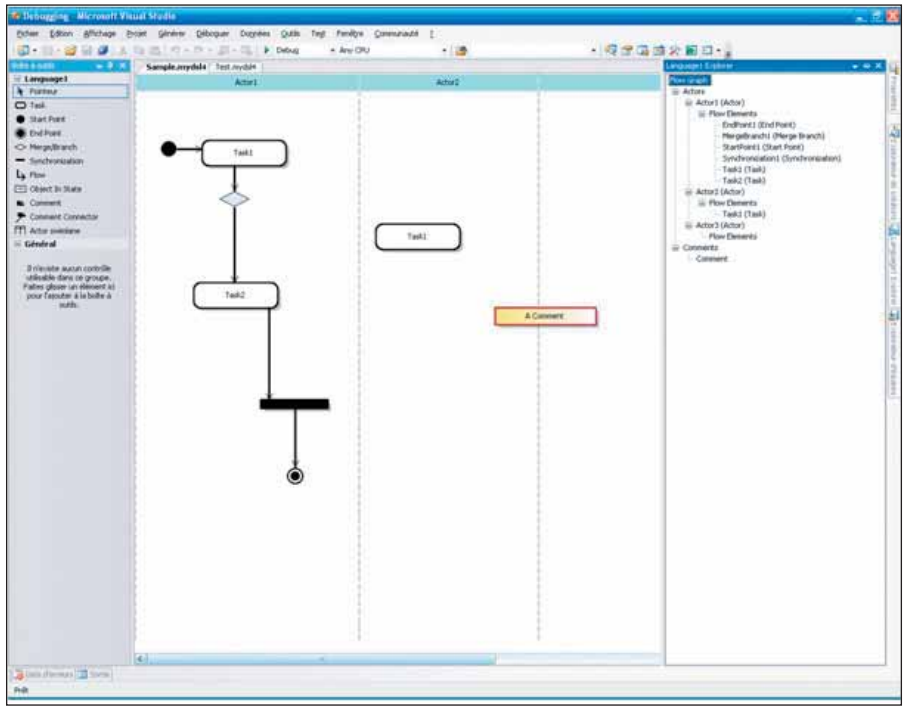


Fig.6

Les problématiques de développement que vous pouvez rencontrer au quotidien ont sûrement des points en commun mais surtout, des spécificités que les software factories peuvent adresser. Capitaliser dans des fabriques logicielles après avoir décomposé les problèmes et intégrer les solutions trouvées.

Pour compléter ce panorama, il manque un outil pour organiser et coordonner les résultats produits. Qui a parlé de Team System ?

■ **Sophiane Souane**
contact@adiba.fr
Fondateur d'Adiba, société française qui propose du conseil et du service en informatique spécialisé sur la plate-forme Microsoft. La mission d'Adiba est d'accompagner ses clients dans leur croissance au travers de la mise en œuvre d'outils d'industrialisation logicielle.

IIS 7 : le nouveau serveur web de Microsoft

L'une des plus grosses différences dans l'architecture APACHE par rapport à IIS vient du fait qu'Apache est basé sur des modules. En effet, la plupart des plug-in pour IIS sont développés en ISAPI mais ne font pas réellement partie du serveur. De plus, il est assez complexe de développer des ISAPI car une certaine connaissance du C++ est absolument nécessaire.

IIS 7 est entièrement modulaire, et les modules peuvent être développés en .Net. On peut ainsi les ajouter et les supprimer du serveur et ce, directement dans le fichier Web.Config. Dans cet article, nous verrons comment développer un module qui listera un répertoire d'images et en fera une gallery, ce qui permettra d'avoir juste à uploader l'image dans le répertoire, pour être prise en compte par la gallery. Une sorte de directory browsing améliorée, faite pour les images.

Mon Premier module.

Dans ce développement, on lira les fichiers qui sont dans le répertoire demandé et on affichera les images dans une gallery, ce qui sera toujours plus beau que la forme actuelle :

Ce sont les nouvelles erreurs d'IIS 7. En fait, la directory browsing n'est pas activée ni autorisée sur le répertoire. On va donc l'activer, en allant dans l'explorateur IIS 7. L'interface d'administration d'IIS se trouve dans les outils d'administration du panneau de configuration. On a une interface un peu différente d'IIS 6 (elle est même radicalement différente) :

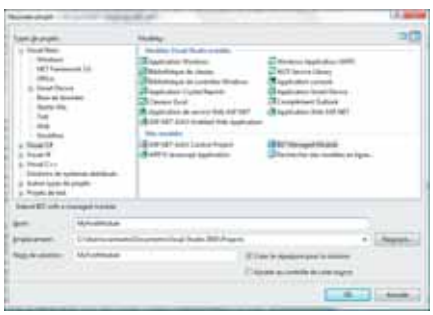
Et on finit par un clic droit/propriétés. Maintenant il faut activer les modules, pour activer des fonctionnalités. Pour cela, on sélectionne " exploration de répertoire " ou " directory browsing " en anglais et on clique sur activer (Figure ci-contre).

Comme vous pouvez le voir, les modules sont paramétrables. Pour ce directory browsing on peut spécifier ce qui va être affiché. Un Web.Config a été créé. Ce Web.Config possède le code suivant :

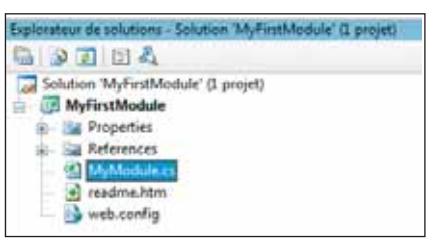
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <directoryBrowse enabled="true" />
    </system.webServer>
</configuration>
```

Comme vous pouvez le voir, un nœud qui se nomme system.webServer permet de configurer le serveur IIS 7 directement dans le fichier de configuration, c'est d'ailleurs ce que l'interface d'administration d'IIS 7 fait quand on active ou paramètre des modules. C'est dans le web.Config que l'on fera le déploiement de notre module plus tard. Mais commençons le développement. Première chose à faire, c'est télécharger le IIS 7 Manages module sur le site IIS.NET à l'adresse suivante http://www.iis.net/downloads/default.aspx?tabid=34&g=6&i=1302

Apres l'avoir installé, je lance Visual Studio et je crée mon projet de type IIS 7 managed module :



Lorsque j'ai fait cela, le template de projet m'a automatiquement créé une classe qui hérite de IHttpModule, un fichier *readme* qui me permet de savoir ce que je dois faire pour créer un module et un fichier *Web.config* pour ajouter le module à mon site (Figure ci-dessous).



Le code de mon module sera relativement simple. On va commencer par gérer l'événement BeginRequest de l'objet HttpApplication. On va donc ajouter ceci dans la fonction Init :

```
application.BeginRequest += new EventHandler(application_BeginRequest);
```

Ce code doit vous être très compréhensible, c'est une gestion courante d'événement. Ensuite, dans le OnBeginRequest, on va récupérer l'application courante, le contexte, l'objet request et response:

```
        void application_BeginRequest(object sender, EventArgs e)
        {
            HttpApplication objApplication = (HttpApplication)sender;
            HttpContext objContext = objApplication.Context;
            HttpRequest objRequest = objContext.Request;
            HttpResponse objResponse = objContext.Response;
        }
```

L'objet Request nous permettra de connaître le chemin du répertoire demandé et l'objet response nous permettra de renvoyer la page HTML formatée avec les images.
Maintenant, je vais mettre en forme toutes les images contenues dans mon répertoire.
D'abord, on récupère les informations du répertoire avec un DirectoryInfo :

```
DirectoryInfo objDirInfo = new DirectoryInfo(objRequest.PhysicalPath);
```

Ensuite, on boucle sur le répertoire pour récupérer les informations des images et les afficher :

```
            if (objDirInfo.Exists)
            {
                foreach(FileInfo objFile in objDirInfo.GetFiles())
                {
                    if (objFile.Extension == ".png" || objFile.Extension ==
".jpg" || objFile.Extension == ".gif")
                    {
                    objResponse.Write("<a href='" + objFile.Name + "'><img
src='" + objFile.Name + "' width=100></a>");
                    }
                }
            }
```

Ceci est assez basique, on peut formater la page pour qu'elle soit plus belle, voici par exemple une page avec une meilleure mise en forme :

```
        void application_BeginRequest(object sender, EventArgs e)
        {
            HttpApplication objApplication = (HttpApplication)sender;
            HttpContext objContext = objApplication.Context;
            HttpRequest objRequest = objContext.Request;
            HttpResponse objResponse = objContext.Response;
            DirectoryInfo objDirInfo = new DirectoryInfo(objRequest.Physical
Path);
            StringBuilder strBuild = new StringBuilder();
            strBuild.Append("<html><head><title>");
            strBuild.Append(objRequest.Path);
              strBuild.Append("</title></head><body><table><tr><td cols
pan='3'><h1>");
            strBuild.Append(objRequest.Path);
            strBuild.Append("</h1></td></tr>");
            if (objDirInfo.Exists)
            {
                FileInfo[] objArFile = objDirInfo.GetFiles();
                for (int miind = 0; miind < objArFile.Length; miind++ )
                {
                    if (miind % 3 == 0 && miind<objArFile.Length-1)
                    {
                        strBuild.Append("<tr>");
                    }
                    FileInfo objFile=objArFile[miind];
                      if (objFile.Extension == ".png" || objFile.Extension ==
".jpg" || objFile.Extension == ".gif")
                    {
                        Image objImag=Image.FromFile(objFile.FullName);
                        strBuild.Append("<td><a href='");
                        strBuild.Append(objFile.Name);
                        strBuild.Append("'><img border=0 src='");
                        strBuild.Append(objFile.Name);
                        strBuild.Append("' width=100></a><br>");
                      strBuild.Append("Width : " + objImag.Width + " Height
:" + objImag.Height);
                        strBuild.Append("</td>");
                    }
                      if (miind % 3 == 0 && miind < objArFile.Length - 1 &&
miind>0)
                    {
                        strBuild.Append("</tr>");
                    }
                }
                strBuild.Append("</table></body></html>");
                objResponse.Write(strBuild.ToString());
            }
        }
```

On ajoute l'appel a notre module dans le fichier de configuration :

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
   <system.webServer>
        <modules>
            <remove name="DirectoryListingModule" />
            <add name="DirectoryListingModule" type="MyIIS7Modules.
MyModule" />
        </modules>

   </system.webServer>
</configuration>
```

Dans ce fichier, je commence par retirer le module de DirectoryBrowsing pour rajouter le mien. La modularité nouvelle d'IIS 7 permet de faciliter grandement le développement sur cette plate-forme. En plus, comme tout est dans le fichier de configuration du site, la simplicité de déploiement, d'activation et de désactivation des modules rend le développement très souple. Le serveur web est entièrement personnalisable et customisable. Je conseille vivement la lecture des différents articles sur le site IIS.NET.

■ **Xavier Vanneste** - *XVanneste@msn.com*
http://www.netfxfactory.org
http://blogs.developpeur.org/xvanneste

Intégrateur WPF : un nouveau métier

Le développement d'applications avec WPF (Windows Presentation Foundation) constitue un changement face aux techniques employées actuellement. Voyons comment et pourquoi l'intégrateur WPF constitue un atout essentiel lors de ces développements.

Il existe 2 techniques pour démarrer un développement avec Windows Presentation Foundation. La première, c'est de laisser le désigner gérer la conduite du projet : c'est lui qui va faire la première ébauche de l'interface qui sera ensuite utilisée par les développeurs pour travailler sur l'application. C'est donc une méthode qui, en quelque sorte, privilégie l'ergonomie car il n'y a rien de mieux qu'un designer pour faire en sorte d'avoir une application ergonome et offrant une bonne expérience utilisateur.

La deuxième méthode consiste à laisser les développeurs diriger la conduite du projet. Ce seront eux qui vont faire une première version de l'interface graphique : dans ce cas là, on préfère privilégier les fonctionnalités et on attend beaucoup du designer, qui aura pour objectif de rendre conviviale et utilisable une application fonctionnelle.

Quelle que soit la méthode employée, le but d'un développement avec WPF est de permettre à chacun de travailler de son côté et, au bout d'une période donnée, on réunit le travail de chaque partie pour assembler l'application.

Rôle de l'intégrateur WPF

C'est justement ce rôle que joue " l'intégrateur WPF ". A la fois développeur pour comprendre le code employé dans l'application, et designer pour pouvoir faire des modifications sur l'interface graphique lorsque cela est nécessaire, il maîtrise les outils liés à l'une ou l'autre des professions (Visual Studio, Expression Studio, etc.). Il est un peu la pièce charnière d'un développement WPF, faisant en sorte que les deux mondes (celui des designers et celui des développeurs) se comprennent parfaitement. Il coordonne ainsi chacune des deux parties et, lorsque cela est nécessaire, se charge de corriger les problèmes qui pourraient empêcher le travail fourni par les deux équipes de s'assembler correctement pour donner l'application finale.

Le rôle d'intégrateur WPF est loin d'être aussi simple qu'il en a l'air : chacune des personnes avec lesquelles il travaille possède son caractère, ses propres habitudes, ses façons de faire, etc. Cette tâche est déjà complexe en temps normal (i.e. lorsque l'on doit faire travailler des développeurs avec des développeurs par exemple!), mais lorsqu'il s'agit de réunir deux mondes que, a priori, tout oppose, la tâche est encore plus ardue.

Qui tient ce rôle ?

Lorsque cela est possible (autrement dit, dans de grandes équipes de développement, qui comprennent un chef de projet, des testeurs, des développeurs, etc.), ce rôle est tenu par une personne à part entière. Il se charge d'intégrer régulièrement (la durée est déterminée au début du projet) le travail produit par son équipe : morceau de code, icône, bout de code XAML, contrôles utilisateur, etc.

Cependant, il faut bien admettre que les équipes de ce genre sont rares : la plupart du temps, il n'est pas anormal de voir une personne jouer plusieurs rôles : un développeur se chargera également des tests techniques, le chef de projet jouera les tests fonctionnels, etc. Dans le cas d'un développement axé autour de Windows Presentation Foundation, il n'est pas rare que le rôle d'intégrateur soit joué par l'une ou l'autre des parties : un développeur ou un designer. Il n'y a pas vraiment de préférence mais, de manière générale, on constate que ce sont tout de même les développeurs qui jouent ce rôle.

Conclusion

Considéré inutile par beaucoup, le rôle d'intégrateur WPF est pourtant très important et peut, lors de projets assez conséquents, faire gagner beaucoup de temps aux développeurs et/ou designers qui n'ont ainsi plus à se concentrer sur l'intégration de leur travail conjoint. Que vous ne puissiez pas avoir un intégrateur à temps complet peut parfaitement se comprendre. Cependant, je ne peux que vous conseiller d'en avoir tout de même un (qui cumule un autre rôle) : le gain de développement sera moins important que si cet intégrateur n'était chargé que de faire cela, mais je peux vous garantir qu'il y aura toujours un gain non négligeable….

Thomas Lebrun – *MVP C#*
Consultant / Formateur – Winwise
thomas.lebrun@winwise.fr
http://blogs.developpeur.org/tom

PROCHAIN NUMÉRO : 8 février 2008

Programmez! Hors-série .Net

Gamme 2008 : SQL Server, Windows Server, Visual Studio

.Net sur Linux avec Mono

Mono, est un projet Open Source, permettant de faire tourner des applications .net sous Linux. Dans cet article nous allons découvrir les bases de Mono. Les exemples seront exécutés sur une distribution Debian.

Pour bien comprendre le projet Mono, il est nécessaire de connaître les principaux concepts de .net. L'élément principal de la technologie Microsoft .net est la CLR (Common Language Runtime), une implémentation Microsoft de la CLI (Common Language Infrastructure). La CLI est un standard ECMA (ECMA-335) qui définit entre autres :

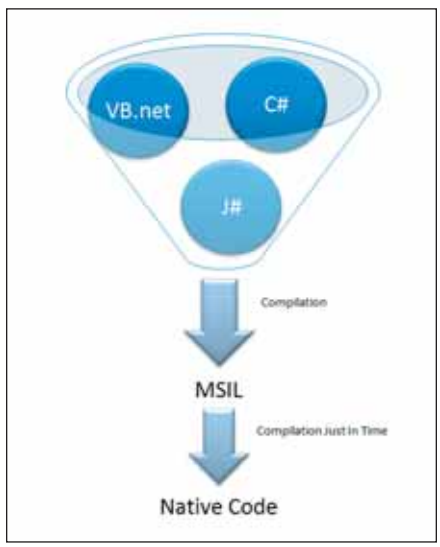
- **La CLS** (Common Language Specification) : définit un ensemble de règles qu'un langage doit implémenter : héritage, événement, etc ...
- **La CTS** (Common Type System) : définit la façon dont les classes sont gérées et utilisées par la machine virtuelle.
- **La CIL** (Common Intermediate Language) : c'est un langage de bas niveau qui peut être interprété par la machine virtuelle de n'importe quelle implémentation de la CLI. La CIL est communément appelé MSIL (Microsoft Intermerdiate Language) qui était le nom avant la standardisation ECMA.
- **La machine virtuelle** qui va compiler à la volée le MSIL en langage machine grâce à un compilateur JIT (Just In Time).
- **La BCL** (Base Class Library): définit les types de base comme System.Object, System.String. C'est le point de départ de la majorité des assemblies .net.

Pour en savoir plus, consulter l'article MSDN " Vue d'ensemble conceptuelle du .NET Framework " : http://msdn2.microsoft.com/fr-fr/library/zw4w595w(VS.80).aspx

Pour résumer, lorsque l'on compile un programme C#, VB.net ou tout autre langage .net, on crée un fichier qui contient du code MSIL : une assembly. Ce code MSIL sera ensuite interprété par la machine virtuelle d'une implémentation de la CLI.

Et Mono dans tout ça ?

Mono est une implémentation libre de la CLI sous licence GPL, LGPL et MIT. Mono fonctionne sur différentes plate-formes telles que Linux, Unix, FreeBSD, Windows, Mac OS X, Solaris. Le compilateur JIT fonctionne avec différents processeurs tel que les x86, SPARC, Power PC, ... En savoir plus sur Mono : www.mono-project.com



Installation de Mono

L'installation de Mono nécessite différents packages que l'on peut télécharger et installer grâce à l'outil apt-get, voici les différents packages :

mono : Le runtime Mono, contient l'implémentation Mono de la CLI, le compilateur C# 1.1 et les principales classes .net

mono-gmcs : compilateur C# 2.0

mono-gac : équivalent de l'utilitaire gacutil de Ms .net

mono-utils : utilitaire pour développeur Mono

apt-get, install mono, mono-gmcs, mono-gac, mono-utils.

A partir de là, Mono est correctement installé, nous pouvons désormais concevoir notre première application C# sous linux. Créons un fichier helloworld.cs avec le code suivant :

```
// helloworld.cs
using System;

class class1
{
    static void Main()
    {
        Console.WriteLine("Hello World from
 linux and C#");
    }
}
```

La commande gmcs permet de compiler ce fichier vers un fichier exécutable :

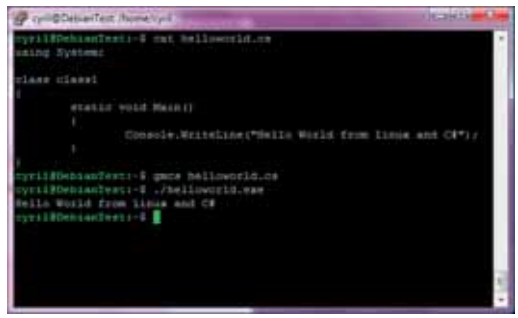
```
gmcs helloworld.cs
```

Par défaut, le nom du fichier sera filename.exe, dans notre cas, helloworld.exe. Pour spécifier le nom de l'exécutable, il faut utiliser l'argument –out :filename.exe. On peut le lancer via la commande Mono :

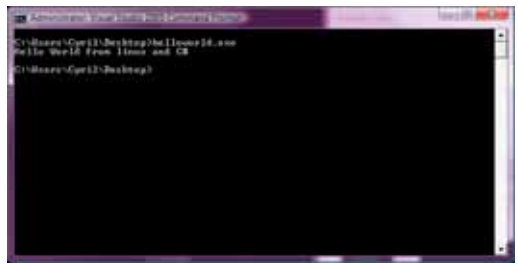
```
mono helloworld.exe
```

Le compilateur rend également le fichier exécutable (chmod +x) on peut alors le lancer directement :

```
./helloworld.exe
```

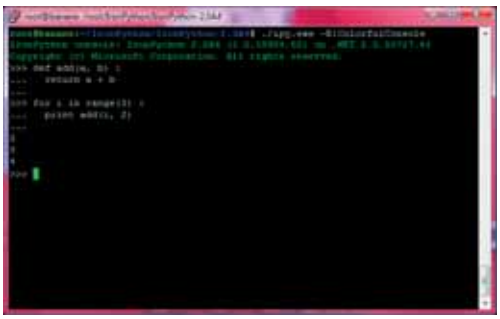


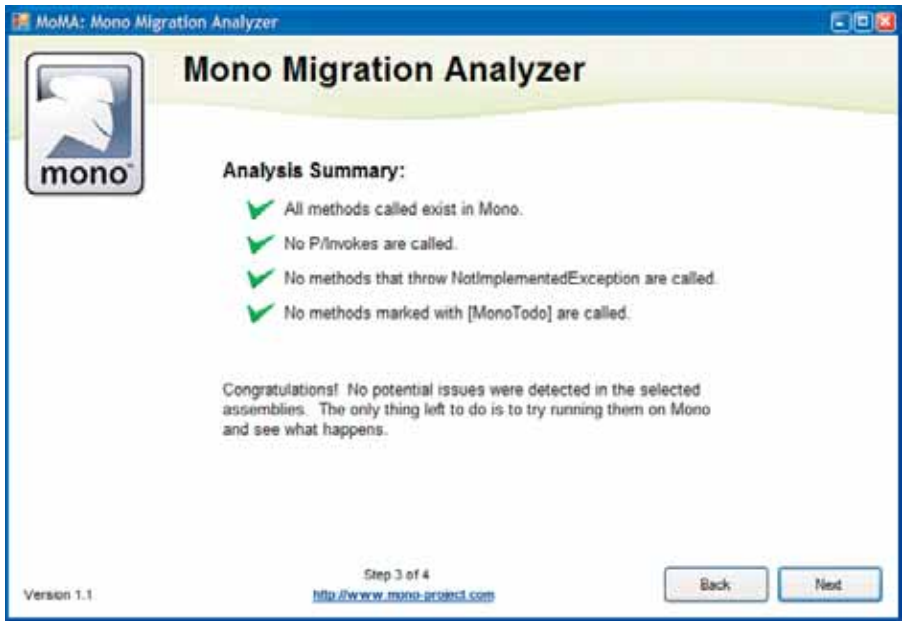
Comme Mono produit du code MSIL compatible avec la CLI, on peut exécuter cette application sur un système d'exploitation Windows qui n'a pas Mono d'installé, mais seulement le framework .net.

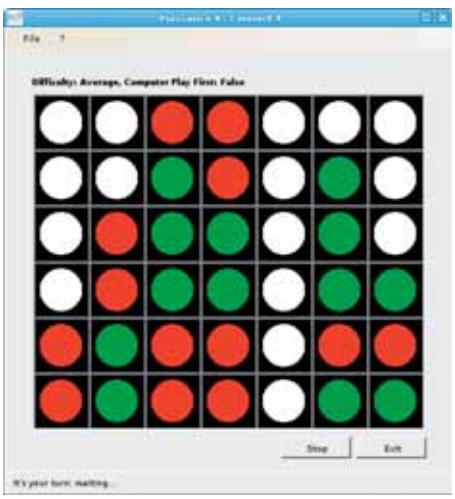


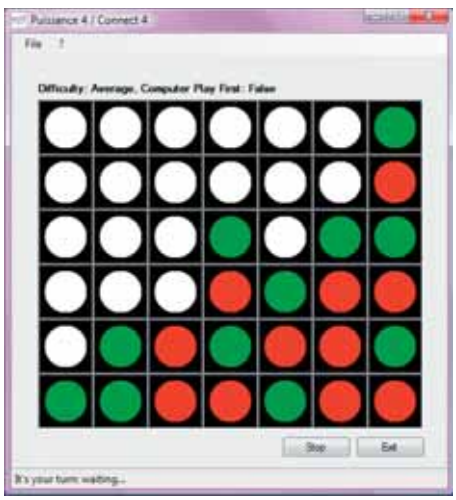
L'interopérabilité entre les deux mondes est totale, on peut utiliser des assemblies compi-

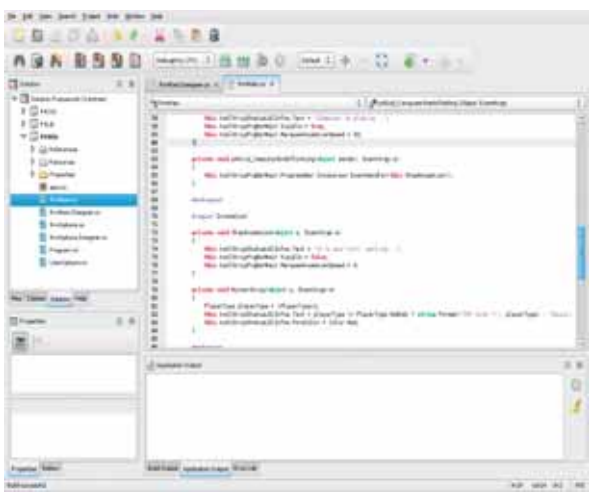
lées avec Windows et les utiliser sous linux et vice-versa. Bien sûr, notre exemple n'est qu'un simple helloworld, mais prenons un plus gros projet initié par Microsoft : IronPython, une implémentation Open Source d'un interpréteur Python tournant sur la DLR (Dynamic Language Runtime), ce projet peut être téléchargé sur codeplex (http://www.codeplex.com/IronPython).











Sans aucune modification du code, on voit que IronPython (et par conséquent toute la DLR) tourne sur Mono.

Quid des applications winforms ?

A l'heure actuelle, la plupart des fonctionnalités de .net 2.0 sont implémentées dans Mono, sauf pour les Windows Form où le support complet est prévu avec la version 2.2 de Mono fin 2007. Pour tester si une application est compatible Mono, il existe un utilitaire qui va vérifier les assemblies : Moma (http://www.mono-project.com/Moma). Cet assistant va vous demander les assemblies à tester, puis vous afficher le résumé.

Testons un jeu de puissance 4 écrit en C# par Bidou : http://www.csharpfr.com/codes/CONNECT-PUISSANCE-TOP-AVEC-ALGORITHME-ALPHA-BETA_41345.aspx

L'assistant indique que notre application est compatible Mono, il nous suffit alors de lancer l'exécutable sur linux.

Moma ne fait que tester certains critères. Comme l'implémentation de Winform n'est pas encore finalisée avec la version actuelle de Mono, il se peut que d'autres problèmes arrivent lorsque vous exécutez l'application sous Mono. Parmi les gros logiciels qui tournent sur Mono on retrouve *paint.net,* un logiciel de dessin.

Développer pour Mono

Puisque Visual Studio génère des assemblies interprétables par Mono, on peut l'utiliser pour développer sous Mono. Si vous n'avez pas Visual Studio et/ou si vous voulez développer avec un environnement qui garantit que votre application fonctionne sous Mono, il existe MonoDevelop : un éditeur écrit en Mono qui tournera donc sous Linux.

Conclusion

Grâce au projet Mono le framework .net ne se confine plus à Windows, Mono est une option envisageable si vous devez développer une application multi-plate-forme.

Dans un prochain article, nous verrons comment configurer le serveur web Apache pour faire tourner des applications ASP.net.

Cyril Durand
Membre de l'équipe de développement CodeS-SourceS
Consultant indépendant
http://blogs.developpeur.org/cyril/

La cryptographie en .Net

*La définition de **cryptographie** dans Wikipedia est la suivante : " la cryptographie est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés ".*

En fait, la cryptologie est l'art de rendre illisibles des informations à ceux qui ne sont pas censés les lire. Il y a plusieurs techniques de cryptographie que l'on peut utiliser en .Net : le cryptage symétrique, le cryptage asymétrique, le hashage. Nous allons examiner chacune de ces techniques.

Le cryptage symétrique

Le cryptage symétrique est une technique de cryptage/décryptage des informations avec la même clé de chiffrage. La manière de crypter et de décrypter est donc identique. Les algorithmes qui peuvent être utilisés pour cela sont : Rijndael, RC2, DES, TripleDES. Nous allons créer une application console qui demandera une chaîne et affichera la chaîne crypter et décrypter dans chacun des algorithmes de cryptage. La cryptographie se trouve sous System.Security.Cryptography. Les classes qui permettront de faire le cryptage sont les classes RijndaelManaged, RC2CryptoServiceProvider,DESCryptoServiceProvider,TripleDESCryptoServiceProvider. On ne crypte pas directement une chaîne de caractères, mais plutôt un tableau de Bytes. La première chose à faire est donc de modifier la chaîne de caractères en tableau de Bytes en utilisant les classes d'encodage :

```
Byte[] encoding = Encoding.UTF7.GetBytes(NonEncryptString);
```

Voici le code de la fonction de codage :

```
static void EncryptString(string NonEncryptString)
{

RijndaelManaged rijndael=new RijndaelManaged();
rijndael.GenerateIV();
rijndael.GenerateKey();
Byte[] RijEncryptString = Encrypt(rijndael.CreateEncryptor(rijndael.Key,
rijndael.IV), NonEncryptString);
string RijDecryptString = Decrypt(rijndael.CreateDecryptor(rijndael.Key,
rijndael.IV), RijEncryptString);



RC2CryptoServiceProvider rc2=new RC2CryptoServiceProvider();
rc2.GenerateIV();
rc2.GenerateKey();
Byte[] RC2EncryptString = Encrypt(rc2.CreateEncryptor(rc2.Key, rc2.IV),
NonEncryptString);
string RC2DecryptString = Decrypt(rc2.CreateDecryptor(rc2.Key, rc2.IV),
RC2EncryptString);



DESCryptoServiceProvider des = new DESCryptoServiceProvider();
des.GenerateIV();
des.GenerateKey();
Byte[] DesEncryptString = Encrypt(des.CreateEncryptor(des.Key, des.IV),
NonEncryptString);
string DesDecryptString = Decrypt(des.CreateDecryptor(des.Key, des.IV),
DesEncryptString);



TripleDESCryptoServiceProvider tripleDes = new TripleDESCryptoService
Provider();
tripleDes.GenerateIV();
tripleDes.GenerateKey();
Byte[] TDesEncryptString = Encrypt(tripleDes.CreateEncryptor(tripleDes.
Key, tripleDes.IV), NonEncryptString);
string TDesDecryptString = Decrypt(tripleDes.CreateDecryptor(tripleDes.
Key, tripleDes.IV), TDesEncryptString);

Console.WriteLine("Rijndael Crypter {0} Decrypter {1}", Encoding.UTF7.
GetString(RijEncryptString),RijDecryptString);
Console.WriteLine("RC2 Crypter {0} Decrypter {1}", Encoding.UTF7.Get
String(RC2EncryptString),RC2DecryptString);
Console.WriteLine("Des Crypter {0} Decrypter {1}", Encoding.UTF7.Get
String(DesEncryptString),DesDecryptString);
Console.WriteLine("Triple DES Crypter {0} Decrypter {1}", Encoding.
UTF7.GetString(TDesEncryptString),TDesDecryptString);


}
```

Les classes nécessitent de générer les clés et les informations de vecteur de transformations, c'est le rôle des fonctions GenerateIV et GenerateKey. Ensuite, on utilise les méthodes CreateEncryptor ou CreateDecryptor pour créer les ICryptoTransform qui se chargeront de transformer le tableau de bytes.

Voici les méthodes de cryptage et décryptage :

```
static Byte[] Encrypt(ICryptoTransform ICT, string NonEncryptString)
{
MemoryStream ms = new MemoryStream();
CryptoStream cs = new CryptoStream(ms, ICT, CryptoStreamMode.Write);
Byte[] encoding = Encoding.UTF7.GetBytes(NonEncryptString);
cs.Write(encoding, 0, encoding.Length);
cs.FlushFinalBlock();
return ms.ToArray();
}

static string Decrypt(ICryptoTransform ICT, byte[] EncryptString)
{
MemoryStream ms = new MemoryStream(EncryptString);
CryptoStream cs = new CryptoStream(ms, ICT, CryptoStreamMode.Read);
StringBuilder DecryptStrb = new StringBuilder();
int byt=0;
```

```
do
{
byt=cs.ReadByte();
if (byt != -1)
      DecryptStrb.Append((char)byt);
}
while(byt!=-1);
return DecryptStrb.ToString();
}
```

On utilise le MemoryStream pour écrire dans le tableau de Byte et on utilise CryptoStream pour crypter avec le ICryptoTransform créé au-dessus. Pour décrypter, on lit Byte à Byte les informations en utilisant le CryptoStream. Le résultat est, qu'avec la même clé, on a crypté et décrypté les informations de la chaîne de caractères. Bien entendu, la clé de cryptage peut être spécifiée et DOIT ETRE sauvegardée pour pouvoir décrypter les informations. Ce qui donne :

Le cryptage asymétrique

Le cryptage asymétrique est très récent, c'est le plus sûr mais aussi le plus lent à mettre en place. Les algorithmes de cryptage Asymétrique sont DSA et RSA. Les classes qui permettent de mettre en place la cryptographie avec ces algorithmes étant DSACryptoServiceProvider et RSACryptoServiceProvider. Reprenons le même principe que l'application du dessus et voyons comment on pourrait le coder : On utilisera la classe RSACryptoServiceProvider. On va créer deux clés, une publique qui servira a chiffrer les informations et une privée pour les déchiffrés. Voici le code :

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
RSAParameters rpPublicKey = rsa.ExportParameters(false);
RSAParameters rpPrivateKey = rsa.ExportParameters(true);
Byte[] RSAEncryptString = AsEncrypt(rpPublicKey,NonEncryptString);
string RSADecryptString = AsDecrypt(rpPrivateKey,RSAEncryptString);
```

Bien, maintenant il nous faut crypter et décrypter ces informations. Voici la méthode qui servira à crypter :

```
static byte[] AsEncrypt(RSAParameters rp, string NonEncryptString)
{
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
rsa.ImportParameters(rp);
byte[]  DataToEncrypt= Encoding.UTF7.GetBytes(NonEncryptString);
return rsa.Encrypt(DataToEncrypt, false);
}
```

On récupère les paramètres de cryptage puis on les passe à notre objet qui chiffrera les informations. Pour décrypter on fait de même, mais avec l'autre clé :

```
static string AsDecrypt(RSAParameters rp, byte[] EncryptString)
{
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
rsa.ImportParameters(rp);
return Encoding.UTF7.GetString(rsa.Decrypt(EncryptString, false));
}
```

Ce qui donne, une fois ajouté a mon code du dessus :

Le chiffrement ou hashage

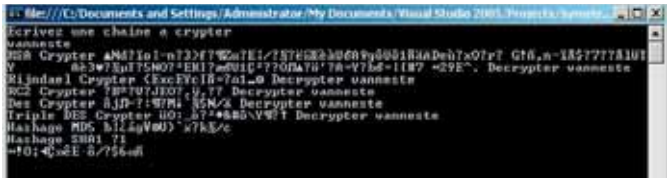
Le hashage transforme des données a crypter en chaîne de caractères illisible, la grosse différence entre le hashage et la cryptographie est que le hashage est irréversible, une fois les données hashées, il est impossible de revenir en arrière. Mais a quoi cela peut-il bien servir alors ? Imaginez que vous mettez les mots de passe de votre entreprise dans une base de données et que quelqu'un tombe dessus. Si les mots de passe sont hashés il lui sera impossible de les lire. Quand quelqu'un se connecte, il suffit de hasher avec le même algorithme les données et de comparer les deux. Autre chose, le hashage est très sensible aux différences, si vous avez deux chaînes de caractères qui sont sensiblement différentes, un espace ou une virgule en plus, vous avez un hash qui est entièrement différent. Cela peut être pratique pour comparer deux fichiers. Pour le hashage, on utilise le MD5 ou le SHA1. Regardons ce que cela donnerait en .Net

```
MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();
byte[] MD5Encrypt=  md5.ComputeHash(Encoding.UTF7.GetBytes(Non
EncryptString));

SHA1CryptoServiceProvider sha1 = new SHA1CryptoServiceProvider();
byte[] SHA1Encrypt = sha1.ComputeHash(Encoding.UTF7.GetBytes
(NonEncryptString));
```

Regardons ce que cela donne.

Essayons la même chose en minuscules :



Comme vous pouvez le voir, les deux chaînes sont sensiblement identiques, mais le résultat de hashage est complètement différent.

Conclusion

La cryptographie est un art à part entière, mais il est clair que les classes du .net nous aident beaucoup. Le cryptage symétrique est de loin le plus performant et le plus simple à mettre en œuvre, quant au hashage, il permet rapidement de voir si un document a été, même sensiblement, modifié. Comme quoi la cryptographie ne sert pas forcement uniquement a cacher les choses.

■ **Xavier Vanneste** - *Xavier.vanneste@ict7.com*
■ **Mohamed Benlmostapha** - *Mohamed.Benlmostapha@ict7.com*

Abonnez-vous

1 ÉCO

Recevez le magazine chaque mois

économisez 20 €

11 Numéros

45 €

Au lieu de 65,45 €
(Prix au numéro)

(Prix France métropolitaine)

-40%

2 Étudiant

Vous devez justifier de votre statut d'étudiant

Economisez 26 €

11 Numéros

39 €

Au lieu de 65,45 €

(Prix au numéro)

(offre réservée France métropolitaine)

-30%

3 Numérique

Lisez chaque mois le magazine

Format PDF
(téléchargement)

11 Numéros

30 €

Tarif Monde entier

Inscription en ligne uniquement

2,73€ le numéro

www.programmez.com

+ Abonnement INTÉGRAL

NOUVEAU

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour 1€ par mois !

Prix de lancement

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Éco, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 12 € (Prix de lancement, identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles /dossiers parus.

OUI, je m'abonne ! ou abonnement en ligne : www.programmez.com

☐ ABONNEMENT 1 an ECO au prix de 45 € TTC. *Tarif France métropolitaine.*
Tarifs hors France métropolitaine : CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres : nous consulter

☐ ABONNEMENT 1 an ETUDIANT (11 n°) : 39 € TTC. *Offre limitée à la France métropolitaine. Photocopie de la carte d'étudiant obligatoire*

☐ + SUPPLEMENT ABONNEMENT 1 an INTEGRAL au prix de lancement de 12 € TTC. (s'ajoute à une des formules d'abonnement)

MONTANT TOTAL DE L'ABONNEMENT : ☐ €

☐ M. ☐ Mme ☐ Mlle Entreprise : ..... Fonction : .....

Nom : ..... Prénom : .....

Adresse : .....

Code postal : ..... Ville : .....

Tél : ..... E-mail : .....

☐ Je joins mon règlement par chèque à l'ordre de Programmez ! ☐ Je souhaite régler à réception de facture

A remplir et retourner sous enveloppe affranchie à :
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75010 Paris.
abonnements.programmez@groupe-gli.com

Programmez! LE MAGAZINE DU DÉVELOPPEMENT

Offre limitée, valable jusqu'au 31 janvier 2008

Le renvoi du présent bulletin implique pour le souscripteur l'acceptation pleine et entière de toutes les conditions de vente de cette offre.

Conformément à la loi Informatique et Libertés du 05/01/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant.
Par notre intermédiaire, vous pouvez être amené à recevoir des propositions d'autres sociétés ou associations.
Si vous ne le souhaitez pas, il vous suffit de nous écrire en nous précisant toutes vos coordonnées.

Votre potentiel, notre passion.™
Microsoft®
LES OGRES SONT PARTOUT. MAIS SI VOTRE APPLICATION EST EN AVANCE SUR LE PLANNING, ILS VOUS LAISSERONT TRANQUILLE.
Votre défi : créer des applications riches, beaucoup plus vite.
Votre arme : Visual Studio® pour Microsoft® Office 2007.
Plus d'informations sur www.releveztouslesdefis.com
RELEVEZ TOUS LES DÉFIS
Microsoft® Visual Studio
© 2007 Microsoft Corporation. Tous droits réservés. Microsoft, Visual Studio, le logo Visual Studio, et « Votre potentiel, notre passion. » sont des marques de Microsoft déposées et/ou utilisées aux États-Unis et/ou dans d'autres pays.