

.NET 2.0

enfin dévoilé !

APPLICATION

Protéger sa page Web avec Advanced HTML Encrypt

.NET

Compression de données avec .Net 2.0

JAVA

Découvrir Aspect J

XML

Utiliser XSLT et XPath

C++

Utiliser la librairie wxWidgets

CRÉER

UN MESSENGER EN FLASH

OPTIMISATION :

DOPEZ VOUS AUX PROCESSEURS

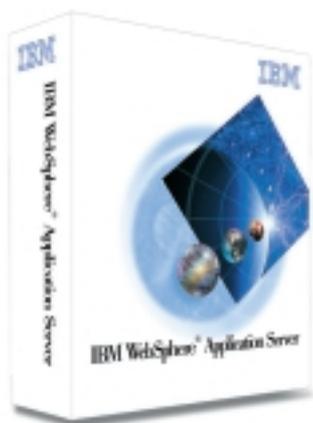


Les 10 ans d'IBM Software !

En 1995, les activités logicielles d'IBM étaient dispersées en diverses entités et filiales. C'est alors qu'un pôle spécial logiciel a été créé : IBM Software. En dix ans, cette entité d'IBM a énormément grossi. Aujourd'hui, elle représente 6 000 brevets, 30 laboratoires de recherche, le rachat et l'intégration d'une quarantaine d'éditeurs. L'un des premiers objectifs d'IBM Software était de gérer et coordonner les 4 000 logiciels et 60 marques différentes du Grand Bleu. Et

à l'époque, le logiciel constituait une activité secondaire. Un énorme travail de rationalisation a été accompli. Ainsi, au lieu des 60 marques, il n'en existe plus que 6, tout en redéfinissant les programmes de recherche et à se repositionner sur la micro-informatique, bref sortir du mainframe. La réorganisation interne s'est immédiatement conjuguée à une active politique de rachat. Il fallait trouver des outils et technologies complémentaires aux logiciels existants. Quatre symboles résument cette décennie : Lotus, Tivoli, Informix et Rational. Une

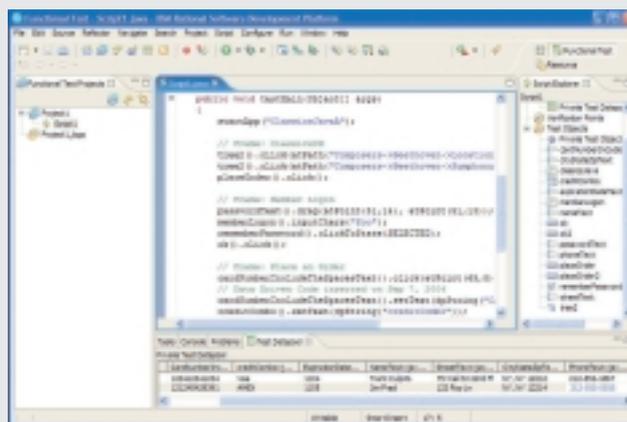
des activités d'IBM Software est aussi de concevoir ou d'aider des technologies. En dix ans, il a contribué ou développé des outils aujourd'hui stratégiques : automatic computing, CORBA, bien entendu Eclipse, EJB, les grilles, Java, SQL et évidemment XML et les Web Services.



IBM Software	Dans l'industrie informatique
1995 10 janvier, naissance de IBM Software Lotus rejoint IBM	Naissance de BEA Sun lance Java Sortie de Windows 95
1996 Tivoli racheté 1 ^{er} centres certifiés Java	1 ^{er} SDK Java Naissance des ActiveX Borland sort JBuilder
1997 Sortie de DB2 Universel v5	XML 1.0 disponible Invention du mot Weblog
1998 Apparition de WebSphere	BEA rachète WebLogic Apple lance iMac Naissance de Google
1999 Support de Linux VisualAge reconnaît l'embarqué	Naissance de J2EE Apple lance Airport Le virus Melissa fait des ravages
2000 Lancement de la plate-forme WebSphere avec intégration à Lotus Domino	Microsoft lance .NET et C#
2001 Rachat d'Informix Eclipse pointe son nez	Naissance de JBoss Apple lance MacOS X
2002 Annonce la stratégie On Demand	Naissance du WS-I Mozilla 1.0 sort
2003 Rachat de Rational Annonce de la gamme Express pour PME / PMI	UMTS arrive enfin! Borland sort C# Builder
2004 Sortie de la plate-forme Atlantic	BEA dévoile son Liquid Computing

Atlantic plus vrai que nature

Ceux qui croyaient que l'environnement Atlantic était une simple illusion optique devront rapidement réviser leur opinion. Atlantic se décompose en trois grosses couches : une couche chef de projet, dirigeant (Portfolio et Team Unifying), puis la couche technique (Eclipse) et enfin, la couche développement, avec l'ensemble des plug-ins. Le sens général est de simplifier le développement, la maintenance, les tests, le déploiement et la conception des projets. On dépasse largement le simple cadre du code. Le matin, quand on arrive au bureau, on se connecte à Atlantic, via la gestion des demandes. Elle est gérée, et centralisée par ClearQuest. L'outil regroupe les requêtes, les



L'outil de création de test

demandes, les bugs. On peut aussi définir les siennes propres. Ensuite, on prend la première demande et on la traite. RUP, la méthodologie projet d'IBM - Rational, est utilisée dans Atlantic, comme un tronc commun dans la gestion projet, le tout en format web pour pouvoir être utilisable à distance. La création du projet peut se faire sous forme Web, ce squelette peut générer les métriques liées (Project Console). Si RUP sert à la définition des projets, Atlantic fournit des modèles RUP prêts à l'emploi. Bref, le site Web du projet sert de base au tableau de bord qui regroupe l'ensemble des données, métriques, fichiers du projet. ClearQuest sert de gestion de demandes. Quand on veut intégrer une nouvelle fonctionnalité, l'environnement utilise la gestion des exigences, en s'appuyant sur un modèle MDA. Pour le développement, le référentiel est ClearCase. Il fournira un arbre des versions, les évolutions d'un projet entre les versions. Le chef de projet aura accès à un arbre multi niveau de granularité. Ainsi, sur une évolution donnée, on voit les fichiers modifiés, ajoutés. On peut aller jusqu'aux lignes de code modifiées, ou encore, voir le modèle UML de deux versions. Une fonction de fusion est disponible. Il est aussi possible de retirer une correction réalisée. Les tests constituent l'argument selon IBM. Dans les tests fonctionnels, il est possible de définir un plan de test avec planification et analyse. Quand on modifie un user case, l'outil de test prévient si le plan de test doit être modifié. Pour faciliter et simplifier ces tests, on dispose de ScriptAssure. Il s'agit d'un robot de test intelligent. Il est capable de réaliser le test, même si on change de place des objets ou les noms. Il identifie les éléments selon plusieurs critères. Cela évite de refaire son script de tests à chaque modification. Si l'outil est prometteur, il nécessite une formation et une bonne pratique. Courant 2005, l'éditeur annoncera une meilleure relation entre le développement et les utilisateurs, afin de mieux comprendre les attentes utilisateurs dans un projet.

SYSTÈME

MacOS X 10.4 : un Longhorn avant l'heure



Apple a sorti la version 10.4 de MacOS X le 29 avril. Depuis des mois, on a pris l'habitude de comparer les fonctions de ce MacOS X au futur Longhorn qui ne sortira qu'en 2006 et encore, avec quelques fonctionnalités en moins. MacOS X 10.4, alias Tiger, propose plusieurs centaines de nouveautés et améliorations. Pour le développeur, Tiger offre une mise à jour importante des outils de développement. On dispose maintenant de Xcode 2.0. Il s'agit de l'environnement de développement intégré d'Apple livré en standard avec le système. Il remplace le ProjectBuilder issu de Next. Si la v1.5 avait apporté quelques fonctions supplémentaires, on attendait une relance de l'IDE. Avec Xcode, on code en C, C++, Objective C, AppleScript et Java. La v2 intègre un module de modélisation afin de créer rapidement le squelette de son projet. Pour la compilation, il s'appuie sur le tout dernier GCC 4.0 qui facilitera l'optimisation du code sur G5 (fonction autovectorisation). Comme Tiger supporte plus largement le 64 bits, Xcode 2 aussi, pour pouvoir adresser la mémoire 64. On pourra créer des applications fat binaires fonctionnant aussi bien en 32 qu'en 64. Il sera aussi aisé de créer des applications à partir d'une compilation distribuée. Pour ce faire, Xcode implémente la technologie distribuée Bonjour. On dispose aussi de nouveaux SDK, afin de prendre en compte les nouvelles fonctions du système (moteur de recherche Spotlight, les composants dashboard, les moteurs Core Data, Core Image, Core Video). Une des fonctions les plus attendues est le Dashboard. Il s'agit d'une mini-application de type Widgets, mêlant application desktop et Web. On peut ainsi récupérer des données sur un site Web et utiliser une fonction d'une application desktop existante. Une application dashboard utilise du HTML, JavaScript et CSS, mais il est aussi possible d'utiliser d'autres langages. L'autre grande nouveauté est l'arrivée du moteur de recherche Spotlight. Le moteur s'appuie directement sur le système de fichier de MacOS X. Il se compose d'une base de données stockant les métadonnées et de l'index des contenus, des API pour implémenter Spotlight dans son code sont disponibles. L'objectif est de pouvoir réaliser des recherches très fines sur le contenu des disques durs. De plus, on bénéficie d'une nouvelle organisation des fichiers. Quand on fait une recherche de fichiers, le résultat se place dans un Smart Folder. Celui-ci se met à jour de lui-même, sans pour autant toucher aux fichiers originaux. On retrouve ici quelques fonctions de WinFS prévues initialement dans Longhorn. Côté Web, Safari arrive en version 2 et adopte RSS, un meilleur support du PDF et une intégration à iPhoto.

Site : www.apple.com

Conférence des développeurs

Comme chaque année, Apple organise pour les développeurs sa conférence mondiale (WWDC, du 6 au 10 juin prochain). MacOS X 10.4 sera à l'honneur dans de nombreuses sessions. Les sessions se répartissent en plusieurs catégories : technologies d'application, fondations système, multimédia, technologie d'entreprise et les outils. Des laboratoires seront aussi disponibles pour les amateurs. On attend avec impatience la conférence inaugurale sans doute menée par Steve Jobs (avec l'annonce du prochain MacOS X ?).

Advanced HTML Encrypt & Password Protect

Nous avons testé pour vous cet outil sécuritaire original qui permet de protéger le contenu d'une page Web.

Les fléaux modernes que sont le phishing, les pagejackers et les spam robots

L'année dernière, Microsoft comblait une brèche d'Internet Explorer qui permettait à un attaquant de pratiquer facilement le phishing. Cette arnaque, que l'on pourrait traduire par "pêche aux consommateurs", est une tentative de récupération illégale de votre nom d'utilisateur et de votre mot de passe dans le but d'accéder à un de vos comptes financiers (ou autre compte non public). Le principe : le fraudeur vous demande tout simplement les clés de votre accès, l'astuce étant qu'il fait tout pour convaincre que la source de cette demande est légitime. Pour y parvenir, il ira jusqu'à prendre l'identité de votre institution bancaire en usurpant son identité. Cette demande est souvent motivée par référence à des soi-disant mesures de sécurité ou de mises à jour de banques de données. Cette brèche d'IE consistait à afficher un vrai faux lien qui trompait l'utilisateur sur l'origine de la demande (si la police demande à voir votre carte d'identité vous la présenterez, mais est-ce vraiment un "vrai" policier qui vous l'a demandée ?). Malgré le colmatage de cette faille, le phishing continue à faire actuellement des ravages.

De même, si vous élaborez une page Web complexe et que vous la mettez publiquement à disposition d'autres internautes, vous ne voulez sans doute pas qu'un pagejacker aille piocher allégrement dans votre code source pour s'en inspirer. Enfin, des robots parcourent sans cesse le web à la recherche d'adresses internet qu'ils extraient du code HTML. Ces adresses finiront ensuite dans l'escarcelle d'une société de marketing pratiquant le spam et vous recevrez du courrier non sollicité.

Le bon docteur AEVITA

Le remède à tous ces maux serait de cadenasser l'accès au code source (HTML / javascript / vbscript) des pages. L'éditeur AEVITA Software s'est penché sur cette question et a mis au point l'outil

Advanced HTML Encrypt and Password Protect. L'application permet à l'utilisateur (qui n'a pas besoin d'être spécialiste, ni programmeur) de protéger par mot de passe une page Web, ou,

et d'en crypter le contenu. Concrètement, après avoir lancé l'application, vous indiquerez en entrée un répertoire ou une série de fichiers HTML. Ensuite, vous devez spécifier le chemin des nouveaux fichiers qui vont être générés (avec écrasement ou non des fichiers originaux). L'onglet suivant vous fait découvrir les possibilités d'encodage. Soit vous n'encodez pas, soit vous optez pour un codage faible, moyen ou fort avec l'algorithme blowfish. L'onglet "password" vous demande si vous désirez ou non qu'un mot de passe soit demandé pour rendre la page visible. Si vous choisissez de protéger votre page par mot de passe, il ne sera pas possible de la visualiser directement.

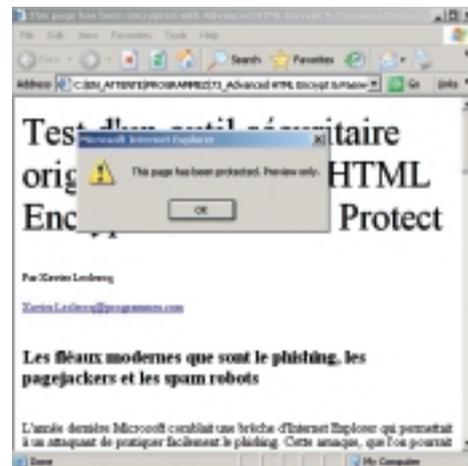
Petite précision : cette option n'est pas compatible avec Firefox 1.0 et nous avons dû utiliser Internet Explorer pour que le script accepte d'afficher le contenu. Par contre, l'encodage d'une page, pour empêcher la lecture du code source HTML, fonctionne avec Firefox (vous pouvez visualiser la page mais non afficher le code source).

Enfin, l'outil possède des options avancées (plus de vingt) qui vous permettront de modifier le layout et d'anticiper les réactions d'utilisateurs. Par exemple vous pouvez insérer votre propre notice de copyright lorsque le surfeur décide de visualiser la page :

```
<!-- Copyright your-domain.com. All Rights Reserved. -->
<!-- Source code not available. -->
```

Ou encore, vous pouvez afficher un message particulier si le bouton droit est enfoncé : "This page has been protected. Preview only". Cette option est particulièrement intéressante si vous ne souhaitez pas que l'on recopie votre texte par copier/coller. Une autre conséquence du blocage du menu contextuel est l'impossibilité de recopier l'adresse de liens.

Ou encore, vous pouvez empêcher la page d'être visualisée avec le navigateur Opera.... Pratiquement, des incompatibilités se présentent dès que l'on sort du cadre d'Internet Explorer (la boîte de dialogue "This page has been protected. Preview only" n'apparaît pas sous FireFox). Si vous travaillez sous Windows XP et que vous avez installé le Service pack 2, IE ne voudra pas non plus exécuter les scripts javascripts, pour-



tant nécessaires au bon fonctionnement du système d'AEVITA. Ce n'est pas bloquant mais vous devrez préciser de ce fait au navigateur que le code javascript qui en masque le contenu n'est pas dangereux à exécuter.

Conclusion

L'éditeur précise que son produit est compatible Internet Explorer, Netscape Navigator, Mozilla et Opera. Dans la pratique, nous avons vu que même avec Internet Explorer le procédé n'est pas toujours transparent comme il le devrait (en raison du SP2). Le produit est cependant original, bourré d'options avancées, et pourrait répondre à pas mal de demandes d'utilisateurs qui désirent protéger le contenu de leurs pages Web. Cependant, nous ne recommandons son usage qu'à l'intérieur d'un intranet. La raison ? Le gonflement démesuré de vos pages HTML : pour vous donner une idée, l'encodage et la protection par mot de passe de ce court article d'environ 5 000 caractères par Advanced HTML Encrypt & Password Protect génère un fichier HTML en sortie de près de 180 Ko...

Site de l'éditeur : <http://www.aevita.com>

Enquête sur les ravages du phishing : <http://www.antiphishing.org/APWG.Phishing.Attack.Report.Feb2004.pdf>

Version d'essai d'Advanced HTML Encrypt & Password Protect :

http://www.aevita.com/ftp/ahtmlpepp_setup.exe

■ Xavier Leclercq

Xavier.Leclercq@programmez.com

.NET 2.0 DÉVOILÉ

la révolution tranquille

Annoncée pour cet automne, la plate-forme de Microsoft, profondément renouvelée, se donne les moyens de son ambition : s'imposer comme le numéro 1 mondial du développement. Portrait de cette révolution tranquille.

Depuis cinq ans, Microsoft tente d'imposer, au moins sur Windows, sa plate-forme .NET. Si au départ, le flou régnait, au fur et à mesure, on s'est aperçu que .NET proposait une vision globale du modèle applicatif et de développement. La première préoccupation fut de s'intéresser au serveur, avant d'aller vers le client et le développeur. Avec Visual Studio .NET 2002 puis 2003, on a pu voir l'évolution du framework et des possibilités offertes. Aujourd'hui, Microsoft propose d'aller encore plus loin, avec la version 2.0 de son framework. En réalité, l'éditeur débarque en force : nouveau framework, nouvelles fonctions dans les langages, nouvelle gamme Visual Studio .NET, outil de gestion " global " du cycle de vie, et le très attendu SQL Server 2005. Il faudra du temps pour digérer toutes ces nouveautés. Mais on n'aura pas de réelles pauses, Longhorn et les nombreuses nouvelles technologies liées à ce système apparaîtront dès 2006, pour continuer en 2007 !

Bien que cette panoplie technologique ne sera disponible qu'en octobre prochain (si tout va bien et sans doute avec quelques décalages de sorties selon les produits), nous avons voulu vous faire découvrir l'ensemble des nouveautés aussi bien sur les IDE qu'au niveau des langages. Si vous ne connaissez pas encore .NET, profitez-en ! Vous utilisez .NET mais vous n'avez pas encore vu la v2, n'hésitez plus ! C'est dès aujourd'hui que l'on s'y prépare. Avec la forte poussée de .NET sur les nouveaux projets, la plate-forme



devient une véritable alternative fonctionnelle et crédible face à J2EE. Même sur le web dynamique, ASP.NET 2 peut titiller PHP, JSP. Alors, faut-il passer à .NET ? Les développeurs VB6 commencent à se ranger à cette évidence. Les efforts de pédagogie de Microsoft sont

louables après un message envers les développeurs, peut être insuffisamment clair et rassurant. L'enjeu pour .NET, et Microsoft, se situe là, dans la capacité de .NET 2.0 d'attirer à lui les réticents, indécis et fidèles de VB. Vaste ambition !

■ François Tonic

1 ► FRAMEWORK ET LANGAGES

Faut-il passer à .NET ?

Cette question, beaucoup de développeurs VB6 se la posent. Mais pas seulement. Avec la sortie de .NET 2.0 dans quelques mois, l'entreprise regarde différemment .NET et les développeurs autres que VB s'interrogent aussi sur l'opportunité. Les seuls à ne pas se poser de réelles questions sont les développeurs .NET actuels qui suivront sans aucun doute le chemin de la v2. La croissance de .NET semble promise à un bel avenir... aux côtés de Java.

Le message de Microsoft sur et autour de .NET vers les développeurs, les développeurs VB et les entreprises a été en partie brouillé par plusieurs éléments. Le premier est une mise en avant de C# par rapport aux autres langages. Cette situation a contribué à effrayer les fidèles de VB. La syntaxe change, les méthodes de développement aussi. Et il faut bien avouer que le passage de VB6 à VB.NET était un véritable fossé, surtout si le code existant était mal structuré. On passait à un nouveau modèle objet et on ne disposait plus de toutes les fonctions de l'ancien VB. Migrer du code VB6 à VB.NET se révélait parfois contraignant, avec une partie importante de l'ancien code à reprendre. Bien entendu, il ne faut pas tout migrer, cela dépendra des applications. Aujourd'hui, avec le futur VB.NET 2.0, les développeurs VB6 retrouveront quelques réflexes et surtout, un environnement plus simple, plus puissant, plus pratique. Au niveau du codage, on dispose de l'objet My qui simplifie énormément la création des objets .NET. Le edit and continue facilite grandement le debugage de son application. Microsoft martèle que .NET se développe aussi bien en C++, qu'en VB.NET, qu'en C#, ou qu'en ASP.NET. Le fait d'avoir communiqué très tôt sur .NET 2.0 (depuis environ 2 ans), il est certain que de nombreux développeurs et entreprises ont attendu cette v2 avant d'envisager toute migration ou utilisation.

1 • Développeur VB6 : oui, non, peut-être...

Pas de doutes à avoir, VB6 est désormais un cul de sac technologique. Le langage et l'outil



n'auront plus d'évolutions. Malgré une pétition sur le maintien de son support par Microsoft, VB 6 fait partie du passé. La question se pose désormais pour les codes existants. Faut-il migrer ? Garder le code tel quel ? Pour le développeur, une formation s'avèrera nécessaire pour bien connaître les méandres de .NET et pour maîtriser l'approche objet des langages .NET et les bonnes pratiques de codage.

Oui, il le faut

Si vous avez un parc applicatif Windows hétérogène avec plusieurs versions de VB et d'autres langages, le passage à une plate-forme unique est un bon choix, même dans le cas d'un parc VB / .NET, l'intérêt est certain. Une unique technologie à maintenir et plus d'éparpillement des ressources. Pour les applications clientes et serveurs, critiques ou non, étant donné que VB6 n'est plus supporté par Microsoft, .NET s'impose de facto si vous restez en environnement Windows. Pour les nouveaux projets, le passage à .NET ou à une autre plate-forme s'impose.

Non, il ne faut pas

Basiquement, quand un code fonctionne bien, pourquoi le changer. Ce réflexe demeure, même pour les applications VB6. Si le code n'est pas critique ou qu'il ne nécessite pas d'évolution, la migration vers .NET n'est nullement une nécessité. Cependant, la question d'une migration sera tôt ou tard à se poser.

2 • Une véritable plate-forme unifiée

Les concurrents le disent : .NET est une plate-forme unifiée, offrant un modèle de développement unique. Pensez donc ! On peut faire avec le même code une application Web, Windows, mobile, que ce soit en client lourd ou léger. Contrairement à Java qui possède trois modèles (même si une convergence se dessine), .NET simplifie grandement le travail de codage. Et les nouveautés apportées par la v2 confortent ce modèle. À partir du même IDE (Visual Studio), on peut cibler ce que l'on veut. De plus, l'intégration toujours plus poussée entre le système, les outils Microsoft et la plate-forme de développement, les interactions possibles sont très grandes (formulaire XML, Office, portail, serveur, base de données, interopérabilité, web service...).

.NET plus cher que Java ?

Difficile de fournir une réponse générique. Il faut juger projet par projet. Sur un nouveau projet, si vous êtes en Windows, .NET est une alternative à Java, C, Delphi, VB.

Si les coûts de licences et de supports ne sont pas anodins, il ne faut pas s'arrêter là. Il faut voir quelle plate-forme répond le mieux aux attentes

1 » FRAMEWORK ET LANGAGES

fonctionnelles et techniques. Aujourd'hui, les langages C# et Java sont très proches, la différence se fera sur la richesse de l'IDE ou les outils annexes (serveurs, intégration...).

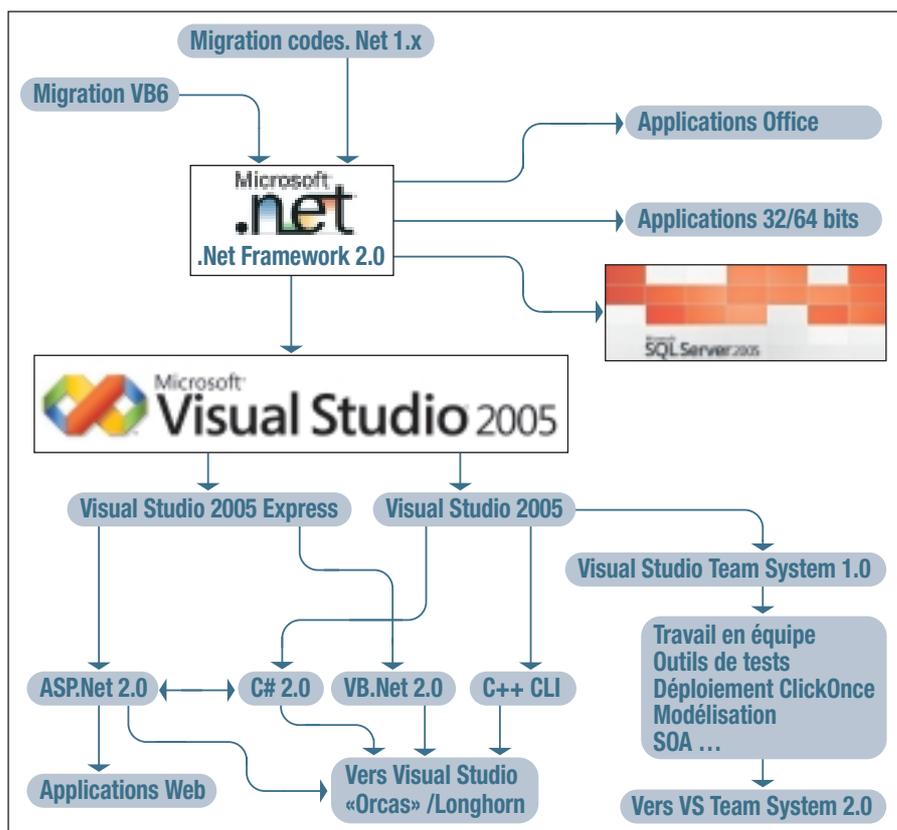
ASP.NET est-il aussi efficace que PHP ?

Microsoft fait actuellement le forcing pour pouvoir proposer aux développeurs Web souhaitant utiliser ASP.NET v1.x ou 2, une offre d'hébergement à la hauteur.

Longtemps handicapée par un manque d'hébergeurs ASP.NET, cette technologie s'affirme avec la v2 : meilleure interaction avec les autres langages CLR, nouvelles fonctions (cache amélioré, page maître...). ASP.NET v2 redonne des arguments intéressants face aux concurrents PHP, JSP, JSF...

Multi-plate-forme ?

Voilà la réelle faiblesse de l'offre .NET de Microsoft. Une absence de multi-plate-forme. Cependant, ce problème peut être en partie comblé par l'utilisation de Mono permettant de coder en ASP.NET et en C# sur plusieurs systèmes. Mono n'est pas à la hauteur fonctionnelle de C# ou ASP.NET 2.0.



Avanade : objectif .NET 2.0

Pas de complexe à avoir, Avanade, spécialisé dans les technologies Microsoft et plus particulièrement autour de .NET, travaille depuis des mois sur la v2 de la plate-forme de l'éditeur et les nouveaux outils. En interne, de nombreux développeurs et consultants travaillent sur la v2. Les premiers tests et formations sont en cours pour être fin prêts à la sortie. Visiblement, les premières impressions sont bonnes, malgré les habituels bugs de jeunesse, mais ils touchent peu ou pas les fonctionnalités. Avanade met en avant un élément que l'on passe souvent sous silence : le début d'intégration des services Windows SharePoint dans .NET. Cela facilitera énormément le travail sur les portails. Malgré les erreurs et bugs de la v1.x, Avanade considère que la plate-forme était bien née, et la v2 apporte bien plus de productivité grâce à des outils tels que Visual Studio .NET 2005 ou SQL Server 2005, ou encore les IDE de la gamme Express et bien entendu Team System. Sur l'existant VB6, le commentaire se veut pragmatique. Selon Avanade, les clients se posent la question de savoir s'il faut passer à .NET et comment le faire. Ainsi, sur les petits projets, il faut apporter de la plus value, et avoir de la visibilité sur le futur de l'application. Le constat est cependant sans appel : VB est mort, il n'a aucun avenir. Avec VB.NET 2.0, Avanade estime qu'il est proche de l'original. Mais, la SSII estime tout de même que Microsoft a été trop prudent et qu'aucune migration automatisée n'était proposée. Il fallait réécrire 60 à 70 % du code ! Avanade pointe aussi les faiblesses du message de Microsoft autour de .NET à ses débuts : il fallait mieux marteler l'aspect multi-langage, inconsciemment, l'éditeur a poussé C# et cela a fait peur à bien des développeurs. Sur le marché .NET, Avanade se montre assez confiant. .NET a pris son envol. Les marchés financiers s'intéressent fortement à lui. Passé l'étonnement de la chose, la finance voit en .NET une plate-forme plus simple que Java et SQL Server y est plutôt bien perçue. Dans les salles de marché, .NET offre de bons temps de réponses et il est possible de créer et de modifier des règles métiers rapidement, tout en faisant du client léger et lourd avec le même code. Le modèle de développement .NET est souvent mis en avant.

Trop lié à Microsoft ?

Oui, dans le sens où l'évolution de la plate-forme dépend directement de l'éditeur. Non, car, sur les outils et composants, de nombreux éditeurs commerciaux proposent des outils .NET (Borland, Mercury, Compuware, IBM...). L'open source n'est pas en reste, de nombreux projets ouverts existent pour .NET. Concernant l'IDE, bien que Visual Studio soit un excellent environnement, Delphi 2005 de Borland offre une alternative particulièrement attrayante.

Et VBA ?

Tout comme VB6, VBA n'a pas d'avenir sous .NET. Le prochain MS Office ne l'intégrera pas. Aujourd'hui, le modèle de développement pour Office est .NET / Visual Studio et le kit dédié.

■ François Tonic

Les tarifs prévisionnels de Visual Studio 2005

VS 2005 Professional Edition : 799 \$
VS 2005 Tools pour Microsoft Office : 799 \$
VS 2005 Standard Edition : 299 \$
VS 2005 Team System : 3 191 \$ (à partir de, selon le volume de licences)
La gamme Express : 49 \$ (chaque IDE)

Lexique de .NET



.NET Compact Framework

Version allégée et adaptée aux terminaux mobiles de type Smartphone, PDA PocketPC.

Assembly ou assemblage

Fichier contenant le code compilé. Il est autodéscriptif. Il remplace l'ancien système des DLL. Il n'existe plus le problème du hell dll, grâce à une gestion du versionning.

C++ CLI

Amélioration du codage C++ managé sous .NET 2.0.

C#

Langage créé par Microsoft. C# est le langage phare de VisualStudio.NET pour concevoir des Services Web. Il est dérivé des langages C et C++ tout en ressemblant au Java de Sun. C# est orienté objet et ressemble par sa syntaxe à Java.

CIL (Common Intermediate Language)

Quand il compile un projet .NET, le compilateur crée un " langage intermédiaire ". Le code CIL généré est identique à tous les langages .NET. Il s'agit du MSIL sous Windows.

CLI (Common Language Infrastructure)

Spécification décrivant comment une application doit être écrite en de multiples langages et pouvant être exécutée sur plusieurs systèmes.

Click Once

Nouvelle technologie d'installation (depuis .net 2). Facilite le déploiement, par le réseau et le Web, d'applications ainsi que leur mise à jour.

CLR (Common Language Runtime)

Le CLR est à .NET ce que la JVM est à Java. Le CLR est une sorte de machine virtuelle exécutant le fichier binaire MSIL. Le CLR crée à partir des fichiers binaires un code en langage machine spécifique au système client, via un compilateur JIT.

CLS (Common Language Specification)

Le CLS permet de rendre compatible la plupart des langages objets avec .NET. Il définit 41 règles communes. Le compilateur pour générer les Services Web s'appuie sur le CLS pour générer le fichier binaire MSIL à partir de n'importe quel code respectant les spécifications CLS.

Code managé, non managé

Sous .NET, il existe deux sortes de code, le code natif .NET, il est dit

code managé et le code non natif .NET, il est dit non managé. C++ est le seul langage .NET à pouvoir mixer les deux sans souci.

CTS (Common Type System)

La diversité des langages et des types impose l'utilisation d'une "table de conversion", le CTS. Si le code du Service Web respecte le CTS, il n'y aura aucun problème.

DLL (Dynamic Link Library)

Un DLL est un fichier contenant des fonctions et des procédures qu'une application utilise. Pour le développeur, la DLL facilite le travail et permet d'appeler une fonction existante directement dans une DLL.

Frameworks

Ensemble de classes conçues pour aider le développeur dans un domaine précis (par exemple : un framework d'interface). Il existe aussi l'Application Frameworks. L'application frameworks facilite le développement d'applications.

JIT (Just in Time)

Le compilateur " juste à temps " permet de générer à la volée un code en langage machine à partir du fichier binaire MSIL. Le JIT adapte le code au système client et permet d'exécuter le Service Web.

MSIL (Microsoft Intermediate Language)

Il s'agit d'un fichier binaire généré par le compilateur .NET à partir des spécifications du CLS. Si le code source peut être de n'importe quel langage ou presque, le compilateur fournit un fichier code uniforme conforme aux spécifications .NET, le MSIL. Il est indépendant du système.

MSI (Microsoft Installer)

Format d'installation Windows. Il demeure en place sous .NET et les applications. ClickOnce ne le remplace pas.

SDK (Software Development Kit)

Le SDK est un ensemble de bibliothèques de développement contenant des objets, classes, méthodes, interfaces, fonctions. Le SDK réunit tous les éléments nécessaires pour concevoir des applications implémentant un protocole, technique ou technologique.

XML (Extensible Markup Language)

Contrairement au HTML qui est un langage figé, XML permet de créer ses propres marqueurs. XML utilise plusieurs techniques pour décrire les formats et les modèles des données, XLS et DTD. XML est un langage central de .Net de Microsoft et des Services Web.

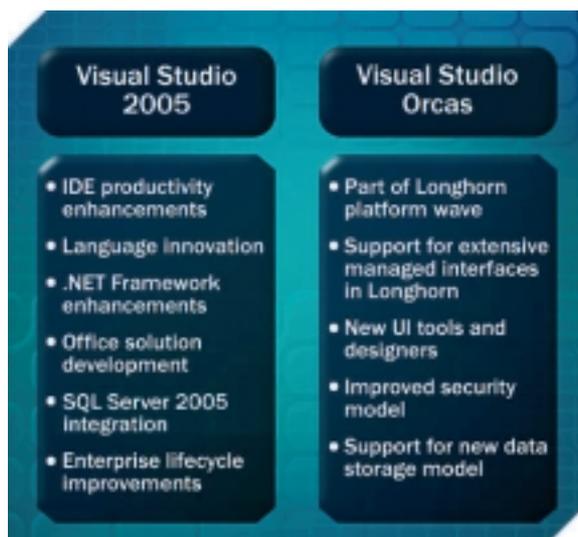
Framework .NET 2.0 : consolidation et évolution

La v2 du framework constitue une étape importante pour Microsoft et sa plate-forme. Ce ne sont pas moins de 1 500 nouvelles classes et améliorations de classes qui apparaissent ! Ce nouveau framework pose les jalons pour les prochaines années et surtout prépare à l'arrivée en 2006 – 2007 des nouveautés suivantes : Longhorn, les technologies Avalon, XAML, Indigo, ObjectSpace, WinFS, etc. La prochaine étape sera sans aucun doute Orcas, le futur Visual Studio. Bref, la v2 de .NET est là pour vous préparer à ce nouveau saut applicatif.

Microsoft utilise désormais l'appellation WinFX SDK pour désigner l'ensemble des technologies et outils de "nouvelle génération". WinFX regroupe .NET 2.0, Avalon et Indigo. Le SDK WinFX inclut en supplément les documentations, exemples et les outils. Bien entendu, vu l'ampleur du .NET v2, impossible de tout décrire. Nous reviendrons dessus dans les mois à venir.

Pour les applications Windows Forms ClickOnce

Pour faciliter le déploiement et les mises à jours, on dispose d'un nouveau modèle : Click Once. Il s'agit de déployer une WinForm aussi facilement qu'une WebForm. Il n'est pas nécessaire d'avoir un compte administrateur pour pouvoir installer. ClickOnce s'utilise soit par une page Web, soit par le réseau, soit sur un CD-Rom. Surtout, pour faciliter la maintenance, ClickOnce scanne automatiquement le réseau pour voir si une nouvelle version est disponible. Au lieu de tout remplacer, ClickOnce, peut mettre à jour uniquement des fichiers. ClickOnce s'appuie sur deux fichiers : un fichier manifeste XML pour l'application et un autre pour le déploiement. Quand on déploie, l'ensemble de l'application est copiée dans le dossier de déploiement, selon les informations fournies par le fichier XML de déploiement. Dans sa version 1.0, ClickOnce ne peut pas installer des pilotes, une application multi-utilisateur, des shared files... Avec la sortie de Longhorn et du futur Orcas, ClickOnce se rapprochera du modèle MSI (sans le remplacer).



Les nouveaux contrôles

Plusieurs nouveaux contrôles apparaissent, les développeurs y seront sans doute sensibles :

- DataGridView : contrôle pour afficher et éditer des données sous forme de tableau. C'est une évolution du DataGrid. Modèle à privilégier pour une WinForm.
- ToolStrip : contrôle de toolbar pour afficher menus, contrôles et contrôles utilisateurs. On peut reprendre le style XP, Office, IE. Remplace l'ancien ToolBar (qui demeure pour assurer la compatibilité).
- SplitContainer : pour créer facilement des splits dans une form. Idéal quand on crée des interfaces complexes.
- BackgroundWorker Component : quand dans une application, on effectue de longs traitements, il faut pouvoir découper son application en threads et en priorités. Désormais, on peut affecter en arrière-plan ce type de tâche

avec BackgroundWorker, comme par exemple pour les chargements d'images ou de fichiers, les transactions de données, les accès disques, etc. Mécanisme idéal pour une application multithread simple. Pour quelque chose de plus costaud, passez par Asynchronous Pattern for Components.

- WebBrowser : ce contrôle permet dans une application WinForm d'afficher une page Web HTML et d'offrir des fonctionnalités de navigation Web (ex. : Me.webBrowser1.Navigate("http://www.microsoft.com")).

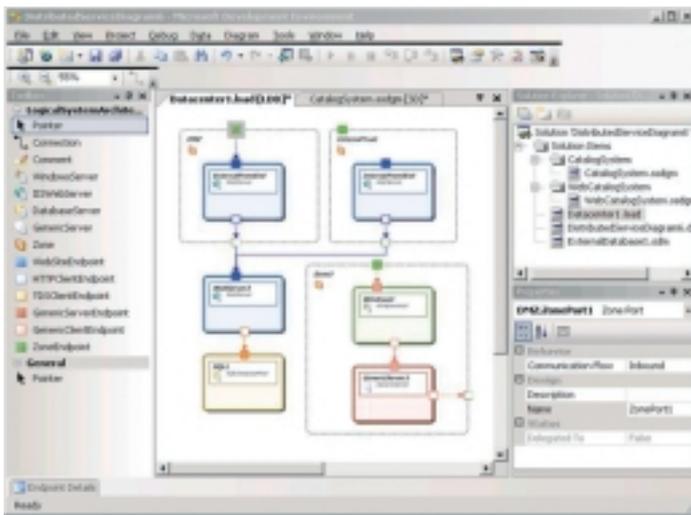
Le 64 bits dès maintenant

Si pour Windows, le 64 n'est pas encore très au point, sur .NET, avec le framework 2.0, on dispose d'une version pure 64 bits. Surtout, cela se fait en toute transparence pour l'utilisateur. Pour le développeur, peu de modifications à réaliser, c'est une option de compilation. La CLR est disponible en 2 versions : extension 64 et pour Itanium. Cependant attention, certaines fonctions posent problème, notamment le Pinvoke, le calcul flottant, le COM 32 (il utilise alors le WoW).

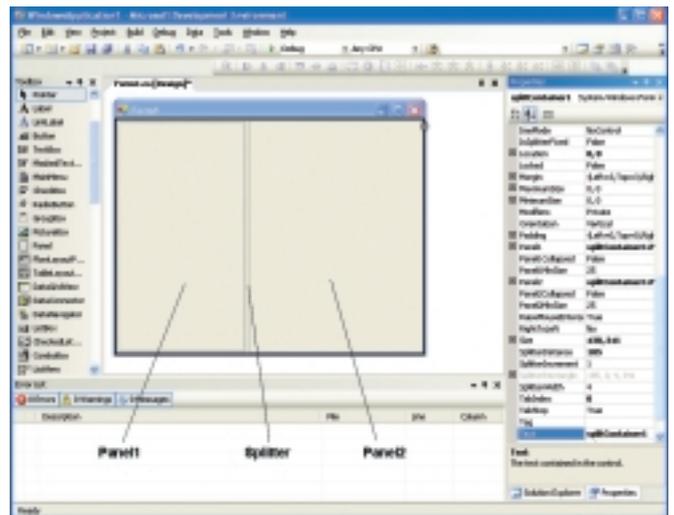
Réseau - matériel NetworkChange

Il est toujours utile de connaître les états d'une connexion réseau et si des changements interviennent. NetworkChange permet de le savoir. Ainsi, on peut détecter si une adresse IP change sur une interface réseau ou si un câble réseau se déconnecte. De plus, via System.Net.NetworkInformation namespace, il

1 ► FRAMEWORK ET LANGAGES



Modélisation d'un Data Center



Le nouveau split

est possible d'accéder aux statistiques réseau IP, Ipv4, Ipv6, TCP et UDP (informations similaires à celles fournies par Ipconfig.exe).

Et aussi

- FTP : on dispose désormais d'un accès direct aux fonctions FTP.
- Remoting : Le remoting de .NET supporte maintenant IPv6 et il est possible d'échanger des types génériques. Différentes améliorations de classes ont été réalisées (support de SSPI pour la sécurité, meilleurs load-balancing...).
- Compression – décompression : Pour pouvoir compresser et décompresser des fichiers / données, on peut maintenant utiliser System.IO.Compression. Il permet d'utiliser le standard Gzip.
- ACL : Comment autoriser ou non l'accès aux ressources d'un ordinateur ? On dispose avec .NET v2 de l'Access Control List. Il est possible de créer et de modifier une ACL directement en code .NET.
- Mail : Pour pouvoir envoyer et recevoir des mails dans son application, .NET v2 propose le support de SMTP (via les espaces de noms : System.Net.Mail et System.Net.Mime).

Internationalisation

La localisation des applications devient une fonction de plus en plus importante dans un projet. .NET v2 propose plusieurs nouvelles fonctions pour améliorer ce support. Pour pouvoir créer une application s'adaptant à des cultures non prévues par défaut dans .NET, on peut alors créer de nouvelles cultures grâce à CultureAndRegion InfoBuilder class. Ainsi, lorsque l'on lance une application, elle chargera la culture adéquate. C'est très important pour pouvoir prendre les bons formatages des dates, heures, boutons, etc. Microsoft en a profité pour prendre en compte les dernières spécifications d'Unicode.

XML

XML subit pas mal de modifications (System.Xml), à savoir :

- nouveau modèle pour les données XML
- refonte de l'architecture XSLT, notamment grâce à l'ajout de la classe XslCompiled Transform qui doit, selon l'éditeur, améliorer les performances par rapport à l'ancienne classe XslTransform.
- Nouveau modèle de création d'objets XmlReader et XmlWriter.

Microsoft prévient qu'avec les multiples changements et améliorations dans les classes XML, il sera nécessaire de migrer le code 1.1 vers la 2.0. Sur les Web Services, on bénéficie maintenant du support de SOAP 1.2 et du WS-I Basic Profile 1.0.

Les génériques

Comme Java 5 le propose, .NET introduit dans sa v2, les generics. Ce concept n'est pas nouveau, il remonte aux templates C++. Le but des generics est de faciliter la réutilisation du code. Ils permettent à des classes et méthodes de travailler avec des types différents. On ne s'occupe plus du typage. Les génériques sont supportés par VB.NET, C# et C++.

Les classes partielles

Les classes partielles ou, partial class, sont une innovation intéressante pour le développeur. Basiquement, cela permet de répartir le code d'une classe sur plusieurs fichiers. Dans chaque fichier source, il y a une section de description de la classe. Lors de la compilation, les éléments seront combinés et compilés.

.NET Compact Framework

La partie mobile évolue aussi. La v2 introduit de nombreuses nouvelles classes issues du framework .NET. Les performances, annoncées comme meilleures, se concentrent sur le JIT, le garbage, XML, les Web Services et les accès aux données. Le Compact Framework 2 supporte maintenant les appels d'objets COM à partir d'un code managé. L'appel à partir de COM d'un objet managé est prévu.

■ François Tonic

Support étendu

Cryptographie	Cryptographie et certificats X509 sont supportés
Assembly	Les Friend Assemblies de C# sont supportés
Réseau	Implémentation de l'authentification NTLM, support de IPv6 et du CredentialCache.
XML	Support de XmlSerializer, System.Xml.Xpath et System.Xml.Schema

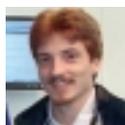
Nouvelles fonctions

Développement	<ul style="list-style-type: none"> - compilation en ligne de commande possible - SDK du Compact Framework inclus dans le SDK du Framework .NET - Nouveaux contrôles d'interfaces
---------------	---

1 ► FRAMEWORK ET LANGAGES

Témoignage d'un bêta-testeur

CodeSourceS : un test grandeur nature

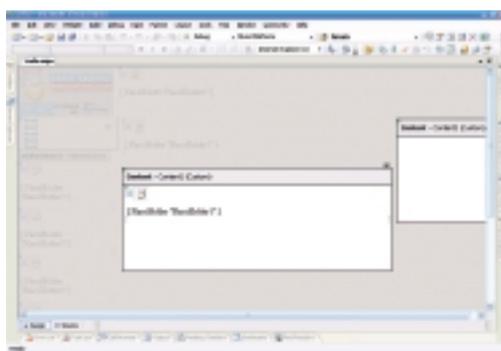


CodeSourceS est un des sites leaders francophones dédiés au développement. Son responsable, homme à tout faire, Nicolas Sorel, prépare activement une grande migration du site vers les technologies .NET 2.0 de Microsoft qu'il teste depuis de longs mois. Expérience visiblement positive.

Aujourd'hui, le site absorbe 200 - 250 000 pages vues par jour, plus de 60 000 visiteurs uniques / jour, 27 000 codes sources partagés sur pas moins de 17 langages, 400 000 messages, des forums, des blogs, un chat et un site d'actualités .NET remis à jour automatiquement via RSS. Si la nuit, l'activité est calme, la journée, le site doit encaisser de très fortes charges. Pour faire tourner tout cela, Nicolas Sorel utilise seulement un gros serveur Web fonctionnant sous Windows Server 2003, et un petit serveur de données pour y stocker les bases SQL Server. Les pages du site sont développées en ASP .NET et utilisent le framework .NET 1.1. À ses débuts, Nicolas utilisait une base MySQL, mais à cette époque, la liaison ODBC fonctionnait mal, d'où le choix de SQL Server, même si "MySQL fonctionne bien" précise Nicolas. Actuellement, CodeSourceS ne représente pas moins de 60 000 lignes de code ! " Le côté positif avec ASP.NET et .NET 1.1, est que le site est une véritable application et non un simple site Web. Le point négatif est qu'il ne gère pas les templates, les skins. Pour changer l'interface des sites, cela demande beaucoup de temps. Et quand je fais une mise à jour de cache, on peut bloquer le site et le serveur s'encombre ".

Vers la v2...

" Malgré les défauts, .NET 1.1 est un bon outil, bien qu'il faille du temps pour s'y adapter (séparation des couches, ndlr), mais on y gagne. Par exemple, ASP était un simple script interprété, ASP.NET fournit une véritable application. " poursuit Nicolas. .NET 2.0 n'est pas une nouveauté pour ce super webmaster. Les premières démonstrations remontent à la



L'utilisation des master pages dans Visual Studio 2005 pour la prochaine version de CodeSourceS.



La même page dans le navigateur.

TechEd 2003. Il devient tout naturellement alpha et bêta testeur. Très tôt, il est impliqué dans les tests du futur outil, notamment avec le projet CodeWise. Il s'agit d'installer un web service sur un site communautaire .NET. Toutes les nuits, Microsoft interroge le service pour récupérer les données liées à .NET, afin de stocker les informations dans des fermes de données. Ces données sont ensuite interrogées par l'aide en ligne de Visual Studio 2005 ! Nicolas se montre plutôt enthousiasme sur la 2.0 de .NET. " J'ai débuté la phase de transition vers le framework 2.0 depuis début février. Tous les nouveaux développements

sont réalisés avec ASP.NET 2.0. Je gagne du temps, notamment avec l'utilisation des Master Pages ou encore pour les phases d'initialisations d'ASP.NET qui sont désormais masquées. On ne perd pas ces acquis entre la 1.1 et la 2.0. Il est très facile de migrer un projet 1.1 en projet 2.0. Sur les performances, impossible de répondre tant que ce ne sera pas mis en production. Cependant, la gestion du cache me semble bien meilleure. Avec ASP.NET 1.1, quand on avait deux navigateurs différents, il créait 2 caches différents. Avec la 2.0, même si on a plusieurs navigateurs différents, on n'aura qu'un seul cache. Sur la base de données, je resterais avec l'actuel SQL Server. La migration vers Yukon sera réalisée plus tard, quand j'aurai du temps. J'avoue que je ne l'ai pas réellement testée. ". CodeSourceS joue aussi la prudence en évitant de tout migrer d'un coup.

Pour le moment, aucun gros problème technique ne semble venir enrayer la mise en œuvre de .NET 2.0 qui sera mise en production dès que la bêta 2 sortira officiellement, la licence autorisant la mise en production avec cette bêta. On pourra alors encore mieux évaluer les évolutions du framework et les fonctions de Visual Studio 2005. Les nouveautés des langages de .NET 2.0 sont aussi de gros plus pour Nicolas Sorel, comme les classes partielles, le déploiement via Click Once et les génériques qui simplifient le travail de typage. Il estime aussi Visual Studio 2005, plus beau, mieux " fini ". " On garde les habitudes de la précédente version mais on dispose de nouvelles fonctions très utiles telles que le refactoring ou les diagrammes, un dataset typé plus joli. Je n'ai pas encore pu tout voir. J'ai notamment hâte de manipuler les outils de tests (ceux de Team System, ndlr) ". Et pour les développeurs restés à VB 6 ? La réponse est plus nuancée, tout dépendra de l'application. " Il faut voir ce que peut apporter une migration vers .NET. Il faut que l'on montre les avantages de la plate-forme et des langages, montrer des cas concrets, sans parler de version de .NET, la 1.1 ou de 2.0 ". ■ F.T.

ASP.NET 2.0 : Le développement Web monte en puissance chez Microsoft

La version d'ASP.NET 1.1 a apporté peu de changements par rapport à la v1.0, si ce n'est l'intégration en standard des Mobile WebForms. La création de pages dynamiques ne consiste plus à concaténer des chaînes de caractères, mais à manipuler le modèle objet de contrôles qui génère de l'HTML parfois riche à la place du développeur.

ASP.NET 2.0 apporte de nombreuses nouveautés et améliorations, en voici les principales.

Les Master Pages

Les fichiers portant l'extension .master permettent à un ensemble de pages d'un site, d'hériter graphiquement d'un même modèle. Ce sont des master pages. Elles se composent de balises HTML, de contrôles serveurs et de code, comme le sont des pages web dynamiques classiques (pages .ASPX). Une master page contient un ou plusieurs contrôles serveur `<asp:contentplaceholder>`. Ces contrôles sont des containers pour du HTML ou d'autres contrôles serveurs. Une page .ASPX faisant référence à une Master Page s'appelle page de contenu. Elle ne peut contenir à sa racine que des contrôles `<asp:Content>` lui permettant ainsi de remplacer le contenu des `<asp:contentplaceholder>`.

Les Thèmes et skins

Les thèmes permettent d'appliquer des chartes graphiques intelligentes à un site. Ils contiennent des feuilles de styles .CSS, des images et des skins. Les skins sont des fichiers permettant de modifier sur tout un site les attributs des contrôles serveur selon leur type ou leur appartenance à un même attribut skinID.

Les Data-Bound Controls

Le DataBinding d'ASP.NET 2 est maintenant bidirectionnel et profite non seulement aux contrôles existants, mais à de nouveaux tels que le TreeView, le Menu ou le GridView, qui est une sorte de super DataGrid paginable, triable, et éditible sans écriture de code ; il rend son prédécesseur obsolète.

Les Data-Source Controls

Les Data-Bound Controls sont peuplés à partir de DataSet ou bien de Data-Source Controls, contrôles capables de rechercher des données XML, relationnelles ou métier. Dans ce dernier



cas, plutôt que de définir des requêtes SQL ou des procédures stockées, il suffit d'écrire une classe comportant des méthodes de sélection, de mise à jour, de suppression ou d'ajout. Le passage d'arguments est déclaratif sous forme de tags et permet de chercher automatiquement la valeur d'un autre contrôle, d'un cookie, d'un champ de formulaire, d'un élément du profil de l'utilisateur, de la QueryString ou de la Session. Les Data-Source Controls assurent également des fonctions de cache, de pagination et de tri.

Les Data-Providers

Ce sont des classes permettant d'accéder à des sources de données OleDb, Access, SQL Server, Oracle, XML ou objets métier. La liste peut être étendue à d'autres sources.

La navigation

On bénéficie désormais d'un système de navigation cohérent et intégré, défini dans un fichier XML qu'il suffit d'appeler web.sitemap ou de créer un objet SiteMap contenant une collection SiteMapNode. Il reste ensuite à déposer tous les contrôles assurant la naviga-

tion sur la master page pour que chaque page puisse en bénéficier.

Ces contrôles sont :

- le SiteMapDataSource, qui assure la correspondance automatique entre l'Url de la page courante et la structure de navigation.
- Le contrôle SiteMapPath qui n'est rien d'autre qu'un fil d'ariane.
- Le contrôle Menu ou TreeView qui affiche l'arborescence du site.

La sécurité

Elle se base sur des services et des contrôles serveurs spécialisés. Chaque application ASP.NET 2.0 bénéficie donc d'un service de gestion des utilisateurs et des rôles, composé d'une base de données, de classes métier, de contrôles serveurs et d'une interface web d'administration. Parmi les composants fournis on trouve des contrôles d'inscription, création, édition, suppression d'un utilisateur, d'authentification par formulaire (login), contrôles de modification, d'oubli de mot de passe avec envoi d'email. Une interface web d'administration permet à l'administrateur du site de gérer les utilisateurs et rôles. Les contrôles fournis s'appuient sur ces services qui sont indépendants des bases de données. Microsoft fournit ces services pour les bases Access, SQLServer, ActiveDirectory. Ces services sont extensibles à d'autres bases telle qu'Oracle.

La personnalisation : les profils

La personnalisation consiste généralement à récolter, persister et restituer des données qui ne sont pas de type métier entre des visites successives d'un utilisateur, pour lui donner la sensation d'être reconnu ou de bénéficier d'un accueil personnalisé. ASP.NET 2.0 comprend désormais un service gestion de profils utilis-

1 » FRAMEWORK ET LANGAGES

teurs qui permet de gérer toutes ces informations très simplement pour le développeur. Il suffit de décrire sous forme de XML la structure des données d'un profil utilisateur dans le fichier web.config

```
<profile>
  <properties>
    <group name="Preferences">
      <add name="ShowQuoteOfTheDay" defaultValue="true" type="System.Boolean" />
      <add name="ShowNews" type="System.Boolean" />
    </group>
    <group name="BillingAddress">
      <add name="Street" type="System.String" />
      <add name="City" defaultValue="Toronto" type="System.String" />
      <add name="StateProv" type="System.String" />
      <add name="ZipPostal" type="System.String" />
    </group>
  </properties>
</profile>
```

Une classe Profile est automatiquement générée à partir de cette structure, qu'il suffit de manipuler ensuite avec le code suivant depuis n'importe quelle page web :

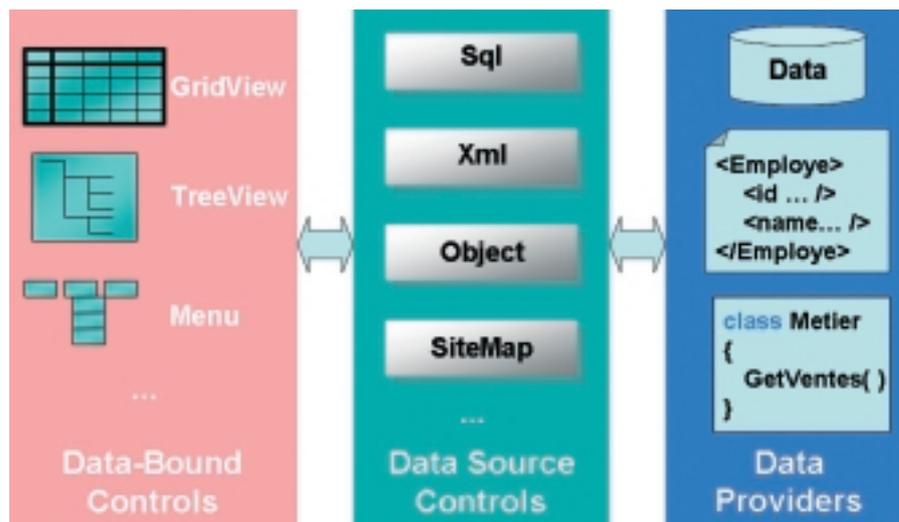
```
Profile.BillingAddress.City='Paris';
Profile.Preferences.ShowNews=true;
```

Les propriétés de l'objet Profile sont accessibles en lecture et en écriture et sont lues et persistées en base de données automatiquement. L'objet Profile est sérialisé automatiquement sous forme XML ou binaire, si bien qu'il n'y a jamais à modifier la structure de la base de données.

Les WebParts

ASP.NET 2.0 contient une douzaine de contrôleurs serveurs spécialisés dans la création d'interfaces de type portails. Un Portail est un site web composé d'une page d'accueil principale découpée en zones contenant chacune différents contenus (WebParts)

ASP.NET 2.0 utilise le même moteur de rendu de WebParts que celui de Sharepoint Portal Server 2003. Le choix des WebParts affichés à l'utilisateur peut tenir compte de ses droits et/ou de ses choix dans un catalogue. Les WebParts peuvent être minimisées, restaurées, fermées ou déplacées par Drag and drop par l'utilisateur. Toutes ces opérations sont auto-



matiquement persistées via la couche de personnalisation, si bien que l'utilisateur retrouvera le portail dans l'état où il l'avait laissé lors de sa dernière visite. La création de WebParts est vraiment très simple puisqu'un UserControl se transforme automatiquement en Webpart à l'exécution si on le dépose dans une WebPartZone. Ça va faire des jaloux chez les développeurs Sharepoint, car ils sont pour le moment obligés de générer des CustomControl pour créer des WebParts, c'est à dire obligés de définir toute l'IHM par code, alors que dans le cas des UserControls, il suffit de les dessiner... ;)

Les changements pour le développeur

Le développement ASP.NET 2.0 se fait, soit avec Visual Studio 2005, soit avec Visual Web Developer. Une application ASP.NET 2 peut contenir du code C#, VB.NET... même si celui-ci est écrit dans du code behind.

En ASP.NET 1.x une DLL était générée pour chaque projet par Visual Studio, et autant de DLL que de pages aspx étaient créées dynamiquement lors de la première exécution. En ASP.NET 2.0, les pages ASPX et les pages de code behind couplées sont automatiquement compilées dans une même DLL. La compilation dans Visual Studio ne sert plus qu'à trouver les éventuels bugs de compilation un peu plus tôt... Il n'y a plus de fichier de projet. La liste des références vers les composants utilisés se trouve maintenant dans le fichier web.config. Si du code doit être accessible par plusieurs pages ASPX, il suffit de le placer dans le répertoire App_Code, afin qu'il soit automatiquement compilé dans une DLL. IIS n'est plus un

pré-requis pour développer une application ASP.NET 2.0, Visual Studio possédant son propre serveur web.

Pour le déploiement, il est maintenant possible de précompiler l'ensemble du site pour s'affranchir du temps de compilation observable lors de la première demande d'une page. On peut également précompiler les pages .ASPX ; ce qui rend même mes tags invisibles en production. Fonctionnalité intéressante pour les éditeurs de logiciels...

La compatibilité ascendante

Des applications écrites pour ASP.NET 1.x et ASP.NET 2.0 pourront cohabiter sur le même serveur sans aucune gêne, chacune s'exécutant dans son framework respectif. Le choix du framework pour chacune des applications web est paramétrable depuis la console IIS, dans un onglet ASP.NET qui lui a été ajouté.

L'ouverture d'un projet ASP.NET dans Visual Studio 2005 permettra d'activer un assistant de conversion de votre ancien projet en ASP.NET 2.0, afin de pouvoir lui faire bénéficier des améliorations fonctionnelles de la nouvelle version. Et enfin, ASP.NET 2.0 est annoncé comme étant 100% compatible ascendant. Une application compilée pour ASP.NET 1.x fonctionnera sans aucune modification sur le framework 2.0 et bénéficiera à l'occasion d'une amélioration des performances de l'ordre de 10% d'après Microsoft.

■ Daniel TIZON - daniel.tizon@winwise.fr
Consultant - Formateur en technologies .NET chez Winwise. Winwise est une SSII spécialisée dans les Technologies Microsoft
<http://www.winwise.fr>

Nouveautés des langages C#, VB.NET & C++

Le .NET Framework 2.0, annoncé en parallèle de la nouvelle version du Visual Studio 2005 apporte de nombreuses nouveautés dans les différents langages. Le mot d'ordre semble orienté vers une réduction de 50% du code à générer !

A la première lecture, une classification sommaire nous permet de dégager les axes d'innovation suivants :

- Innovation des langages
- Perfectionnements des compilateurs
- Perfectionnements de la productivité
- Perfectionnements du débogueur

Comme nous pourrons le constater, les nouveautés et innovations des langages deviennent difficilement dissociables de Visual Studio 2005. Nous pourrions même, en faisant abstraction de l'approche purement puriste, constater que les nouveautés majeures reposent sur les combinaisons des langages et de leur cadre de développement respectif, soit les différentes versions du VS.NET. Bien qu'innovants, force est de reconnaître que des langages comme C# et VB.NET tendent à s'uniformiser et à devenir tellement semblables en terme de réalisations possibles qu'il devient difficile de tirer des arguments purement techniques sur le choix décisif de l'un ou l'autre.

Innovations du langage C# 2.0

C# ou C Sharp bénéficient de nettes avancées dans la programmation et sa structure. On peut citer en vrac : les Génériques, les classes partielles, les méthodes anonymes, les itérateurs, les qualifications d'alias et des classes statiques, ...

1. Génériques

Présenté comme étant une des caractéristiques prépondérantes, soutenue par Bill Gates, du C# 2.0, les Génériques apportent la capacité de réutilisation générique des algorithmes. Couramment utilisés en C++ comme Template, l'absence des Génériques sous C# fut un reproche des utilisateurs auprès de Microsoft (même reproche fait par les développeurs Java à Sun, d'où leur implémentation dans Java 5, ndr). Avant l'apparition des génériques, on aurait écrit le code suivant :

```
public class List
{
    private object[] elements;
    private int count;

    public void Add(object element) {
        if (count == elements.Length) Resize(count * 2);
        elements[count++] = element;
    }

    public object this[int index] {
        get { return elements[index]; }
        set { elements[index] = value; }
    }

    public int Count {
        get { return count; }
    }
}
```

En l'exploitant de la manière suivante :

```
List intList = new List();

intList.Add(1); // Boxing
intList.Add(2); // Boxing
intList.Add("Three"); // Génère une erreur à
// l'exploitation du code

int i = (int)intList[0]; // Le Cast est nécessaire
```

Cette méthodologie, bien que fonctionnelle, soulève quelques problématiques : le transtypage explicite nécessaire, la diminution des performances de ce code lors de l'étape implicite de boxing et unboxing des éléments, la possibilité d'empiler des éléments de types différents pouvant engendrer des problématiques de transtypage, uniquement décelables lors de l'exécution du code, nous parlerons dans ce cas de code non Type-Safe.

L'entrée des Génériques permet d'écrire le code suivant :

```
public class List<T>
{
    private T[] elements;
    private int count;

    public void Add(T element) {
        if (count == elements.Length) Resize(count * 2);
        elements[count++] = element;
    }

    public T this[int index] {
        get { return elements[index]; }
        set { elements[index] = value; }
    }

    public int Count {
        get { return count; }
    }
}
```

Et en exploitation :

```
List<int> intList = new List<int>();

intList.Add(1); // Pas de boxing
intList.Add(2); // Pas de boxing
intList.Add("Three"); // Erreur levée lors de
// la compilation

int i = intList[0]; // Aucun Cast nécessaire
```

Cette nouveauté évite les problèmes mentionnés. On bénéficie ainsi d'un code performant, Type-Safe n'engendrant aucune opération de boxing/unboxing sans transtypage des éléments.

2. Classes partielles

Les classes partielles représentent une nouveauté intéressante et plus particulièrement dans le cadre du travail collaboratif. Les classes pourront être écrites sur plusieurs fichiers, grâce à la syntaxe " public partial class ...". Une nuance à ce titre entre VB.NET

1 ► FRAMEWORK ET LANGAGES

et C#, VB.NET tolère une classe déclarée en "Public Class".

Exemple :

```
// ClassPart1.cs
using System;

public partial class ClassPart
{
    public void Methode()
    {
    }
}

// ClassPart2.cs
using System;

public partial class ClassPart
{
    public void AutreMethode()
    {
    }
}
```

3. Méthodes anonymes

Les méthodes anonymes permettent d'instancier une méthode sans la nommer. Contrairement aux Génériques, cette nouveauté n'engendre aucune nouvelle implémentation au sein de l'IL (Intermediate Language) mais se situe au niveau du compilateur.

```
button.Click += delegate { MessageBox.Show("HelloWorld"); };
```

L'exemple suivant montre une méthode anonyme avec des arguments.

```
button.Click += delegate(object sender, EventArgs e) {
    MessageBox.Show(((Button)sender).Text);
};
```

4. Itérateurs

Les itérateurs simplifient le processus d'itération de collection. En C#, il est relativement simple d'itérer des collections finies d'éléments, en utilisant le mot clé "foreach". Ceci nécessite l'implémentation des interfaces IEnumerable et IEnumerator. L'objectif premier des itérateurs est d'alléger le travail d'écriture des zones fixes de code et de rendre plus simple l'exposition de collection énumérables.

```
class List<T> : IEnumerable<T>
{
```

```
private T[] elements;
public IEnumerator<T> GetEnumerator()
{
    foreach (T element in elements)
    {
        yield element;
    }
}
```

Il ne faut pas oublier d'implémenter "System.Collections.Generic" afin d'exploiter la version Generic de IEnumerator : "using System.Collections.Generic; ". Vous aurez aussi remarqué l'utilisation d'un nouveau mot clé "yield" qui est utilisé dans notre méthode comme retour de IEnumerator, IEnumerable ou tout autre Generic équivalent. Tout comme les méthodes anonymes, les itérateurs gèrent des arguments.

5. Qualificateur global d'espace de nom

L'accès à un membre dans l'espace de nom global s'avère être utile quand le membre pourrait être caché par une autre entité du même nom. Le qualificateur global d'espace de nom résout ce problème par l'introduction de l'opérateur "::" qui peut être employé comme espace de nom ou type préfixe nommé.

```
namespace Acme
{
    namespace System
    {
        class Example
        {
            static void Main()
            {
                ::System.Console.WriteLine("Hello");
            }
        }
    }
}
```

6. Classes Statiques

Les classes statiques prévoient de remplacer les design patterns de création de classes de type "sealed" avec un constructeur "private" contenant uniquement des méthodes statiques. Le code suivant :

```
public sealed class Environment
{
    // Keep class from being created
    private Environment() { }
}
```

Devient :

```
public static sealed class Environment
{
}
```

L'avantage d'employer une classe statique au lieu du modèle de conception ci-dessus est que le compilateur peut maintenant rapporter une erreur si des instances de méthodes sont accidentellement déclarées.

7. Types Nullables

Si vous n'aviez jamais imaginé assigner des valeurs nulles à vos types "valeur", détrompez-vous ! Vous pouvez désormais assigner des valeurs nulles à des types valeur en définissant un type nullable. Ceci en ajoutant un point d'interrogation "?" immédiatement à droite du type, lors de la définition de votre variable. Les types nullables dérivent du type générique System.Nullable<T>. T étant votre type. Deux propriétés étendent l'utilité des types nullables : "HasValue" et "Value". HasValue évalue le caractère "nulle" de votre type tandis que Value retournera la valeur fondamentale, que celle-ci soit nulle ou non.

//les deux déclarations suivantes se réfèrent au même type nullable d'integer

```
int? myInt1 = null;
Nullable<int> myInt2 = null;

if (myInt1 == null)
    Console.WriteLine("myInt1 is null.");
if (myInt2 == null)
    Console.WriteLine("myInt2 is null.");
```

// Nous pouvons examiner le type valeur pour nous assurer d'obtenir une valeur de cette manière

```
myInt1 = 1;
if (myInt1.HasValue)
    Console.WriteLine("myInt1 has a value = {0}", myInt1.Value);
else
    Console.WriteLine("myInt1 is null.");
```

Innovations du langage VB.NET 2.0

Les nouveautés sont toutes aussi importantes que celles de C# 2.0. Certaines sont communes, d'autres non. Une des plus remarquées concerne l'objet My.

1. My

L'objet "My", exclusivité de VB.NET, permet d'avoir accès rapidement à un ensemble de fonctionnalités.

Exemple de chargement d'une fichier son :
My.Computer.Audio.Play("c:\redo\fichier.wav")
Nous y retrouverons : My.

- Application : informations de version, logs, ...
- Computer : impression, base de registre, audio, fichiers, ...
- User : nom d'utilisateur, groupe, domaine, ...
- Resources : icônes, images, ...
- Settings : paramètres des applications et utilisateurs
- Forms : collection de formulaires
- WebService : collection des web services référencés dans le projet

2. Surcharge d'opérateurs

.NET Framework autorise la surcharge des opérateurs. Ce dispositif permet de consommer des opérateurs dont la fonctionnalité a été surchargée par une classe particulière, ce dispositif n'était pas disponible dans les versions précédentes, et permet aussi au développeur de surcharger les fonctionnalités des opérateurs standards des langages comme +, -, *, / et bien d'autres.

```
Dim p1 As New Point(100, 100)
Dim s1 As New Size(20, 30)
```

```
Dim p2 As Point = p1 + s1
Dim p3 As Point = p1 - s1
```

p2 contenant ainsi {120, 130} et p3 contenant {80, 70},

Cette surcharge n'est pas limitée à ces quelques opérateurs, mais s'étend à d'autres opérateurs comme /, ^, &, Like, Mode, And, Or, Xor, Not, <<, >>, = (comparaison), <, <=, >, >=, CType, IsTrue et IsFalse.

3. Mots-clés IsNot & Using

Une des innovations se trouve également dans l'introduction de ces deux mots-clés. Voici ce que nous devons écrire :

```
If Not (obj Is Nothing) Then
```

Nous écrirons maintenant :

```
If obj IsNot Nothing Then
```

Plus intuitif ! Dans VB.NET 2005 apparaît le mot clé Using qui se chargera de créer un bloc

de code qui sera disposé lors de la fin de l'utilisation de ce bloc, peu importe comment vous échapperez ce bloc de code. Un exemple repris de la msdn :

```
Private Sub Form1_Paint( _
    ByVal sender As Object, ByVal e As Paint
    EventArgs) _
    Handles MyBase.Paint
    ' Create a new font.
    Using myFont As New Font("Verdana", 12)
    ' Create a new solid brush.
    Using myBrush As New SolidBrush( _
        Color.FromArgb(123, 0, 123))
        e.Graphics.DrawString( _
            "Hello, World!", myFont, myBrush, 0, 0)
    End Using
    End Using
End Sub
```

4. Types Integer non signés

VB.NET 2005 prend ainsi nativement en charge les types signés et non signés ainsi que l'ensemble des méthodes nécessaires à la conversion de ceux-ci. Les types SByte, UShort, UInteger et ULong.

5. L'état Continue

VB.NET 2005 intègre le " Continue ". Cette syntaxe est déjà présente en C#. Elle permet de passer directement à l'itération suivante sans exécuter le code de votre Do, While, For.

Innovations du langage C++

Terminons enfin sur un rapide passage en revue des innovations du langage C++.

1. Disparition du __

Les __ des extensions managées sont remplacés par des mots-clés.

Quelques exemples :

CLI Type	Managed Extension	C++/CLI
reference class	__gc class R	ref class R
value class	__value class V	value class V

2. Tracking Handle (^) remplace le pointer (*)

Pour la déclaration d'un type managé, le pointeur n'est plus utilisé comme c'était le cas en V1.

// V2 declaration of a CLI reference type

```
String ^ ps = "a string literal";
```

3. Apparition du mot-clé gcnew

// V2 Syntax

```
StreamReader ^file = gcnew StreamReader( file );
NativeClass * pnc = new NativeClass( args );
```

4. Apparition du mot-clé nullptr

// V2: OK ... we set obj to refer to no object

```
Object ^ obj = nullptr;
```

// V2: OK ... we initialize obj2 to a Int32^

```
Object ^ obj2 = 1;
```

5. Déclaration simplifiée des tableaux

// V2 Syntax

```
void PrintValues( array<Object>^ myArr );
```

```
void PrintValues( array<int,3>^ myArr );
```

6. Propriétés unifiées

// V2 Syntax

```
public ref class Vector sealed{
    float _x;
public:
    property double x
    {
        double get(){ return _x; }
        void set( double newx ){ _x = newx; }
    } // Note: no semi-colon ...
};
```

7. Les opérateurs

// V2 Syntax

```
public ref class Vector sealed {
public:
    Vector( double x, double y, double z );
    static bool operator ==( const Vector^,
const Vector^ );
    static Vector^ operator /( const Vector^,
double );
    static Vector^ operator +( const Vector^,
const Vector^ );
    static Vector^ operator -( const Vector^, const
Vector^ );
};
```

```
int main()
```

```
{
    Vector^ pa = gcnew Vector( 0.231, 2.4745, 0.023 ),
    Vector^ pb = gcnew Vector( 1.475,4.8916,-1.23 );
```

```
    Vector^ pc1 = pa + pb;
    Vector^ pc2 = pa - pc1;
    Vector^ pc3 = pc1 / pc2->x();
```

```
    if ( pc1 == pc2 ) // ...
```

```
}
```

■ Grégory Renard

Directeur Développement Wygwan



2 LES OUTILS

Migrer de .NET 1.x à .NET 2.0



La version VS 2005 vient de sortir en bêta 2 et vous vous posez légitimement la question de migrer ou non vos projets vers cette version et le framework 2.0. Tentons de répondre de manière succincte à ces différentes questions.

1. Questions préalables

Avant toute migration potentielle, quelques questions se doivent d'être posées :

- Pourquoi migrer vers .NET 2.0 ?
- Comment migrer vers .NET 2.0 ?
- Mes connaissances 1.x seront-elles obsolètes en 2.0 ?
- Mon projet en 1.1 tournera-t-il correctement sous la 2.0 ?
- Les versions 1.1 et 2.0 pourront-elles cohabiter sur un même serveur ?
- Quelles sont les upgrades à opérer sur mes machines pour la 2.0 ?
- ... et bien d'autres questions !

2. Pourquoi migrer vers .NET 2.0, ses implications ?

Avant de se lancer corps et âme dans la migration d'un projet quelconque .NET 1.x vers la version 2.0, il est légitime de prendre un peu plus connaissance de notre environnement de développement futur.

Hormis les nouveautés des langages traités dans un autre article de ce numéro, il est intéressant de parcourir en quelques points les principales nouveautés du .NET 2.0, nous permettant ainsi de statuer sur la perspicacité d'une migration de notre projet 1.x.

- Amélioration des IDE respectifs de .NET 2.0
- Estimation de réduction d'exécution et de temps de lancement de 50% par rapport à la version .NET 1.1
- Réduction marginale des coûts de fonctionnement des processus
- Améliorations considérables de NGen (.NET Native Image Generator) au sein du .NET Framework 2.0
- Nouveautés des APIs
- UTF8Encoding, la traduction effectuée 2.5x plus rapidement, qui est une grande victoire pour ASP.NET
- Invocation des délégués 2x plus rapides
- etc.

Tout concorde ainsi à nous pousser à basculer en douceur vers le Framework 2.0. Mais la migration de nos serveurs peut-elle se passer aussi facilement que cela ? A cette question Microsoft répond que les systèmes d'exploitation ne devront subir aucun upgrade et qu'un minimum de changements est nécessaire.

3. Migrer de la CLR 1.x vers la CLR 2.0

La CLR1.x (Common Language Runtime), version commerciale de la CLI (Common Language Infrastructure, Standard ECMA), plus connue sous le nom .NET Framework, cohabitera sans crainte avec sa sœur la CLR 2.0. Nous retrouvons ici une compatibilité ascendante quasi parfaite (hormis quelques cas exceptionnels dans le cadre d'Interopérabilité poussée, de .NET Remoting violents, ...). Microsoft nous garantit que l'installation de la CLR 2.0 ne modifie en rien l'association de la CLR 1.x et des applications compilées dans cet environnement ! Après de nombreux tests, je ne puis que confirmer que je n'ai levé aucune exception.

La migration, ou peut-être la conversion, si nous faisons un rapide comparatif avec les différentes évolutions de la suite office, pourra s'envisager de différentes façons :

- Migration partielle
- Migration totale

Migration Partielle

La migration partielle d'un projet s'envisagera au travers de la scission de l'interface utilisateur, de votre framework métier et des assemblées référencées.

Les parties métiers pouvant être ou non recompilées dans la version 2.0.

Passons en revue ces différentes possibilités :

- Exécution de l'assembly 1.1 sous 2.0 sans recompilation de l'exécutable

Après modification du fichier de configuration de l'application afin de lui ajouter la balise de

notification de support de la CLR 2.0, il nous suffira de la relancer. Nous nous retrouvons bien ainsi dans une optique de fonctionnement en side-by-side, les breaking changing entre la version 1.x et la 2.0 sont très faibles ! En ce qui concerne les contrôles tels que les DataGrids, quoiqu'obsolètes dans la CLR 2.0, sont toujours opérationnels. Nous pouvons en déduire de cette manière que la majorité des projets développés sous la version 1.x tourneront sous une version 2.0.

- Evolution de l'interface utilisateur vers la version 2.0, l'assembly restant en Version 1.x

Ce second cas peut intervenir comme une phase intermédiaire de basculement de votre solution globale vers le .NET 2.0. Cette solution sera principalement retenue en vue de vous permettre d'incorporer les nouveaux contrôles du .NET Framework 2.0

- Recompilation des classes métier des assemblées référencées en 2.0

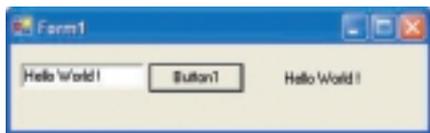
Ce cas est certainement un des seuls ne fonctionnant pas, et de surcroît cela semble logique, votre application 1.x ne pouvant consommer une assembly 2.0. Un message d'erreur sera alors levé par l'IL (CLR) qui préférera arrêter l'exécution de l'exécutable plutôt que de risquer de consommer des dll plus récentes. Ex : Consommation des Génériques !

Vous comprendrez que ces différents cas de figures ne sont qu'une solution transitoire et que notre objectif à terme est de devoir migrer l'intégralité sous la CLR 2.0.

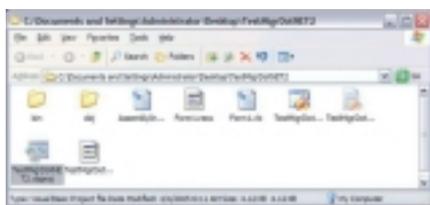
Migration totale

La migration de vos solutions de la CLR 1.x vers la CLR 2.0 pourra se dérouler au travers des assistants "Wizards" de migration de VS 2005. Nous en rencontrerons deux différents : un premier attaché aux projets de type WinForm, Librairies, ... et un second aux projets de type WebForm.

Voici un Exe réalisé sous la CLR 1.1 :



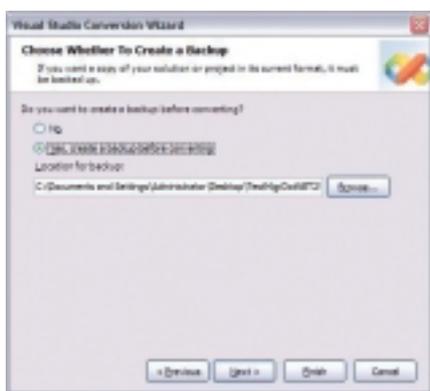
Par simple copier-coller, je déplace le répertoire de la solution sur ma machine de développement sous la CLR 2.0. Nous remarquons d'emblée que la version de la solution apparaît directement sur l'icône de celle-ci !



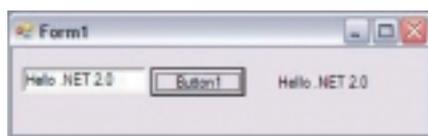
Dès l'ouverture de la solution, le wizard apparaît !



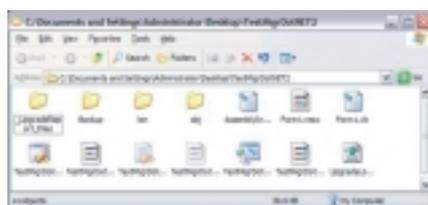
Un des points intéressants de ce wizard est le second écran, proposant d'effectuer une sauvegarde préalable de votre solution 1.1 avant la migration, juste pour les distraits.



Après 2 Next's, un Finish et un close, la solution sera migrée vers la CLR 2.0 et surtout vers une solution de type VS 2005. Un rapport de conversion vous sera généré, vous mentionnant les erreurs rencontrées lors de celle-ci.



Vérifions les MétaDatas via ILDASM : `// Metadata version: v2.0.50110`. Notre application a bien été migrée sous la CLR 2.0. Nous constaterons également que notre icône de solution est bien référencée par VS 2005.



Hormis le rapport de conversion, vous pourriez obtenir une présence de warnings vous stipulant qu'il y a de meilleures méthodes de développement dans la CLR 2.0 pour certaines fonctionnalités développées sous la CLR 1.x. Notons également que la conversion préservera votre application et n'engendrera pas de transformation de vos classes en classes partielles. Ainsi, le designer de VS 2005 est capable de lire les classes réalisées dans sa version précédente.

Vous retrouverez l'ensemble des changements sur GotDotNET : <http://www.gotdotnet.com/team/changeinfo/Backwards1.1to2.0/default.aspx>

4. Migrer d'ASP.NET 1.x vers ASP.NET 2.0

La migration d'un projet Web ASP.NET se trouve sensiblement dans la même lignée des applications de type WinForm.

Vous le comprendrez, notre serveur ne devra pas subir de mise à jour conséquente, hormis l'installation de la CLR 2.0. Nous garderons les configurations suivantes :

- Windows 2000 & IIS 5.0
- Windows XP & IIS 5.1
- Windows Server 2003 & IIS 6.0

a. Exécution d'un projet ASP.NET 1.x sous ASP.NET 2.0 sans recompilation

Tout comme les projets de type WinForm, une première méthode de migration sera d'exécuter notre projet ASP.NET 1.1 sous la CLR 2.0 au travers de notre IIS.

Pour effectuer ceci, nous pourrions opérer une manipulation traditionnelle au travers de l'exécutable "aspnet_regiis.exe" de la CLR 2.0 sur notre projet ASP.NET (Site Web). Afin de nous

simplifier cette manipulation, l'équipe de développement ASP.NET 2.0 de chez Microsoft nous a mis à disposition une interface intuitive !

Nous y accéderons avec l'onglet ASP.NET depuis le panneau d'administration du projet web IIS (Service Internet) ou nous viendrons modifier la liste de choix relative à la version de l'ASP.NET (induisant la version de la CLR) afin d'identifier celle à exploiter pour mon projet Web !

Une solution Web ASP.NET développée en version 1.1 tournera ainsi intuitivement sous la version ASP.NET 2.0.

Cet onglet nous permettra également d'accéder aux fichiers "machine.config" et "web.config" de votre application web.

Quelques subtilités seront à retenir en vue de migrer au mieux votre projet ASP.NET 1.1 vers une version normée 2.0 :

- Passage en classes partielles et nouveau modèle de Code Behind
- Incorporation du JavaScript en ressources du projet
- Ventilation des fichiers dans les répertoires de type "App_"
- Pré-compilation avec aspnet_compiler.exe

La migration d'un projet ASP.NET 1.x vers une version 2.0 pouvant faire l'objet à elle seule d'un article complet, je ne pourrais que vous conseiller de parcourir le document complet de migration d'ASP.NET 1.x vers ASP.NET 2.0 proposé par Microsoft à l'adresse suivante : <http://msdn.microsoft.com/asp.net/migration/default.aspx?pull=/library/en-us/dnvs05/html/migratefromaspnetto2.asp>

b. Wizard de migration d'ASP.NET 1.x vers ASP.NET 2.0

La seconde solution de migration d'une application web ASP.NET 1.x vers ASP.NET 2.0 est d'ouvrir votre solution 1.x (VS.NET 2003) à partir de votre VS 2005. Lors de l'ouverture de celle-ci, un nouveau Wizard, dédié aux projets Web s'ouvrira en vue de vous assister à la migration. Après quelques Next's, Finish et Close, un rapport de migration apparaîtra. Il vous restera à compiler votre projet Web ASP.NET pour vérifier qu'il est bien passé sous la version 2.0

■ Gregory Renard



Visual Studio 2005 : le meilleur des IDE



Visual Studio 2005 est le nouvel environnement de développement de Microsoft. Prévus pour le milieu de l'année, cette version est en rupture nette avec la version précédente : plus de fonctionnalités, plus d'outils de productivité pour les développeurs et les managers et enfin une prise en compte du cycle de vie du logiciel. Visual Studio 2005 est une première mise en œuvre concrète du concept de software factory élaboré par Microsoft.

Visual studio 2005 (VS), est proposé en plusieurs versions : Express, Standard, Professional et Team System. La version Express permettra des développements pour un coût très réduit. Les autres sont destinées à des petites équipes ou à des groupes plus importants pour la version Team System.

Interface graphique

Tout commence avec l'interface graphique : Microsoft propose, comme à son habitude, un nouveau look and feel pour son nouvel environnement de développement. La meilleure innovation est apportée au niveau du positionnement des barres d'outils : des indicateurs apparaissent à l'écran afin d'aider à leur placement, facilitant ainsi la personnalisation de l'environnement de travail. En revanche, le changement des raccourcis clavier risque d'en agacer plus d'un, mais, heureusement, Microsoft permet de les modifier au profit de ceux de l'ancienne version, voire de choisir ceux d'Emacs ! La navigation entre fichiers est également facilitée comme en témoignent la nouvelle fenêtre de navigation (ctrl-tab) ou les options proposées sur l'onglet des fichiers.

Editeur de code

VS propose dans sa version 2005 un menu contextuel dans la fenêtre d'édition, afin de remodeler très facilement des blocs de code. Il est désormais possible de renommer une méthode et d'impacter les sources dépendantes, de générer une méthode à partir d'une sélection (en générant les paramètres idoines), d'extraire une interface à partir d'une classe, de transformer une variable locale en paramètres etc. Ces fonctionnalités manquaient cruellement dans les versions précédentes, ce

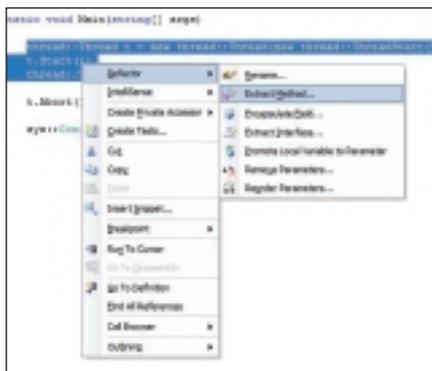


Fig. 1

qui imposait de nombreuses interventions manuelles de la part des développeurs pour faire évoluer les applications. L'automatisation de toutes ces tâches offre aux développeurs un gain de temps appréciable. La facilité d'intervention permet, par ailleurs, d'augmenter la fréquence des actions de refactoring au cours

de la vie d'une application. (Fig. 1) Le refactoring concerne la structure du code, mais VS2005 est également capable d'améliorer très sensiblement la présentation des codes sources. En effet, l'outil de reformatage s'est enrichi d'une pléthore d'options qui lui permettent de s'adapter à pratiquement tous les styles possibles. Qui n'a jamais été confronté à différents styles de développements dans une même application, voire un même fichier ? Le respect des styles pose, en effet, toujours problème car il demande un travail peu motivant pour les développeurs.

Désormais, espaces, position des accolades, indentations des labels, etc., tout est configurable ! C'est un grand pas en avant pour la démarche d'uniformisation des codes sources dans les projets. (Fig. 2)

L'édition du code est une chose, sa représentation graphique en est une autre. VS2005

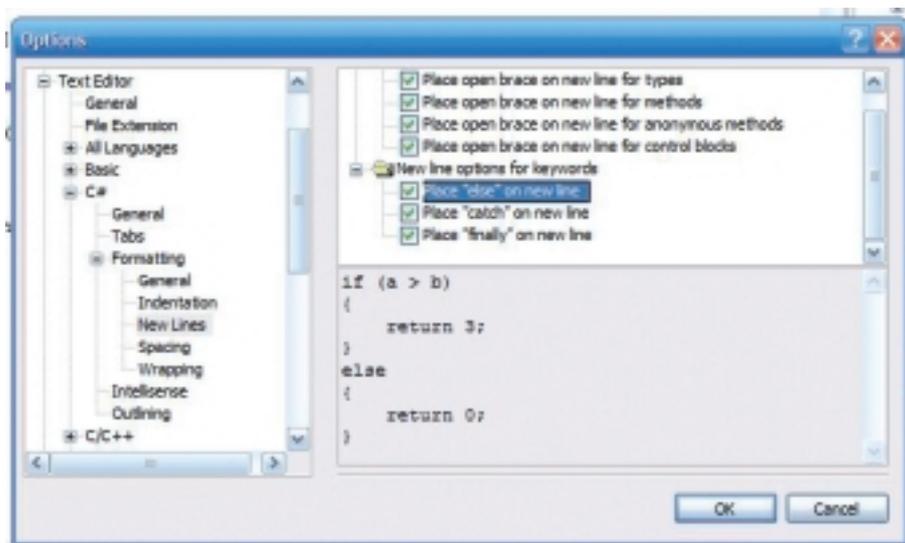


Fig. 2

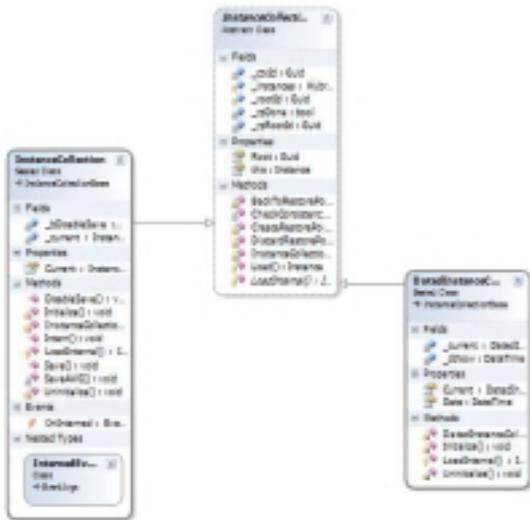


Fig. 3

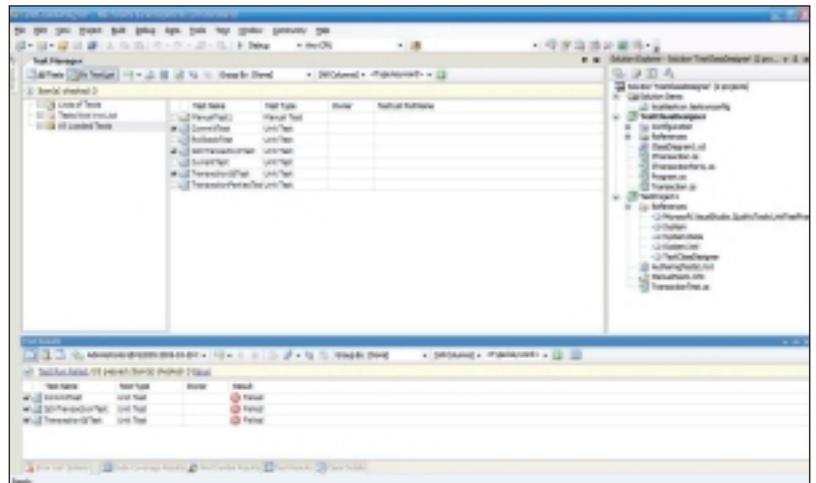


Fig. 5

intègre désormais, en natif, un éditeur pour diagrammes statiques de classes (class designer). Cet éditeur est capable d'extraire des informations à partir d'une base de code source existante (retro-engineering) afin d'en donner une représentation plus synthétique : relations, méthodes, interfaces. C'est un très bon outil qui permettra de gagner énormément de temps au cours de la phase de prise de connaissance d'un ensemble de codes sources. Il offre également la possibilité d'éditer des classes directement à partir de cette vue, ce qui permet de concevoir un ensemble de classes très rapidement, sans avoir à écrire une seule ligne ! (Fig. 3)

De nos jours, peu d'applications nouvelles peuvent se passer d'XML. En effet, la manipulation d'instances XML ou XSD fait généralement partie du quotidien des développeurs et VS2005 apporte de réelles solutions de productivité dans ce domaine.

- IntelliSense sur le XML en fonction du schéma associé,
- Possibilité de débayer XSLT en fournissant une instance XML en entrée,
- Validation des instances XML en fonction du schéma associé,
- Possibilité de mettre des éléments XML en commentaire rapidement comme dans n'importe quel autre code...

Outils pour l'adoption de bonnes pratiques

Dans cette catégorie, on peut ranger tous les artifices qui permettent de faciliter le développement.

La gestion des ressources était auparavant un

vrai casse-tête. Il fallait la plupart du temps recourir à des outils externes comme, par exemple, l'excellent Ressourcer de Lutz Roeder, afin de pouvoir l'ajouter à son application. L'éditeur de ressource de VS2005 copie largement cet outil et permet ainsi d'intégrer très rapidement. Manque encore à l'appel la possibilité d'internationaliser en utilisant des assemblées satellites. Parallèlement à l'édition des ressources, VS2005 génère automatiquement une classe pour les manipuler dans vos applications. L'usage de la classe Ressource Manager est ainsi totalement encapsulé, ce qui facilite l'adoption de la localisation dans les applications.

Les informations de paramétrage ne sont pas non plus en reste. Tout comme pour les ressources, VS2005 propose un éditeur et génère une classe, afin de gérer leur chargement et leur persistance. On pourra noter la facilité



Fig. 4

avec laquelle il est possible de gérer des informations au niveau de l'application, comme au niveau de son utilisateur.

Gestion des sources

Visual Source Safe est toujours présent... au grand dam de ses détracteurs ! Quoi qu'on en dise, cet outil a toujours rendu service aux petites équipes, grâce à sa simplicité de gestion et de mise en œuvre. Cette version offre de meilleures performances et une fiabilité accrue. Mais l'apport le plus intéressant est un mode de type client/serveur par HTTP à la Source Offsite : il est désormais possible de travailler à distance, sans avoir à recourir à des outils tiers.

Reste que Microsoft a, cette fois-ci, investi dans un nouvel outil qui fait partie de la gamme Team Foundation Server (TFS). Basé sur SQL Server pour la partie stockage, et résolument orienté client/serveur, cet outil est destiné à des équipes plus importantes et nécessitant une plus grande montée en charge.

(Fig. 4)

Tests

Des frameworks de tests unitaires ont fait leur apparition dans les équipes de développements et de qualité comme, par exemple, NUnit côté .NET. Ces outils ont prouvé leur efficacité pour assurer des tests de non régression et améliorer la qualité logicielle de manière substantielle. L'intérêt de ces outils réside principalement dans leur capacité à être intégrés dans des environnements automatisés de compilation d'applications. Côté IDE, mis à part l'excellent TestDriven.NET, aucun outil ne

2» LES OUTILS

permet d'effectuer facilement des tests.

Mais comme tout arrive à point à qui sait attendre, VS2005 intègre un outil de tests unitaires, mais ne s'arrête pas là. Parmi ses points forts, on notera :

- La compatibilité avec NUnit qui facilite la migration des jeux de tests,
- La possibilité de lancer les tests directement à partir de l'environnement,
- La visualisation du rapport de test,
- La possibilité de générer un test automatiquement en sélectionnant une méthode,
- L'analyse de la couverture du code.

Le dernier point est certainement la plus grande avancée dans ce domaine. Même si les outils de couverture de code ne sont pas toujours un bon indicateur de la qualité des tests, ils permettent néanmoins de découvrir très facilement des cas non testés. La facilité d'accès aux tests permettra à tout un chacun de définir et d'appliquer les tests de son choix afin de valider de nouveaux développements. Les tests seront ainsi exécutés au moyen du gestionnaire de tests qui permettra de sélectionner et de visualiser les résultats in situ. (Fig. 5)

Designers

VS2005 intègre des designers assez surprenants au premier abord, mais d'une redoutable efficacité! Imaginez la possibilité d'ajouter un service web ou une base de données à votre application en quelques clics. Imaginez que vous puissiez décrire rapidement vos plates-formes du point de vue des services proposés ou, mieux encore, d'être en mesure de déployer virtuellement vos applications sur vos plates-formes pour en tester la cohérence.

Avec VS2005, c'est possible !

VS2005 intègre un designer d'application qui permet de formaliser les schémas d'utilisation

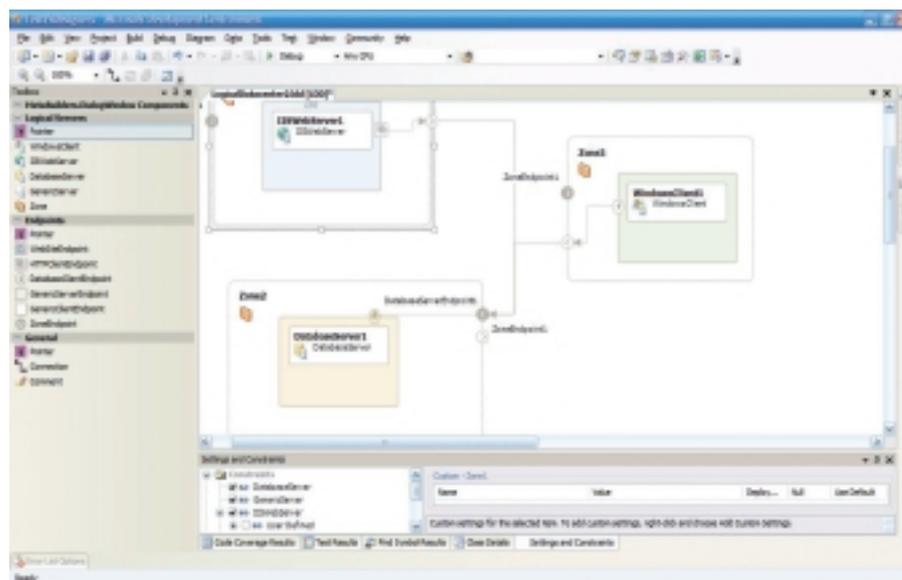


Fig. 6

des services applicatifs sous la forme d'un ensemble de contraintes nécessaires à leur déploiement. Par exemple, pour pouvoir exposer un service web, il est nécessaire, d'une part, de disposer d'un serveur web installé, et, d'autre part, de s'assurer de la disponibilité de l'ensemble des composants réseau qui permettent d'y accéder. Il faut noter que cet outil n'est pas seulement un outil de conception ; il permet également d'implémenter les services en générant du code automatiquement pour le développeur.

Le designer Logical DataCenter permet de modéliser l'architecture cible en définissant les éléments d'infrastructure qui supportent des services particuliers (IIS, SQL Server, .NET, Office...).

Le designer de déploiement permet de faire correspondre les besoins de l'application avec les ressources techniques disponibles dans le data center. L'outil permet, entre autres, de

détecter automatiquement les incohérences avant de procéder au déploiement. (Fig. 6)

Conclusion

Visual Studio avait accumulé un retard certain par rapport à d'autres environnements, comme Eclipse. Dans sa version 2005, Microsoft a tenu compte des attentes de ses clients et est allé au-delà en proposant les premières briques de sa vision de " Software Factory ".

Même si l'outil ne permet pas d'implémenter complètement ce concept, espérons que cette nouvelle version permettra aux développeurs de s'appropriier cette vision et de participer activement à son aboutissement.

■ **Pierre Chalamet** - pierre.chalamet@neoxia.com
Architecte chez NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.



www.programmez.com

www.programmez.com

www.programmez.com

Nouveaux services :

- Offres d'emploi
- Codes sources

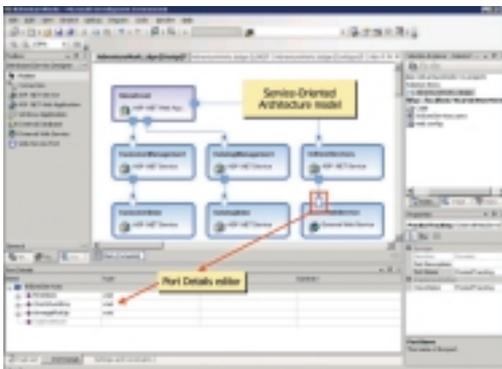
Des centaines de nouveautés, actualisées en permanence, avec nos partenaires

Visual Studio Team System

Jusqu'à présent, Microsoft ne fournissait pas ou peu de choses autour du test, de la modélisation, du travail en équipe, de la gestion des sources. Avec Team System, l'éditeur tente une incursion dans un marché qui lui est inconnu. Bien qu'incomplet, l'environnement est prometteur. Reste à savoir si les développeurs et entreprises se laisseront séduire. L'éditeur propose donc une gestion de cycle de vie assez complète (appelée SDLC par Microsoft pour Software Development Life Cycle).

Plutôt orienté entreprise, Team System a pour objectif de fournir une "software factory", c'est à dire, réduire le temps de développement, automatiser un maximum de fonctions, avoir un environnement intégré, réaliser du code et des applications de qualité. Team System mise sur les bonnes pratiques, le travail en équipe, la gestion de configuration et de projet ainsi que l'intégration continue et l'utilisation massive de métriques.

Pour cette première version, Team System ne



possède pas de fonctions de gestion des exigences formelles, il faudra se rabattre sur les solutions du marché, à l'instar d'un CaliberRM (Borland). Team se découpe en trois éditions : Team Architect, Team Developer, Team Test. La partie serveur a pour nom Team Foundation. Les trois éditions possèdent un socle commun : Visual Studio Professional, ensuite, chaque version possède des modèles plus ou moins spécifiques. Pour fonctionner, Foundation n'est pas indispensable. Ce que l'on ne sait pas toujours, est que Team System s'appuie sur MSF 4.0. MSF (Microsoft Solutions Framework) est un framework offrant les bonnes pratiques utilisées dans Team System. C'est ce que Microsoft nomme "méthode

agile". Par défaut, on dispose de MSF et MSF Agile. Mais comme le framework est extensible via un SDK et des API, il est possible d'implémenter sa propre méthode agile ! Foundation constitue la pièce maîtresse de l'ensemble. Il s'agit d'un référentiel centralisant l'ensemble des données et informations liées aux projets. Il utilise SQL Server 2005 pour la partie stockage.

Une automatisation poussée pour faciliter la vie

Foundation est la tour de contrôle. Il stocke les projets, les métriques, les méthodes, les sources, la documentation. Surtout, il ne se limite pas à du classique client / serveur. Il propose un portail projet accessible depuis n'importe où ! L'outil est orienté services. Car on peut invoquer les services de Foundation via des Web Services. Il utilise WSS (Windows SharePoint Services) et des outils de reporting de projets pour suivre l'évolution de chaque tâche et du projet global.

La création de projets est assez simple. On crée un projet Team System. On se connecte au référentiel puis on choisit la méthode vou-

lue. L'environnement génère automatiquement la structure du projet (site portail, infrastructure de reporting...). Les templates sont eux aussi créés automatiquement. On choisit les développeurs intervenant sur le nouveau projet, avec les niveaux d'accès. Puis, on définit et configure le site du projet et les Work Item. Il s'agit d'actions, de tâches, de bugs, de besoins que l'on définit par rapport aux projets en cours. Foundation s'occupe ensuite de faire la synchronisation et la notification (par mail essentiellement) que les développeurs reçoivent dans son environnement.

Comme tout est intégré dans l'environnement, les nouvelles fonctions de tests, le sont aussi. Il s'agit d'un item comme un autre. On peut construire un véritable squelette de tests et des scénarios. Si Foundation n'est pas utile pour les fonctions de tests, pour les scénarios de montée en charge ou d'automatisation, le serveur est indispensable. Ainsi, durant la nuit ou la période précise, on peut exécuter des automates de tests sur le serveur après compilation. Pour le moment, on sait que Team System ne gère que les projets .NET 2.0. On ne sait pas encore si on pourra faire de la compilation .NET 1.x. Côté modélisation, Microsoft ne croit pas (pas encore ?) à UML. Cependant, on dispose d'un designer d'application, de déploiement et de tests. Le designer d'application est bi-directionnel. Cela signifie que si on modifie le code, le modèle est modifié et vice versa. Enfin, on dispose désormais d'un "ANT / NANT like" avec MS Build. Il s'agit d'un fichier de configuration projet en XML. Une des utilités est de pouvoir prendre le projet et ce fichier et de faire une compilation sur un autre poste, même s'il ne dispose pas de Visual Studio.

■ François Tonic

Team Architect v1	Team Developer v1	Team Test v1
<ul style="list-style-type: none"> - support WSE - support Click-once - Designer de déploiement - Moteur de validateur - Designer d'application 	<ul style="list-style-type: none"> - support analyseur code statique (code managé, non managé) - profiler de code - code coverage - tests unitaires (similaire à NUnit) 	<ul style="list-style-type: none"> - test de charge - scénario de tests
Team Architect v2	Team Developer v2	Team Test v2
<ul style="list-style-type: none"> - support d'Indigo - Designer Business Process - Moteur de pattern 	<ul style="list-style-type: none"> - profiling au niveau système et monitoring d'application - amélioration de l'analyse statique 	<ul style="list-style-type: none"> - test fonctionnel (Avalon, WinForms, Win32, IE) - gestion Test Lab - historique du code coverage

2» LES OUTILS

SQL Server 2005 : le killer d'Oracle et DB2 ?



Annoncé depuis environ 3 ans, SQL Server 2005, alias Yukon, aura fait couler beaucoup d'encre. L'enjeu pour Microsoft est immense. Produit phare des nouveautés 2005, Yukon doit permettre à l'éditeur de sortir SQL Server de son image actuelle, floutée par Access.

Encore aujourd'hui, SQL Server n'est pas vu comme un SGBDR sérieux, à l'instar d'un DB2 ou d'un Oracle. Yukon, avec les nombreuses avancées techniques peut enfin casser cette image non professionnelle mais, le pari n'est pas pour autant gagné. Il faut déjà qu'il sorte comme prévu à la fin de l'été et que les premières impressions se confirment. Difficile de toute décrire en une seule page. Ici nous ne retiendrons que l'essentiel. Très prochainement, un article plus complet sur SQL Server 2005 sera publié.

Yukon est une mise à jour importante de SQL Server. Cela pourrait se résumer ainsi : intégration de la CLR .NET, implémentation native de XML, procédures stockées et triggers, Web Services, amélioration support ADO.NET, modèle objet SMO, services de reporting, implémentation standard de SQL Server Mobile Edition...

Sécurité

Microsoft a beaucoup travaillé sur le modèle de sécurité de Yukon, afin de réduire la surface d'attaque et de mieux sécuriser les données. Il faut retenir deux notions importantes : Safe, Unsafe. Par défaut, Yukon est en configuration Safe, c'est à dire un accès restreint aux bases et fichiers. Seul le DBA peut alors fournir les droits d'accès. En Unsafe, on autorise les accès. Mais là, l'administration peut

autoriser tel ou tel accès, interdire celui aux ressources propres à SQL Server, selon l'utilisateur, etc. La granularité offerte aux DBA est donc très large.

.NET

Il s'agit d'une des nouveautés les plus attendues, DB2 le propose depuis plusieurs mois. Ainsi, on pourra créer des procédures stockées, des triggers, des fonctions avec les langages reconnus par la CLR .NET. On bénéficie donc avec Yukon d'une programmation managée .NET. On pourrait coder tout cela à partir de Visual Studio 2005. Grâce à ce modèle objet, on bénéficie d'une plus grande souplesse pour écrire des applications orientées données, qu'avec Transact-SQL.

Par exemple, Yukon prend en compte le ramasse-miettes. Selon Microsoft, l'intégration .NET a 4 avantages : meilleures performances et montées en charge, un modèle de développement unifié, meilleure sécurité (code managé), possibilité de définir types et aggregates. À vous de choisir entre Transact-SQL et .NET (et son code managé).

Pour les objets et les manipulations complexes, le managé est adapté. Mais attention à ne pas mettre dans une base tout en objet .NET, cela n'a aucun sens et les performances peuvent en souffrir.

XML

XML est un autre gros morceau de Yukon. Supporté nativement, il est possible de stocker les données sous cette forme, mais il n'est pas le format par défaut de SQL Server 2005. Qui dit XML, dit aussi Web Service. On peut développer des Web Services dans une base. On peut aussi accéder à une base Yukon directement par http. Yukon expose une interface sous forme d'un service web permettant d'exé-



cuter requêtes, fonctions, procédures. Yukon introduit un nouveau type de données dédié à XML (XML Data Type). Sous Yukon, les données XML sont stockées dans des BLOB pour en faciliter la manipulation et le stockage. On bénéficie aussi du Xquery pour réaliser des requêtes sur des données XML. Il utilise le langage Xquery. Yukon supporte aussi l'extension du langage Xquery, DML (XML Data Modification Language).

Reporting

Les services de reporting constituent une avancée importante de Yukon. Il doit faciliter la récupération d'informations de base pour son business. Ces services fournissent les fonctions complètes pour créer, gérer et fournir des rapports. On disposera pour ce faire, d'outils dédiés, d'un moteur de reporting et d'interfaces ouvertes et extensibles.

■ François Tonic

SQL Server 2005 Mobile Edition (ou SQL Mobile)

Un des axes de Yukon concerne la mobilité et les échanges de données entre la base, les applications .NET et les terminaux PocketPC / Smartphone (support de la future version de Windows Mobile Smartphone) et TabletPC. SQL Mobile a subi une remise à niveau des performances et des fonctions. La synchronisation entre les données et la base a été améliorée : support de souscription multiple pour la même base, support du multi-utilisateur, présence d'une API de statut de la synchronisation. L'architecture de stockage a été entièrement réécrite pour mieux s'adapter aux mobiles. SQL Mobile s'intègre naturellement aux autres outils v2005. Il est possible d'administrer la base mobile sur le terminal ou sur un desktop. La gestion, la création des bases et des requêtes ainsi que le déploiement reprennent le modèle de SQL Server 2005.

Visual Studio Express C# : un codage simplifié

Avec le prochain Visual Studio 2005, le développeur C# dispose d'un système RAD orienté code plus productif que jamais. Sans doute le meilleur outil de programmation C# !



Au moment de rédiger ces lignes, les prix de vente des différentes versions de VS 2005 ont été dévoilés par Microsoft. En résumé Visual Studio 2005 Professional Edition ainsi que Visual Studio 2005 Tools for the Microsoft Office System coûteront \$799 (upgrade : \$549), Visual Studio 2005 Standard Edition \$299 (upgrade : \$199), et Visual Studio 2005 Express Edition \$49. SQL Server express (le successeur de MSDE) restera gratuit. Les IDE de la famille Express sont payants mais ce n'est pas vraiment une surprise. Le prix est fort heureusement très modique même pour un étudiant. Nous pensons que l'édition Visual Studio 2005 Express est d'un excellent rapport qualité/prix. Bien entendu si vous voulez apprendre le C# il existe d'autres solutions comme ajouter un plug-in C# à Eclipse, ou bien développer avec MonoDevelop.

1 La désactivation des messages d'avertissements

Prenons le code suivant et compilons-le (ctrl-Shift-B sous VS 2005 Express bêta 2).

```
public class clx
{
    public int i;
}

public class clz
{
    public static void Main()
    {
        int j = 0;
        // j++;
        clx a;
        // clx a = new clx();
    }
}
```

Deux messages d'avertissement sont affichés : un de type "CS0219" (la variable 'j' est assignée, mais sa valeur n'est jamais utilisée), et un seconde de type "CS0168" (la variable 'a' est déclarée mais jamais utilisée). Nous pouvons obtenir ce numéro de code d'erreur en demandant le détail du warning (show error help), ou bien plus simplement en visualisant la sortie des messages du compilateur après un build. Pour ne plus voir apparaître ces messages d'avertissements, il suffit d'enlever les commentaires // se trouvant devant les lignes j++; et clx a = new clx();. En C#2, il est possible d'ajouter le code inline #pragma warning disable

Système d'exploitation : Windows XP
Environnement : Visual Studio C# 2005 Bêta 2
Langage : C#
Besoin : Vous devez télécharger la bêta 2

219,168 en début du bloc de code incriminé, et #pragma warning restore 219,168 pour restaurer l'affichage de ces avertissements en fin de bloc de code. Remarquez que les avertissements sont une bonne chose et que le fait de les masquer ne doit être tenté qu'en toute connaissance de cause.

2 Les tableaux fixes

Avec VS 2005, il est maintenant possible de définir des tableaux fixes à l'intérieur d'une structure. Si vous tentez de compiler le code qui suit sous VS.NET 2003, le compilateur vous affiche l'erreur "Syntax error, bad array declarator. To declare a managed array the rank specifier precedes the variable's identifier". Vous devez autoriser le code via le switch /unsafe que ce soit sous VS.NET 2003 ou VS 2005.

Sous VS 2005 le code se compilera sans problème. Un avertissement sera toutefois émis, du style "Field 'Program.FixedStruct.overflow' is never assigned to, and will always have its default value 0". En fait, comme il n'y a pas de vérification de dépassement, overflow sera bien garni de la valeur 3.

```
using System;
unsafe class Program
{
    unsafe struct tableauFixe
    {
        public fixed int array[10];
        public int overflow;
    }

    static void Main(string[] args)
    {
        tableauFixe fixedStruct = new tableauFixe();
        fixedStruct.array[10] = 2;
        fixedStruct.array[10] = 3;
        Console.WriteLine(fixedStruct.overflow);
        Console.ReadLine();
    }
}
```

2» LES OUTILS

3 Le Type Nullable

VS 2005 introduit un nouveau type qui est appelé "Nullable" :

```
struct Nullable<T>
{
    public bool HasValue;
    public T Value;
}
```

En C#, un type T peut être suivi par un point d'interrogation, ce qui aura pour effet de remplacer toute expression T? par Nullable<T>.

Autrement dit, la ligne `int? a = null;` est équivalente à `Nullable<int> a = null;`. Et plutôt que d'écrire `Nullable<int> i = new Nullable<int>(4);`, vous pouvez écrire `int? i = 4;`

Dans la même veine, plutôt que d'écrire `if (i.HasValue)` vous pouvez écrire `if (i != null)`.

Ainsi :

```
Nullable<int> a = new Nullable<int>(4);
Nullable<int> b = new Nullable<int>(3);
Nullable<int> i = (a.HasValue && b.HasValue) ? new Nullable<int>
(a.Value + b.Value) : Nullable<int>.NullValue;
```

Devient :

```
int? a = 4;
int? b = 3;
int? i = a + b;
```

4 Le Refactoring

Vous disposez d'un menu refactor, qui vous rendra plus productif que jamais, en vous permettant de retravailler le code source en vue d'améliorer sa structure. Attention, la version C# express (du moins la bêta 2) ne présente qu'Extract method et rename. Cette première méthode de refactoring consiste à extraire une méthode (Extract Method). Considérons le code suivant :

```
...
// Ferme les streams
GZipDeflateStream.Close();
fsSource.Close();
fsDest.Close();
...
```

Sélectionnons ce bloc, puis à l'aide du menu contextuel, ou du menu refactor, appliquons Extract Method. Le code sera alors restructuré pour prendre l'aspect suivant :

```
...
FermeLesStreams(fsSource, fsDest, GZipDeflateStream);
...
private static void FermeLesStreams(Stream fsSource, Stream fsDest,
Stream GZipDeflateStream)
{
    // Ferme les streams
    GZipDeflateStream.Close();
    fsSource.Close();
    fsDest.Close();
}
```

Tout ceci, dans le but de pouvoir réutiliser la méthode dans une autre partie du code. Vous rendez le code plus lisible, tout en supprimant de cette manière les blocs de codes redondants. Maintenant, vous pouvez à tout moment renommer le nom d'une variable, d'une méthode ou propriété. Il suffit de sélectionner le nom puis de sélectionner rename dans le menu contextuel des options de refactoring. C'est évidemment beaucoup plus intéressant que de procéder manuellement un remplacement (via le menu Edit / Find and replace).

Si nous prenons le code suivant :

```
...
int a;
...
```

L'option Encapsulate Field crée une propriété "emballée" autour du champ "a" :

```
int a;
public int A
{
    get
    {
        return a;
    }
    set
    {
        a = value;
    }
}
```

Plus spectaculaire, la promotion d'une variable locale au rang de paramètre. Prenons le code suivant :

```
using System;

namespace Exemple_refactoring
{
    class Programme
    {
        static void Main()
        {
            Console.WriteLine(Ajout(1, 2, 3, 4, 5));
            Console.ReadLine();
        }
        static int Ajout(int A, int B, int C, int D, int E)
        {
            int F = 6;
            int Somme = A + B + C + D + E + F;

            return Somme;
        }
    }
}
```

Pointons sur la ligne `int F = 6;` et appliquons-lui l'option de refactoring "Promote Local Variable to Parameter" (la variable locale est transformée en paramètre). Ensuite, sélectionnons la ligne de la fonction Ajout

et appliquons l'option "Reorder Parameters" pour transformer le code `static int Ajout(int F, int A, int B, int C, int D, int E)` en `static int Ajout(int A, int B, int C, int D, int E, int F)`, ce qui modifiera aussi l'ordre d'appel des arguments (6,1, 2, 1, 3, 4, 5,1 en 1,2,3,4,5,6).

```
using System;

namespace Exemple_refactoring
{
    class Programme
    {
        static void Main()
        {
            Console.WriteLine(Ajout( 1, 2, 3, 4, 5 ,6 ));
            Console.ReadLine();
        }
        static int Ajout(int A, int B, int C, int D, int E, int F)
        {

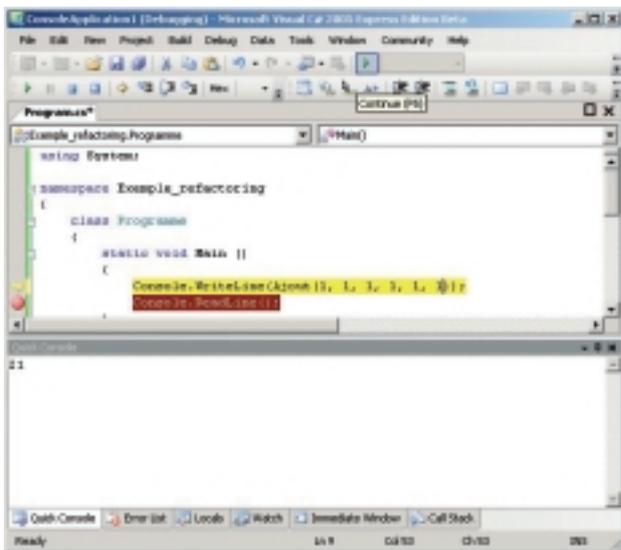
            int Somme = A + B + C + D + E + F;

            return Somme;
        }
    }
}
```

Vous pouvez aussi retirer un paramètre, même si l'intérêt dans ce cas est plus limité (Remove Parameters).

5 Edit and continue

Cette fonctionnalité très attendue est disponible à partir de Visual Studio (Express) 2005 bêta 2, et deviendra vite indispensable à n'importe quel développeur qui y goûtera. En effet, elle permet de modifier le code sans devoir le recompiler. Reprenons le bout de source précédent et plaçons-y un point de contrôle d'exécution (BreakPoint) sur la ligne `Console.ReadLine();`. Compilons et exécutons. L'exécution



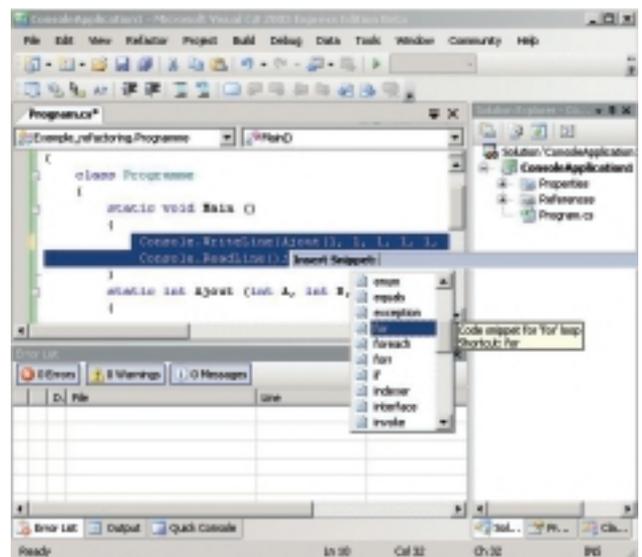
s'arrête sur la ligne marquée de ce BreakPoint. Nous pouvons maintenant modifier le code et, par exemple, changer `Ajout(1, 2, 3, 4, 5 ,6)` en `Ajout(1, 1, 1, 1, 1, 1)`. Nous déplaçons la flèche jaune (le pointeur d'exécution), placée pour l'instant sur le point rouge sur la ligne du dessus. Cette flèche indique où l'exécution va reprendre. Cliquons maintenant sur "continue" et la "quick console" nous indique "6" au lieu de "21". Nous pouvons à nouveau changer nos valeurs. A aucun moment le code n'a été recompilé. Toutefois, cette technique a ses limites. Par exemple, si nous tentons d'ajouter une autre méthode le logiciel nous affiche "Error 1 Adding a 'method' is not an allowed operation".

6 Les snippets

L'éditeur de code C# inclut une complétion automatique de code, du type "remplissez les blancs" pour les structures les plus courantes, comme les boucles ou les blocs de gestion d'exception. Si nous sélectionnons le bloc constitué par les lignes `Console.WriteLine()` et `Console.ReadLine()`; nous pouvons insérer un "snippet", c'est à dire une séquence de code prédéfinie. En sélectionnant le menu contextuel nous choisissons insert snippet puis "for" et le code se transforme automatiquement :

```
for (int i = 0; i < length; i++)
{
    Console.WriteLine(Ajout(1, 1, 1, 1, 1, 1));
    Console.ReadLine();
}
```

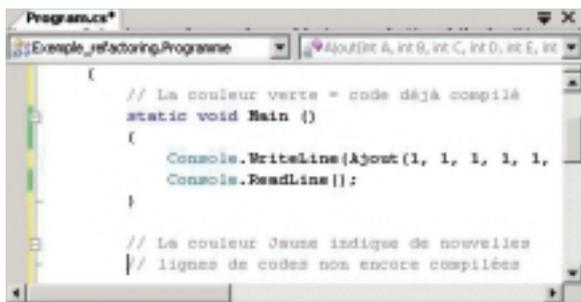
Reste à garnir `length` avec une valeur adéquate. Pour ce faire, il suffit de voyager avec TAB. Nous pouvons définir par exemple `CompteurAjout` à la place de "i" et automatiquement le reste de l'itérative se transformera en `for` (`CompteurAjout = 0; CompteurAjout < 3; CompteurAjout++`).



7 Le change tracking

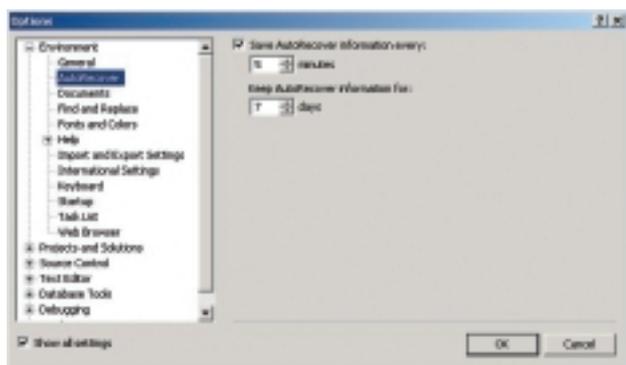
Une fine ligne verticale jaune apparaît dans l'éditeur lorsqu'un bloc de code n'a pas encore été compilé, sinon la ligne verticale est de couleur verte.

2» LES OUTILS



8 La sauvegarde automatique

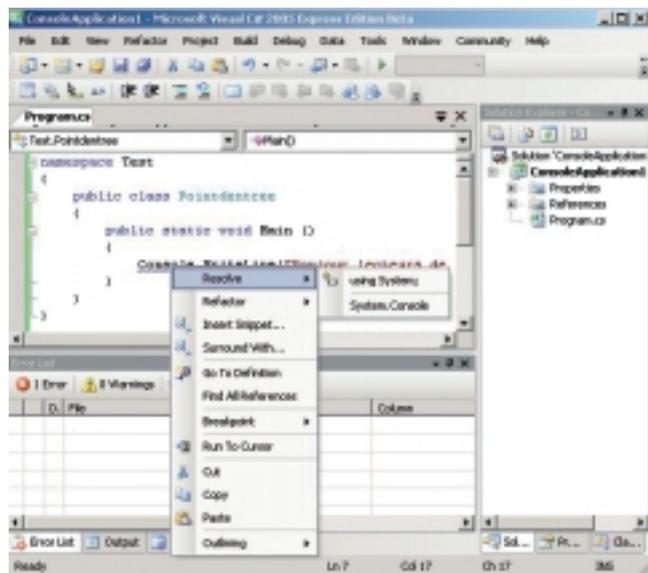
En éditant "Outils / Options / Autorecover", vous pouvez fixer l'intervalle entre deux sauvegardes. Si VS 2005 crashe, cette option permettra de retrouver votre travail dans l'état dans lequel vous l'aviez laissé.



se. Pourquoi ne pas avoir introduit cette fonctionnalité plutôt ? VS C# 2005 conservera ces informations x jours (second paramètre pouvant être fixé dans options / autorecover).

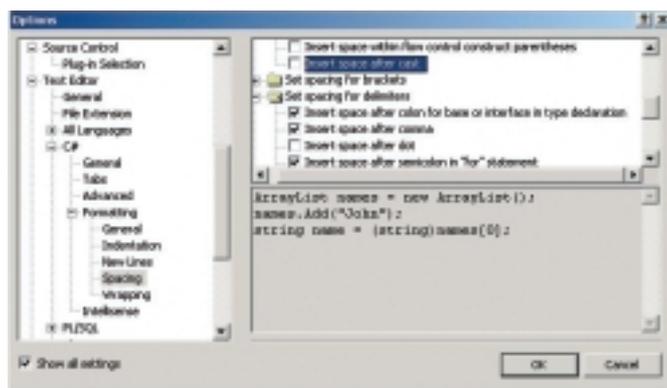
9 La résolution automatique des contextes d'appels

Si vous compilez le bout de code suivant sans indiquer l'utilisation de l'espace de nom System, le compilateur produira l'erreur attendue "The



name 'Console' does not exist in the current context". Avec VS C# 2005, il vous suffit de cliquer sur la ligne d'erreur, puis d'invoquer le menu contextuel pour corriger l'erreur. Deux solutions vous seront proposées, soit l'ajout automatique d'un using System; soit l'écriture de System.Console.WriteLine("Bonjour lecteurs de 'Programmez!'");

```
namespace Test
{
    public class Pointdentree
    {
        public static void Main()
        {
            Console.WriteLine("Bonjour lecteurs de 'Programmez!'");
        }
    }
}
```



10 Embellissement de votre code

Visual C# 2005 propose toute une panoplie de règles de formatage de votre code source. Par exemple, vous pouvez définir ou non un espace entre le nom de la méthode et de ses parenthèses. Ou encore, vous pouvez décider que la déclaration d'un bloc comme public int Age { get { return age; } } reste une seule ligne ou bien soit scindé en plusieurs lignes :

```
public int Age
{
    get
    {
        return age;
    }
}
```

Les possibilités sont très nombreuses et comme le programmeur C# passe beaucoup de temps à embellir son code, ces options de formatage ne sont pas une mauvaise idée.

Téléchargement des dernières bêtas : <http://lab.msdn.microsoft.com/vs2005/>

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

Optimisation, code et processeur : le difficile compromis

Il existe de multiples manières d'améliorer les performances d'une application. Le rôle du processeur n'est pas à négliger et le développeur peut s'en préoccuper durant le développement ou à la phase compilation. Mais faut-il réellement tirer parti du processeur, ou au contraire être le plus générique possible et éviter toute incompatibilité ?

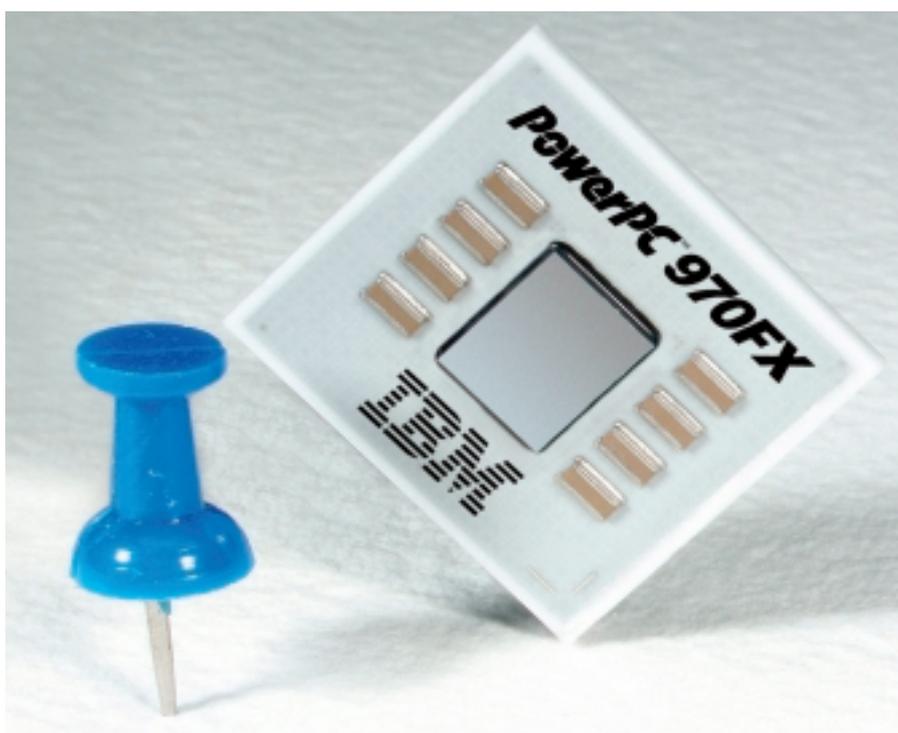
Pour qu'une application fonctionne convenablement, on peut définir le triplicte suivant : un code propre et optimisé, de bons outils, un bon processeur (avec des sous-systèmes adéquats). Il faut le bon compromis entre ces trois éléments. Si un des critères n'est pas bon, ou mal calibré, les mauvais comportements arrivent rapidement. L'objectif ici n'est pas de savoir programmer votre processeur, mais de savoir quand et comment en tirer profit. Car, si on veut réellement tirer parti d'une puce, il faut maîtriser le C++ ou l'assembleur et les structures / architectures du processeur et des unités vectorielles. Et aller coder à la main les appels et fonctions...

Choisir son application à optimiser

En premier lieu, toute application n'a pas besoin d'être optimisée par rapport aux processeurs. Ciblez vos applications les plus exigeantes en traitement de données (audio, voix, vidéo, compression – décompression, requête http, 2D / 3D, 64 bits, etc.). Dans ce cas, la question se posera d'utiliser des fonctions ciblées du processeur. Si vous avez un simple site Web, une application cliente interrogeant une base de données avec des flux moyens ou peu importants, les gains de performances seront souvent faibles. Dans les applications serveur importantes et critiques, l'optimisation processeur peut en revanche être une option à évaluer. Même si dans ce cas, il faut aller vérifier que le serveur active les fonctions processeurs.

Quelle optimisation réaliser ?

Comme dit plus haut, les méthodes d'optimisation sont multiples. Elles sont souvent complémentaires. Il y a les méthodes rapides et peu exigeantes et des méthodes plus lourdes.



Il n'est pas toujours nécessaire de passer par la programmation pure et dure pour optimiser son code. De plus, il ne faut pas forcément tout optimiser. Des fonctions peuvent l'être sans passer par la case processeur. Par exemple, la couche données et d'accès aux données s'optimisent souvent par un meilleur requêtage ou un middleware adapté. Ciblez le code critique consommateur de temps CPU et de ressources. C'est ce code précis qu'il faudra optimiser par rapport aux possibilités processeur.

Architecture logicielle adaptée

L'architecture de son application, et donc de son code, conditionne de facto une portion importante des possibilités d'optimisation. Les

processeurs actuels proposent une architecture très évoluée (NX, multi core, multiprocesseur, vectoriels, hyper threading, etc.). Or, pour pouvoir utiliser optimalement ces mécanismes, et donc, améliorer le fonctionnement de son code, il faut IMPERATIVEMENT réaliser une architecture logicielle parfaitement adaptée. Cela passe par un fonctionnement multithread de son application. C'est la base même de l'optimisation. Si l'application n'est pas multithreadée, il est impossible, ou quasi, de pouvoir exploiter les possibilités du processeur. Plus la conception de son code est propre, stable, plus il sera aisé d'utiliser au mieux le processeur. Si vous avez un code mal structuré, le compilateur aura du mal à optimiser de lui-même, ou alors mal, et vous aurez un long

travail de corrections. Il faut aussi utiliser les guides des fondeurs qui édictent les bonnes pratiques : comment utiliser les bibliothèques vectorielles, comment compiler, comment profiler son code, etc. Le processeur ne réalise aucun miracle, une application mal écrite, mal structurée, se drainera. Et vous passerez du temps et de l'argent à modifier le code existant.

Le compilateur

Le compilateur s'occupera de faire du pipelining logiciel, de l'analyse procédurale, de l'inlining, de l'optimisation de boucles, de la vectorisation. Vous pouvez aussi vérifier que le compilateur utilise la norme IEEE 754. Celle-ci définit la qualité du calcul flottant et de la portabilité des applications. Il est notamment présent dans les domaines du calcul scientifique. L'utilisation de IEEE 754 se nomme optimisation conservatrice. Vous pouvez utiliser un compilateur " généraliste " de type GCC ou opter pour du spécifique : Intel sur processeur Intel par exemple, IBM pour du PowerPC / Power, etc. Dans une optimisation compilateur, inutile de tout optimiser. Vous pouvez concentrer l'optimisation processeur sur des fichiers sources précis, sur une portion de code bien délimitée. Nous allons voir comment et pourquoi élaborer une tactique de compilation.

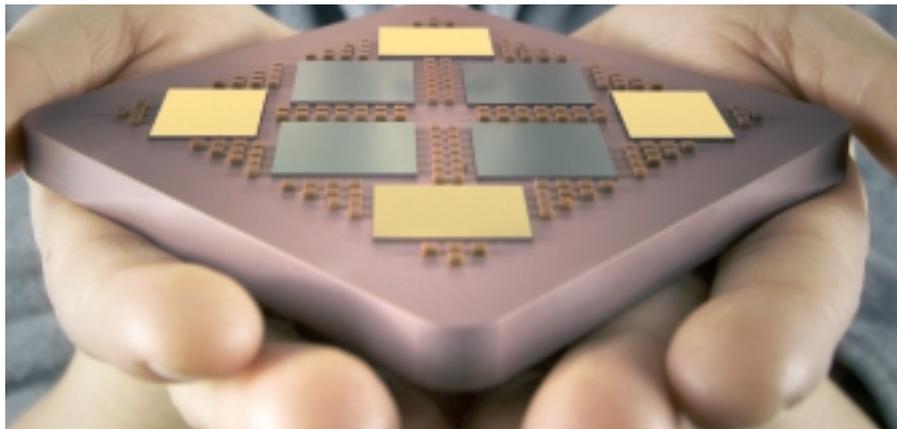
1 • La recompilation

La recompilation est une méthode simple et souvent rapide. Il s'agit de compiler son projet avec le nouveau compilateur. En général, cela permet de supporter de nouveaux processeurs et d'améliorer l'exécutible final. Le gain de performances n'est pas phénoménal, mais souvent appréciable. Cependant, faites des tests de compatibilité et de comportement.

2 • Les compilations optimisées et directives de compilation

Dans les compilateurs "modernes", notamment en C++, l'optimisation peut se réaliser en utilisant les mécanismes de Link-time code generation. Pour faire simple, cela permet d'éviter de compiler séparément chaque source, puis de réaliser les liens pour chaque fichier. Avec cette technique, on génère d'abord un langage intermédiaire, puis on passe à l'édition de liens.

La génération du binaire se réalise en une fois lorsque les liens sont créés. Bémol : cela nécessite plus de mémoire vive et le temps de compilation se rallonge. Vous avez aussi la



Dans l'embarqué...

Dans le monde de l'embarqué, on utilise aussi des processeurs. Les PowerPC sont souvent utilisés, tout comme Motorola, ARM ou même Intel. Dans ce type de programmation très pointu, et donc, très optimisé, on développe souvent en C, C++, et maintenant de plus en plus en Java. Il faut souvent des outils " spéciaux ". Soit on opte pour des solutions propriétaires et très ciblées, par exemple les outils IBM pour Power et PowerPC (kit d'évaluation, High C/C++ Compiler, debugger RISC...). Le plus souvent, on optera pour des solutions multi ciblage sur le langage et surtout le support processeur. Ainsi des éditeurs comme Green Hills Software, QNX, Apogee Software ou encore Wind River proposent des IDE et outils (debug, analyseurs, etc.) supportant C, C++, Fortran, parfois Java, et capables de générer des projets pour PowerPC, StrongArm, etc. Dans l'embarqué, de facto, le code sera optimisé dès le départ pour prendre en compte les contraintes de l'embarqué et des systèmes RTOS.

Et sous MacOS X ?

Le monde Apple et MacOS X n'oublie pas de proposer aux développeurs des solutions d'optimisation. Les processeurs G4 et G5 possèdent plusieurs unités vectorielles (connues sous le nom d'AltiVec). Le code vectoriel AltiVec fonctionne aussi bien sur G4 que G5 (tests à prévoir tout de même avec recompiler pour espérer tirer parti du G5). Là aussi, on rencontre les mêmes soucis que sur processeur Intel ou AMD. Il faut paralléliser le fonctionnement, le calcul, donc passer par du multithreading dans son code. Pour aider à utiliser AltiVec, on utilise un profiler pour identifier le code à vectoriser (par exemple Shark), utiliser des bibliothèques déjà optimisées (Accelerate.framework). MacOS X 10.4 intègre dans ses outils de développeurs tous les éléments nécessaires à une programmation optimisée (il inclut notamment GCC 4.0). Tiger (MacOS X 10.4), propose Core Image dédié à l'impression, aux images, à la vidéo, effets spéciaux. Il s'agit de cacher la complexité de programmation de ce type de fonctions. Au lieu d'attaquer les API de bas niveaux, le développeur utilise uniquement ce " middleware " qui s'occupe ensuite de l'accélération matérielle nécessaire. Cette accélération passe aussi bien par les processeurs GPU que par les unités AltiVec des G4 ou G5. Il met alors en place une exécution parallèle. Pour la partie vidéo, on dispose du Core Video qui fait le pont entre Quicktime et le bas niveau Quartz Extreme qui s'appuie sur l'accélération matérielle. Quel que soit le core utilisé, l'objectif est le même : réduire la charge du processeur central.

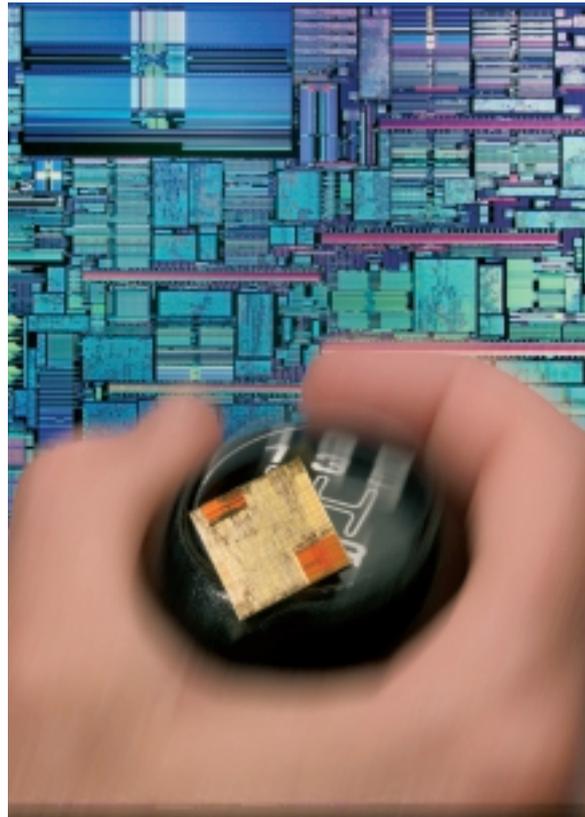
possibilité d'utiliser l'optimisation guidée par profil (ou PGO en anglais).

On génère des profils en lançant l'application sur des entrées réelles. Ces profils serviront ensuite au compilateur pour générer un code optimisé pour ledit exécutable. La première étape est d'instrumenter l'exécutable (on compile les fichiers avec l'option /GL puis on lie le tout avec un commutateur). L'instrumentation consiste à implanter dans le code des sondes qui vont récolter les informations. Puis on forme l'exécutable proprement dit selon les scénarios d'utilisateurs, et c'est à partir de cela que l'on crée les profils. On procède ensuite à une recompilation du projet avec l'optimisation PGO activée. Les directives de compilations apportent une optimisation appréciable, à condition de savoir réaliser l'équilibre entre génération générique d'une application stable et optimisée (et donc indépendante du processeur), d'une optimisation instable et dépendante. C'est toute la difficulté des directives (et options) de compilations. Ces directives s'utilisent de deux manières soit directement dans le code, soit en passant par la configuration du compilateur. Si vous avez des routines précises à optimiser, inutile de coder en assembleur soi-même. Vous pouvez mettre en œuvre les intrinsèques (ou intrinsics). Il s'agit de macros encapsulant de l'assembleur.

MMX et cie...

1 • C'est quoi ?

Les extensions de type MMX, SIMD, AltiVec... sont des fonctions supplémentaires aux processeurs. Ils sont souvent rassemblés dans des unités vectorielles. Elles permettent de



manipuler d'importantes quantités de données et favorisent les longs calculs, tout en déchargeant la CPU. Cela s'utilise dans les traitements audio, video, voix, compression / décompression, cryptologie, compression réseau, etc.

Que l'on soit en AltiVec, MMX ou 3DNow, lorsque l'on code en dur ces fonctions, on utilise le C/C++ (le plus souvent le C++). L'utilisation de telle fonction nécessite en plus une bonne maîtrise de l'architecture vectorielle et des algorithmes. Microsoft supporte les technologies d'Intel et de AMD (via les intrinsics dans le compilateur). Même remarque

pour les fonctions Streaming SIMD Extensions (ou SSE), tout se passe là encore en C/C++.

2 • Vectorisation - Autovectorisation

La vectorisation est une technique éprouvée, optimisant calculs et données. Il faut une bonne maîtrise du code et des techniques de vectorisation, donc de l'algorithmie. Si le compilateur moderne peut vectoriser de lui-même (= autovectorisation), cette technique a tout de même des limites et gêne le debugage. Aujourd'hui, l'autovectorisation se généralise dans les compilateurs. Intel l'avait déjà intégrée. C'est au tour de GCC (gamme 4.x) de la proposer. L'autovectorisation du compilateur est capable de détecter les parties du code utilisant les unités vectorielles des processeurs (MMX/SSE, AltiVec...).

Cela signifie que le compilateur détecte et vectorise le code en question afin d'optimiser le code et donc les appels. On remplace le code normal par des instructions vecteurs. Sous GCC 4.x, on

utilise le framework Tree-SSA, utilisant le SSA (Static Single Assignment). En SSA, une variable se voit assigner une valeur une seule fois. On touche là aux problèmes d'algorithmie. SSA récupère les informations nécessaires d'une arborescence (GIMPLE) venant de l'arborescence GENERIC issue de l'analyseur syntaxique. Le Generic constitue l'arbre d'organisation du programme que l'on veut compiler. SSA permet de faire une analyse complexe pour l'optimisation.

Dans un premier temps, il est raisonnable de passer par l'autovectorisation, même si elle a aussi ses défauts. Elle ne saura pas toujours

PowerPC, POWER

Fournisseurs de processeurs, IBM propose aussi une gamme d'outils pour PowerPC et POWER. On dispose de compilateurs, de kit d'évaluation et d'analyseur / debuggeur (ex. : RISCWatch). IBM et Absoft travaillent ensemble sur les compilateurs. On dispose ainsi des solutions XL Fortran ou XL C/C++. La version C/C++, fonctionnant aussi sous MacOS X, cible les processeurs PowerPC (970) et POWER (5). Il implémente la librairie OpenMP et génère des applications 32, 64, tout en fournissant des fonctions d'optimisation. Il supporte les gammes serveurs p, i et zSeries d'IBM. Il supporte aussi les unités vectorielles de type AltiVec. Les binaires sont compatibles GCC. Par défaut, le debug est assuré par gdb. Il intègre aussi maintenant le

debugger visuel d'Absoft, Fx2 Debugging Solution. Ce debug supporte les codes 32, 64, MacOS X, Linux, Windows, le multiprocesseur et le multithread. Cependant, pour les IDE et analyseurs, on passera plutôt par des outils tiers.

À noter qu'IBM implémente dans certains mainframes des puces essentiellement dédiées au code Java, le zAAP (zSeries Application Assist Processor). Il délivre un environnement d'exécution Java au z/OS. Le processeur central ne traite plus, ou quasiment plus, de code Java. Très utile pour les applications critiques et dans la montée en charge. Il dispose d'une JVM IBM. En principe, il n'y a pas à modifier son application pour considérer ce processeur lors de l'exécution de code Java.

trancher sur un code (dépendances, travail sur plusieurs pointeurs, etc.). Dans ce cas, elle émet des warnings (si le compilateur est bon) aidant le développeur à comprendre les problèmes de structure. Bien entendu, on peut toujours forcer le compilateur et regarder ensuite les dégâts.

Et la mobilité ?

Avec l'explosion des applications mobiles, les processeurs mobiles ne cessent d'évoluer. La programmation multithread est encore plus indispensable que sur desktop ou serveur. Si l'application est monothread, les tâches s'exécuteront à la suite, le temps CPU ne sera pas ou très peu saturé et la consommation d'énergie se fera sur une longue durée. Le multithread permet de mieux saturer le processeur, d'effectuer plusieurs tâches en parallèle. On aura peut-être des sauts de consommation, mais celle-ci sera plus courte dans le temps. À partir de votre code, vous pouvez accéder à différents paramètres (et fonctions) du processeur : état de la consommation, état des protocoles de communications et des ports... Intel propose un Mobile Platform SDK. Il comprend les bonnes pratiques, des exemples, des outils de tests et de monitoring. Il fonctionne sur Windows XP Pro et Windows Mobile 2003. L'interface se fait en C++. Cependant on dispose d'interfaces Java et .NET. On dispose de 8 classes permettant d'accéder aux terminaux et capacités systèmes. On dispose aussi d'un module pour gérer et visualiser les flux réseaux. Basiquement, via le SDK, on peut contrôler et récupérer les données sur :

- les interfaces réseaux du terminal
- les protocoles utilisés et actifs
- les informations sur les batteries

Java Advancing Imaging

Java n'échappe pas à l'utilisation d'extensions processeurs. Il est possible, via l'API Java Advancing Imaging, d'utiliser les fonctions MMX. Pour cela, on passe par la librairie de traitement mediaLib de Sun (support MMX valable en Win32). Pour les autres plates-formes ne supportant pas MMX, on passe par une version C de la mediaLib. Cette librairie s'utilise pour l'imagerie, l'audio et la vidéo, les opérations mathématiques et la 2D / 3D. Cette librairie évite de coder en C++ / assembleur, les fonctions SIMD.

- les informations sur le processeur et la présence de SSE2 ou Streaming SIMD 3.

On dispose aussi du Wireless MMX que l'on retrouve dans les processeurs mobiles Xscale. Ce sont des instructions MMX, mais sur mobiles pour l'énergie, l'image, le son, la vidéo, etc. Intel travaille actuellement à la version 2. Wireless MMX intègre le MMX des puces de bureaux en y ajoutant des fonctions spécifiques. Il utilise des SIMD 64. Intel a fait en sorte que les instructions Wireless MMX et MMX soient le plus possible équivalentes, limitant la réécriture des codes.

Les outils indispensables

Pour réaliser de l'optimisation vectorielle ou non, il faut passer par des outils et librairies. Les fondeurs et constructeurs proposent de telles solutions, payantes ou gratuites. Comme toujours, à vous de juger et de tester les outils les plus adaptés à vos besoins.

Intel

Intel propose des compilateurs, outils d'analyses et de performances, des librairies (y compris des SDK), des outils de threading et de clustering. Une gamme élargie qui prend en compte les dernières technologies processeurs.

AMD

AMD propose aussi une série d'outils pour prendre en compte ses processeurs, même si le choix paraît bien restreint comparé à Intel.

• Compilateurs

Pour les compilateurs, AMD utilise ceux d'éditeurs tiers. Sous Windows, le principal est celui de Visual C++ qui supporte les intrinsèques AMD et les instructions 64. Sous Linux, on pourra utiliser GCC 3.x. La v4 n'est encore officiellement listée par AMD.

• Debug, analyseurs, performances

Pour les bugs, on pourra utiliser GDB 6.x, les outils Windows Debugging Tools 32 ou 64. Pour Linux, Unix, Solaris, Fx2 (AbSoft) supporte le multi-langage, TotalView Debugger (Etnus). AMD est meilleur pour les librairies. On dispose de ACML v2 pour les fonctions mathématiques 64 (version 32 disponible). Une seconde librairie, Numerical Libraries (NAG) est dédiée au 64 et aux calculs complexes. En Open Source, on bénéficie en plus des librairies Atlas 3.5.x (algorithme linéaire), GNU glibc (optimisée AMD64). Pour l'analyse de code, AMD propose AMD CodeAnalyst. Il permet un profiling 32 et 64 de son code. Fonctionne sous Windows et Linux.

■ François Tonic

Compilateur

C++ Compiler Disponible pour Windows et Linux. Supporte Intel Extended Memory 64 et permet de compiler pour desktop, serveur et mobile. Existe des versions Windows CE.NET et Palm.

Outils de performances

Vtune Indispensable pour profiler son code. Il fonctionne sous Linux et Windows. Il facilite l'analyse de son code et les fonctions à optimiser. Comptable Visual Studio et Eclipse

Librairies

IPP Library Integrated Performance Primitives, multi-plate-forme. Optimise les fonctions audio/vidéo, la voix, l'image, la cryptographie...

Math Kernel Library fonctions mathématiques dédiées aux applications utilisant énormément les calculs mathématiques, telles que la finance ou le monde scientifique. La librairie utilise les dernières capacités des processeurs. Support Xeon et EM64. Version cluster disponible.

Mobile Platform SDK Pour prendre en compte dans son code les fonctions spécifiques des terminaux et processeurs mobiles. Interface C++, .NET et Java.

Outils de threading

Thread Checker Pour monitorer les threads et les données des threads. Fonctionne sous Windows et Linux

Thread Profiler Outil d'analyse fine des threads.

EDS : le géant de l'outsourcing accélère en France

Filiale de General Motors, EDS est l'une des principales SSII mondiales : 20,6 milliards de dollars de chiffre d'affaires en 2004 et 130 000 collaborateurs dans 60 pays. En France, la société emploie plus de 4 000 personnes et a réalisé en 2003, un chiffre d'affaires de 500 millions d'euros. Elle prévoit 450 recrutements en 2005.

Présente en France depuis 1985, EDS a procédé à de nombreuses acquisitions, dont celle d'Answare en 2002. Elle possède également la Française de maintenance, qui assure notamment la maintenance de la Française des jeux. EDS gère dans l'hexagone 160 000 postes de travail et terminaux et administre quelque 50 000 serveurs.

La société est organisée autour de quatre activités : conseil, via EDS consulting Services (3 700 personnes dans le monde, dont 130 en France), gestion des infrastructures, services applicatifs et Business Process Transformation (mise en œuvre de meilleures pratiques, services industrialisés).

En 2005, "EDS France veut se focaliser sur le marché des applications" explique Gérard Lary, directeur de l'entité EDS Answare. Une nouvelle organisation a été mise en place pour adresser ce marché.

La société procède à deux types de recrutement sur deux pôles d'activité, explique Gérard Lary : des profils de directeurs de projets, chefs de projets et développeurs pour son activité de développement et maintenance d'applications, et des profils d'experts pour son activité de gestion des infrastructures, son cœur de métier.

200 postes ouverts sur les nouvelles technologies

EDS France a recruté l'an dernier 300 nouveaux collaborateurs et en prévoit cette année entre 400 et 450, soit 200 à 250 créations de postes sur les métiers en fort développement : services applicatifs, la gestion des infrastructures



Gérard Lary

ou encore les services de support terrain. Les deux tiers des recrutements concernent les services applicatifs.

les profils recherchés concernent aussi bien des jeunes diplômés que des professionnels confirmés et touchent des fonctions très variées : consultants, au sein d'EDS consulting, Services architectes des systèmes d'information, chefs de projet ERP, ingénieurs en nouvelles technologies (J2EE et .net principalement). Le profil des consultants ?

Il s'agit en majorité de cadres (80 %) et pour moitié de jeunes diplômés. Sur la partie nouvelles technologies, sont recherchés des débutants ou des profils ayant 4 à 5 ans d'expérience pour intervenir sur des projets de développement applicatif et développement de logiciels spécifiques, à Paris comme en région. La moitié des recrutements concerne la région parisienne, mais les besoins en région sont importants : EDS dispose de onze agences en France et 60 sites d'assistance client.

Carole Hammou, responsable du recrutement

chez EDS France, explique qu'après une pré-qualification, les CV retenus sont transmis aux opérationnels, qui sélectionnent les candidats qui les intéressent. Suit un double entretien, avec les ressources humaines et avec un responsable opérationnel, voire un troisième, en fonction du profil. Gérard Lary souligne que les recrutements ne se font pas pour une mission particulière ou un poste particulier. Selon lui, "un bon profil va créer son poste".

Il indique que la majorité des recrutements se fait par cooptation

Créer un sentiment d'appartenance

La capacité d'évolution au sein d'une SSII aussi grande est bien sûr importante : La société souhaite se développer fortement en France et y propose une gamme de métiers assez variés : conseil, développement chez le client, intégration au forfait, gestion d'infrastructure au sein d'EDS ou chez le client. Les opportunités de mutation géographique existent, en France, ou des possibilités de missions à l'étranger, pour des profils plus confirmés, directeur de projets par exemple. Comme toute société américaine, EDS a développé une information corporate à destination de ses collaborateurs, mais veut également créer un esprit d'appartenance au travers d'initiatives plus inhabituelles, par exemple des réseaux trans-pays sur certains sujets. La communauté Open Source d'EDS regroupe ainsi des développeurs indiens, mexicains. " Nous avons beaucoup d'initiatives pour qu'il y ait des groupes au-delà des barrières linguistiques et géographiques ", souligne Gérard Lary.

Autres vecteurs de fidélisation des salariés : des plans de formations " conséquents ", un dialogue entre les dirigeants et les collaborateurs. " le taux de fidélisation est mesuré par le taux de cooptation ", remarque Gérard Lary, le premier des recrutements se fait par cooptation.

■ Carole Pitras

Programmez!

Nouveaux services :

- Offres d'emploi
- Codes sources

Des centaines de nouveautés, actualisées en permanence, avec nos partenaires

www.programmez.com

Sylis conjugue baisse des coûts et esprit citoyen

Après plusieurs années difficiles, la SSII d'origine lilloise a entamé un repositionnement, afin de se renforcer auprès des grands comptes et de se développer sur le marché des PME-PMI. Elle crée un premier centre de développement fonctionnant en 2x8. Une usine de développeurs ?

Présente en France, en Belgique et aux Pays Bas, Sylis compte 1750 collaborateurs, dont 1200 en France et a réalisé l'an dernier un chiffre d'affaires de 130,1 millions d'euros. Confrontée depuis plusieurs années à la crise du secteur et à une concurrence accrue sur des prestations en régie, elle a décidé de se repositionner autour d'offres génériques sur une cible de grands comptes (Widesourcing, tierce maintenance applicative, tierce recette applicative) et autour d'offres packagées, avec des solutions ERP, CRM ou décisionnel de Microsoft, sur une cible de PME-PMI. Elle travaille également à la mise en place d'une offre de conseil.

Actuellement, les prestations en régie génèrent encore la moitié du chiffre d'affaires de la société. L'objectif est de passer à 30% dans les deux années à venir, et de monter à 60% pour les projets au forfait, les 10% restant étant générés par la nouvelle activité conseil.

Le Widesourcing : nearshore+offshore

Pour disposer d'une offre à moindre coût, Sylis crée actuellement à Tourcoing une plate-forme pour assurer le développement applicatif sur différentes technologies : J2EE, .net, Unix/NT, mainframe, en plus de compétences sectorielles et fonctionnelles. L'objectif est de proposer aux clients des prestations de 15 à 30% moins chères que celles réalisées chez le client. Ce centre devrait compter 50 personnes fin 2004 et 100 à terme, essentiellement des Bac+2 encadrés par des Bac+5+ ; Une des particularités est la place réservée aux handicapés : 30% des postes. Selon Jacques Tricot il s'agit d'une main d'œuvre " très motivée mais



absente de la profession ". Les compétences technologiques de cette plate-forme seront les suivantes : J2EE, .net, Unix/NT, main-

frame. La société a également développé une approche offshore, en partenariat avec des sociétés roumaines, autour des nouvelles technologies, avec des sociétés marocaines, pour les clients voulant de l'offshore sans avoir à faire un cahier des charges en anglais, et des sociétés indiennes, pour les clients ayant déjà l'habitude de l'offshore. Au reproche éventuel de délocalisation, Sylis répond que pour six postes basés en offshore, elle crée un poste de haut niveau. Ce sont notamment des chefs de projets qui assurent l'interface entre les plates-formes Nearshore et offshore d'un côté, et le client de l'autre. Une équipe de 30 personnes, chargée de l'analyse des risques, des préconisations et du livrable des applications. Un des aspects sur lequel insiste Sylis est la qualité : déjà partiellement certifiée Iso 9001, elle compte étendre cette certification et démarre une démarche de certification CMM-I. Elle compte atteindre le niveau 2 en un an et progressivement le niveau 4.

Un processus RH optimisé

Sylis a mis en place à Bordeaux un centre de gestion des candidatures qui réceptionne tous les CV et les traite en 48h, afin de mettre à disposition des agences régionales des dossiers de collaborateurs " prêts à embaucher ". L'entretien d'embauche est toutefois réalisé dans l'agence qui recrute. Sylis prévoit de recruter une centaine de personnes au premier semestre, dont une quarantaine pour la nouvelle plate-forme, et 350 en 18 mois. Profils

recherchés : ingénieurs, confirmés et juniors, architectes, chefs de projets et directeurs de projets. Elle recrute également en Belgique et en Hollande, et cherche pour ces deux filiales des spécialistes des produits Microsoft.

La SSII dispose de son propre centre de formation, sur les aspects de méthodologie ou .net, ou a recours à des formations externes sur la partie conduite du changement, ou concernant des formations technologiques (Erp, Web, méthodes, décisionnel C/S, SGBD, Systèmes & réseaux).

Sylis revendique "une vision des rapports économiques, avant tout humaine et citoyenne" et indique miser "sur un développement personnel des facultés, afin que chacun de ses collaborateurs puisse s'épanouir dans son quotidien professionnel". Cet aspect humaniste s'illustre dans l'adhésion de Sylis au pacte mondial de l'ONU, visant à promouvoir le développement durable (environnement, droits de l'homme, droit du travail), dans sa volonté de promouvoir l'embauche de personnes à mobilité réduite, ou encore dans son partenariat avec l'Université Catholique de Lille, afin de "disposer " d'une meilleure connaissance des techniques de communication et de management ". Dans le cadre de sa restructuration, Sylis a instauré un " contrat entreprise/collaborateur " définissant les droits et devoirs des deux parties. Identifiés comme devoirs de l'entreprise : la gestion du plan de carrière, le plan de formation. En contrepartie elle affirme son droit à " exiger l'esprit de service ". Le " critère discriminant " est, selon Jacques Tricot, " l'envie de faire du service ". Il constate que durant les années de crise cette envie a parfois pu s'éteindre et un plan de reclassement des salariés démotivés a été engagé, en utilisant l'outplacement.

■ Carole Pitras

Manipulation de documents XML : au delà du balisage

La plupart des programmeurs savent aujourd'hui qu'un document XML est un texte balisé. C'est d'ailleurs la définition officielle qui est donnée dans la recommandation XML où le document XML est décrit comme un objet textuel (textual objet), dont certains caractères servent à représenter le balisage (markup) et d'autres constituent les données textuelles (character data) structurées à travers le balisage.

Il est cependant utile de rappeler/noter que la plupart des traitements importants appliqués aux documents XML (validation avec des schémas XML, transformation avec XSLT, etc.) n'opèrent pas sur cette représentation balisée.

Prenons le cas d'une transformation de documents avec XSLT. Pour de nombreux programmeurs, cette opération consiste à appliquer une feuille de style XSLT à un document XML balisé pour obtenir un autre document balisé, typiquement un document HTML pouvant être affiché par un navigateur (figure 1). Cette vision est superficielle. En réalité XSLT, dans ses deux versions (XSLT 1.0 et XSLT 2.0 est en cours d'élaboration par le W3C.), opère toujours au niveau d'un modèle de données. XPath 1.0 dans le cas de XSLT 1.0 opère sur le modèle XPath 1.0 alors que XSLT 2.0 opère sur le modèle dit "XPath/XQuery" (figure 2).

Les modèles de données XPath 1.0 et XPath/XQuery

Au delà de leurs différences (le modèle "XPath/XQuery" est beaucoup plus riche et formalisé que le modèle utilisé par XSLT 1.0 et repose sur la notion de "séquence"). Ces modèles de données considèrent un document XML comme une structure d'arbre pouvant être composée de sept types de nœuds :

- document node (root node dans le modèle de XPath 1.0)
- element node
- attribute node
- comment node
- processing instruction node
- text node
- namespace node

Les opérations décrites dans les règles XSLT opèrent sur une instance du modèle de données,

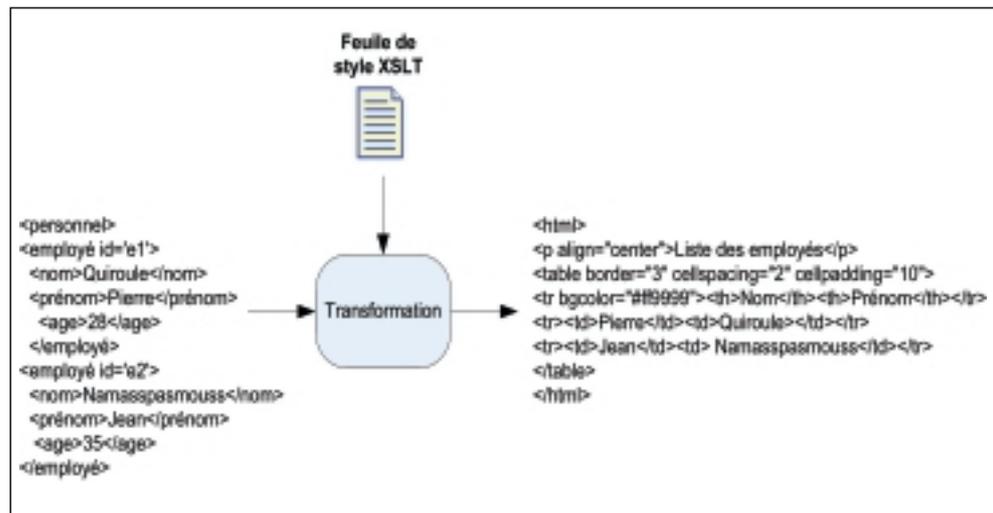


Figure 1. Vision naïve d'une transformation avec XSLT

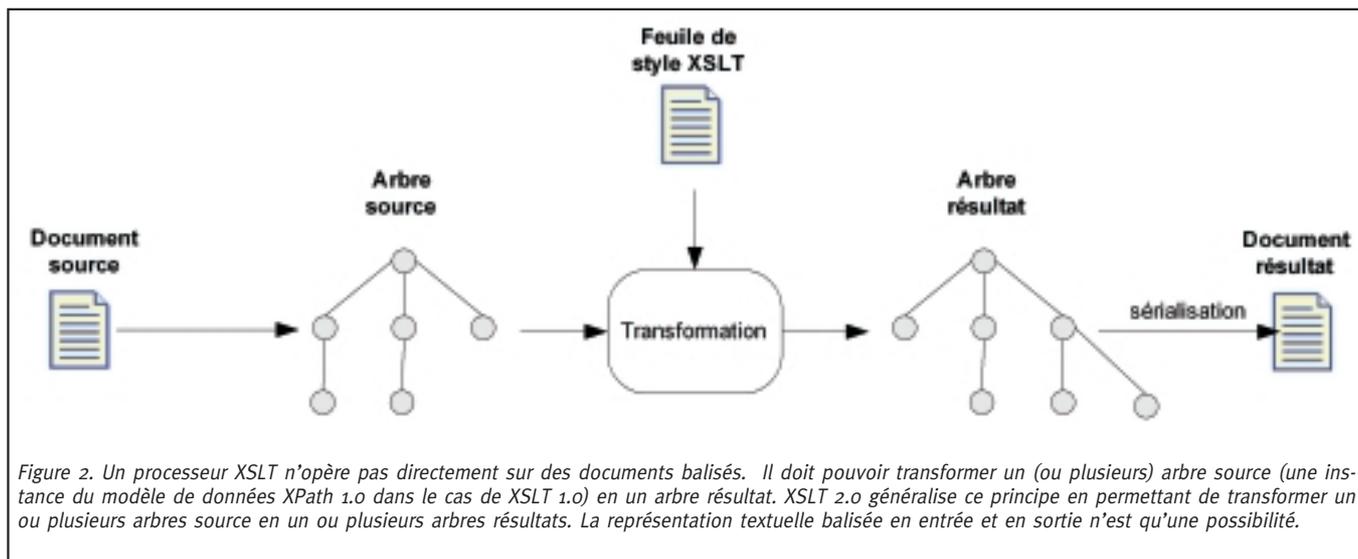
nées, l'arbre source (source tree), pour produire une autre instance du modèle de données, l'arbre résultat (result tree) (l'arbre dont il est question n'a rien à voir avec l'arbre DOM). L'arbre source peut avoir été obtenu à partir d'un texte balisé (mais pas forcément...) Quant à l'arbre résultat, il peut sous certaines conditions être "sérialisé", c'est-à-dire retranscrit sous forme d'un flux de caractères (extrémité droite de la figure 1). Il faut cependant avoir conscience que cette sérialisation n'est pas une obligation.

A chaque nœud est associé un ensemble de propriétés. Ainsi, le modèle XPath 1.0, comme le modèle XPath/XQuery considèrent que chaque nœud a une propriété nommée string-value. La valeur de cette propriété est déterminée de façon différente en fonction du type de nœud concerné, comme cela est schématisé dans le tableau suivant :

Tableau : 1

Type du nœud	String value
Document	La valeur obtenue en concaténant tous les nœuds de type text du document
Element	La valeur obtenue en concaténant tous les nœuds de type text qui descendent de l'élément
Attribute	La valeur " normalisée " de l'attribut au sens de la recommandation XML
Comment	Le contenu du commentaire (le contenu textuel situé à l'intérieur des délimiteurs ouvrant et fermant)
Text	Le texte du nœud
Pi	Tout ce qui est situé entre la cible et le délimiteur fermant
Namespace	L'URI de l'espace de noms ainsi que le préfixe associé

Certains des sept types de nœuds (element, attribute, namespace, processing-instruction) ont également une propriété nommée expanded-name dans XPath 1.0 qui correspond au nom de ces nœuds. Ce nom " développé " (expanded) est une paire consistant en un URI d'espace de noms (éventuellement vide) et



une partie locale. Dans le cas d'un élément ou d'un attribut, la partie locale est le nom du type de l'élément et l'URI est l'identifiant de l'espace de noms dont relève (éventuellement) cet élément ou cet attribut. Pour d'autres types de nœuds pouvant aussi avoir un nom étendu, ce dernier est obtenu de façon différente. Par exemple, pour un nœud namespace la composante URI est toujours vide tandis que sa partie locale correspond au préfixe auquel l'espace de noms a été (éventuellement) lié. Comme autre propriété importante d'un nœud, on peut enfin citer son nœud parent (propriété parent), la liste de ses nœuds fils (propriété children) ou la liste des nœuds espaces de noms associés à un nœud element. Là encore, toutes les propriétés ne sont pas définies pour tous les types de nœuds : par exemple, un nœud text n'a pas de propriété children. Les propriétés essentielles que nous venons de citer existent dans les deux versions du modèle, qu'il s'agisse du modèle XPath 1.0 ou du modèle " XPath/XQuery ". Etant lié aux schémas XML, ce dernier va plus loin et définit d'autres propriétés, dont par exemple la valeur typée (typed value) qui peut être associée à un nœud element, attribute ou text.

XSLT nous présente son modèle

Un programme XSLT très simple peut suffire à mettre en évidence les instances modèles de données que nous venons de décrire. XSLT va ainsi nous montrer lui-même la représentation sur laquelle il opère...
Considérons le document XML ci-dessous qui décrit un ensemble d'employés.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="datamodel
.xsl" ?>
<!-- fichier de démo -->
<emp:employés xmlns:emp="http://www.exemples
.fr/employes/">
<emp:employé idEmp="e1">
<emp:nom>Quiroule</emp:nom>
<emp:prénom>Pierre</emp:prénom>
</emp:employé>
<emp:employé idEmp="e2">
<emp:nom>Namasspasmouss</emp:nom>
<emp:prénom>Jean</emp:prénom>
</emp:employé>
</emp:employés>
<!-- ajouter d'autres employés ici -->
</emp:employés>
```

Vu en termes de modèle de données, ce document contient des nœuds de type element, attribute, namespace, comment, text, processing-instruction.

Nous allons écrire un programme XSLT permettant de donner un aperçu de l'instance sous-jacente du modèle de données :

- en représentant les nœuds de façon visible par leur identifiant unique ;
- en donnant pour chaque nœud, les principales propriétés applicables à ce nœud.

Pour rendre perceptibles les nœuds du modèle, nous allons utiliser la fonction XSLT generate-id(node-set ?) qui, comme son nom le suggère, est capable de générer une chaîne permettant d'identifier de façon unique un nœud de l'arbre. Chaque nœud sera ainsi représenté par son identifiant unique, ce qui est conforme

aux deux modèles de données XPath qui spécifient que tout nœud dans une instance de modèle est unique. Concrètement, pour chaque nœud de l'arbre on produira donc le fragment `identifiant : <xsl:value-of select="generate-id()"/>`

La structure générale du programme est très simple. Rappelons que le but n'est pas d'apprendre à écrire des programmes XSLT, mais d'exhiber la représentation sur laquelle opèrent ces programmes... On se contente donc d'itérer sur chacun des sept types de nœuds par une série de xsl:for-each. Par exemple, pour itérer sur les nœuds element on utilise `<xsl:for-each select="/*" *...</xsl:for-each>`. De façon similaire, pour itérer sur les nœuds namespace on utilise `<xsl:for-each select="//namespace::*" *...</xsl:for-each>`, etc.

La partie intéressante concerne en fait la mise en évidence des propriétés des nœuds. Prenons par exemple la propriété correspondant au nom d'un nœud. Nous avons dit que dans les modèles XPath le nom d'un nœud est donné sous forme d'une paire (URI de l'espace de noms, partie locale du nom).

Ces informations sont données, du moins pour les types de nœuds pouvant posséder un nom, par les fonctions XPath namespace-uri() et local-name(). Les autres propriétés importantes (liste des nœuds fils, nœud parent, nœuds espaces de noms associés à un élément, etc.) ne sont pas accessibles par des fonctions, mais peuvent être obtenues par des chemins XPath comme par exemple namespace::* ou child::node() pour accéder respectivement aux nœuds fils ou aux nœuds namespace::*

ce associés à un nœud element donné.

Examinons maintenant quelques informations produites par ce programme et vérifions si elles sont conformes à ce que nous avons dit des modèles de données dans la partie " Les modèles de données XPath 1.0 et XPath/XQuery ".

Prenons tout d'abord l'exemple du nœud correspondant à l'élément dont la balise ouvrante et fermante est étiquetée par emp:employés et qui constitue la racine de la structure logique :

- **identifiant** : IDAOYI4
- expanded-name : (<http://www.exemples.fr/employes/>, employés)
- string-value : Quiroule Pierre Namasspasmouss Jean Golotte Valérie
- parent : IDAIYI4
- children : IDAQYI4, IDAUYYI4, IDAYYI4, IDA2YI4
- namespaces : IDAGII4AOYI4, IDAPYI4AOYI4

Il a pour nom développé (expanded-name) la paire (<http://www.exemples.fr/employes/>, employés), ce qui est normal, puisque nous avons dit que dans les modèles XPath le nom d'un nœud est donné sous forme d'une paire (URI de l'espace de noms, partie locale du nom). Comme le montre la déclaration xmlns:emp="http://www.exemples.fr/employes/" située au début du document XML, un élément emp:employés est bien un élément du type employé appartenant à l'espace de noms <http://www.exemples.fr/employes/> qui a été lié au préfixe emp.

Sa valeur textuelle (string-value) est Quiroule Pierre Namasspasmouss Jean, ce qui est bien la valeur obtenue en concaténant tous les nœuds de type text qui descendent de l'élément, comme nous l'avons dit dans le tableau 1. Ce nœud a enfin un nœud parent (la racine du document XML, et deux nœuds fils. Comme c'est un nœud element, il a aussi des nœuds namespace associés d'identifiants IDAGII4AOYI4 et IDAPYI4AOYI4 (*Rappelons que ces identifiants sont des identifiants calculés à la volée par la fonction generate-id() et qui peuvent varier d'une exécution et d'une implémentation à l'autre. Leur unicité pour une instance donnée est cependant garantie*). Examinons justement ces nœuds. Le premier (identifiant IDAGII4AOYI4) est un nœud correspondant à l'espace de noms d'URI <http://www.w3.org/XML/1998/namespace> qui est automatiquement associé à tous les éléments XML et implicitement déclaré avec le

XSLT et modèle XPath

Code XSLT permettant de mettre en évidence les principales propriétés d'un nœud élément conformément au modèle XPath

```
<xsl:for-each select="//*">
<ul>
<li><b>identifiant : <xsl:value-of select="generate-id()"/></b></li>
<li>expanded-name : {<xsl:value-of select="namespace-uri(.)"/>, <xsl:value-of select="local-name(.)"/>}</li>
<li>string-value : <xsl:value-of select="string(.)"/></li>
<li>parent : <xsl:value-of select="generate-id(..)"/></li>

<li>children : <xsl:for-each select="child::node()"><xsl:value-of select="generate-id(.)" /><xsl:if test="position()
!=last()">, </xsl:if></xsl:for-each></li>

<li>namespaces : <xsl:for-each select="namespace::*"><xsl:value-of select="generate-id(.)" /><xsl:if test="position()
!=last()">, </xsl:if></xsl:for-each></li>
</ul>
</xsl:for-each>
```

préfixe xml. Si vous exécutez la feuille de style vous constaterez en effet que tous les nœuds element possèdent un nœud (distinct) namespace correspondant à cet espace de noms.

Le second (identifiant IDAPYI4AOYI4) correspond au préfixe qui a été déclaré explicitement par l'utilisateur à travers l'attribut xmlns:emp="http://www.exemples.fr/employes/" au début du document XML. Dans les deux cas, on constate que le nom développé est bien une paire, dont la composante URI est nulle et dont la partie locale est donnée par le préfixe déclaré dans le document balisé. La valeur textuelle est également conforme à la spécification donnée dans le [tableau 1](#).

- **identifiant** : IDAGII4AOYI4
- expanded-name : (, xml)
- string-value : <http://www.w3.org/XML/1998/namespace>
- parent : IDAOYI4
- **identifiant** : IDAPYI4AOYI4
- expanded-name : (, emp)
- string-value : <http://www.exemples.fr/employes/>
- parent : IDAOYI4

Comme autres exemples de valeurs conformes à ce qui a été énoncé dans la partie " Les modèles de données XPath 1.0 et XPath/XQuery ", on peut également considérer ce que produit l'exécution du programme au sujet des nœuds de type comment ou processing-instruction.

On vérifie ainsi qu'un nœud de type comment n'a pas de nom et que la valeur textuelle de ce

type de nœud correspond à la chaîne située à l'intérieur des délimiteurs ouvrant et fermant.

- **identifiant** : IDAHAo4
- string-value : fichier de démo
- parent : IDACAo4
- **identifiant** : IDAWAo4
- string-value : ajouter d'autres employés ici
- parent : IDAOYI4

Quant aux nœuds de type processing-instruction, on remarque que leur valeur textuelle est constituée par tout ce qui est situé entre la cible et le délimiteur fermant. Des constatations similaires pourraient être faites pour les types de nœuds (attribute, text) non abordés dans les exemples précédents.

- **identifiant** : IDAGAo4
- expanded-name : (, xml-styleheet)
- string-value : type="text/xml" href="datamodel.xml"
- parent : IDACAo4

Conclusion

Il resterait beaucoup de choses à dire sur le sujet. Nous n'avons pas examiné les rapports entre les modèles de données XPath et l'Infoset ou le PSVI. Pour aller plus loin sur ces questions, le lecteur intéressé pourra se reporter à notre livre*. Au total, on aura compris que, derrière les apparences, le document XML est un objet complexe à manipuler avec précaution, mais sans modération...

■ François Role

* *Modélisation et manipulation de documents XML*, Hermes Science, 2005

Introduction à AspectJ

Vous ne savez pas ce qu'est la programmation par aspect ? Cet article se propose de vous la faire découvrir par l'intermédiaire d'AspectJ, un sous-projet d'Eclipse (<http://eclipse.org/aspectj/>).

La programmation par aspect permet d'ajouter des comportements génériques à des classes métiers pré-existantes. Ces ajouts de comportement ne sont pas censés modifier le comportement métier de ces classes. La programmation par aspect est donc une sorte de "décoration de code" non intrusive. La programmation par aspect permet aussi de centraliser des traitements qui ne sont pas des comportements propres au métier mais plutôt liés à une forme d'organisation autour du métier ou à des problématiques plus techniques, comme l'ajout d'une politique de sécurité, par exemple, ou la gestion de la persistance. On peut bien entendu utiliser la programmation par "aspect" de façon plus large, mais c'est au risque de perdre complètement le principe d'encapsulation et de casser la logique objet.

Pour ajouter des comportements sans être intrusif, il ne faut pas modifier les classes existantes. Le mécanisme envisagé par l'"aspect" consiste à pouvoir déclencher des traitements systématiquement au moment de l'invocation de certaines méthodes. Ces méthodes sont quelconques, issues d'une classe lambda, elles sont uniquement répertoriées comme un élément déclenchant. Pour l'exécution des traitements, il faut aussi préciser à quel moment est effectué le traitement (avant ou après l'appel des méthodes désignées pour le déclenchement ?) et avec quels paramètres. Les paramètres ne peuvent bien évidemment être issus que de l'exécution et sont donc, soit des paramètres de la méthode, soit l'instance de la classe elle-même.

Quelques définitions

Un "aspect" est l'agrégation de notions qui permettent de définir quand et comment les traitements sont invoqués :

1. **"pointcuts"** : points de rendez-vous définis par l'interception du déclenchement d'une ou plusieurs méthodes.
2. **"advice"** : traitement associé à un point de rendez-vous, déclenché avant, après (avec ou sans erreur) l'invocation d'une méthode associée aux "pointcuts".
3. **"paramètres"** d'un **"advice"** : paramètres d'une méthode classique, mais provenant des objets et méthodes du code métier en cours d'exécution.
4. **"inter-types"** : permet l'ajout d'attributs et de méthodes à une classe sans changer son comportement métier. Ces inter-types se déclarent au sein de l'aspect qui va affecter la ou les classes concernées. La programmation orientée "aspect" permet la capture de conditions choisies par le développeur pour déclencher des traitements transverses que l'on ne peut pas capter avec un langage traditionnel tel que Java. En Java, il est difficile de poser des contraintes transverses (crosscutting) qui ne soient pas associées à une interface qu'implémente la classe. La programmation par "aspect" offre un mécanisme d'observation du comportement des classes pour déclencher des traitements, alors que l'orienté objet voit principalement les relations entre les traitements, via l'appel de méthodes et l'héritage.

Présentation de l'exemple illustrant ces notions

Le contexte fonctionnel présenté ci-dessous va nous permettre d'illustrer la mise en œuvre de la programmation par aspect. Il s'agit d'une bibliothèque de classes graphiques.

Les "Figures" sont constituées d'éléments unitaires : "FigureElements", qui sont des lignes ou des points. La classe "Screen" est en charge de l'affichage au travers de la méthode "display". Les exemples présentés dans la suite de l'article sont réalisés autour des éléments figurant sur le diagramme ci-contre :

Les Pointcuts (point de rendez-vous)

Un point très sensible de la programmation par "aspect" est la définition des points de rendez-vous, qui définissent sur quels critères doivent être déclenchés des traitements transverses (crosscutting).

AspectJ fournit plusieurs types de points de rendez-vous, mais nous ne traitons ici que de l'appel des méthodes.

On peut considérer que la syntaxe du point de rendez-vous est une sorte de requête de sélection comportant des "et", des "ou" et des "not". La syntaxe de définition des points de rendez-vous tient compte de la signature complète des méthodes, exceptions comprises.

Il n'est pas exclu que plusieurs points de rendez-vous soient associés à une même méthode.

Voici un exemple de déclaration des caractéristiques d'un point de rendez-vous avec "AspectJ".

```
call(void Point.setX(int))
```

Cette syntaxe permet d'indiquer que le "point de rendez-vous" correspond à l'interception de l'appel de la méthode dont la signature est : void Point.setX(int).

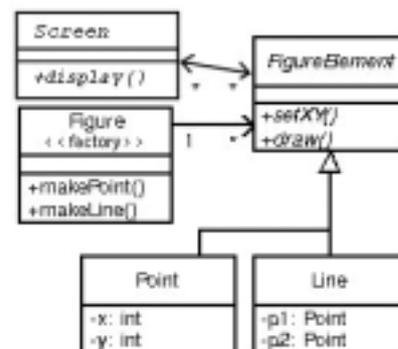
Un point de rendez-vous peut être défini à partir de plusieurs méthodes avec des "or" des "ou" ou des "not" (respectivement &&, || et !). Par exemple :

```
call(void Point.setX(int)) ||  
call(void Point.setY(int))
```

correspond à un point de rendez-vous pour les invocations des méthodes setX ou setY.

Un point de rendez-vous peut avoir un nom, pour définir le point de rendez-vous de nom "move" pour tout déplacement d'un élément graphique, on fait comme suit :

```
pointcut move();
```



```
call(void FigureElement.setXY(int,int)) ||
call(void Point.setX(int))           ||
call(void Point.setY(int))           ||
call(void Line.setP1(Point))         ||
call(void Line.setP2(Point));
```

...ainsi le mot clef "pointcut" permet de définir un point de rendez-vous et l'identifiant `move()` référence ce point de rendez-vous assez complexe qui rassemble toutes les possibilités de déplacement d'un élément graphique.

Les points de rendez-vous précédents étaient basés sur une énumération de signatures de méthode. Ceci s'appelle le name-based crosscutting. Néanmoins AspectJ propose d'autres mécanismes de définition de point de rendez-vous, sans forcément désigner les méthodes par leur nom exact. Il est possible de définir des "filtres" qui permettent de désigner un ensemble de méthodes, ce mécanisme est appelé le property-based crosscutting. La plus simple expression du property-based crosscutting consiste à indiquer une partie du nom d'une méthode.

Ci-dessous un exemple simple de point de rendez-vous de type property-based :

```
call(void Figure.make*(..))
```

Cette ligne de code permet de définir comme point de rendez-vous toute méthode de la classe "Figure" commençant par "make" et n'ayant pas de paramètres.

Dans notre exemple, cela correspond aux méthodes `makePoint` et `makeLine`.

Un autre exemple de ce type "rendez-vous" :

```
call(public * Figure.*(..))
```

Cela définit un point de rendez-vous pour toutes les méthodes publiques de la classe "Figure".

Les Advices (conseils)

Nous avons vu comment définir les points de rendez-vous, mais rien à ce stade ne permet de lancer un quelconque traitement. Pour implémenter un comportement associé à un point de rendez-vous, on définit un "advice". L'"advice" fournit le code à exécuter pour ce ou ces points de rendez-vous.

AspectJ définit différents types "d'advice" afin que l'exécution du code soit réalisée "avant, ou après" le point de rendez-vous.

Exemple de "before advice" :

```
before(): move() {
    System.out.println("about to move");
}
```

Le code s'exécute avant que le code des méthodes pointées par `move()` soit invoqué. Dans le cadre de cet exemple, on visualise au niveau de la console : "about to move", avant le déplacement effectif des objets graphiques.

De la même façon, il existe un "after advice" qui s'exécute après que la méthode référencée au niveau du point de rendez-vous a donné son résultat. Or, comme une exécution peut se terminer plus ou moins bien, AspectJ propose trois types d'advice : `after returning`, `after throwing`, et l'`after` tout simple.

Ainsi par exemple :

```
after() returning: move() {
    System.out.println("just successfully moved");
}
```

...envoi sur la console "just successfully moved" si les méthodes pointées par `move()` s'exécutent normalement.

Les paramètres

Un "advice" peut consommer des paramètres, tout comme une méthode. Dans l'exemple ci-dessous, l'"advice" est associé au point de rendez-vous référencé par `call` et ses paramètres sont définis en utilisant les mots clefs `target` et `args`.

```
after(FigureElement fe, int x, int y) returning:
    //...des Pointcuts...
    call(void FigureElement.setXY(int, int))
    && target(fe)
    && args(x, y)
{
    //...un corps principal...
    System.out.println(fe + " moved to (" + x + ", " + y + ")");
}
```

Ici, nous nous apercevons que trois variables sont exposées par le point de rendez-vous au niveau de l'appel de la méthode `setXY` :

- La `target` (la cible) est l'instance de "FigureElement" par laquelle il y a un appel à la méthode `setXY(int, int)`. L'instance courante est référencée par le mnémonique `fe` qui est déclaré comme un paramètre de l'advice.
- les `args` (arguments) qui ont pour nom `x` et `y` désignent les deux entiers passés en paramètres de la méthode `setXY` et identifiés comme paramètres de l'"aspect".

Les paramètres de l'advice sont donc définis via les mots-clefs `target` et `args`. Voici un autre exemple d'advice qui appelle un point de rendez-vous nommé :

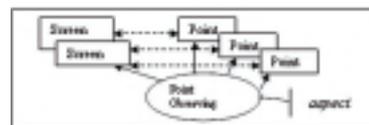
```
pointcut setXY(FigureElement fe, int x, int y):
    call(void FigureElement.setXY(int, int))
    && target(fe)
    && args(x, y);

after(FigureElement fe, int x, int y) returning: setXY(fe, x, y) {
    System.out.println(fe + " moved to (" + x + ", " + y + ")");
}
```

Déclaration des inter-types et présentation d'un aspect complet

Les inter-types permettent de définir des nouveaux attributs ou de nouvelles méthodes d'une classe. Ces attributs ou méthodes sont dans l'aspect, mais affectent la classe. Ils deviennent alors potentiellement réutilisables au travers des classes pointées par l'aspect.

Les inter-types sont susceptibles de porter des changements d'états, de provoquer des opérations statiques et de compilations. Nous remarquons que les inter-types sont susceptibles d'agir au travers de multiples classes par héritage et relations entre classes.



Voici une illustration d'inter-type. Supposons que nous voulions avoir un objet "Screen" qui observe le changement des objets "Point".

```
aspect PointObserving {
```

```
//inter-type observers = sorte d'attribut de la classe Point
//Concerne un ou plusieurs de ses points.
private Vector Point.observers = new Vector();
...
}
```

L'inter-type " observers " est privé (private), pour que seul l'aspect "PointObserving" puisse le voir.

On définit, comme suit, les méthodes (inter-types aussi) static addObserver et removeObserver pour ajouter ou retirer un écran qui "observe" un point et qui est alors ajouté à la liste " observers " de l'instance de " point " concernée :

```
aspect PointObserving {
    //inter type observers concernant la classe Point et pouvant affecter
    un ou ../. plusieurs de ses points.
    private Vector Point.observers = new Vector();

    //un point p pointe sur un écran s via observers.
    public static void addObserver(Point p, Screen s) {
        p.observers.add(s);
    }

    //on retire la référence du point p pointant sur l'écran s via observers
    public static void removeObserver(Point p, Screen s) {
        p.observers.remove(s);
    }
    ...
}
```

Continuons dans la définition de l' " aspect " par l'ajout d'un point de rendez-vous qui capture les appels de modification d'un point et de l'"advice" qui va provoquer l'affichage à l'issue de la modification d'un déplacement d'un point.

Voici le code illustrant l'aspect dans son ensemble :

```
aspect PointObserving {
    //inter-type observers concernant la classe Point et pouvant affecter un
    ou ../. plusieurs de ses points
    private Vector Point.observers = new Vector();

    //un point p pointe sur un écran via observers.
    public static void addObserver(Point p, Screen s) {
        p.observers.add(s);
    }

    //on retire la référence du point p pointant sur l'écran s via observers
    public static void removeObserver(Point p, Screen s) {
        p.observers.remove(s);
    }

    //Point de rendez-vous sur l'appel d'un assesseur " set " sur un
    //..point p
    pointcut changes(Point p): target(p) && call(void Point.set*(int));

    //Après chaque changement sur un point, réaffichage du point p pour
    //.. tous leurs écrans.
```

```
after(Point p): changes(p) {
    Iterator iter = p.observers.iterator();
    while ( iter.hasNext() ) {
        updateObserver(p, (Screen)iter.next());
    }
}
//affichage du point par l'écran
static void updateObserver(Point p, Screen s) {
    s.display(p); }
}
```

Nous remarquons alors que pour chaque changement sur un point, via un accesseur Point.set*(int), un réaffichage (display) systématique sur tous les écrans de la liste des " observers " est lancé.

Nous remarquons dans ce cas de figure que ni la classe des " Point ", ni celle des " Screen " n'a été changée.

Finalement, nous constatons qu'un aspect est constitué de pointcuts (point de rendez-vous), des advices et des inter-types dans une unité modulaire. Ces unités ressemblant à des sortes de classes permettent des traitements transverses appelés crosscutting.

Les aspects peuvent aussi avoir des méthodes, des attributs et des initialiseurs en plus des membres de crosscutting. Nous remarquons que puisque seuls les membres de crosscutting sont déclarés dans l'aspect, ces membres voient donc bien leurs effets localisés par l'aspect sur les autres classes objets concernées.

Conclusion

AspectJ peut au premier abord paraître complexe. Cependant, schématiquement, c'est un outil d'interception d'événements liés à l'exécution normale de classes Java. Ainsi, c'est un excellent outil pour mettre en place des mécanismes transverses (sécurité, persistance, ...), à condition de définir très correctement les " points de rendez-vous ". Une chose est certaine, il faut faire très attention à ne pas utiliser les aspects pour venir patcher du code (bien que la tentation soit grande), ce serait une catastrophe pour sa maintenabilité. Les aspects doivent rester le moyen de mettre en place des traitements transverses, mais cela suppose que la conception des classes d'origine soit propre et rigoureuse, mettant en avant des mécanismes systématiques et facilitant ainsi la possibilité de définir les bons points de rendez-vous. Ce type de programmation peut s'avérer tout à fait utile pour gérer une politique très fine d'accès à l'information quand celle-ci n'a pas été pensée au départ. Dans ce cadre, les méthodes pour accéder à l'information sont en général rapidement identifiées, en revanche la finesse avec laquelle doivent être faits les contrôles évolue souvent avec les changements d'organisation...

Les aspects sont complémentaires à la programmation orientée objet. Il reste à savoir si les avantages associés à leur utilisation compenseront les risques de les voir utilisés pour patcher sauvagement du code !

■ Nicolas Sciarra

Consultant, Homsys-Aston, filiale de Homsys Group

Homsys-Aston conseille, met en œuvre et propose des formations sur des projets orientés Business Intelligence, Développement Objet, Portails, Workflow et performance des Infrastructures. Le Groupe est implanté à Paris, Lyon, Marseille, Bordeaux, Toulouse, Grenoble et Rennes et fédère plus de 500 collaborateurs. (www.homsysgroup.com)

homsys aston
Une filiale de Homsys Group

Compressons nos données avec Whidbey

Le Framework 2.0 offre pas moins de 1500 nouvelles classes. Parmi celles-ci figure le nouvel espace de nom `System.IO.Compression` qui nous permet d'archiver nos fichiers au format Gzip.

1 Qu'est ce qui se cache derrière l'espace de nom `System.IO.Compression` ?

A l'heure actuelle, il est courant qu'un ordinateur neuf soit équipé d'un disque dur de 160 à 300 Go. La compression de données est cependant nécessaire lorsqu'il s'agit de transmettre des fichiers à travers un canal de transmission plus lent comme le réseau. Microsoft a implémenté les classes `GzipStream` et `DeflateStream` dans le but de permettre aux développeurs de comprimer physiquement des données sans perte. Il existe en effet deux types de compresseurs. Le premier type, dit "lossless", réduit le volume de données en entrée, sans perte d'informations lorsque les données sont ensuite décompressées pour restituer le fichier initial. Le deuxième type est le compresseur "lossy" qui réduira le volume d'informations tout en ne garantissant pas la restitution de toutes les informations par voie de décompression (jpeg par exemple est un algorithme de type "lossy").

Mais attention, si en théorie, le chiffre de valeurs, c'est à dire le rapport entre la longueur des données comprimées et la longueur des données initiales, doit être inférieur à 1, ce ne sera pas toujours le cas comme nous allons le voir...

Retenons qu'il existe de nombreuses méthodes de compression de données. Toutes ne sont pas d'efficacité égale car elles dépendent de la nature des informations composant les fichiers. Tel algorithme utilisé pour réduire la taille d'un fichier de données graphiques ne sera pas exploitable pour réduire la taille d'un fichier de données alphabétiques. Les classes `GzipStream` et `DeflateStream` recourent toutes les deux à l'algorithme `Deflate` comme il est décrit dans le RFC (Request for comments) n°1951. L'algorithme `deflate`, libre de brevets, est une combinaison des algorithmes `LZ77` et `Huffman`. Le format des fichiers est soit `Gzip` (`GzipStream`, extension `gz`), soit `zlib` (`DeflateStream`). Pour rappel, `Gzip` signifiant GNU zip, est un logiciel libre de compression. La seule différence est que `Gzip` est extensible et occupe un peu plus de place sur disque. Nous vous conseillons `Gzip` que vous pourrez relire facilement sous Windows ou Linux.

Nous utilisons Microsoft Visual C# 2005 édition Express Bêta 2. Nous commencerons par déclarer le nouvel espace de nom `System.IO.Compression`. Avec le Snippet Compiler dont nous parlerons brièvement plus loin, vous pouvez choisir le framework, mais non avec Visual C# 2005. Si le nombre d'arguments est inférieur à 4 (`args.Length < 4`), et (&) que nous avons demandé une décompression (`str_Compression_ou_decompression.ToLower().CompareTo("-c") == 0`), il y aura affichage de la syntaxe. Idem si le nombre d'arguments n'est pas égal à 3 pour une décompression. Nous utilisons une "astuce" pour déterminer quel algorithme de décompression sera appliqué (pas de quatrième argument dans ce cas). En effet, si une exception de type `InvalidDataException` est provoquée par la tentative de décompression avec un flux `DeflateStream`, nous appelons dans le catch `GzipStream`.

Système d'exploitation : Windows XP

Environnement : Visual Studio C# 2005 Bêta 2

Langage : C#

Besoin : Vous devez télécharger la bêta 2 et le compilateur de Snippets

2 Créons deux snippets pour tester la nouvelle méthode `ReadAll()`

Entre parenthèses, pour la lecture des données non compressées dans le tampon, nous aurions pu procéder autrement. En effet, le framework 2.0 utilise une autre méthode de lecture, beaucoup plus simple. Voici globalement comment il faut procéder avec le framework 1.x (le fichier `C:\news.txt` doit exister et contenir quelques lignes de textes) :

Snippet1.cs

```
using System;
using System.IO;

namespace Exemple_FrameWork1
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamReader sr = null;
            string line;
            string nomFichier;
            nomFichier = @"C:\news.txt";
            try
            {
                if (! File.Exists(nomFichier))
                {
                    Console.WriteLine("Le fichier {0} n'existe pas", nomFichier);
                    return;
                }

                // Ouverture du fichier et écriture du contenu du fichier sur la console
                sr = new StreamReader(nomFichier);
                Console.WriteLine("Début du fichier");
                line = sr.ReadLine();
                while (line != null)
                {
                    Console.WriteLine(line);
                    line = sr.ReadLine();
                }
                Console.WriteLine("Fin du fichier");
            }
        }
    }
}
```

```

}
finally
{
    // Fermeture streamreader
    if (sr != null) sr.Close();
}
}
}
}

```

Et voici ce que l'on peut coder maintenant avec le framework 2.0 :

```

Snippet2.cs
using System;
using System.IO;

namespace Exemple_FrameWork2
{
    class Program
    {
        static void Main(string[] args)
        {
            string strContenuFichier;
            strContenuFichier = File.ReadAll(@"C:\news.txt");
            Console.WriteLine(strContenuFichier);
            Console.ReadLine();
        }
    }
}

```

Une seule ligne de code est suffisante.... avec ReadAll.

3 Quelques explications sur le code du compresseur / décompresseur

Toutefois, nous avons utilisé GzipDeflateStream.Read ou Write qui sont tous les deux limités par leur troisième argument de valeur entière. Par conséquent, vous ne pouvez manipuler un fichier plus long que 2.147.483.647 octets, c'est à dire 2 Go. Pour modifier les arguments, vous irez sous VS 2005 C# Express dans les propriétés de Compression.cs pour y ajouter par exemple les arguments

```

-c "C:\DATAS\Compression\ipcop-1.4.2.iso" "C:\DATAS\Compression\ipcop-1.4.2.iso.gz" -gzip
puis
-d "C:\DATAS\Compression\ipcop-1.4.2.iso.gz" "C:\DATAS\Compression\ipcop-1.4.2.iso"

```

Avec le fichier ISO, tel que celui testé (l'image ISO de la distribution Linux pare-feu ipcop), le chiffre de valeurs est plus élevé que 1. Autrement dit, le fichier compressé occupe plus de place que le fichier non compressé...

```

>>>>Test de l'espace de nom System.IO.Compression du framework 2.0
Un instant... Lecture du fichier source ...
Compression...GzipStream
Taille d'origine : 41055,00 Ko

```

```

Taille finale : 61248,24 Ko
Chiffre de valeurs : 1,49

```

Par contre, si nous compressons le fichier source (Compression.cs) le chiffre de valeurs sera inférieur à 1 (il y a gain de compression) :

```

>>>>Test de l'espace de nom System.IO.Compression du framework 2.0
Un instant... Lecture du fichier source ...
Compression...GzipStream
Taille d'origine : 6,07 Ko
Taille finale : 1,48 Ko
Chiffre de valeurs : 0,24

```

Nous avons compressé le même fichier avec la commande gzip sous Linux et nous obtenons 1381 octets contre 1526 dans le cas de GzipStream et 1508 dans le cas de DeflateStream. Le format gz est lisible par 7-Zip sous Windows, sinon vous pouvez toujours le décompresser :

```

>>>>Test de l'espace de nom System.IO.Compression du framework 2.0
DECOMPRESSION...de type 'Gzip'. Archive 'unzipped' vers C:\DATAS\Compression\texte.txt

```

Voici un extrait du code source de Compression.cs (le code source complet est disponible sur le site de Programmez)

```

#region Using directives
using System;
using System.IO;
using System.IO.Compression;
#endregion

namespace Compression
{
    class Syntaxe
    {
        static public void Entête()
        {
            Console.WriteLine(">>>>Test de l'espace de nom System.IO.Compression du framework 2.0");
        }

        static public void Corps()
        {
            Syntaxe.Entête();
            Console.WriteLine("Syntaxe : Compress -c Chemin_Fichier_Source Chemin_Fichier_Destination -GZip");
            Console.WriteLine("Compress -c Chemin_Fichier_Source Chemin_Fichier_Destination -Deflate");
            Console.WriteLine("Compress -d Chemin_Fichier_Source Chemin_Fichier_Destination");
        }
    }

    class Compression_Framework_2
    {

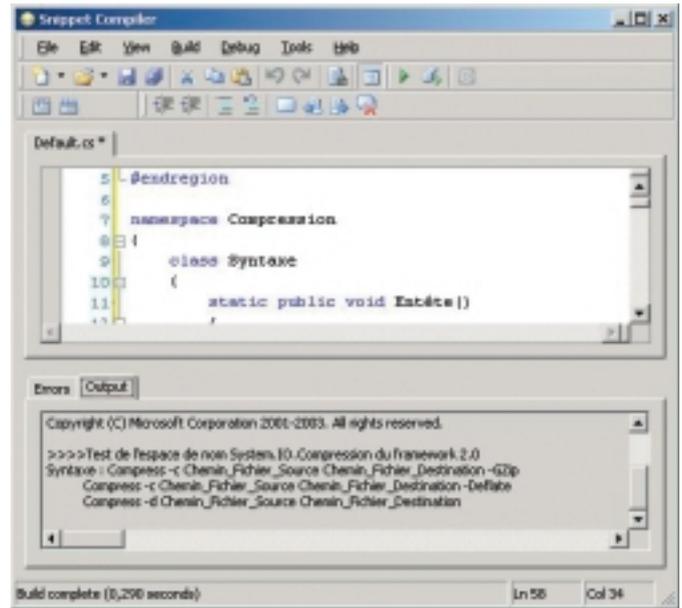
```

```
static void Main(string[] args)
{
    Stream fsSource = null;
    Stream fsDest = null;
    Stream GZipDeflateStream = null;
    ...
}
```

```
...
Console.WriteLine("DECOMPRESSION...");
arrTamponFichierSource = new byte[1024];
int increment = 1024;
fsSource.Position = 0;
int NbrOctetsLus = 0;
GZipDeflateStream = new DeflateStream(fsSource, CompressionMode.
Decompress);
try
{
    NbrOctetsLus = GZipDeflateStream.Read(arrTamponFichierSource,
0,increment);
    Console.WriteLine("de type 'Deflate'. ");
}
catch (InvalidDataException)
{
    fsSource.Position = 0;
    GZipDeflateStream = null;
    try
    {
        GZipDeflateStream = new GZipStream(fsSource,
CompressionMode.Decompress);
        NbrOctetsLus = GZipDeflateStream.Read(arrTamponFichierSource,
0, increment);
        Console.WriteLine("de type 'Gzip'. ");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception, erreur : {0}", ex.Message);
        return;
    }
}
fsDest.Write(arrTamponFichierSource, 0, NbrOctetsLus);
...
```

4 Pourquoi utiliser un compilateur de snippets ?

Un aspect intéressant avec VS 2005 est que vous pouvez créer un projet et le tester, sans que celui-ci soit sauvegardé. Ce n'est que lorsque vous appuyez sur l'icône de sauvegarde que le projet devient persistant sur disque ce qui évite de créer une multitude de noms de répertoires allant de `WindowsApplication1` à `WindowsApplication1xxx`. Jeff Key s'est penché sur le problème. Il possède comme beaucoup d'entre nous un grand nombre de bouts de codes de tests. Ceux-ci servent à tester un code en dehors d'un projet plus vaste, ou encore de tester une fonctionnalité du langage. La propre documentation de Microsoft fourmille de bouts de codes de tests. Comment les compiler sans devoir lancer à



chaque fois une instance de Visual Studio ? Avec Visual Studio 2003 il faut bien avouer qu'il s'agit d'une tâche bien lourde. VS C# 2005 Express représente un pas en avant, même si l'outil Snippet Compiler va encore plus loin dans le concept de compilation d'un bout de code. En effet, tout d'abord il est possible de choisir son environnement d'exécution (Framework 1.x ou 2.x), et vous pouvez compiler aussi bien du C# que du VB.NET ou de l'ASP.NET. Ensuite, vous pouvez importer un projet VS.NET, et construire des exécutables WinForm, console, ou DLLs. Les erreurs de compilation s'affichent et vous pouvez exporter le code en HTML ou RTF.

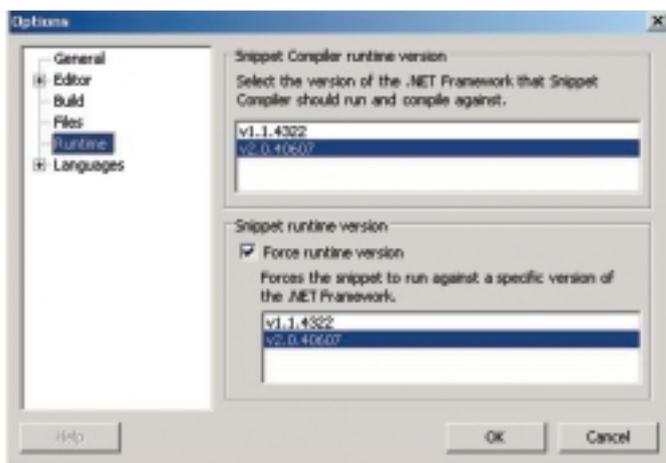
Concrètement, le Snippet Compiler est présent en permanence dans votre barre de tâches. Si vous avez besoin de compiler rapidement un bout de code vous l'avez toujours à votre disposition. Au repos, son occupation CPU (engendre 7 threads) est nulle et il n'accapare que 2 Mo de RAM, contre plus de 13 Mo pour VS C# 2005 Express (et 23 threads). Par exemple, vous voulez tester la généricité avec le framework 2.0, c'est à dire la possibilité pour un objet d'être utilisé tel quel dans différents contextes. Vous n'avez qu'à copier/coller le bout de code dans le Snippet Compiler, puis à cliquer sur le bouton "start". C'est tout.

```
using System;

public class Stack<S>
{
    public Stack (S s)
    {Console.WriteLine(s);}

    public void Push (S s)
    {Console.WriteLine(s);}
}

public class X
{
    static void Main ()
    {
        Stack<int> s1 = new Stack<int> (3);
    }
}
```



```

s1.Push (4);

Stack<string> s2 = new Stack<string> ("Hello");
s2.Push ("Test");
}
}

```

Dans la fenêtre OUTPUT s'affiche alors le texte suivant :

```

C:\Documents and Settings\XL> "C:\WINDOWS\Microsoft.NET\
Framework\v2.0.40607\csc.exe"
/t:exe /utf8output /R:"C:\WINDOWS\Microsoft.NET\Framework\
v2.0.40607\mscorlib.dll"
/R:"C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\System.dll"
/R:"C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\System.
Drawing.dll"
/R:"C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607
\System.Windows.Forms.dll"
/R:"C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\System.Xml.dll"
/R:"C:\WINDOWS\Microsoft.NET\Framework\v2.0.40607\System.Dat
a.dll" /out:"C:\Documents and Settings\XL\Local Settings\Temp\Snippet
CompilerTemp\aa19c803-7533-4f2c-848d-6d0b8d011584\output.exe"
/D:DEBUG /debug+ /optimize- /w:4 "C:\Documents and Settings\
XL\Local Settings\Temp\SnippetCompilerTemp\2fec50f3-fc6f-4e05-9c91-
c92528aa6cdd\0"

Microsoft (R) Visual C# .NET Compiler version 8.00.40607.42
for Microsoft (R) Windows (R) .NET Framework version 2.0.40607
Copyright (C) Microsoft Corporation 2001-2003. All rights reserved.

3
4
Hello
Test

```

Évidemment vous devez avoir sélectionné le runtime 2.0 (menu options). Une remarque : vous pouvez constater que le snippet compiler exécute le compilateur du framework en utilisant le paramètre utf8output. Ce switch n'est pas définissable sous VS. Il est pourtant indispensable lorsque vous désirez afficher des accents dans une fenêtre console alors que votre version de VS.NET est anglaise.

Vous pouvez maintenant facilement tester les snippets listés en début d'article (snippet1.cs et snippet2.cs), ainsi que le code source Compression.cs.

5 Et la compression sous Mono ?

En attendant que Mono puisse reconnaître l'espace de noms System.IO.Compression, vous pouvez intégrer l'assemblage GPL SharpZipLib à vos applications tournant sous Mono.

Voici un exemple de code utilisant l'assemblage 100% C# SharpZipLib :

```

using System;
using System.IO;

using ICSharpCode.SharpZipLib.Checksums;
using ICSharpCode.SharpZipLib.Zip;
using ICSharpCode.SharpZipLib.GZip;

class MainClass
{
    public static void Main(string[] args)
    {
        string[] filenames = Directory.GetFiles(args[0]);

        Crc32 crc = new Crc32();
        ZipOutputStream s = new ZipOutputStream(File.Create(args[1]));

        s.SetLevel(6); // de 0 (pas de compression) à 9 (compression maximale)

        foreach (string file in filenames) {
            FileStream fs = File.OpenRead(file);

            byte[] buffer = new byte[fs.Length];
            fs.Read(buffer, 0, buffer.Length);
            ZipEntry entry = new ZipEntry(file);

            entry.DateTime = DateTime.Now;
            entry.Size = fs.Length;
            fs.Close();

            crc.Reset();
            crc.Update(buffer);

            entry.Crc = crc.Value;

            s.PutNextEntry(entry);

            s.Write(buffer, 0, buffer.Length);

        }

        s.Finish();
        s.Close();
    }
}

```

■ Xavier Leclercq - Xavier.Leclercq@programmez.com

Des Web Services à café

2^{ème} partie

Après avoir travaillé avec les bases du maniement des Web Services en Java, nous abordons ce mois-ci des notions plus avancées : clients dynamiques et Web Services avec état.

Le mois dernier, nous avons appris comment mettre en place un Web Service en Java. Le serveur d'application était l'implémentation J2EE 1.4 de référence fournie par Sun. Nous continuons avec cet outil aujourd'hui. Nous rappelons que c'est un très bon outil pour se former à la technologie, mais qu'il n'est pas à utiliser en production. Nous avons écrit un client statique pour notre Web Service, et nous avons constaté que le code était lourd et le maniement également, car le service doit être parfaitement connu et même déployé pour que le client statique puisse être compilé et que des stubs soient générés lors de l'opération. Aujourd'hui, nous allons voir deux méthodes pour écrire des clients dynamiques, autrement dit, deux procédés pour nous affranchir de ces horribles stubs. Nous verrons également comment construire un Web Service avec état. Vous trouverez tous les codes d'exemples sur le site www.programmez.com

1 Un service enrichi

Avant d'écrire nos clients, nous allons enrichir notre service. Nous le dotons d'une méthode renvoyant un nombre aléatoire compris entre 1 et une limite (valeur incluse) reçue en argument. Cette nouvelle méthode vous sera très utile pour jouer au loto :-). L'interface de notre service devient :

```
package programmez.fred.ws;

import java.rmi.*;

public interface ProgrammezServiceIF extends Remote {
    public String helloWorld() throws RemoteException;
    public int random(int limite) throws RemoteException;
}

Son implémentation est triviale:

package programmez.fred.ws;

public class ProgrammezServiceImpl implements ProgrammezServiceIF {
    public String helloWorld() {
        return "\nProgrammez!\nEt Abonnez vous :-)\n";
    }
    public int random(int limite) {
        int n = new java.util.Random().nextInt(limite);
        return n+1;
    }
}
```

Par ailleurs, nous rappelons que nous devons travailler avec un CLASSPATH pointant sur les classes d'implémentation de J2EE 1.4. Pour vous aider dans ce fastidieux labeur, nous fournissons un script nommé setenv sous Linux et setenv.bat sous Windows. Nous avons déjà donné

Pré-requis à la lecture de cet article

- télécharger l'implémentation de référence de Sun de J2EE 1.4.
- avoir une JDK 1.4.2 ou supérieure installée sur votre machine.
- posséder des notions sur les Web Services.
- savoir déployer un Web Service dans le serveur d'application de l'implémentation de référence de Sun, ou sinon, avoir lu la première partie de cet article, parue dans Programmez ! 74

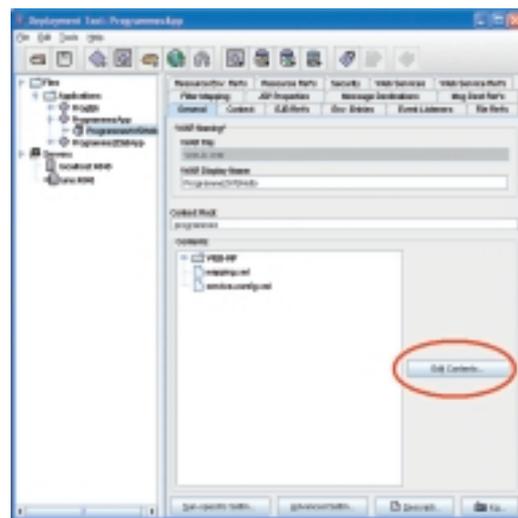


Fig. 1 : Avant de procéder au redéploiement, ne pas oublier d'intégrer les nouveaux fichiers .class à l'archive de l'application.

ce script le mois dernier. Depuis, nous avons remarqué que selon la version de l'implémentation de référence, 2 jars, dom.jar et xercesImpl.jar ne figurent parfois pas dans le sous répertoire endorsed, mais simplement dans le répertoire lib avec les autres jars. Le lecteur voudra donc bien adapter notre script en accord avec la version de l'implémentation qu'il a installée sur sa machine. Après compilation, le redéploiement est une formalité, à condition de ne pas oublier d'intégrer les nouveaux fichiers .class à l'archive .ear de l'application. Pour cela, dans l'outil de déploiement, cliquez sur le composant Web de l'application, puis sur le bouton 'Edit content' comme montré ci-contre, afin d'ajouter nos nouvelles classes qui vont écraser les anciennes. Même chose pour les descripteurs XML et pour le fichier WSDL qui doit être recompilé. (Fig.1) Ceci fait, déployer comme expliqué le mois dernier.

2 Premier cas de client dynamique lorsque le WSDL est connu

Comme expliqué au paragraphe 4.1 des spécifications des Web Services en Java, les clients dynamiques peuvent être de deux types selon que le document WSDL est totalement ou partiellement connu. Nous mettons d'abord en oeuvre le premier cas, dit stub/proxy. La présence du mot stub ici ne doit pas étonner, bien que nous ayons dit que nous nous affranchissions de ceux-ci. Dans ce cas, les stubs sont générés dynamiquement par l'implémentation (ouf ;). La méthode présente des

avantages par rapport à un client statique :

- il n'est pas nécessaire de déployer le service pour pouvoir compiler le client.
- il n'est pas nécessaire d'écrire un descripteur XML pour le client.
- il n'est pas nécessaire de connaître au préalable la localisation (URL) du service pour compiler le client. Cette URL peut fort bien être passée a posteriori, par exemple, sous forme d'arguments de ligne de commande.

Mais aussi des inconvénients :

- le code est un peu plus lourd que dans le cas des stubs statiques, quoique moins lourd que dans le cas que nous étudierons plus loin. L'absence de descripteur de déploiement XML compense largement l'alourdissement du code.
- Le fichier WSDL doit être connu, c'est à dire accessible au client par son URL.
- le .class de l'interface doit être accessible au client, c'est à dire pointé par le CLASSPATH.

Voici le code du client de notre service (fichier DemoDynClient1.java)

demodynclient1.java

```
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.rpc.Service;
import javax.xml.rpc.ServiceFactory;

public class DemoDynClient1 {
    public static void main(String[] args) {
        try {
            String nom_service = "ProgrammezService";
            String url = "http://localhost:8080/programmez/ws?WSDL";
            String namespaceURI = "urn:programmezService";
            String nom_port = "ProgrammezServiceIFPort";

            URL wsdlURL = new URL(url);
            ServiceFactory factory = ServiceFactory.newInstance();
            Service service = factory.createService(wsdlURL,
                new QName(namespaceURI,
                    nom_service));
            ProgrammezServiceIF proxy =
                (ProgrammezServiceIF)service.getPort(
                    new QName(namespaceURI, nom_port),
                    ProgrammezServiceIF.class);

            System.out.println(proxy.helloWorld());
            System.out.println();
            System.out.print(
                "Voici un des nombres du prochain tirage du loto: ");
            System.out.print(proxy.random(49) + " ");
            System.out.println();
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Ce code s'articule autour des classes QName, ServiceFactory et Service. QName encapsule des noms qualifiés. Ces noms sont ceux que nous avons donnés lors du déploiement et ils sont bien entendu intégrés au document WSDL. ServiceFactory, comme son nom l'indique, est une fabrique d'objet Service, cette dernière classe analysant le document WSDL et construisant le proxy à partir du WSDL et de l'interface java. Le proxy est enfin transtypé sur l'interface (narrowing) et il ne reste plus qu'à invoquer les méthodes du service.

3 Deuxième cas de client dynamique, lorsque le WSDL n'est pas connu.

Les spécifications nous informent que lorsque le client est agnostique (sic!), c'est à dire quand il n'a pas connaissance de la localisation du document WSDL et qu'il ne peut donc pas l'analyser comme dans l'exemple précédent, nous pouvons accéder aux méthodes d'un Web Service via la DII ou interface d'invocation dynamique. Le maniement de cette interface ne va pas sans rappeler le mécanisme de réflexion de Java. Cette méthode apporte les avantages suivants :

- les mêmes avantages que ceux apportés par le client stubs/proxy.
- le document WSDL n'a besoin que d'être partiellement connu. Le nom et le nom qualifié du service, le nom du type de port, l'URL du point terminal et les types des paramètres et des valeurs de retour sont les seules informations indispensables. La connaissance de la localisation du WSDL n'est pas nécessaire lors de l'exécution du client.
- La classe de l'interface du service n'est pas requise.

Mais il y a aussi des inconvénients :

- Le code est considérablement alourdi.
- Les types pour les méthodes des services sont définis à l'exécution et le compilateur ne peut pas les vérifier.

Pour être complets, nous pourrions également utiliser cette DII dans le cas précédent, car qui peut le plus peut le moins et lorsque le client connaît le WSDL, le proxy sait générer des instances de l'interface Call (voir méthode getCalls du proxy). Cette interface Call est au centre de tout client DII. Pour un premier exemple, nous écrivons notre client comme s'il ne connaissait pas le WSDL de notre service (fichier DemoDynclient2.java):

DemoDynClient2.java

```
import javax.xml.rpc.Call;
import javax.xml.rpc.ParameterMode;
import javax.xml.rpc.Service;
import javax.xml.rpc.ServiceFactory;
import javax.xml.namespace.QName;
import java.net.URL;

public class DemoDynClient2 {
    public static void main(String[] args) {
        Call call;
        try {
            ServiceFactory factory = ServiceFactory.newInstance();
            Service service =
                factory.createService(new QName("ProgrammezService"));
            call = (Call) service.createCall();
            call.setPortTypeName(new QName("ProgrammezServiceIF"));
            call.setTargetEndpointAddress("http://localhost:8080/programmez/ws");
        }
    }
}
```

```

call.setProperty(Call.SOAPACTION_USE_PROPERTY, new Boolean(
true));
call.setProperty(Call.SOAPACTION_URI_PROPERTY, "");

// Ci dessous:
// C'est normalement l'encodage par défaut
call.setProperty(Call.ENCODINGSTYLE_URI_PROPERTY,
    javax.xml.rpc.NamespaceConstants.NSURI_SOAP_ENCODING);

// Initialiser le tableau de paramètres
Object[] params = new Object[]{new Integer(49)};
// En cas d'utilisations consécutives de Call
call.removeAllParameters();
// Définir le type de retour
call.setReturnType(
    new QName("http://www.w3.org/2001/XMLSchema", "int"));
call.setOperationName(
    new QName("urn:programmezService", "random"));
// ajout d'un paramètre
call.addParameter(
    "int_1",
    new QName("http://www.w3.org/2001/XMLSchema", "int"),
    ParameterMode.IN);

Integer result = (Integer)call.invoke(params);
System.out.println(
"\nVoici un des nombres du prochain tirage du loto: "
+ result.toString());
    System.out.println();
}
catch (Exception ex) {
    ex.printStackTrace();
}
}
}

```

ServiceFactory et Service sont de nouveau présentes, mais nous aboutissons cette fois à une instance de Call, notre interface d'appel. Cette interface doit d'abord être configurée. Ensuite, nous créons un tableau d'objet dont les éléments seront les arguments de l'appel. Nous définissons encore le type de la valeur de retour. Enfin, nous ajoutons un à un les paramètres (arguments de l'appel); ceux-ci devant évidemment correspondre aux objets du tableau mentionné plus haut. Il ne reste plus qu'à invoquer la méthode.

4 Un exemple concret de DII

Le site www.x-methods.com expose de nombreux Web Services de démonstration. Ces services sont écrits en des langages à priori quelconques et justement la DII nous permet de ne pas nous préoccuper de cela. Nous allons nous amuser à écrire un client pour le service qui retourne la température d'une ville américaine identifiée par son code postal. Il existe également un service qui, à partir d'un nom de ville, retourne le code postal. Si vous voulez vous exercer ;) vous allez sur www.x-methods.com et vous verrez que l'URL du point terminal du service qui nous intéresse est :

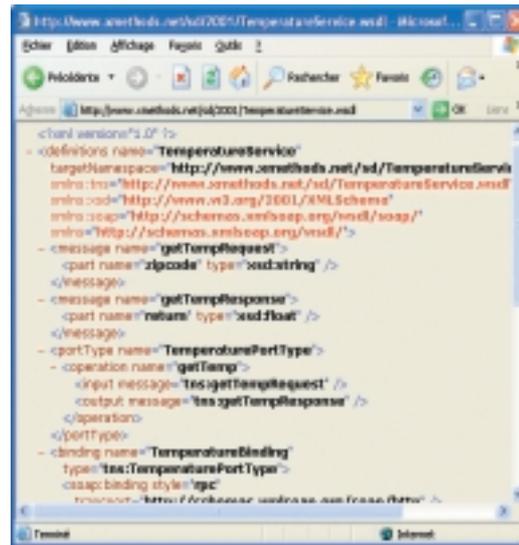


Fig. 2 :
Le document WSDL
du service
TemperatureService à
www.x-methods.com.

<http://services.xmethods.net:80/soap/servlet/rpcrouter>

Adresse peu parlante s'il en est, en raison de la présence d'un mécanisme de routage. Si vous continuez la visite de ce site vous verrez que le document WSDL du service réside à une adresse totalement différente :

<http://www.xmethods.net/sd/2001/TemperatureService.wsdl>

Saisissez cette adresse dans votre navigateur et examinez le document (Fig.2). Bien sûr, cette analyse pourrait être automatisée, par exemple avec DOM, mais c'est un autre sujet, que nous ne traiterons pas aujourd'hui. Il est facile de trouver le nom du service, son port et son point terminal. Ceci fait, il est aisé de modifier l'exemple précédent pour interroger le service. L'URL du WSDL n'est jamais donnée à ce client. Vous trouverez le code complet dans DemoDynClient3.java. Nous ne donnons ici que les lignes de code significatives.

DemoDynClient3.java

```

// Extrait de DemoDynClient.java

call.setPortTypeName(new QName("TemperaturePortType"));    call.set
TargetEndpointAddress(
"http://services.xmethods.net:80/soap/servlet/rpcrouter");

// Initialiser le tableau de paramètres
Object[] params = new Object[]{"20001"};
// Définir le type de retour
call.setReturnType(
new QName("http://www.w3.org/2001/XMLSchema", "float"));
call.setOperationName(
new QName("urn:xmethods-Temperature", "getTemp"));
// ajout d'un paramètre
call.addParameter(
"zipcode",
new QName("http://www.w3.org/2001/XMLSchema", "string"),
ParameterMode.IN);

Float result = (Float)call.invoke(params);
System.out.println("\nVoici la température à Washington DC: "
+ result.toString() + "°F");

```

Au moment de la rédaction de cet article, il faisait 53° Fahrenheit à Washington DC, dont le code postal est 20001 (Fig. 3). Remarquons bien que si cette méthode de travail nous dispense de se préoccuper de savoir si le service a été écrit en Java, il n'en reste pas moins que le CLASSPATH du client doit pointer sur l'implémentation, comme dit plus haut.

5 Un Web Service avec état.

Les spécifications nous informent qu'un Web Service peut implémenter l'interface optionnelle ServiceLifecycle qui gère le cycle de vie du service. Cette interface comporte deux méthodes, init et destroy, que le conteneur se chargera d'appeler à la création et à la destruction du service, respectivement. Les spécifications nous apprennent en outre "qu'une implémentation JAX-RPC de point terminal dans un container suit le cycle de vie standard des servlets". Ceci implique que l'objet

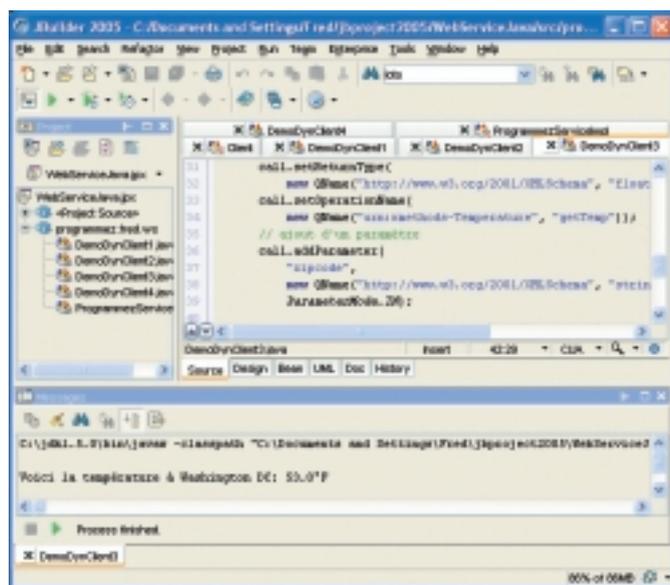


Fig. 3 : Il faisait 53° F à Washington DC au moment de la rédaction de cet article :-)

context reçu par la méthode init de ServiceLifeCycle est en fait de type ServletEndpointContext dans notre cas, et qui dit Servlet dit accès à une session HTTP dans lequel nous pouvons conserver un état. Pour l'exemple, nous avons défini un autre service. Son unique méthode incrémente un compteur à chaque invocation.

```
public interface ProgrammezServiceEtatIF extends Remote {
    public int compteur() throws RemoteException;
}
```

Voici maintenant l'implémentation du service:

ProgrammezServiceEtatImpl

```
package programmez.fred.wsetat;

import javax.servlet.http.HttpSession;
import javax.xml.rpc.ServiceException;
import javax.xml.rpc.server.ServiceLifecycle;
import javax.xml.rpc.server.ServletEndpointContext;

public class ProgrammezServiceEtatImpl
```

```
implements ProgrammezServiceEtatIF, ServiceLifecycle {

    private ServletEndpointContext sec;

    public void init(Object context) throws ServiceException {
        sec = (ServletEndpointContext)context;
    }

    public void destroy() {}

    public int compteur() {
        HttpSession session = sec.getHttpSession();
        Integer valeur = (Integer)session.getAttribute("compteur");
        if(valeur == null) {
            valeur = new Integer(0);
        }
        int result = valeur.intValue();
        result++;
        session.setAttribute("compteur", new Integer(result));
        return result;
    }
}
```

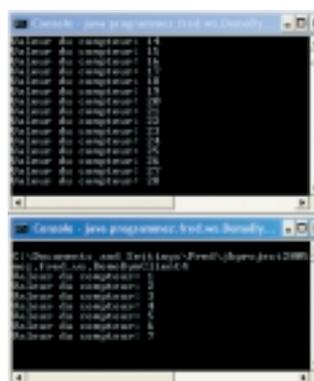


Fig. 4 : Exemple de Web Service avec état. Deux clients lancés en parallèle disposent chacun de leur session.

Le contexte de Servlet est obtenu à l'initialisation. Il servira à obtenir un objet HttpSession depuis la méthode compteur.

Celle-ci lit un attribut de session, l'incrémente puis le stocke dans la session et ainsi de suite. Deux clients lancés en parallèle (Fig.4) travaillent avec un valeur de compteur qui est conservée dans leurs sessions respectives. Le client est du code DII comme vu plus haut, avec seulement cette ligne supplémentaire :

```
call.setProperty(call.SESSION_MAINTAIN_PROPERTY, new Boolean(true));
```

qui indique que le client veut participer à une session avec le service. Nous en avons terminé avec les Web Services en Java et leur implémentation JAX-RPC. Avant de nous quitter, signalons qu'il est encore possible de déployer un EJB Session sans état (et seulement ce type d'EJB) sous la forme d'un point terminal de Web Service. Si le bean dispose en outre d'une interface Home, on pourra l'utiliser aussi de façon classique.

■ Frédéric Mazué - fmazue@programmez.com

Ressources

- L'implémentation de référence et les spécifications J2EE 1.4 <http://java.sun.com/j2ee/1.4>
- Les spécifications JAX-RPC1 <http://java.sun.com/xml/jaxrpc>
- Les spécifications Web Services for J2EE 1.1 Requirements <http://jcp.org/en/jsr/detail?id=109>

Découvrir la librairie graphique wxWidgets

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Professionnel ou amateur passionné, si vous avez besoin d'écrire des interfaces graphiques portables en C++, ou si plus simplement vous avez besoin d'un toolkit riche et facile à manipuler, wxWidgets est fait pour vous. Découvrons le ensemble.

C++ n'est pas mort, loin s'en faut. C++ est et restera certainement encore très longtemps un langage irremplaçable pour nombre de développements. Contrairement à Java, C++ n'offre pas un toolkit équivalent à Swing ou même à AWT pour écrire des interfaces utilisateurs graphiques portables. C++ n'offre même rien du tout. Au fil du temps, de nombreux toolkits ont été créés pour satisfaire ce besoin. Aujourd'hui, deux librairies tirent leur épingle du jeu : Qt et wxWidgets, chacune possédant d'immenses qualités. Votre serviteur à une préférence pour wxWindows, mais il est vrai que lorsque Qt 4 et sa licence GPL sous Windows auront vu le jour, le choix sera cornélien. En attendant, découvrir wxWidgets n'est assurément pas du temps perdu.

1 Présentation de wxWidgets

wxWidgets est un projet qui fait ses preuves depuis longtemps, car il a été lancé il y a maintenant 12 ans sous le nom de wxWindows. Le projet a été récemment rebaptisé wxWidgets en raison du (ridicule) courroux de Microsoft au sujet de l'emploi du mot 'windows' dans wxWindows. Mais quand on connaît wxWidgets, on est tenté de se demander si Microsoft n'a pas eu simplement du mal à digérer la supériorité de wxWidgets sur son piteux toolkit MFC. Il est vrai que, vu de loin, le code wxWidgets a une certaine ressemblance avec le code MFC, notamment en ce qui concerne les tables de routages de messages, définies par un jeu de macros. Mais la ressemblance n'est vraiment qu'apparente, car avec wxWidgets on dispose d'un cadre très riche, allant, à l'instar de Qt, très au delà de la simple boîte à outils graphiques. Utiliser les classes se fait en direct. Pas d'invocation à une méthode Create, suite à une instanciation de contrôle d'objets ou de gymnastique laborieuse pour mettre en place une barre d'outils. Surtout, wxWidgets est remarquablement portable. Les plates-formes supportées sont Windows, Unix, Linux et Mac. D'autres plates-formes devraient être bientôt supportées. Pour les systèmes de type Unix, les ports sont construits autour de Gtk+, X11 et Motif, ce qui doit faire le bonheur de tout le monde. En outre, et ce n'est pas négligeable, wxWidgets est en C++ 100% standard, sans extension comme Qt. J'utilise wxWidgets depuis de nombreuses années sous Linux et Windows, parfois en compilation croisée et je peux dire n'avoir jamais rencontré la moindre difficulté avec la portabilité du code. Avec wxWidgets vous pouvez écrire du code GPL ou des applications commerciales sur n'importe quelle plate-forme. Ne manquez pas d'aller visiter <http://www.wxwidgets.org>

2 Installer wxWidgets

Nous traiterons le cas de Linux et de Windows. Nous installons la dernière version stable, soit la 2.4.2. Commençons par Linux. Même

si'il existe des binaires sous la forme de paquetages RPM, le mieux est d'installer et compiler la librairie à partir des sources. Ceci permet de profiter au mieux des possibilités de configuration offertes par l'outil GNU configure, et cela permet facilement d'appliquer d'éventuels patches. Avant de procéder il sera sans doute judicieux d'enlever les paquetages RPM plus ou moins périmés qu'aura installés votre distribution. Il s'agit probablement du port Gtk+, vous pouvez donc connaître le nom exact des paquetages en procédant ainsi :

```
rpm -qa | grep wxGTK
```

Cette commande listera probablement trois paquetages que vous retirez de votre système Linux en faisant fi des dépendances. Par exemple :

```
rpm -e --nodeps wxGTK-2.4.1
```

et de même pour les autres paquetages.

Puis après avoir téléchargé et décompressé l'archive de source dans un répertoire faites :

```
cd repertoire
./configure
make
su <et donner le mot de passe de root>
make install
ldconfig
exit
```

Sous Linux, wxWidget va se construire autour de Gtk+ par défaut. Mais vous pouvez vous en assurer en procédant ainsi :

```
./configure --with-gtk
```

et surtout essayez un ./configure --help pour connaître et profiter de toutes les finesses possibles si vous vous sentez à l'aise.

Compiler une application sous Linux est tout ce qu'il y a de simple, car wxWidgets vient avec un outil de configuration qui est la réplique de gtk-config et qui sert à analyser votre environnement et à passer les informations collectées à votre compilateur. Ainsi, pour une petite application d'un seul fichier monapp.cpp. La compilation et l'édition de liens se font d'un seul coup comme ceci :

```
g++ monapp.cpp `wx-config --cxxflags --libs` -o monapp
```

Remarquez bien que les apostrophes sont inversées. Si celles-ci vous posent un problème, vous pouvez encore faire :

```
g++ monapp.cpp $(wx-config --cxxflags --libs) -o monapp
```

Etudions maintenant le cas de Windows et plus particulièrement le cas

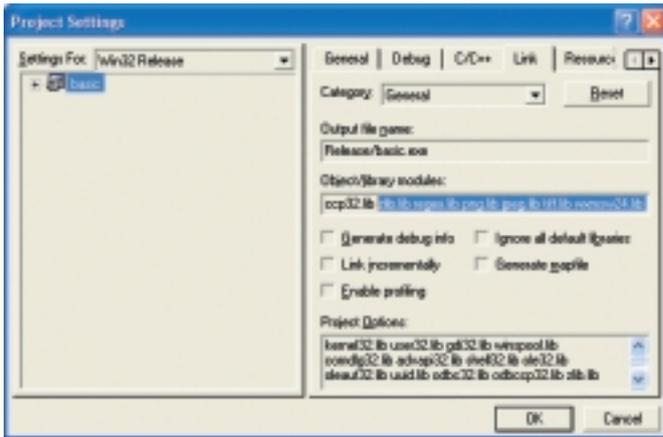


Fig 1 : Dans un projet wxWidgets ajoutez quelques bibliothèques à l'intention de l'éditeur de liens.

de Visual C++ 6.0. Toutefois, ce compilateur n'est pas le seul à pouvoir être utilisé, bien au contraire. wxWidgets vient avec un jeu de makefiles que vous trouverez à la racine de son arborescence. Il existe des makefiles pour Borland C++, Mingw32, etc. Tout ce que vous avez à faire avant de lancer l'utilitaire make de votre compilateur préféré est de définir une variable d'environnement pointant sur le répertoire de wxWidgets :

```
set WXWIN=c:\wxWindows-2.4.2
```

Venons en à Visual C++ (J'ai également compilé la bibliothèque avec Visual Studio .Net sans difficulté). A la racine de wxWidgets se trouve un projet qui fabriquera de multiples cibles. Bibliothèques statiques, dynamiques (dll) en mode release ou débogage. Vous pouvez construire tout cela d'un seul coup, en sélectionnant 'BuildBatch Build'. Toutefois, il est bon de procéder à deux opérations préliminaires. Par défaut, l'utilisation d'en-têtes précompilés est désactivée afin d'économiser de l'espace disque. Si vous ne voulez pas que la compilation de toutes ces cibles s'éternise, même avec une machine très puissante, il est judicieux de rétablir la pré-compilation. Pour cela, allez dans la configuration de chaque projet, puis dans l'onglet C/C++, sélectionnez "Precompiled Headers" dans la liste déroulante, puis, en dessous cochez "Use pre-compiled header file (.pch)" et juste en dessous donnez "Through header: wx/wxprec.h".

La seconde opération consiste à configurer la bibliothèque. Sous Linux cette opération est faite via le classique configure, mais il n'existe rien de tel sous Visual C++. Vous devez donc ouvrir le fichier %WXWIN%\include\msw\setup.h et le modifier à la main. Par exemple, vous pouvez souhaiter pouvoir bénéficier de l'intégration d'OpenGL dans wxWidget. Cette intégration est désactivée par défaut. Dans le fichier setup.h vous trouverez :

```
// Recommended setting: 1 if you intend to use OpenGL, 0 otherwise
#define wxUSE_GLCANVAS 0
```

Donnez :

```
#define wxUSE_GLCANVAS 1
```

Ensuite, pour compiler une application wxWidgets/OpenGL vous n'aurez qu'à lier les bibliothèques systèmes glu32.lib et opengl32.lib. Parcourez tout ce fichier setup.h afin de configurer votre bibliothèque aux petits oignons.

3 Compiler une application wxWidgets sous Visual C++

Après génération de la bibliothèque, vous trouverez une réplique de setup.h sous \lib\msw\wx, \lib\mswd\wx, etc. Le compilateur devra pouvoir trouver ces en-têtes. Toutes mes cibles étant configurées de la même façon, j'applique la méthode du paresseux et copie simplement le fichier setup.h original sous include\wx. faites de même :-)

L'éditeur de liens doit être renseigné sur les bibliothèques à lier à l'application. Vous devrez donc ajouter (Fig 1):

```
zlib.lib regex.lib png.lib jpeg.lib tiff.lib wxmsw24.lib
```

sous l'onglet "Link". Spécifier wxmsw24.lib implique que l'application sera liée à la dll wxmsw24.dll qui devra être trouvée à l'exécution, c'est-à-dire, soit déposée sous c:\windows\system32, soit pointée par le PATH. Enfin, il faut donner quelques directives de précompilation sous l'onglet C/C++ (Fig 2) :

```
WIN32,NDEBUG,_WINDOWS,WXUSINGDLL=1
```

NDEBUG en version Release, DEBUG sinon et fort logiquement.

4 Analyse du code d'une application wxWidgets minimale.

Voici (Encadré basic.cpp) le code d'une application de type 'Hello World'. Vous trouverez ce code sur le site www.programmez.com. Ce code est le point de départ de toutes applications wxWidgets. L'on commence par dériver la classe wxApp qui encapsule l'application. La méthode redéfinie OnInit de la classe est le lieu pour initialiser l'application, notamment en y instanciant la fenêtre principale. Si la méthode retourne TRUE, l'application démarre en lançant sa pompe à événements et s'arrête dans le cas contraire.

Nous remarquons qu'il n'y a pas de fonction main dans notre code. Celle-ci est écrite par le préprocesseur lorsqu'il étend la macro IMPLEMENT_APP. La classe de la fenêtre est très simple. On y remarque une autre macro qui déclare la table de routage des événements. La classe est implémentée également avec macros. Notons que BEGIN_EVENT_TABLE doit toujours recevoir une classe et sa classe de base en para-

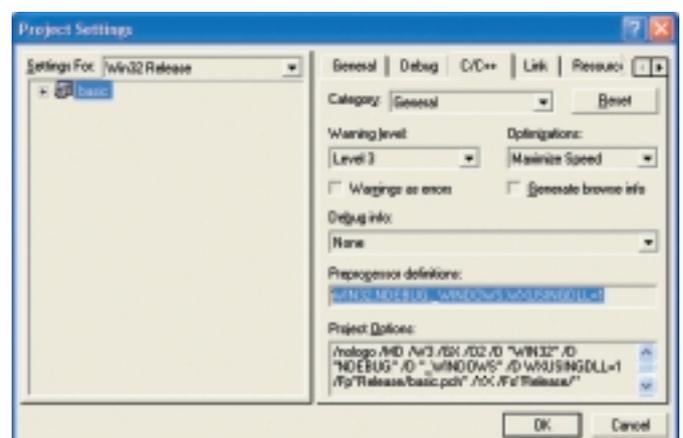


Fig 2 : Dans un projet wxWidgets ajoutez quelques directives à l'intention du préprocesseur.

mètres. Regardons le code de OnInit et le constructeur de notre fenêtre. Nous y voyons que tous les contrôles, de la fenêtre au simple bouton, sont instanciés sur le tas. C'est la règle avec wxWidgets (comme avec Qt d'ailleurs). Nous ne voyons aucun appel à delete dans le code. En effet, quand une fenêtre (principale ou non) est détruite elle se charge de libérer tous ses contrôles enfants. Enfin, quand un contrôle est instancié par son constructeur il est du même coup réalisé, c'est à dire que sa fenêtre associée est créée au niveau du système d'exploitation. Pas d'appel à une méthode supplémentaire Create, comme cela se voit avec d'autres toolkits plus besogneux :) Il est impossible de présenter wxWidgets dans sa totalité ici. Mais avant que nous ne nous quittions, je vous suggère de visiter la documentation des classes wxSizer et dérivées. Ces classes sont des gestionnaires de mise en forme à l'instar des layouts de Java ou des QLayout de Qt. Ces classes vous permettront d'obtenir un comportement identique de vos fenêtres quel que soit le système d'exploitation sous-jacent. Bon C++!

Ressources

Il existe des outils de conception graphique pour les fenêtres wxWidgets.

En voici deux gratuits :

- wxHatch : <http://biolpc22.york.ac.uk/wx/wxhatch/help/>
- WxwxGlade : <http://wxglade.sourceforge.net/>

En voici deux commerciaux, à des prix raisonnables et écrits par les auteurs de wxWidgets eux-mêmes :

- wxDesigner : <http://www.roebling.de/>
- DialogBlocks : <http://www.anthemion.co.uk/dialogblocks/>

basic.cpp

```
#include "wx/wx.h"

class MyApp: public wxApp
{
    virtual bool OnInit();
};

IMPLEMENT_APP(MyApp)

enum
{
    ID_QUIT = 1, ID_ABOUT, ID_BUTTON,
};

class MyFrame: public wxFrame
{
public:
    MyFrame(const wxString& title,
            const wxPoint& pos, const wxSize& size);
    void OnQuit(wxCommandEvent& event);
    void OnAbout(wxCommandEvent& event);
    void OnButton(wxCommandEvent& event);
private:
    DECLARE_EVENT_TABLE()
};
```

```
BEGIN_EVENT_TABLE(MyFrame, wxFrame)
EVT_MENU(ID_QUIT, MyFrame::OnQuit)
EVT_MENU(ID_ABOUT, MyFrame::OnAbout)
EVT_BUTTON(ID_BUTTON, MyFrame::OnButton)
END_EVENT_TABLE()

bool MyApp::OnInit()
{
    MyFrame *frame = new MyFrame("Hello World",
wxPoint(50,50), wxSize(450,340));
    frame->Show(TRUE);
    SetTopWindow(frame);
    return TRUE;
}
```

```
MyFrame::MyFrame(const wxString& title,
const wxPoint& pos, const wxSize& size)
: wxFrame((wxFrame *)NULL, -1, title, pos, size)
{
    wxMenu *menuFile = new wxMenu;
    menuFile->Append(ID_ABOUT, "&A propos...");
    menuFile->AppendSeparator();
    menuFile->Append(ID_QUIT, "E&xit");
    wxMenuBar *menuBar = new wxMenuBar;
    menuBar->Append(menuFile, "&Fichier");
    SetMenuBar(menuBar);
    CreateStatusBar();
    SetStatusText("Programmez! avec wxWindows!");
    wxPanel *panel = new wxPanel(this);
    wxButton *button =
        new wxButton(panel,
ID_BUTTON, "Programmez!", wxPoint(150, 120));

    void MyFrame::OnQuit(wxCommandEvent& WXUNUSED(event))
    {
        Close( TRUE );
    }

    void MyFrame::OnAbout(wxCommandEvent& WXUNUSED(event))
    {
        wxMessageBox("Ceci est une application wxWindows rudimentaire",
"Programmez!", wxOK | wxICON_INFORMATION);
    }

    void MyFrame::OnButton(wxCommandEvent& event)
    {
        wxMessageBox("Abonnez vous! :-)", "Programmez!",
wxOK | wxICON_INFORMATION, this);
    }
}
```

■ Frédéric Mazué - fmazue@programmez.com

Programmez un messenger en Flash

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Macromedia est un éditeur spécialisé notamment dans le développement Internet. Son logiciel phare se nomme Flash. Il est devenu une référence pour la conception d'animation online animée et interactive. Mais peu de développeurs connaissent également le développement offline avec Flash. Et pourtant les possibilités de création offline avec ce logiciel existent. Savez-vous qu'il est par exemple possible de réaliser un programme de messagerie instantanée en Flash ?

Développement offline avec Flash ?

Ce terme n'est pas tout à fait explicite ni tout à fait pertinent. Par développement offline, nous désignons en réalité le développement d'une application non spécifiquement dédiée à Internet, mais à un système d'exploitation. Habituellement, pour ne pas dire dans 99% des cas, les animations Flash sont exploitables via un navigateur Internet (Internet Explorer, Firefox ou Safari). Les animations Flash générées, sont des fichiers au format SWF (pour Shockwave Flash). Ces fichiers SWF sont par la suite directement intégrés dans du code HTML pour une diffusion sur le Web.

Mais il est tout à fait possible de développer en Flash un programme exécutable non encapsulé dans un navigateur. La solution la plus simple, mais aussi la plus basique en terme de fonctionnalité, se nomme la "projection Flash" : vous développez une animation via le logiciel de Macromedia et vous l'exportez au format exécutable (.exe pour Windows) au lieu d'exporter un fichier SWF dédié à Internet. Cette possibilité est intéressante, mais n'étant pas un outil de développement dédié, les exécutables générés avec Flash sont très limités en terme de fonctionnalités et ne peuvent pas décemment rivaliser avec un environnement de développement comme Director (spécialiste de la création de cd-rom) Visual Studio, Delphi ou Eclipse.

La solution SWF2EXE

Pour combler ces lacunes, des éditeurs tiers ont développés des outils complémentaires à Flash. Ces outils utilisent le concept de conversion SWF2EXE, qui comme son nom l'indique, permet de transformer une animation SWF en programme exécutable d'extension EXE. Ces outils étendent ainsi le champ d'application de Flash dans des domaines inconnus pour ce der-

nier, comme des fonctionnalités dédiées, par exemple à la gestion du système d'exploitation, ou bien encore, à la gestion des dossiers et fichiers sous Windows. Mais pourquoi utiliser Flash et un outil SWF2EXE pour faire le travail qu'un environnement dédié pourrait faire ? La question est pertinente mais la réponse fuse, elle tient en 3 mots : simplicité, rapidité et souplesse.

Simplicité, car le développement avec Flash est probablement l'un des plus simples en terme de RAD : la prise en main de ce logiciel peut sembler parfois un peu déroutante au premier abord mais une fois les fondamentaux acquis, il est très simple de concevoir une animation Flash.

La rapidité est une autre caractéristique du développement avec Flash, son système de time line intégré permet de concevoir une application intuitivement, sans même de véritables notions de programmation avancée. D'ailleurs, Flash était avant tout un outil dédié aux designers web et aux infographistes souhaitant concevoir des animations vectorielles pour Internet, pas du tout comparable à un environnement de développement et de programmation. Certes, Flash a évolué depuis la version 5 et ses caractéristiques dédiées à la programmation fonctionnelle se sont largement améliorées, pouvant même aujourd'hui s'étendre à la programmation orientée objet,



Fig.1 : développer un navigateur Internet avec... Flash ?

via un langage de script normalisé ECMA et qui porte le nom évocateur de langage ActionScript. Mais la communauté des utilisateurs professionnels de Flash reste encore largement composée d'animateurs et d'infographistes, n'ayant pas d'arrière plan technique dans le domaine de la programmation pure et dure. Il est donc indispensable de proposer aux utilisateurs de Flash un outil simple et intuitif, optimisant l'aspect créatif.

La souplesse du développement avec Flash découle des caractéristiques intrinsèques et historiques de cet environnement. Etant un outil d'animation vectoriel à la base, il est possible de presque tout faire au niveau créatif et design. En effet, alors que la plupart des environnements dédiés, comme Visual Studio ou Delphi n'offrent pas aisément de grande liberté dans la conception d'interfaces originales, Flash permet de presque tout faire au niveau design, avec des fonctionnalités simples d'animations vectorielles que nous ne trouvons nulle part ailleurs.



Figure n°2 : programme messenger, paramétrage initial

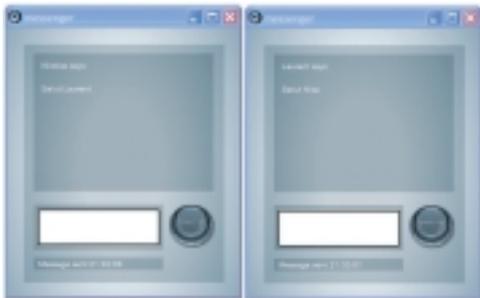


Figure 3 : programme messenger en action

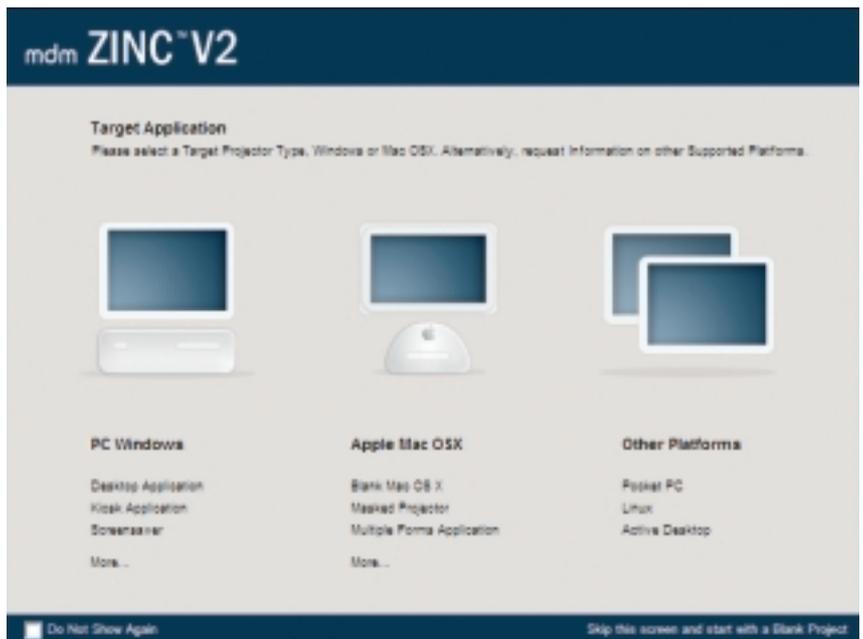


Figure 4 : sélection de la cible

Introduction à la technologie Zinc

Il existe quelques solutions technologiques pour transformer un fichier Flash en programme exécutable, tout en étendant ses fonctionnalités. L'une de ces solutions portait le nom évocateur de Flash Studio Pro, l'un des leaders dans le domaine des logiciels SWF2EXE. Dernièrement, Flash Studio Pro a été renommé et porte le nom moins évocateur de Zinc, version 2 en cours. Zinc est réputée pour être l'une des solutions les plus rapides et souples pour le développement RAD en complément de Macromedia Flash. En effet, Zinc transforme les animations Flash en des applications "desktop" pour Windows par exemple, en ajoutant plus de 600 nouvelles fonctions à vos projets Flash. Zinc peut ainsi complètement personnaliser chaque aspect de votre application, comme les dimensions, le style, les bordures et les icônes. Vous pouvez télécharger une version d'évaluation du logiciel Zinc V2 sur le site suivant : <http://www.multimedia.com/software/zinc/>. Le logiciel Macromedia Flash est également disponible en version d'évaluation ici : <http://www.macromedia.com/fr/software/flash/>

Socket Communication avec Zinc et Flash

En guise d'exemple, nous vous présentons un petit programme nommé " messenger " qui permet d'échanger des messages entre deux interlocuteurs connectés sur Internet. Zinc

fournit en effet une technologie capable d'envoyer et de recevoir des informations à partir de n'importe quel ordinateur utilisant le protocole UDP. Ce moyen de communication est basé sur le transfert d'informations via un réseau TCP/IP, ce qui signifie qu'un paquet contenant le message et l'adresse IP de destination, ainsi que le numéro du port, est construit à chaque fois que la communication est requise et les données envoyées sur le réseau.

Sans aller plus loin dans les détails techniques, sachez qu'un programme exécutable utilisant la gestion des sockets réseaux avec Zinc doit respecter la structure suivante :

- 1 – Information de destination : adresse IP
- 2 – Numéro du port de l'ordinateur client
- 3 – le message à envoyer

Le code source du programme se trouve dans le fichier messenger fla accompagnant cet article :

Frame 1 :

```
mdminit();
mdm.getipaddress("localip");
sendport = "4000";
listenport = "4001";
stop();
```

Frame 2 :

```
mdm.getLocaltime("time");
stop();
```

Bouton 1 :

```
on (release) {
    mdm.socketlistenon(listenport);
    gotoAndPlay("chat");
}
```

Bouton 2 :

```
on (release, keyPress "<Enter>") {
    if (messageout == "") {
    } else {
        messageout2 = name add " says:" +
        newline + messageout;
        mdm.socketsend(remoteip,sendport,
        "messagein@eq@"+messageout2);
        messageout = "";
        mdm.getLocaltime("time");
        status = "Message sent " + time;
    }
}
```

Pour tester ce programme sur le même ordinateur, vous pouvez exécuter les programmes messenger1.exe et messenger2.exe sur le même ordinateur et entrer la même adresse IP comme sur la figure n°2 (en inversant égale-

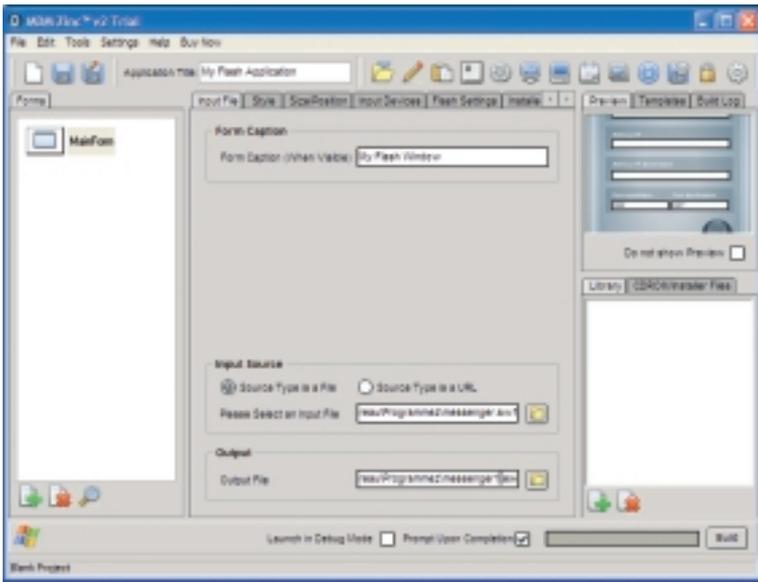


Figure 5 : Interface du logiciel Zinc

vous pouvez choisir une application cible (PC Windows, Mac OS X, Pocket PC, Linux Fig. 4) L'interface du logiciel est simple, mais efficace et volontairement dépouillée. Deux champs de saisie importants retiennent notre attention. La sélection du fichier flash SWF en entrée et le nom du programme exécutable en sortie (Figure 5). Un clic sur le bouton " build " lance la génération de l'exécutable. Plusieurs paramètres de votre application sont possibles directement dans Zinc. Les possibilités de personnalisation sont nombreuses. N'hésitez pas à visiter l'interface de ce logiciel qui reste très intuitive.

Allez plus loin avec Flash et Zinc

Les fonctionnalités de la technologie Zinc ne se limitent pas à la simple manipulation de socket réseaux. En effet, avec Zinc vous pouvez étendre vos applications Flash à la gestion de contenu vidéo, à la gestion d'une base de données complète, notamment via des requêtes SQL, ou bien encore à la création d'applications connectées à la couche DirectX de Windows. Bref, Flash et Zinc peuvent vous réserver encore bien des surprises...

■ Laurent Jayr



ment les ports 4000/4001 sur messenger1.exe et 4001/4000 sur messenger2.exe, Fig : 2. Dans l'exemple de la figure n°2, l'adresse IP correspond à l'adresse d'un réseau local. Bien entendu, si vous souhaitez tester le programme via Internet et deux ordinateurs distants, il vous faudra entrer cette fois ci les véritables adresses IP des deux ordinateurs connectés à

Internet. Idéalement, le programme Flash devrait déterminer ces adresses IP automatiquement avant chaque début de session. (Fig. 3)

Interface du logiciel Zinc

Au lancement du programme Zinc (ouvrez en guise d'exemple le fichier messenger.zinc),

Abonnement "éco" 1 an
soit l'équivalent de **3 Numéros GRATUITS**



économisez 30%

Tarif normal : 65 €
prix de vente au numéro

Offre Eco **45€** seulement !
(Tarif France Métropolitaine)

Abonnement 1 an + Clé USB NOUVELLE VERSION
128 Mo

Ce stick mémoire haut de gamme est multi-standard (Windows, Linux, Mac OS) à transfert ultra rapide et plug & play.



économisez 50%

Nouvelle version : inclut en supplément le logiciel commercial Travel It Easy, (pour XP), suite d'utilitaires pour utilisation nomade (sécurité, sauvegarde au format Outlook Express etc...)

Tarif normal : 99 €

Offre Anniversaire **49,50€** seulement !
(Offre limitée à la France Métropolitaine)