



RSS,  
REST,  
RESTFUL,  
Twitter

## Gardez le fil sur Internet

L'avenir de Java est-il assuré ?  
**Toutes les nouveautés  
de Java 7**

### Webmaster

Comment migrer  
vers IE 9 ?



### Smartphone

■ Google Maps  
sur iPhone



■ Gérer la persistance  
des données  
sur Android

### Carrière

L'emploi est  
timidement reparti !



**USB**  
Maîtriser les ports  
USB sous Windows

**Office**  
Les nouveaux outils  
VSTO 2010

**Windows Azure**  
Mettre en oeuvre AppFabric

**PHP**  
Symfony 2.0 :  
le meilleur framework PHP ?

M 04319 - 137 - 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



N°1 EN FRANCE

- Crée des applications Windows, Linux, Mac, ... compatibles Internet, Intranet, Android, Windows Phone 7
- Éditeur d'analyses (UML, ...)
- RAD (patterns)
- Lien avec toutes les bases de données (ODBC, OLE DB), Oracle, SQL Server, AS/400, Informix, DB2..., lien natif MySQL, PostgreSQL,
- Puissante base de données Client/Serveur HyperFileSQL gratuite incluse
- Générateur d'états PDF, Codesbarres
- Accès natif SAP R/3, Lotus Notes
- Gestion de planning
- Gestion des Exigences
- Langage de 5<sup>e</sup> génération (L5G)
- SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL
- Domotique
- Liaisons série et USB
- Débogage à distance
- Profiler
- Refactoring
- Multilingue automatique (64 langues)
- Gestionnaire de versions
- Gestion de l'Infrastructure Logicielle
- Installateur 1-clic & Push
- Suivi des suggestions utilisateurs
- Tout en français
- etc...

# WINDEV® 16

CRÉEZ VOS APPLICATIONS PERSONNALISÉES POUR PC, MAC, LINUX, INTERNET, MOBILES,...

VOTRE CODE EST MULTI-PLATEFORMES :  
Windows, .Net, Java, PHP, Linux, Mac, J2EE, XML, Internet, SaaS, Windows Phone 7, Pocket PC, Android, ...

# DÉVELOPPEZ

# 10 FOIS PLUS VITE

**996**  
NOUVEAUTÉ

**VERSION EXPRESS GRATUITE**  
Téléchargez-la !

Elu «Langage le plus productif du marché»

Fournisseur Officiel de la Préparation Olympique

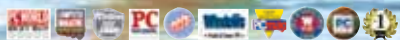


**DEMANDEZ LE DOSSIER GRATUIT**

260 pages + DVD + Version Express gratuite + 116 Témoignages.  
Tél: **04.67.032.032** ou **01.48.01.48.88**  
info@pcsoft.fr



**www.pcsoft.fr**



► **Dossier gratuit 260 pages** sur simple demande. Tél: **04.67.032.032** info@pcsoft.fr

## \\ actus

En bref .....	6
Agenda .....	8

## \\ matériel

Xbox Kinect : déjà des projets de hack .....	10
--	----

## \\ webmaster

Les itinéraires dans une application iPhone .....	12
Silverlight 5 : GPU et développement améliorés .....	15
Migrer de IE8 à IE9 .....	16

## \\ gros plan

<b>Java7 : Retour sur le futur</b> .....	24
--	----

## \\ outils

### Modélisation

Générer des web services Axis Spring+Hibernate à l'aide d'AndroMDA et de RSM (IBM) .....	32
---	----

## \\ carrière

Le cru 2010 des Epitech Innovative Projects .....	36
L'emploi est timidement reparti .....	38

## \\ dossier

### Gardez le fil sur internet

Ecrire un client pour l'API Twitter .....	41
Rest , Reslet : garder le contact .....	44
Architecture Restfull SharePoint 2010 .....	47

## \\ architecture

AppFabric : l'accélérateur de développement d'applications multi-tiers .....	49
--	----

## \\ code

La persistance sous Android .....	53
Embarquer une station web .....	56
Office 2010 : le nouveau VSTO .....	58
Interopérabilité en PHP .....	64
La boîte à outils (presque parfaite), Rails 3 (2 <sup>e</sup> partie) .....	66
Découverte de Samsung Bada (2 <sup>e</sup> partie) .....	69
Communiquer avec les périphériques USB sous Windows .....	72
Symfony 2.0 : PHP comme vous ne l'avez jamais vu ! .....	78

## \\ temps libre

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



6



9



24



40



36



69

L'info continue sur [www.programmez.com](http://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.



# EXPERIENCE

Belles visualisations de données qui donne vie à vos applications.



Utilisez notre Motion Framework™ pour observer vos données dans le temps et donner à vos utilisateurs un nouvel aperçu de leurs données. Visitez [infragistics.com](http://infragistics.com) pour l'essayer dès aujourd'hui !



**NetAdvantage®** ULTIMATE

for ASP.NET, Windows Forms, WPF, Silverlight,  
WPF Data Visualization, Silverlight Data Visualization

**Infragistics®**

Infragistics Ventes France  0800 667 307  
Infragistics Europe Ventes +44 (0) 800 298 9055  
Infragistics India +91-80-6785-1111  
[t@infragistics](http://t@infragistics)







## Le jeu des 7 systèmes

2010 aura été l'année Google. Succès d'Android sur les mobiles, Chrome devenant le vrai concurrent d'Internet Explorer en lieu et place de Firefox, les applications web s'étant imposées et contraignant les concurrents à proposer des alternatives. Sans compter les dons de code, de protocoles, d'outils à la communauté Open Source comme Wave, offert à la fondation Apache, ou le soutien à Eclipse via la création d'un site dédié aux extensions Eclipse[\*].

Le défi de Google en 2011 sera de s'affirmer sur les systèmes d'exploitation. Android connaît déjà un superbe succès sur le mobile et les tablettes. Même si tout n'est pas parfait, le système évolue régulièrement. L'éditeur tentera de refaire le « coup » avec ChromeOS. C'est un super navigateur web, doté de fonctions système et pouvant lancer des applications, en s'appuyant sur le cloud computing. Le concept nous avait paru un peu bancal lors de sa présentation officielle. Google a rectifié certains aspects mais les critiques demeurent, le retard de plusieurs mois se confirme et Google, prudent, ne s'attend pas à un succès immédiat. Il faut dire que Windows garde 90 % du marché, MacOS X continue sa croissance et Linux, sur le desktop, reste dans les limbes. Le défi est digne de l'Everest.

Une question de fond demeure autour de ChromeOS : pourquoi créer deux lignes de système (ChromeOS et Android) ? Et cette question n'a toujours pas reçu de réponses claires. Ce n'est donc pas un hasard si dès novembre 2009, des voix internes à Google, évoquèrent l'idée d'une fusion de ChromeOS dans Android. Sergey Brin lui-même (co-fondateur de Google) l'exprima clairement : ChromeOS et Android sont destinés à fusionner. Car finalement, les deux reposent sur une fondation Linux, possèdent des composants logiciels et techniques similaires. Mi-décembre dernier, Paul Buchheit - créateur de Gmail et concepteur originel de AdSense - prédisait sur Twitter : « ChromeOS sera tué l'année prochaine (ou fusionné avec Android) ».

Car finalement, si ChromeOS ne propose rien de mieux qu'Android, pourquoi continuer dans cette voie ? ChromeOS sera-t-il le nouveau Google Wave ?

On en revient toujours à la même question : pourquoi deux systèmes ? A priori, chacun correspond à des marchés différents. Mais est-ce si sûr ? Android peut très bien convenir à des tablettes et netbooks. Et le système peut introduire des mécanismes propres à ChromeOS.

De son côté, Apple semble répondre partiellement à la question. Aujourd'hui, la Pomme possède deux systèmes : MacOS X et iOS. Mais iOS repose sur les fondations de MacOS X. Il partage donc des briques techniques essentielles et un modèle de développement identique hormis pour les SDK et API. L'annonce de MacOS X Lion (le prochain MacOS X) marque un tournant intéressant : MacOS X va reprendre de plus en plus de concepts propres à iOS pour arriver à une fusion entre les deux. Car finalement, ce qui différencie les deux systèmes c'est l'interface, l'expérience utilisateur et l'usage, mais pour le reste... Il faut cependant que le rapprochement soit fait intelligemment car on n'utilise pas un Mac comme un iPad. Et il est plus facile de maintenir et de développer un code unique puis de l'adapter à chaque terminal que d'avoir un code radicalement différent pour chacun... Et la convergence va commencer dès janvier avec la disponibilité du Mac App Store qui est un clone du App Store...

La guerre des boutiques ne fait que commencer !

■ François Tonic

[\*] <http://code.google.com/a/eclipselabs.org/hosting/>

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.

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,

S. Saurel. Experts : V. Daubry, J. Scher, V. Bellet, D. Diaz,

J. de Oliveira, J.-L. Boucho, D. Guignard, C. Villeneuve,

J. Dollon, G. Durelle, R. Dumitrague, A. Olivier,

P. Antivackis, J. Louvel, T. Templier, T. Boileau,

X. Vanneste, A. Detrie, Y. Moté.

Illustrations couverture : © istockphoto-CostinT

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com.

Dépôt légal : à parution - Commission paritaire :

0712K78366 ISSN : 1627-0908. Imprimeur :

S.A. Corelio Nevada Printing, 30 allée de la

recherche, 1070 Bruxelles Belgique. Directeur de

la publication : J-C Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10  
Tél. : 01 55 56 70 55

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 numé-

ros 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 exclusivement sur

[www.programmez.com](http://www.programmez.com)

**L'INFO PERMANENTE**  
[WWW.PROGRAMMEZ.COM](http://WWW.PROGRAMMEZ.COM)



**PROCHAIN NUMÉRO**  
**N°138 février 2011**  
parution 29 janvier

✓ **Plateforme Atom**

Développez pour la plateforme Intel Atom et la boutique AppUp

✓ **SGBD**

NoSQL : la mort des bases de données relationnelles et SQL ?

✓ **Spécial TechDays**

Découvrez les dernières annonces de Microsoft

Cahier spécial Team Foundation Server

**MEILLEURS VŒUX POUR L'AN NOUVEAU !**

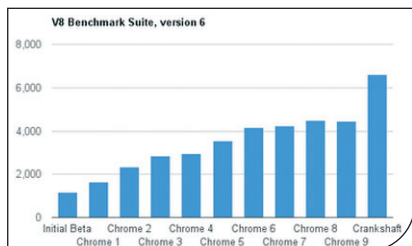
**VOUS POUVEZ COMPTER SUR NOUS !**

$$2 = 0 + 1 + 1$$

■ **Python 3.2** est en phase de développement. On est actuellement en version bêta. La nouveauté majeure de la 3.2 est le support de la programmation concurrente ! Celle-ci sera regroupée dans un package fort pertinemment nommé 'concurrent' et qui contient pour l'instant un seul module : 'futures'. Ce module est directement inspiré de ce que l'on trouve en Java sous `java.util.concurrent.package`.

Futures est une abstraction pour les threads, processus et appels de procédures distantes. Des exécuteurs permettent de lancer les tâches en parallèle et de les gérer dans des pools.

■ Google veut bouleverser les moteurs JavaScript. Et l'éditeur introduit **Crankshaft (V8)**. Le compilateur « juste à temps » du moteur est modulaire pour



gagner en performance : un compilateur de base sans optimisation « optimale », un profileur pour identifier les zones de code sensibles, un compilateur optimisateur pour optimiser les hotspots et enfin un désoptimisateur. Ce fonctionnement paraît complexe mais permettra de mieux tirer parti de chaque composant du moteur JavaScript et de travailler de manière indépendante. Mozilla saura-t-il faire mieux avec le projet Tracemonkey ?

■ **Codendi 4.2** est en version stable. La principale nouveauté est une réécriture complète de tout le système de suivi. Celui-ci fournit ainsi de nouvelles fonctionnalités plus efficaces : interface utilisateur et administration plus intuitive, création rapide de tableaux de bord, workflow, etc.

## La guerre des suites bureautiques aura lieu

Cette fois c'est certain, OpenOffice et LibreOffice vont se livrer la guerre bureautique ! LibreOffice, fondé par le Document Foundation, suite à un conflit avec Oracle autour d'OpenOffice, a dégainé le premier avec la première version de sa suite bureautique : LibreOffice 3.3. Tout naturellement elle se base sur OpenOffice 3.3 avec un code optimisé et de nouvelles fonctions qui marquent une première étape vers l'ambitieuse roadmap 2011. Le but est de renouveler entièrement le concept même de suite bureautique vieille de 20 ans. La première tâche sera tout de même de nettoyer le code pour l'optimiser, l'améliorer, car les performances d'OpenOffice n'étaient pas forcément son meilleur atout. Le vœu de la fondation est de libérer le contenu du format et de la présentation, de réduire la dépendance de la suite envers Java. Avec comme volonté de remettre l'utilisateur au cœur ! Jusque-là rien de révolutionnaire tout de même. Mais la fondation n'a pas encore dévoilé ce qu'elle espère réaliser. Attendons de voir. L'interface sera, espérons-le, un chantier prioritaire. Car Microsoft avec Office a réussi un virage à 180° de l'interface en introduisant notamment la notion de ruban qui après des débuts chaotiques est un réel succès.

Mais pendant ce temps, Oracle n'a pas dit son dernier mot avec Open Office dont il reste propriétaire. Tout d'abord, c'est l'annonce de OpenOffice 3.3. Cette version améliore de nombreuses fonctions du traitement de texte et du tableau : 1 million de lignes dans un tableau, insertion d'un dessin dans un graphique, nouvelle commande mathématique (nospace), moteur de recherche unifié, nouvelle interface d'impression. Mais surtout Oracle annonce deux versions d'Open Office : édition standard et édition entreprise. La version entreprise inclut les

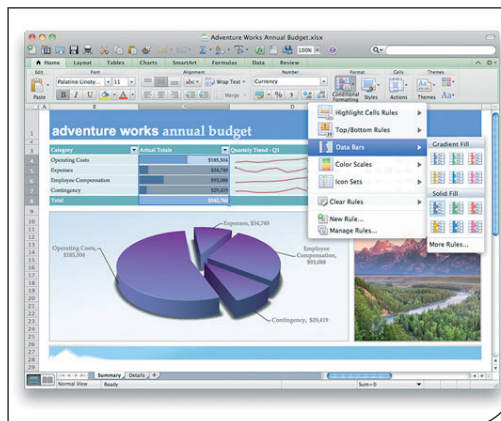


outils entreprises (kit de développement, les connecteurs multiples vers la BI, les bases de données, le support utilisateur). Les deux versions sont payantes.

### Le Cloud arrive

Mais l'annonce la plus importante reste Oracle Cloud Office. Il s'agit d'une suite bureautique Web et mobile permettant une collaboration Web 2.0 et l'accès mobile aux documents. La compatibilité avec Microsoft Office et l'intégration avec Oracle Open Office permet d'éditer hors ligne des présentations complexes, des documents textuels et des feuilles de calcul, de façon transparente. Elle a été conçue pour tirer parti d'une architecture Web extrêmement souple, capable de monter en puissance très facilement. Les clients et partenaires peuvent s'appuyer sur Oracle Cloud Office en choisissant un déploiement sur site, à la demande ou en mode SaaS.

Pendant que Oracle et Document Foundation se battent, Microsoft avance tranquillement sur la bureautique en mode cloud avec Office Web Apps et surtout la stratégie Office 365, tout en renouvelant l'interface et la partie BI et serveur de la suite, deux éléments qui font cruellement défaut à Libre et Open Office...







BEST-SELLER

**FusionCharts** à partir de € 147

Graphiques Flash &amp; JavaScript (HTML5) interactifs pour les applications Web.

- Animez vos applications Web avec des graphiques interactifs et pilotés par les données
- Créez des graphiques AJAX avec des possibilités d'exploration en quelques minutes
- Exportez les graphiques au PDF et les données en CSV directement depuis les graphiques
- Créez des jauges, des tableaux de bord, des graphiques financiers et plus de 550 types de carte
- Adopté par plus de 17 000 clients et 330 000 utilisateurs dans 110 pays

BEST-SELLER

**Spread for Windows Forms** à partir de € 739

GrapeCity PowerTools

Feuille de calcul pour les applications Windows Forms, compatible avec Microsoft Excel.

- Accélérez le développement avec les concepteurs de feuilles de calcul, l'Assistant de prise en main et les concepteurs de graphiques
- Renseignement automatique : anticipation de la frappe dans la cellule
- Nouveau - outil intégré de création de diagrammes avec 85 styles
- Nouveau - préserve les .XLS et restaure les fonctions non supportées
- Inclut des apparences prédéfinies ainsi que la possibilité de créer des apparences personnalisées

BEST-SELLER

**Nevron Chart for .NET Enterprise** à partir de € 526

NEVRON

Fonctionnalités riches de création de tableaux à vos applications Windows Forms et ASP .NET.

- Graphiques en 2 et 3D : histogrammes, lignes, escaliers, aires, secteurs, points, bulles, bourse, flottants, radar, polaires, max./min., mailles, grilles, formes, courbes lisses, barres flottantes, Venn et erreur. Nombreuses variations : nuages de points XY et XYZ, barres empilées, etc.
- Axes avec barre de défilement, légende intégrée, annotations et filigranes sur graphique
- Support de la conception Visual Studio, nombreuses fonctionnalités avancées (éditeurs de style)

BEST-SELLER

**IBM ILOG Elixir Standard** à partir de € 813

IBM

Étendez les plateformes Adobe Flex et AIR avec des contrôles de visualisation interactifs.

- 10 contrôles d'UI : graphique 3D, jauge, carte, carte de clics, calendrier, OLAP et tableau croisé dynamique, organigramme hiérarchique, arborescence, graphique en radar et chronologie
- Intégration transparente avec l'environnement de développement intégré d'Adobe Flash Builder 4
- Controls can be used individually or in combination, alongside other Flex components
- Utilisation individuelle ou combinée des contrôles avec les autres composants Flex

# LeWeb 2010 : 3 000 participants, 60 pays !

Les 8 et 9 décembre derniers, LeWeb10 a enflammé le monde du web européen. L'événement a rassemblé sur deux jours plus de 3 000 personnes venant de 60 pays ! Des dizaines de sessions plénières et d'ateliers se sont succédés sans interruption du matin au soir.

Il y en avait pour tout le monde et tous les goûts : Renault, informatique par la pensée, la mobilité, un peu de XML... Seul vrai regret, que nos confrères ont soulevé, l'absence d'une véritable ligne directrice, d'un thème unique. Cette année, le "thème" fut la plate-forme. Mais laquelle ? Comment la comprendre ? Le Web est-il LA plate-forme ? Peut être, mais comment monétiser le web ? Ne faut-il pas aller sur le mobile, le cloud ? Une table ronde a d'ailleurs tenté de présenter ce thème de la plate-forme : "comment construire sa plateforme". Est-ce par les technologies, les

API, les outils, l'usage que l'on en fait ? Et finalement quelle est sa finalité ? On peut dire qu'il s'agit d'avoir la bonne application au bon moment pour répondre aux utilisateurs et d'avoir un business model.

Une des conférences les plus intéressantes et les plus spectaculaires fut celle de la société Interaxon qui a démontré le contrôle de l'ordinateur par la pensée. Cette technique existe depuis quelques années et plusieurs éditeurs (surtout américains) proposent déjà logiciels et matériels. Microsoft a tenté de démontrer que Windows Phone 7 était une bonne alternative à Apple et son iPhone et qu'il y avait la place pour lui. Nokia a aussi joué sa partition sur sa vision du mobile. Le grand absent était Apple, mais tellement présent avec les iPhone et iPad visibles un peu partout et dans les conférences ! Trop fort ! Renault a aussi marqué LeWeb10 par sa présenta-



© LeWeb10

tion de la vision du transport de demain et notamment par la présence de sa voiture électrique. Intel était là avec sa boutique applicative : AppUp. L'éditeur / fondateur veut pousser son AppStore maison basé notamment sur Atom et capable de fonctionner sur la télévision, les netbook, smart-

phone, desktop. Le développeur est largement courtisé pour profiter de cette nouvelle opportunité. A vous de vous faire une idée de l'événement en visionnant les vidéos !

Toutes les conférences en vidéo : <http://www.youtube.com/lewebparis#g/>

## agenda \

### JANVIER

- Du 17 au 18 janvier 2011, FIAP Jean Monnet, 30 rue Cabanis 75014 Paris - **SemWeb.Pro** : une conférence dédiée à l'univers du Web Sémantique. FIAP Jean Monnet, 30 rue Cabanis 75014 Paris <http://www.semweb.pro/appeal-a-communication>.

### FÉVRIER

- Du 08 au 10 février, Palais des Congrès de Paris : TechDays 2011 Rendez-vous incontournable des développeurs, décideurs et professionnels de l'informatique, Les **Microsoft TechDays** sont une occasion unique de se former sur toutes les nouveautés Microsoft et de découvrir les tendances du marché. [http://www.microsoft.com/france/mstechdays/?CR\\_CC=200012242](http://www.microsoft.com/france/mstechdays/?CR_CC=200012242)

### AVRIL

- Du 7 au 9 avril, Paris, **Hackito Ergo Sum 2011**. Conférence sur la sécurité, le hacking ouvre dès à présent les appels à contribution (jusqu'au 20 février). Site : <http://hackitoergosum.org>

## Le W3C investit sur l'internet mobile

**L**e consortium W3C a dévoilé mi-décembre les bonnes pratiques pour les applications web mobiles (ou MWABP). Il s'agit d'une recommandation W3C pour les développeurs, éditeurs, fournisseurs afin d'assurer de meilleures applications mobiles quelle que soit la plate-forme mobile utilisée. « Ce document est une mine d'or de conseils : j'ai eu l'occasion de les mettre en application pour m'assurer que l'application mobile «antisèche» du W3C (W3C cheatsheet) fonctionnait non seulement sur des appareils mobiles, mais aussi sur tous les autres types d'appareils » déclare Dominique Hazaël-Massieux, responsable de l'initiative

Web mobile pour le W3C. Une des ambitions est de pouvoir écrire une fois le code et de l'appliquer sur l'ensemble des systèmes mobiles ! Les applications Web mobiles ont un avenir brillant. L'essor du marché des applications mobiles a de ce fait accru l'intérêt porté au Web comme plate-forme de développement pour ces appareils (comme le relève un livre blanc rédigé par un analyste du GIA). Les applications Web remplacent déjà les applications natives présentes sur beaucoup d'ordinateurs. L'autre effort est de centraliser le développement des applications autour de HTML 5, CSS 3 et SVG.



## Le projet **Mono** continuera à évoluer en 2011

**M**iguel de Icaza rassure la communauté (Mono et Moonlight, les piles .Net et Silverlight open source). Et le travail ne manque pas : continuation des développements autour de Mono, des outils Mono, de Moonlight, possible rajout des langages F#, IronRuby et UnityScript, poursuite de l'implémentation des nouveautés .Net et Silverlight de Microsoft, nouvelles évolutions de MonoTouch et MonoDroid.

La dernière version de Mono-Develop, la 2.4, supporte les projets .Net 4, Gtk# sur MacOS X, intégration native améliorée sur Mac et Windows. Pour cet environnement, une extension F# est désormais disponible depuis mi-novembre : F# MonoDevelop Add-In qui permet de développer et de compiler du code F# directement sur MonoDevelop. Pour l'interface, on peut utiliser Gtk#. Une autre opportunité de langage pour Miguel concerne IronRuby et

IronPython. Ces deux projets ne font plus partie des développements de Microsoft et ont été offerts à la communauté. Reste à voir comment implémenter ces langages dans le projet Mono mais l'objectif est aussi d'étendre la communauté et de supporter Linux, Windows et MacOS X. La partie parallélisme dans Mono sera aussi améliorée dans les prochains mois.

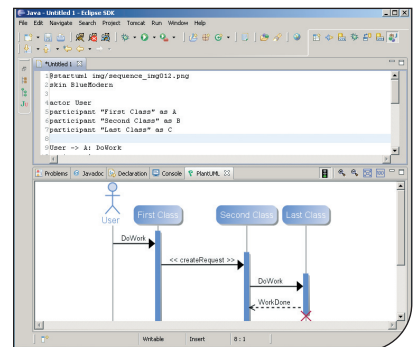
Sur la partie Moonlight, la pile open source de Silverlight, le travail continue même si la cadence de Microsoft, notamment avec l'annonce de Silverlight 5, peut donner des cauchemars. Le but de l'équipe est toujours de faire évoluer le projet et supporter au fur et à mesure les nouvelles fonctionnalités. Ainsi, l'accélération matérielle est désormais présente, ce qui devrait aider les applications Moonlight à améliorer leurs performances. DeepZoom, la fonction de zoom infini, aura droit à une sérieuse améliora-



tion. Sur la partie mobile, iOS et Android, le travail se poursuit aussi. MonoTouch, dédié à iOS d'Apple, évolue régulièrement. Actuellement, la version 3.2.2 supporte iOS 4.2, les nouvelles API et améliore de nombreuses API et fonctions. MonoDroid pour Android suit le même objectif, développer en C# / Mono des applications mobiles. La version 1 supporte le moteur Mono 2.8, OpenGL, par exemple. La v2, déjà annoncée, sera une version plus mature avec la prise en compte des demandes des développeurs. Bref, les divers projets Mono n'ont pas fini d'évoluer en 2011 !

■ Un nouveau langage est en train de voir le jour : **Reia**. Ses concepteurs ont pour ambition de marier le meilleur des deux mondes. D'abord le monde d'Erlang, le champion de la programmation concurrente et distribuée, avec ses capacités de tolérance aux fautes et de chargement de code à chaud. Côté syntaxe et modernité du langage, Reia se rapprochera de Ruby pour la machine virtuelle d'Erlang et il apportera une syntaxe plus facile, l'introspection, la meta-programmation et les blocs.

■ **PlantUML** est un outil écrit en Java qui permet de générer des diagrammes UML à partir d'un langage texte simple et intuitif. Sept

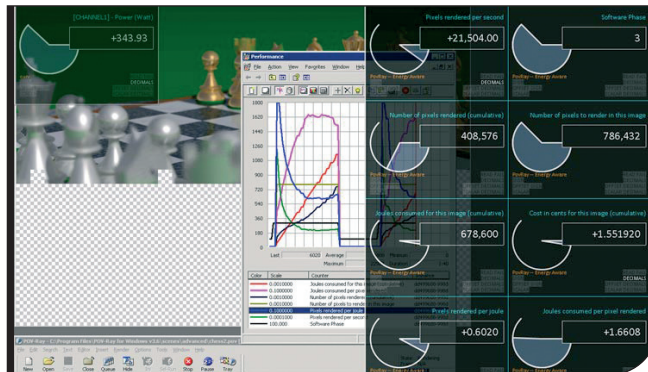


types de diagrammes sont supportés : séquence, cas d'utilisation, classe, activité, composant, état et objet. Le document qui servira à la génération peut être par exemple écrit sous Word ou bien depuis un plugin Eclipse. Les diagrammes générés sont au format PNG ou SVG. Il est encore possible de les générer en ASCII Art.

■ Oracle sort **MySQL 5.5**, une bonne nouvelle pour tous les utilisateurs de MySQL. L'accent a été mis sur les performances allant jusqu'à un facteur 5 d'accélération ! Les premières réactions étaient plutôt positives comme celles de Kaj Arno, co-fondateur de SkySQL Ab, qui se dit ravi de la disponibilité de la 5.5 en licence GPL.

## Intel renouvelle son **Energy Checker SDK**

**L**e fondeur Intel continue de travailler sur le GreenIT. Pour cela, une nouvelle version de son Energy Checker SDK est disponible depuis la mi-décembre. Ce dernier doit permettre aux développeurs de voir l'efficacité d'un système, de la consommation énergétique du matériel mais aussi des applications. Le but est d'aller loin dans l'analyse pour voir où des optimisations seraient possibles. On bénéficie d'une instrumentation logicielle via des API que les applications peuvent embar-



quer. La version 2.0 peut désormais fonctionner sur des terminaux embarqués et ceux fonctionnant sous MeeGo, support du langage

Objective-C, le moniteur utilise HTML 5 pour les mises à jour dynamiques. Site : <http://software.intel.com/en-us/articles/intel-energy-checker-sdk/>



■ Intel a dévoilé fin novembre de nouveaux processeurs Atom qui ont la caractéristique d'être configurables. Cela signifie qu'il est possible pour un intégrateur, constructeur d'obtenir des design sur mesure ! C'est possible avec la gamme de processeurs Atom E600C. Ils se prêtent plus facilement aux modifications de conception sans changements matériels compliqués, soit des coûts de développement en baisse. En raison de ce regroupement, ils se caractérisent aussi par d'importants gains de place et une meilleure maîtrise des stocks ainsi que par un flux de fabrication simplifié et un contexte mono-fabricant.

■ AMD continue à travailler pour faciliter le travail du développeur sur ses processeurs et GPU. Courant novembre, le fondateur a dévoilé ATI GPU Services Library. Ce kit se destine aux technologies AMD Eyefinity et CrossFireX. Le développeur pourra ainsi contrôler l'état et la configuration du GPU, coder des applications Eyefinity sans disposer du matériel requis, etc.

Pour en savoir plus :

<http://developer.amd.com/gpu/ags/Pages/default.aspx>

■ Le mini PC sous Linux, Linotop, est disponible dans la 3e évolution. Compact, puissant avec un processeur Intel ATOM à 1.6 GHz, fanless (sans ventilateur), il est particulièrement adapté pour un poste de travail sans maintenance, un kiosque d'accès à Internet ou pour de l'affichage dynamique. Il se connecte à un écran VGA, DVI, HDMI et possède 6 ports USB. 34 euros HT. Idéal pour un barebone !



■ Le constructeur Dane-Elec dévoile la nouvelle version de son my-Ditto qui intègre un disque dur réseau, une clé propriétaire pour avoir accès partout à ses données depuis le web avec une sécurité maximale. Désormais, il supporte les

téléphones Android, création d'une prévisualisation des documents, possibilité de brancher un disque dur additionnel USB, outil de backup automatique.

Site : <http://www.my-ditto.com>

## Xbox Kinect : déjà des projets de hack



La première console de jeux temps réel sans manette, Xbox Kinect se vend très bien. Mais dès sa sortie, les bidouilleurs et développeurs se sont penchés sur son cas : comment manipuler la Kinect, installer des logiciels, récupérer ses flux ? Et il faut dire que l'on trouve déjà beaucoup de choses : openkinect piano, Kinect MIDI controller, Kinect 2 Ledwall, ou encore comment contrôler Windows 7 avec la Kinect ou utiliser la console avec son iPad !

Le projet Open Kinect vise à créer des pilotes open source, notamment via un concours lancé peu après la sortie de la console. Et les premiers résultats se sont rapidement fait connaître. Un des hackers (Marcan42) a pu en quelques heures récupérer des flux vidéo des caméras de la console. Et même des hackeuses s'y mettent comme LadyAda (voir son blog très intéressant : <http://www.adafruit.com/blog/2010/11/04/the-open-kinect-project-the-ok-prize-get-1000-bounty-for-kinect-for-xbox-360-open-source-drivers/>). Et le concours Open Kinect Project est doté de plusieurs milliers de dollars. Il y a quelques années, la Wii avait eu aussi beaucoup de succès auprès des développeurs et des hackers en tout genre pour utiliser la WiiRemote directement sur son PC ! Même si Microsoft n'est pas ravi de cette effervescence, cela contribue au succès de la Kinect et l'éditeur ne semble pas interdire la création de

drivers pour sa nouvelle console. Sinon, que fournit Microsoft autour de la Kinect ? D'abord, il faut savoir que les couches matérielles et logicielles ne sont pas modifiables mais Microsoft laisse la possibilité de créer de nouveaux usages. Par contre, le SDK n'est pas disponible librement, il est réservé aux éditeurs de jeux.

### Pilotes disponibles

Même sans SDK accessible, cela n'empêche pas les développeurs de s'y intéresser. Code Laboratories prend en compte la Kinect via NUI Platform (<http://codelaboratories.com/nui/>). Le projet a démarré en .Net (WPF et C#). Pour le moment, on dispose des flux vidéo et de l'accéléromètre, prochainement de la partie audio. Mais les développeurs ne s'arrêtent pas à la Kinect. Si tous ces projets sont intéressants en soi, reste à trouver les usages, les applications.

Mi-décembre, PrimeSense, le créateur des pilotes Kinect, a distribué officiellement les drivers de la console. Ils sont compatibles avec Linux et Windows. De plus, l'éditeur a rendu publique son API, OpenNI et la documentation technique. Les développeurs vont pouvoir s'écarter. Sélection de divers projets autour de la Kinect : <http://www.2-high.info/post/2010/11/25/Open-Kinect>

Site Open Kinect : <http://www.openkinect.co.uk/>



# Advanced database technology for breakthrough applications



## Laissez vos applications s'envoler...

Avec InterSystems Caché, faites décoller vos applications. Elles bénéficieront immédiatement de performances hors du commun, deviendront massivement scalable et ne nécessiteront plus d'administration fastidieuse.

InterSystems **Caché**® base de données post-relationnelle, mais aussi serveur d'application, framework Ajax, ... se fonde sur une technologie Objet avancée qui permet de construire beaucoup plus facilement des applications XML, Web Services, AJAX, Java et .NET.

InterSystems Caché est aussi une base SQL jusqu'à 5 fois plus rapide que les bases de données relationnelles classiques en accès SQL et bien plus en accès Objet !

Grâce à son Architecture de Données Unifiée unique, Caché élimine le

besoin de mapping objet-relationnel, réduit les temps de développement, et facilite l'évolution et la maintenance de votre application.

Déployé sur plus de 100.000 systèmes de par le monde pour des applications de 2 à 50.000 utilisateurs, Caché est disponible sur toutes les plates-formes majeures du marché.

**Nouveau:** L'intégration de Caché avec JAVA - JNI permet littéralement d'exploser les performances et d'offrir enfin aux programmeurs JAVA un moteur sans compromis et digne de leurs réalisations.

Depuis plus de 30 ans InterSystems vous apporte des technologies avancées qui vous permettent de construire des applications qui font la différence.

# INTERSYSTEMS

Téléchargez votre version gratuite complète InterSystems Caché - sans limite de temps: [InterSystems.fr/avancee](http://InterSystems.fr/avancee)

# Les itinéraires dans une application iPhone



Jusqu'à cet été, les applications iPhone qui souhaitaient offrir un itinéraire sur une carte Google étaient condamnées à faire sortir l'utilisateur pour débrayer sur l'application maps. Et un utilisateur qui quitte une application a de bonnes chances de ne pas la relancer.

**L**e 21 juin 2010, l'iPhone passe en version 4. Tous les changements ne figurent pas dans la liste des changements officiels. Et parmi les fonctionnalités non annoncées dans le changelog, il y en a une qui nous permettra de gérer l'affichage d'itinéraires directement sur une carte, sans quitter l'application ! C'est ce que nous allons vous proposer de coder. Mais nous avançons un peu vite : revenons

un peu en arrière... Avec l'OS 3, il était déjà possible d'afficher un itinéraire. Mais accrochez vous, il fallait :

- Effectuer une connexion http vers l'API Google Direction, pour recevoir l'itinéraire entre deux points, et parser la réponse.
- Pour chaque point, convertir la coordonnée GPS en points, en fonction de la taille de votre mapView.
- Afficher une vue transparente par-dessus la carte, et relier les points.
- Ne pas oublier de « déplacer » les points simultanément au déplacement de l'utilisateur, si celui-ci a le malheur de vouloir scroller.

Je vous passe les algorithmes mis en œuvre lorsque l'utilisateur zoome... Cela va sans dire, le rendu était bien moins soigné que sur l'application Plan. Ce qui est bien, mais pas top.

L'OS4 révolutionne tout cela. Dans les notes de version, on ne trouve que deux modifications mineures aux cartes. Pourtant, une API complète a été ajoutée : les Overlays. Il s'agit de calques que vous pouvez appliquer sur votre carte, et qui se transformeront de manière synchronisée avec cette dernière. Sur ces calques, vous pouvez ajouter des points, des lignes, des boutons, voire même redessiner complètement le look par défaut.

Dans la suite de cet article, nous allons détailler les points critiques pour afficher un itinéraire dans une application iPhone :

- la récupération d'itinéraire
- la création d'un calque à afficher sur la carte
- le paramétrage de cet affichage.

Nous terminerons en parlant design de code : comment faire tout cela de manière lisible et réutilisable.

Le code complet est disponible sur le site du magazine.

## Quel est le plus court chemin entre deux points ?

Demander à Google l'itinéraire entre deux points se fait en appelant l'URL suivante :

```
#define GOOGLE_ROUTE_URL @"http://maps.google.com/maps/api/directions/json?origin=%f,%f&destination=%f,%f&sensor=false&mde=walking"
```

Et bizarrement il ne vous renverra pas une droite, mais une route à suivre. Voici un extrait du JSON que l'on reçoit en retour :

```
{
  «status»: «OK»,
  «routes»: [ {
    «summary»: «D115»,
    «legs»: [ {
      «steps»: [ {
        «travel_mode»: «WALKING»,
        «start_location»: {
          «lat»: 43.0108900,
          «lng»: 2.9920800
        },
        «end_location»: {
          «lat»: 43.0138400,
          «lng»: 2.9882000
        },
        «polyline»: {
          «points»: «aqoeGokgQKt@cBrCg@d@Qn@WpFMh@eA|@{DpBm@e@AX»,
          «levels»: «B????@???B»
        },
        «duration»: {
          «value»: 385,
          «text»: «6 minutes»
        },
        «html_instructions»: «Prendre la direction \u003cb\u003eouest\u003c/b\u003e \u003cspan style=\»white-space: nowrap\»\u003e- continuer pendant\u003cb\u003e0,3&nbsp;mi\u003c/b\u003e\u003cspan\u003e»,
        «distance»: {
          «value»: 503,
          «text»: «0,5 km»
        }
      }
    ],
    «duration»: {
      «value»: 157476,
      «text»: «1 jour 20 heures»
    }
  }
  [...]
],
  «duration»: {
    «value»: 157476,
    «text»: «1 jour 20 heures»
  }
}
```



```

«distance»: {
  «value»: 209029,
  «text»: «209 km»
},
«start_location»: {
  «lat»: 43.0108900,
  «lng»: 2.9920800
},
«end_location»: {
  «lat»: 41.9996700,
  «lng»: 2.0002700
},
«start_address»: «11130 Sigeon, France»,
«end_address»: «08517 Santa Maria de Merlès, Espagne»,
«via_waypoint»: [ ]
} ],
«copyrights»: «Données cartographiques ©2010 Tele Atlas»,
«overview_polyline»: {
  «points»: «aqoeGokg [...] @du@»,
  «levels»: «BAA@@@AA [...] @@AAB»
},
«warnings»: [ «Le calcul d'itinéraires piétons est en bêta.
Faites attention – Cet itinéraire peut ne pas comporter de
trottoirs, ni de voies piétonnes.» ],
«waypoint_order»: [ ]
} ]
}

```

Pour le parser, nous vous recommandons d'utiliser la librairie JSON disponible à l'adresse suivante : <http://code.google.com/p/json-framework/>. Ce fichier contient un objet « legs ». Ce dernier possède une liste de « step », correspondant à une droite entre deux points. « start\_location » en est l'origine, et « end\_location » la destination. En considérant un « step » sur deux, nous pouvons donc reconstituer les points de notre ligne brisée, qui sera notre itinéraire. Ces points formeront une liste de coordonnées GPS. Ce sera un tableau de MKMapPoint. Voici le code permettant de constituer ce tableau :

```

-(NSArray*) routeStepsArrayFromJSONString {
    NSDictionary *jsonData = [jsonData JSONValue];
    NSArray *routes = [jsonData objectForKey:@"routes"];
    NSDictionary *legsDict = [routes objectAtIndex:0];
    NSArray *legs = [legsDict objectForKey:@"legs"];
    NSDictionary *stepsDict = [legs objectAtIndex:0];
    NSArray *steps = [stepsDict objectForKey:@"steps"];
    return steps;
}

-(MKMapPoint*) mapPointCArrayFromJSONString {
    NSArray *steps = [self routeStepsArrayFromJSONString];

    MKMapPoint *mapPointCArray = malloc(sizeof(CLLocationCoordinate2D)
    * [steps count]*2 -1);
    numberOfPoints = [steps count]-1;

    int index=0;
    for (NSDictionary *stepDict in steps) {

```

```

        [self addRouteStepDict:stepDict toMapPointCArray:map
        PointCArray atIndex:index];
        index = index+2;
    }

    return mapPointCArray;
}
[fichier OctoRouteParser.m]

```

**Note :** Le fichier JSON contenait également un attribut « overview\_polyline ». Il s'agit de l'ensemble des coordonnées cryptées dans une seule string. Le code à adapter pour la décoder est disponible à l'adresse suivante : <http://sapessi.com/2010/06/decode-a-gpolyline-in-objective-c/>. Cette solution peut être un peu plus optimisée, mais plus complexe à mettre en oeuvre.

Nous obtenons donc un tableau de MKMapPoint, et nous en connaissons le nombre de points. Il ne nous reste alors plus qu'à créer une MKPolyline avec ces points, et à l'ajouter à notre MKMapView. Voici un code permettant de le faire facilement

```

-(void) routeLoadSucceededWithRoutePoints:(MKMapPoint*)route
Points count:(int)pointNumber {
    MKPolyline* routeLine = nil;
    routeLine = [MKPolyline polylineWithPoints:routePoints count
:pointNumber];

    // add the overlay to the map
    if (nil != routeLine) {
        [self removeOverlays:self.overlays];
        [self addOverlay:routeLine];
    }
}
[fichier MKMapView+OctoRoute.m]

```

Notre MKPolyline est un calque à afficher par-dessus la carte. Cependant, si l'iPhone ne sait pas comment l'afficher, il laissera ce calque transparent. Nous devons donc en définir l'affichage.

## Que la Polyline soit !

Pour que notre contrôleur reçoive des messages de la carte, il devra implémenter le protocole MKMapViewDelegate. La carte doit également lui être liée dans Interface Builder. Se référant à la documentation officielle, nous constatons que toutes les méthodes de ce protocole sont optionnelles. L'une d'entre elles nous intéresse particulièrement :

```

- (MKOverlayView *)mapView:(MKMapView *)mapView viewForOverlay:
(id <MKOverlay>)overlay Asks the delegate for the overlay
view to use when displaying the specified overlay object.

```

Disponible depuis l'OS 4.0, cette méthode demande au contrôleur de lui retourner une vue pour un calque donné. Cette vue devra être une MKOverlayView, qui est une vue classique, avec quelques méthodes de plus, principalement pour déterminer sa position sur la carte. Enfin, pour l'affichage d'un calque de type MKPolyline, l'API nous propose une vue spécifique : MKPolylineView.

Nous implémenterons donc une méthode qui :

- Récupère le calque MKPolyline représentant la route à afficher
- Le lie avec une MKPolylineView
- Paramètre cette vue, et la retourne.

Apple a tenu à nous simplifier ces tâches !

```
- (MKOverlayView *)mapView:(MKMapView *)mapView viewForOverlay:
(id <MKOverlay>)overlay
{
    MKPolylineView *routeLineView = [[[MKPolylineView alloc] initWithPolyline:overlay] autorelease];
    routeLineView.fillColor = [UIColor redColor];
    routeLineView.strokeColor = [UIColor redColor];
    routeLineView.lineWidth = 3;

    return routeLineView;
}
[fichier RootViewController.m]
```

Et la polyline fut.

## Soyons catégoriques !

Le modèle "Modèle-Vue-Contrôleur" (MVC) est préconisé pour le développement d'applications iPhone. Dans celui-ci, la vue gère l'affichage. Elle est native pour l'iPhone : c'est ce qui traduit le fichier .xib en de jolis effets à l'écran. Le modèle gère l'accès aux données. Enfin, le contrôleur est un liant, chargé de gérer les événements de la vue, ainsi que la synchronisation entre les différents éléments.

Notre contrôleur aura donc en charge :

- De demander à la couche de données d'ajouter un calque sur la carte
- De paramétrer la vue pour afficher ce calque.

Il contiendra le minimum de code pour réaliser ces deux fonctions. Le paramétrage de la vue a été explicité dans la partie précédente. Intéressons nous à la demande faite à la couche de données.

```
- (void)viewDidLoad {
    [super viewDidLoad];

    CLLocationCoordinate2D start;
    start.latitude = 42.0;
    start.longitude = 2.0;

    CLLocationCoordinate2D end;
    end.latitude = 43.0;
    end.longitude = 3.0;

    [self.mapView addRouteFrom:start to:end shouldZoom:NO];
}
```

Nous souhaitons donc qu'un objet de type MKMapView réponde au message "addRouteFrom:to:shouldZoom:". Deux solutions sont proposées pour cela : étendre MKMapView par héritage, ou la compléter via l'usage des category.

Étendre MKMapView, c'est créer une nouvelle classe qui en hérite. Nous pourrions la nommer CustomMapView. Cela permet de redéfinir les méthodes existantes, ou d'en créer de nouvelles. Cependant, les objets souhaitant bénéficier de ce nouveau comportement devront utiliser ce nouveau type. Concrètement, sur du code existant cela signifie remplacer tous vos objets MKMapView par des objets CustomMapView, le refactoring induit peut être coûteux.

Objective-C propose une alternative pour enrichir directement des objets existants : les category. Une category peut également modifier des méthodes existantes et en créer de nouvelles. Cependant, il est impossible d'y définir de nouvelles variables d'instance.

Pour plus d'information sur le concept de Category :

[http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Articles/ocCategories.html#//apple\\_ref/doc/uid/TP30001163-CH20-SW1](http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Articles/ocCategories.html#//apple_ref/doc/uid/TP30001163-CH20-SW1)

Par convention, une category se trouve dans un fichier nommé ClasseDeBase+NomDeLaCategory. La syntaxe est la suivante :

```
#import <Foundation/Foundation.h>
#import <MapKit/MapKit.h>

@interface MKMapView (OctoRoute)

- (void) addRouteFrom:(CLLocationCoordinate2D)startCoord
to:(CLLocationCoordinate2D)endCoord zommed:(BOOL)shouldZoom;

@end
```

C'est dans cette catégorie que nous définirons nos méthodes. Ainsi, une simple MKMapView sera capable de tracer un itinéraire.

Nous pouvons désormais proprement et simplement afficher un itinéraire sur une carte. Mais qu'en est-il de la légalité de la chose ?

## Des accords entre Apple et Google

Certains d'entre vous se voient peut-être déjà en train de proposer leur propre application de guidage routier (qui ont toujours fait partie du top 10 des applications les plus rentables de l'appstore). Nous sommes effectivement techniquement capables de créer une application de Guidage Par Satellite (GPS). Cependant, les termes et conditions de l'API Google Maps concernant ce sujet sont stricts :

En outre, le Service ne doit pas être utilisé : pour ou avec un système de guidage routier en temps réel (y compris, sans s'y limiter, le guidage routier étape par étape et tout autre système d'itinéraire activé à l'aide d'un capteur), ou pour ou avec les systèmes ou fonctions associés au contrôle automatique ou autonome du comportement du véhicule. Autrement dit, dès lors que vous récupérez la position géographique de l'utilisateur depuis le GPS pour créer un système de guidage, vous risquez d'être en désaccord avec les conditions d'utilisation de l'API MapKit de Google. **Vous êtes dans la légalité tant que l'origine et la destination de votre itinéraire n'est pas donnée par le capteur GPS du téléphone.**

Vous pouvez consulter les termes complets du service maps en français à cette adresse :

[http://www.google.com/intl/fr\\_fr/help/terms\\_maps.html](http://www.google.com/intl/fr_fr/help/terms_maps.html)

Et spécifique à l'iPhone :

<http://code.google.com/intl/de/apis/maps/iphone/terms.html>

Les termes et conditions de l'iPhone font également référence à l'utilisation de l'API de localisation pour la réalisation d'un système de guidage. (voir plus haut).

[http://developer.apple.com/programs/terms/ios/standard/ios\\_standard\\_agreement\\_20100909.pdf](http://developer.apple.com/programs/terms/ios/standard/ios_standard_agreement_20100909.pdf)

A votre créativité... prêts ? Codez !



■ Vincent **Daubry**, Expert iPhone pour OCTO Technology, Cabinet spécialisé en architecture de Systèmes d'information.

■ Jonathan **Scher**, Expert iPhone et Agilité pour OCTO Technology, Cabinet spécialisé en architecture de Systèmes d'information.  
<http://blog.octo.com/>, <http://www.universite-du-si.com/>







# Silverlight 5 : GPU et développement améliorés

Lors de la PDC 2010 (en octobre dernier), des rumeurs sur un possible abandon pur et simple de Silverlight au profit de HTML 5 circulaient, et l'offensive de Microsoft sur HTML 5 n'arrangeait rien. Mais avec l'annonce de Silverlight 5, l'éditeur fait taire les critiques, même si sur la partie mobile il se limite à Windows Phone 7.

**S**ilverlight 5 apporte une quarantaine de nouveautés et se présente comme une version majeure avec plusieurs évolutions que l'on retrouve dans Flash – Flex. Très clairement, cette v5 se place contre la plate-forme d'Adobe. Sur l'agenda, une première bêta sera disponible d'ici mi-2011 pour une mise à disposition 2e semestre 2011, soit un peu plus d'un an après la v4. Les axes majeurs évoqués sont :

- décodage matériel de la vidéo : support intensif des GPU pour accélérer aussi bien l'affichage, l'animation que le décodage des flux vidéo, notamment haute définition avec toujours le même objectif : réduire l'occupation de la CPU et optimiser la qualité et les performances. En théorie, sur un netbook, on pourrait visionner une vidéo 1080p.
- Trickplay : cette fonction permet d'accélérer, de modifier la vitesse d'une vidéo, sans modifier la lecture des pistes audio.
- support de remote control : sympa, on pourra utiliser une télécommande.
- amélioration de l'affichage des textes, des animations et du modèle de développement
- support de Windows pour IIS Media Services.

## Un développement orienté entreprise

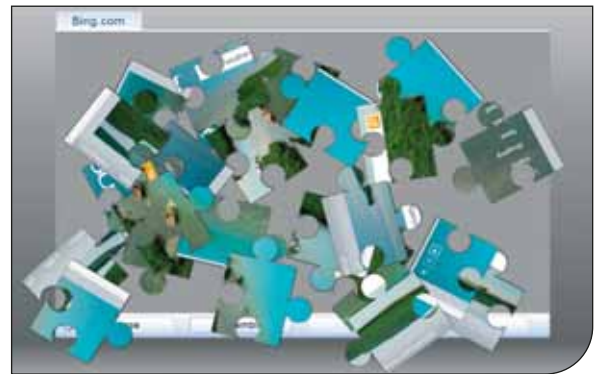
Bien que Silverlight s'appuie sur Visual Studio et la gamme Expression, le modèle de développement n'était pas toujours efficace et parfois trop lent. La v5 tente de corriger les défauts tout en introduisant des nouveautés non négligeables. Tout d'abord, le

data binding est renforcé notamment envers le MVVM. Ce rapprochement permettra de mieux intégrer les données dans Silverlight. Le développeur pourra faire du debug de data binding (enfin !), avoir des points d'arrêt, mieux traquer les erreurs. On bénéficiera aussi des « Implicit Data Templates » pour créer un support de données spécifiques. Pour les applications professionnelles, RIA Services et WCF sont complétés : support de WS-Trust, MVVM amélioré, pile réseau repackagée pour réduire le temps de latence (vital pour des applications nécessitant beaucoup de transactions et de trafic réseau). Il serait même possible de tester l'interface directement dans Visual Studio. Les tests étaient une des faiblesses de la plateforme.

Quelques autres nouveautés :

- multicolonne pour le texte
- composition des textes plus fluides et plus lisibles
- support du 64-bit
- l'accélération graphique reprend la couche introduite par IE9
- meilleur support HTML (sur le contenu) vers une application Silverlight
- interaction avec Microsoft Office (par exemple : lancer un programme Office)
- mêler code managé et non managé (typiquement C et C++) via Plinvoke

La partie graphique subit un lifting important avec l'accélération graphique par GPU pour l'affichage et pour la 3D. Le « out of Browser » (en dehors du navigateur), à l'instar d'un Adobe Air, profite de la v5 pour pouvoir afficher des fenêtres filles, utiliser Plinvoke pour les appels Win32.



Clairement, Silverlight 5 se veut une fusion entre Silverlight 4, Flash et Flex. Et Microsoft veut contrer Adobe en entreprise. Surtout, l'éditeur fait taire les doutes que l'on avait sur l'avenir de Silverlight depuis la dernière conférence PDC. Si l'éditeur veut mettre en avant HTML 5, ce n'est pas pour autant que Silverlight sera abandonné, les deux technologies sont, pour le moment, trop éloignées. HTML 5 ne peut remplacer Silverlight. Cependant, Silverlight aurait tout intérêt, tout comme Expression, à s'intégrer et à interagir avec HTML 5, car sur ce terrain, Adobe (et même Apple) veut aller vite ! Dans le même temps, Microsoft a dévoilé WCF RIA Service Toolkit décembre 2010. Cette version améliore l'installation (plus simple et s'appuyant sur Visual Studio Extension). Les bugs de génération T4 ont été corrigés. Un autre bug est corrigé avec la nouvelle version du Windows Azure SDK qui provoquait des arrêts de stockage (TableDomainService).

Pour en savoir plus :

<http://www.microsoft.com/silverlight/future/>

<http://weblogs.asp.net/scottgu/archive/2010/12/02/announcing-silverlight-5.aspx>

■ François Tonic



# Migrer de IE 8 à IE 9

L'année 2010 est indiscutablement un tournant important pour les produits Microsoft et Internet Explorer en est un excellent exemple. Il y a un an j'écrivais un article sur la compatibilité applicative pour Internet Explorer 8.0 qui montrait déjà les efforts en termes de standardisation, performance et sécurité établis avec cette mouture. Aujourd'hui nous sommes à quelques mois de la sortie d'Internet Explorer 9.0 qui a déjà marqué les esprits. Et ce n'est qu'un début !

**A**vec IE 9.0, nous avons clairement rattrapé notre retard et même pris de l'avance ! Le challenge était loin d'être évident avec un marché extrêmement concurrentiel et un retard non négligeable de notre côté. IE 9.0 est en version bêta depuis septembre, nous continuons de publier des mises à jour du moteur. À l'heure où est écrit cet article, nous en sommes à la preview 7. Pour rappel, nous publions ces previews toutes les 8 à 10 semaines depuis le mois d'avril 2010. Cette preview est d'autant plus importante qu'elle nous permet de prendre l'ascendant pour la première fois sur nos concurrents en matière de rapidité d'exécution des pages Web. Dans les lignes qui suivent, nous allons nous intéresser aux problématiques de compatibilité, aux outils à utiliser mais également aux nouveautés côté développeurs.

## Pourquoi parle-t-on de compatibilité applicative ?

Quand on développait des pages Web dans le passé et notamment à l'ère d'Internet Explorer 6.0, on développait du code propre à chaque navigateur du marché. Nous utilisions un ensemble de techniques qui permettaient d'optimiser le rendu de notre page pour IE, Firefox, Opera... L'idée de la standardisation est de pouvoir développer un code qui soit interprété par les moteurs de rendu des navigateurs de manière équivalente. Cette définition des standards relève du consortium web W3C où les grands acteurs du Web aujourd'hui comme Google, Mozilla et bien entendu Microsoft participent. Cependant si ce point tend à devenir réalité avec les dernières versions des navigateurs, à partir d'IE8 pour Microsoft, il faut continuer de gérer les migrations depuis IE6/IE7 vers IE8 et dans les mois à venir vers IE9. La migration depuis IE6 est très certainement le cas le plus compliqué à gérer mais aussi le plus fréquent. Nous avons mis en place de la documentation, des outils et des méthodes d'accompagnement de nos clients pour faciliter cette période de migration.

## Les normes respectées par Internet Explorer aujourd'hui et demain

Comme vous le savez le développement Web nécessite la connaissance de langages, protocoles, dans l'objectif d'obtenir les comportements souhaités pour une page Web.

Standards	Internet Explorer 8.0	Internet Explorer 9.0
HTML 4.01	Total	Total
CSS 2.1	Total	Total
ECMAScript 3	Total	Total
ECMAScript 5	Partiel	Total
HTML 5	Partiel	Total
CSS 3 Partiel Total SVG 1.1	-	Total

Microsoft a montré clairement sa stratégie vis-à-vis du HTML 5 lors de notre dernier événement PDC (Professional Developers Conference) en octobre. Nous investissons énormément dans les spécifications et l'implémentation du standard Web de demain ! Le HTML 5 est bien parti pour ouvrir une nouvelle ère du Web.

Pour vous en donner un avant-goût n'hésitez pas à tester les démonstrations mises en ligne avec votre bêta IE9 : <http://ie.microsoft.com/testdrive/Default.html> et <http://www.beautyoftheweb.com>.

## Le Challenge de la migration

Plusieurs cas de figure sont à prendre en compte. Comme vous vous en doutez, tout dépend de votre point de départ, il va clairement conditionner l'investissement à donner pour la période de migration. Le tableau ci-dessous traduit les situations les plus fréquentes.

Point de départ	Point d'arrivée	Difficulté	Points à prendre en compte
IE 6.0	IE 7.0	Modérée	Les commentaires conditionnels Version du Vector User Agent Respect partiel HTML 4 Respect partiel CSS 2 Améliorations HTTPS Mode protégé
IE 6.0	IE 8.0	Importante ++	Tous les points IE6 à IE7 Respect total 4.01 Respect total CSS2.1 ECMAScript 3 et partiel 5 DEP/NX, filter XSS Mode mixte http/HTTPS Sécurité AJAX renforcée Architecture LCIE
IE 6.0	IE 9.0	Importante ++	Tous les points IE6 à IE8 Evolutions User Agent Evolutions certains objets Evolution CSS Certains événements DOM ne sont plus d'actualité
IE 7.0	IE 8.0	Importante	Les commentaires conditionnels Version du Vector User Agent Respect total 4.01 Respect total CSS2.1 ECMAScript 3 et partiel 5 DEP/NX, filter XSS Mode mixte http/HTTPS Sécurité AJAX renforcée Architecture LCIE
IE 7.0	IE 9.0	Importante	Tous les points IE7 à IE8 Evolutions User Agent Evolutions certains objets Evolution CSS Certains événements DOM ne sont plus d'actualité
IE 8.0	IE 9.0	Modérée	Evolutions User Agent Evolutions certains objets Evolution CSS Certains événements DOM ne sont plus d'actualité



Ce tableau résume les points suivants :

- Migrer vers IE 9.0 sera une opération modérée si vous avez ou être en train de migrer vers IE8.0 : le travail de standardisation le plus important va se faire au passage vers IE8, les problématiques d'architecture (LCIE...) et de sécurité (mode protégé sous Vista/Win7...) sont équivalentes.
- Attendre la sortie d'IE9.0 pour migrer sur IE8.0 n'est pas forcément la bonne stratégie car les problèmes de compatibilité à fixer pour IE8.0 sont les mêmes pour IE9.0 et le delta IE8/IE9 est assez faible. Migrer vers IE8 actuellement constitue donc l'étape majeure de votre migration vers IE9.0 lorsqu'il sortira en version finale.
- Les problématiques de migration se résument en 4 points :
  - La gestion de version du navigateur (commentaires conditionnels, version Vectors, User Agent...)
  - Standardisation (HTML 4.01/5, CSS 2.1/3, ECMAScript 3/5...)
  - Sécurité (HTTPS/http, DEP/NX, filtre XSS, AJAX, ActiveX)
  - Architecture : LCIE

L'objectif de cet article n'est pas de détailler chacun de ces points mais de présenter les points les plus fréquents.

## Les problèmes les plus fréquents

### Les menus cassés

Ce cas arrive fréquemment et fait suite à la logique implémentée pour que le texte apparaisse devant le background du menu.

Dans ce cas, il est indispensable que l'item du menu puisse trouver la valeur du z-index de son background. Souvent on récupère la valeur initiale z-index de l'objet mais cela peut poser des problèmes. En effet, les anciens navigateurs vont fonctionner avec l'ancienne valeur mais les navigateurs standardisés aujourd'hui vont utiliser la valeur initiale. Dans le premier exemple, on ne vérifie que l'ancienne valeur, le code corrigé permet de prendre en compte la valeur « auto » utilisée par les navigateurs standardisés.

Avant	Après
<pre>If(zIndex == 0){   // Code }</pre>	<pre>} If(zIndex == 0    zIndex== « auto »){   // Code }</pre>

### Centrer une page

Dans les versions précédentes IE8, il était possible d'utiliser la propriété text-align pour centrer une page. Cependant cette méthode n'est pas respectueuse des standards dans le sens où les éléments DIV et TABLE sont considérés comme des niveaux de bloc et ne prennent plus en compte la propriété text-align spécifiée à un niveau hiérarchique supérieur comme BODY. Une méthode de résolution consiste à initialiser la propriété margin à auto.

### Les filtres CSS

Cette méthode était utilisée pour spécifier des règles/comportements pour Internet Explorer. On utilisait les filtres CSS notamment pour régler des problèmes de standardisation quand on développait des pages pour IE6 et Firefox. Exemple :

```
*html.myDiv{
  background-color :blue ;
}

<div class="myDiv">Test</div>
```

Dans l'exemple ci-dessus, sous Internet Explorer 6.0, les objets dont la classe de style a pour valeur *myDiv* auront une couleur de fond bleue et cette propriété sera tout simplement ignorée pour les navigateurs alternatifs et à partir d'IE7. Cela peut donc avoir d'importants impacts lors de la migration, notamment quand on imagine que ces propriétés CSS peuvent se concentrer sur le positionnement d'éléments et pas seulement de la couleur de fond.

## La fonction getElementById

S'il existe bien une fonction couramment utilisée en JavaScript pour procéder à des actions sur des éléments du DOM d'une page c'est bien la fonction *getElementById(objectId)*.

Cette fonction est définie selon les standards W3C comme étant sensible à la casse, ce qui n'était pas le cas dans le passé. Ainsi, si l'id passé en paramètre ne correspond pas scrupuleusement à l'id d'un objet du DOM de la page courante, vous pourriez bien voir des erreurs JavaScript apparaître, des comportements inattendus comme un menu qui ne fonctionne plus.

## Anomalie de superposition

Internet Explorer dans sa version 6.0 était plus permissif et faisait preuve de pro-activité pour interpréter des comportements mal ou non spécifiés par les développeurs. Pour standardiser IE, ce genre d'interprétation a tout simplement été abandonnée, le navigateur suit avec précision ce qui a été codé par les développeurs.

Ainsi une erreur bien connue est la définition de la largeur/hauteur d'un panel par rapport à son contenu. Le W3C spécifie que les propriétés appliquées au parent doivent tenir compte du contenu, qu'il ne doit pas se modifier automatiquement par rapport à ce même contenu. Si le contenu dépasse donc la largeur spécifiée du parent, il dépassera. Sous IE6, ce parent était redimensionné automatiquement par le moteur de rendu.

## Les outils

### L'outil de développement intégré

Afin de faciliter le travail du développeur Web, un outil de développement a été intégré à partir d'IE8. Ce même outil continue de bien évoluer sous IE9 :

- Manipulation DOM
- Exploration HTML détaillée et intuitive d'une page
- Exploration du CSS (Cascading Style Sheet)
- Debug et manipulation JavaScript
- Profiling d'une page

Ces outils sont parfaitement intégrés au produit et il n'est nullement nécessaire d'installer un complément, une mise à jour pour les utiliser. Une fois que vous commencerez à les utiliser, il n'est même plus envisageable de s'en débarrasser tellement ils vous apporteront une flexibilité et un réel appui dans vos développements ! On comprend très facilement l'intérêt qu'ils vont jouer également pour nous assister pendant la migration.

Comment y accéder ?

#### Méthode 1

Un simple raccourci avec la touche F12 dans IE8/IE9

#### Méthode 2

Depuis le menu, cliquez sur **Outils** puis **Outils de développement [Fig.1 et 2]**

Depuis cette nouvelle page, vous avez accès à l'ensemble des outils pour les développeurs Web.



## Onglet HTML

L'onglet HTML vous permettra d'identifier le code HTML de la page en cours. Son principal atout n'est pas de nous montrer basiquement le code HTML que l'on pourrait très bien obtenir simplement avec un clic droit **Voir source** mais bien d'inspecter la page. Par exemple, si vous vous déplacez au sein de la hiérarchie HTML, en cliquant sur un élément, par exemple une table, vous obtiendrez l'ensemble des détails sur cet élément. Ils se trouvent sur la partie droite de la fenêtre de l'outil de développement. Notez bien que les outils ci-dessous ne se concentrent pas que sur de la visualisation mais permettent d'interagir avec la page. Si vous décidez de décocher une case d'une propriété CSS, vous verrez le résultat sur la page immédiatement. Que demander de plus pour trouver la propriété CSS à l'origine d'un problème de positionnement ou d'affichage ?

### Styles

Styles permet d'identifier les règles qui sont appliquées à l'élément sélectionné dans le code HTML. A noter que l'ordre des règles respecte l'ordre dans lequel elles sont appliquées. Pratique pour déboguer, notamment lors de surcharge CSS. [Fig.3]

### Suivre les styles

Cet outil s'adresse évidemment aux styles CSS à la différence de **Styles**, l'outil classe les règles CSS par propriété. Par exemple vous recherchez toutes les occurrences sur la hauteur, il sera préférable d'utiliser cet outil. [Fig.4]

### Disposition

L'outil de disposition se focalise sur le positionnement des éléments. Par exemple, vous ne comprenez pas pourquoi votre padding ne s'applique pas, utilisez l'outil

**Disposition** sur votre élément. [Fig.5]

### Attributs

Cet outil vous permet de définir des attributs pour l'élément sélectionné. Très pratique lorsque vous voulez simuler un comportement dans votre page. Vous pouvez bien entendu en ajouter, supprimer ou modifier. [Fig.6]

## L'onglet CSS dresse l'ensemble des propriétés CSS utilisées par la page

L'onglet Script permet d'inspecter l'ensemble des fonctions qui sont employées dans la page. Une fonction intéressante sera de pouvoir lancer le débogage d'une page au sein du code JavaScript. On retrouvera l'ensemble des fonctionnalités propres à du débogage comme l'ordonnancement via des points d'arrêts, la possibilité de contrôler les actions lecture/pause, vérifier les valeurs de variables à un état précis... De la même manière que les outils de l'onglet HTML vus précédemment, vous pouvez agir directement sur les styles de la page en cochant/décochant les cases propres à chaque propriété CSS [Fig.7].

## L'onglet Script

L'outil Script vous aidera à déboguer le code JavaScript de votre page. En tant que développeur Web, vous connaissez toute la complexité de déboguer du JavaScript ! Cet outil permet de :

- Définir des points d'arrêts pour contrôler l'exécution de vos scripts
- L'affichage du Console notamment pour l'affichage d'information durant l'exécution. A noter que les messages d'erreur sont logués dans le fichier console.log
- L'affichage et la gestion des variables locales pour vérifier si telle ou telle variable a bien telle ou telle valeur à un moment T
- Afficher les fonctions en cours d'appel. Cela permet de voir quelles sont les fonctions et le code associé en cours d'exécution.

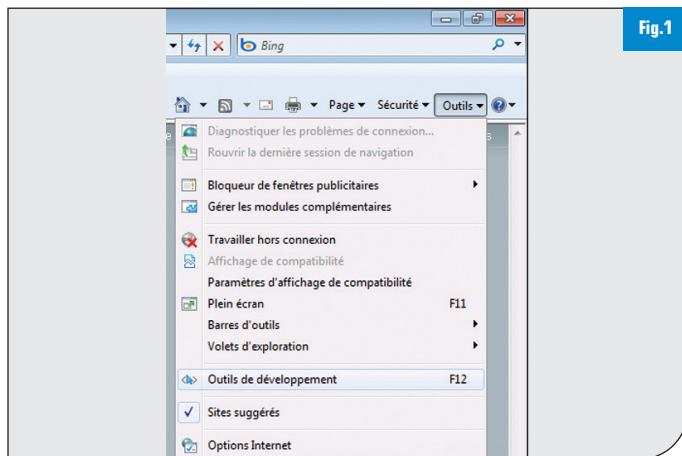


Fig.1

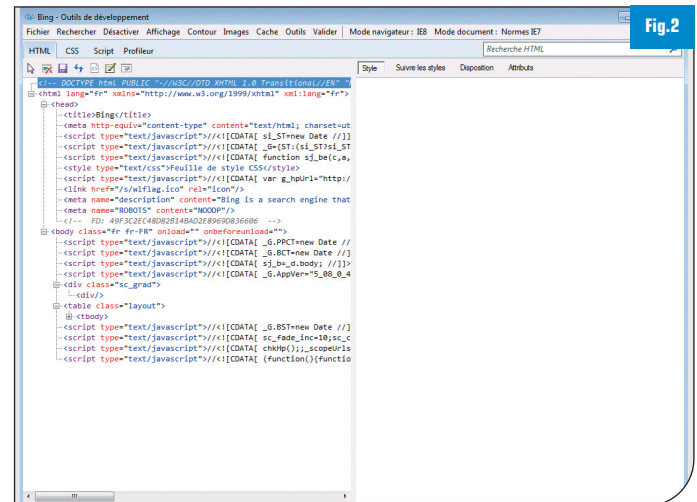


Fig.2

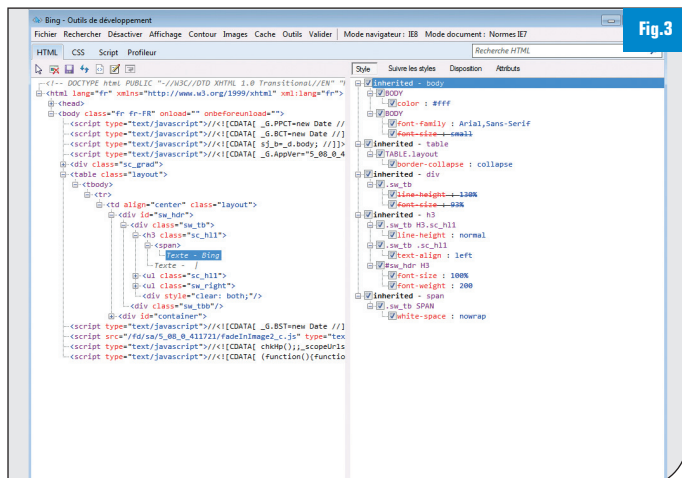


Fig.3

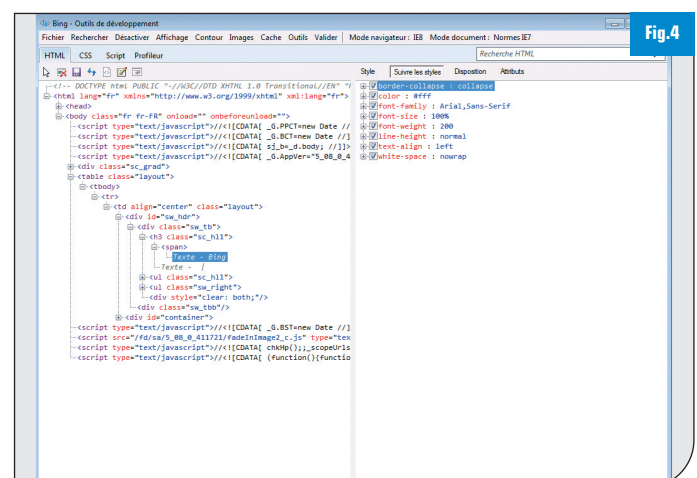


Fig.4



## L'onglet Profilleur

Un outil intéressant pour améliorer votre page Web. Le profileur va vous permettre d'identifier rapidement les fonctions ou bouts de code qui sont les plus consommateurs au sein d'une page. Depuis le profileur vous avez un lien direct vers la fonction et la ligne concernée. Le profileur vous donne la possibilité de gérer plusieurs sessions notamment avec la notion de rapport. Vos rapports peuvent être également exportés en format CSV. Vos exports peuvent ensuite vous aider à consolider vos résultats et fournir des rapports complets d'analyse de vos sites Web.

## Le sélecteur

Le sélecteur est un mode de l'outil de développement qui vous permet de sélectionner n'importe quel élément de manière visuelle directement sur la page Web. Une fois sélectionné, l'outil de développement actualise ses données et vous positionne dans la portion de code HTML concernée. C'est un mode réellement intéressant pour avoir accès rapidement au code correspondant. Je me rappelle personnellement de mes débuts dans le développement Web il y a une dizaine d'années où nous devions faire nous-mêmes la corrélation pas à pas entre ce qu'on avait dans l'aperçu et le code HTML qui peut être parfois réellement complexe. Dans l'illustration [Fig.8], on a utilisé le sélecteur sur la boîte de texte pour saisir nos mots-clés de recherche sous Bing. Automatiquement les définitions HTML et CSS correspondantes sont apparues.

## Gestion des liens

Quand on travaille sur le développement d'un site Web et en particulier de portails, un des éléments fondamentaux et utilisé N fois est le

lien hypertexte. L'outil de développement permet de remplacer le contenu texte des liens par les URL. Cela peut présenter un vif intérêt lorsque l'on souhaite vérifier les liens sans devoir regarder les sources ou sans passer la souris dessus.

Une autre fonctionnalité fournie qui permet de compléter cette approche est le rapport de liens [Fig.9].

Une page s'affiche et répertorie l'ensemble des liens de la page en cours. On peut ainsi vérifier très rapidement la cohérence texte affiché/URL ou toute autre correspondance. Ces fonctionnalités sur les liens sont accessibles depuis le menu Affichage :

- Chemins de lien
- Rapport de lien

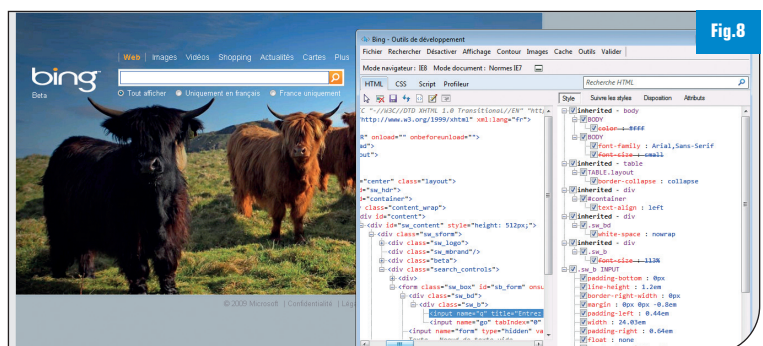
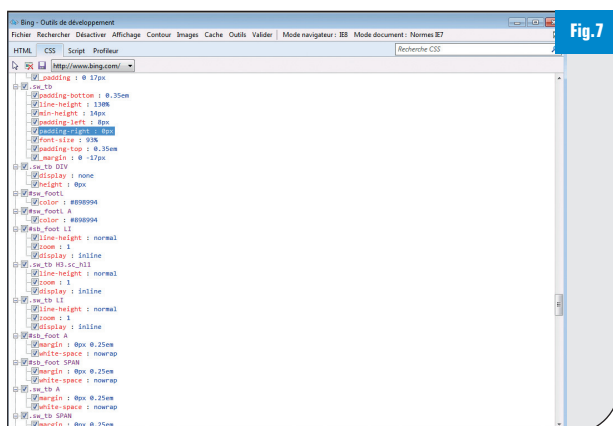
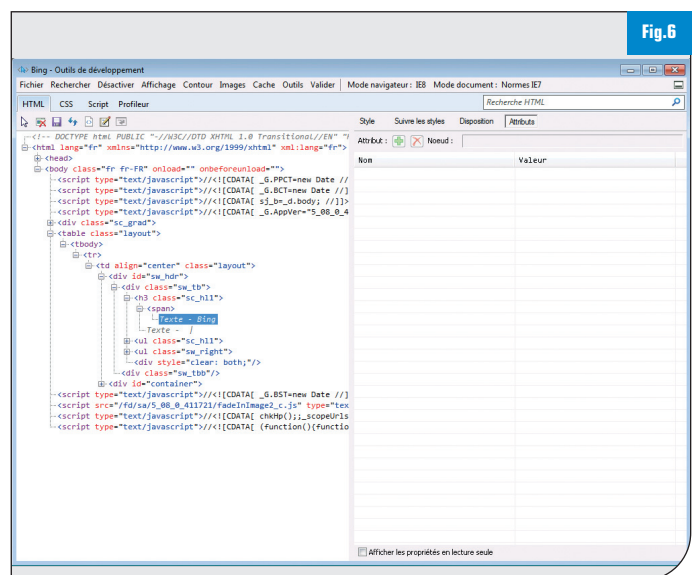
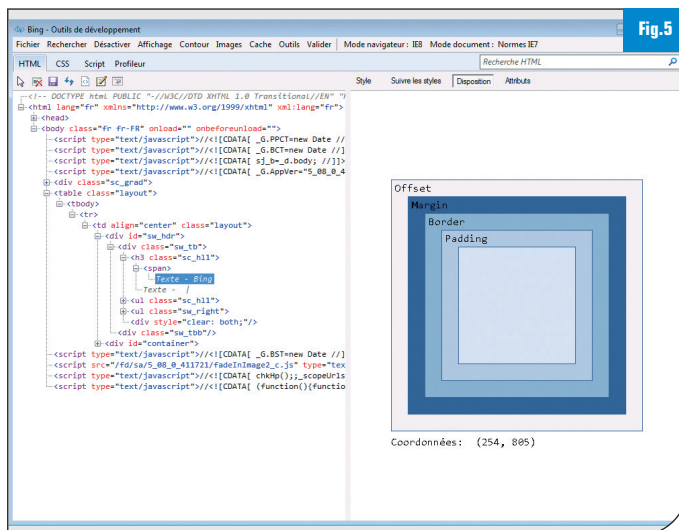
## Architecture de la page

L'architecture de la page peut être visuellement identifiée par de la coloration du contenu. En effet, on peut par exemple, afficher le contour des tableaux ou des DIV de la page. On voit une illustration de cette fonctionnalité [Fig.10].

Les éléments dont le contour est vert correspondent aux différents DIV qui architecturent la page.

En plus de l'architecture visuelle, on a bien entendu accès à la structuration DOM de la page. Ces différentes options sont disponibles depuis les menus **Affichage** et **Contour** :

- Structuration DOM : **Affichage** > **Source** > ...
- Structuration des éléments : **Contour** >
  - Tableaux
  - DIV
  - Images ...





## Gestion du cache et des cookies

Le cache est un composant très important des serveurs Web mais également des clients et donc des navigateurs internet.

Le cache est de plus en plus utilisé dans les applications Web d'aujourd'hui et permet d'optimiser les performances d'affichage des pages en stockant des pages, éléments de pages localement afin de ne pas devoir les recharger à chaque affichage.

Vous vous en doutez, l'inconvénient majeur du cache pour un développeur peut être de maintenir des développements dans un état qui n'est pas l'état réel. En effet, il arrive souvent qu'on ne comprenne pas qu'un développement ne soit pas pris en compte, très souvent le cache se trouve être le « coupable » de ce genre de comportement. L'outil de développement IE8 permet de contrôler le cache et notamment de le vider à la demande. Pour rappel, une combinaison souvent efficace est également d'utiliser Ctrl+F5.

L'avantage de l'outil est de pouvoir ajouter une notion de granularité. En effet on peut gérer le cache du navigateur entièrement ou pour un domaine donné selon la page où on se trouve. Nous retrouvons exactement la même notion sur la suppression des Cookies.

Ces fonctionnalités sont accessibles depuis le menu **Cache**

Pour conclure, une fonctionnalité intéressante est de pouvoir afficher les cookies qui dépendent de la page en cours. On peut ainsi vérifier la date d'expiration, son nom, sa valeur, le domaine d'application et son chemin via un rapport global (voir ci-dessous). Un rapport s'affichera dans une nouvelle page HTML, ce n'est pas sans rappeler le rapport de liens vu précédemment [Fig.11].

## Validation & conformité

Pour conclure sur l'outil de développement, nous allons voir les fonctions de validation fournies. Comme vous le savez, on peut développer une page Web de façon très différente tout en ayant un résultat visuel relativement similaire. Cependant, le HTML permet cette flexibilité mais si on souhaite avoir des pages visibles de la même façon, par un plus grand nombre, le principe de base est de se conformer à certaines règles afin d'obtenir des pages conformes aux règles de développement Web, établies notamment par le W3C.

Pour faciliter ce travail de mise en conformité, on peut valider sa page voire son flux de données via l'outil de développement.

On peut choisir son niveau de granularité sur la validation :

- HTML : respect des normes HTML

- CSS : respect des normes CSS
- Flux : vérifier si la définition du flux est cohérente
- Liens : vérifier si les liens existent bien
- Tous (validations multiples)

Par défaut, nous utilisons les outils de validation du W3C ou FeedValidator mis à la disposition des développeurs.

## Les nouveautés de IE9

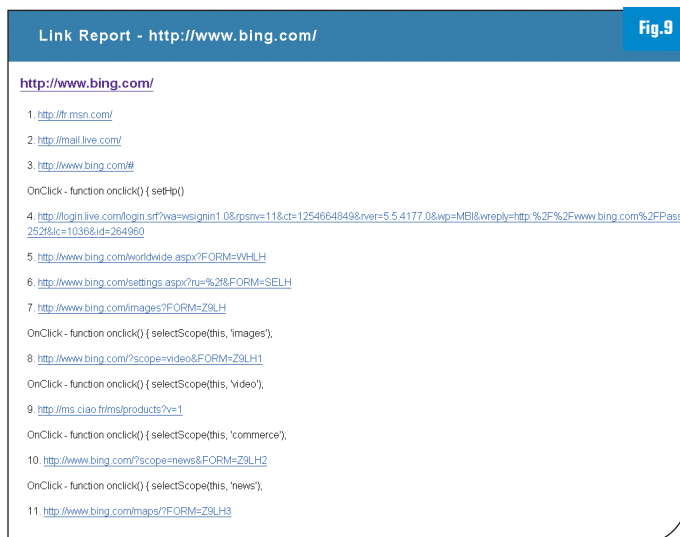
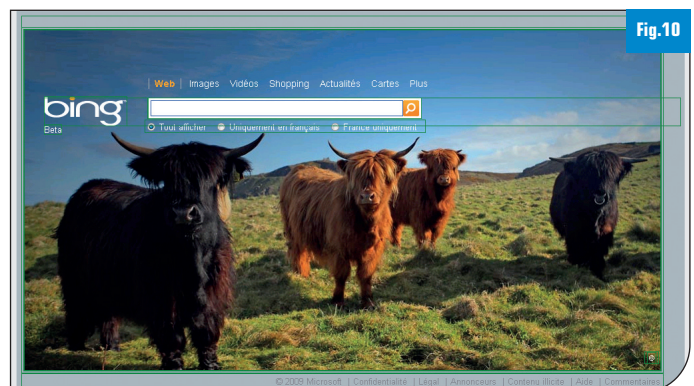
Et oui l'outil continue d'évoluer avec notre nouvelle mouture, bien évidemment en prenant en charge les nouveaux standards tels que définis à ce jour comme HTML5/CSS3/ECMAScript5 mais aussi en rajoutant son lot de fonctionnalités.

Un nouvel onglet **réseau** fait aussi son apparition. Son objectif est simple : capturer le trafic http entre le client et les serveurs. Cet outil intégré qui rapporte des informations similaires à ce que l'on peut faire avec un outil comme Fiddler permet de :

- Décomposer les différentes requêtes http sous-jacentes
- Analyser les résultats des réponses des serveurs Web
- Analyser les tailles et temps de réponse pour chacune de ces requêtes
- Avoir une vue analytique des temps à la fois
  - Pour estimer leur ordre d'exécution
  - La répartition du temps de chacune de ces requêtes [Fig.12]

Nous allons pouvoir aller plus loin en double cliquant sur une requête, on accède immédiatement à l'ensemble des détails de cette requête :

- Détails du timing : début/attente/chargement...



NAME		SRCHD
VALUE		D=925312&MS=925312&AF=NOFORM
DOMAIN		bing.com
PATH		/
EXPIRES		04/10/2011 15:52:16

NAME		SRCHUSR
VALUE		AUTOREDIR=0&GEOVAR=&DOB=20091004
DOMAIN		bing.com
PATH		/
EXPIRES		04/10/2011 15:52:16

NAME		RMS
VALUE		T=272384
DOMAIN		.com
PATH		/
EXPIRES		At the end of the Session

NAME		_FS
VALUE		mkt=fr-FR&ui=fr-FR
DOMAIN		.com
PATH		/
EXPIRES		At the end of the Session

NAME		OVR
VALUE		fl=0&DomainVertical=0&Cashback=cbtest4&MSCorp=kievfinal&GeoPerf=0
DOMAIN		bing.com
PATH		/
EXPIRES		At the end of the Session



- ## Moniteur de Processus – ProcMon

Les cas où cet outil nous sera d'une grande utilité est lorsqu'on embarque des composants avec un fonctionnement en boîte noire, typiquement des ActiveX tiers ou BHO. Ces composants peuvent présenter des problèmes lors d'une migration sous Windows 7/IE8-9 avec les évolutions de sécurité et notamment le mode protégé qui a pour objectif d'isoler le navigateur du noyau du système d'exploitation.

## Modes de compatibilité

Ce type de problématique est très important lors d'une migration du poste de travail. Imaginez que des applications Web sont fondamentales au business d'une entreprise avec plusieurs milliers de postes clients...on peut se trouver très rapidement freiné par une migration, quand bien même elle serait envisagée !

Les cas typiques de problèmes rencontrés lors de migration vers IE8 sont clairement autour du développement CSS et JavaScript. Le navigateur étant moins permissif que dans le passé, on peut avoir des comportements inattendus et notamment sur l'affichage du contenu. Ce genre de problèmes couvre tout de même au moins 70 à 80% de ceux rencontrés à ce jour. Clairement ce ne sont pas la plupart du temps des problèmes critiques mais les modes de compatibilité seront là pour vous aider dans votre démarche.

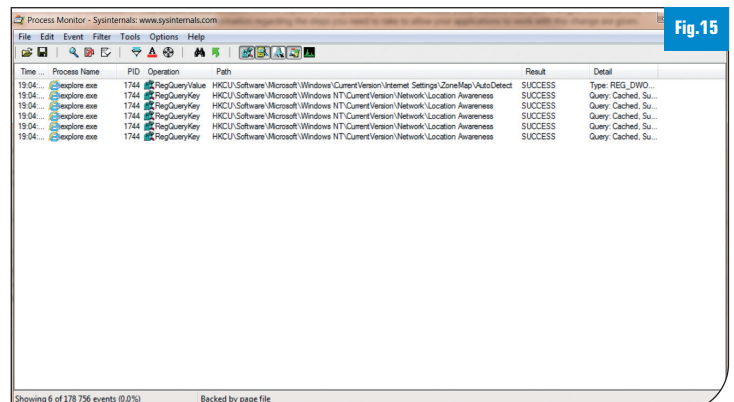
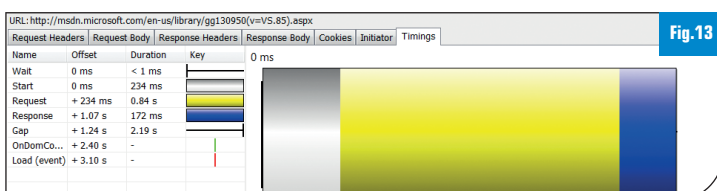
- Quirks
- Standard

Avec IE9, les possibilités sont :

- Mode quirks – IE 5.5
- Mode IE7 standards
- Mode IE8 standards
- Mode IE9 standards

La première méthode consiste à indiquer au navigateur que telle page devra utiliser tel mode. Cela est rendu possible par l'utilisation d'une balise meta bien particulière.

URL: http://msdn.microsoft.com/en-us/library/gg130950(v=VS.85).aspx									
	Request Header	Response Body	Response Headers	Cookies	Initiator	Timings	<b>Fig. 14</b>		
Direction	Key	Value		Expires	Domain	Path	Sec.	HTTP O...	
Sent	ToCHas..	ms310241(n)/aa155073(n)/aa737439(n)/gg130953(VS.85)							
Sent	WT_IVR	0=:1=en-us fr-fr;2=en-us/library;3=fr-fr/subscriptions/...							
Sent	BFFC	:FR;							
Sent	s_cc	true							
Sent	s_sq	mstomsdtnr2Cmsdstomdsnr9o2Cmsdstomdsnmktenus...							
Sent	ADS	SN=17SA21EF							
Sent	MC1	GUID=6ba3637d9ddc24c8177dc1e8bee063f8HASH=7d6...							
Sent	A	tBj=AxUJAFAAAAA+CbAwwcrzD1GSfwKQSm3GpwffHl8...							
Sent	MUID	BFCC728AD384C4BEDEF70CDDA773C							
Sent	WT_FPC	id=94.245.127.13-3448709072.31010743-i=129151508...							
Sent	_unam	289c965-12be547c348-4b93c411-1							
Sent	omnidID	do40base_28b4_4afc_e3a6_9eed996ae5ec							
Sent	WT_NV...	0=msdn technet:1:=2=							
Sent	msdn	L=1033							





La balise meta doit être placée dans le header de la page et avant les autres éléments, exception faite pour title et les autres balises meta. La fonction JavaScript document.documentMode permet d'afficher la version du navigateur utilisée.

Internet Explorer 5.0 – Mode de compatibilité

```
<meta http-equiv="X-UA-Compatible" content="IE=5">
```

Internet Explorer 7.0 – Mode de compatibilité

```
<meta http-equiv="X-UA-Compatible"
content="IE=7.5">
```

Internet Explorer 8.0 – Mode de compatibilité

```
<meta http-equiv="X-UA-Compatible" content="IE=100">
```

Préciser à IE8 de prendre le mode de compatibilité le plus récent

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
```

Internet Explorer 7.0 - Emulation

```
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7">
```

A noter que si vous définissez plusieurs meta *X-UA-Compatible*, seule la première balise sera interprétée. Cependant, si vous souhaitez rendre vos sites disponibles avec plusieurs modes de compatibilité, il vous suffira de les spécifier en les séparant par un point-virgule :

```
<meta http-equiv="X-UA-Compatible" content="IE=7.5; IE=8" >
```

La question suivante est bien entendu : comment définir un mode de compatibilité suivi par N sites, N applications, c'est trop compliqué, trop impactant et cela prend trop de temps de rajouter ce genre de code dans chacune des pages ! Et bien on va le définir au niveau du serveur, on peut le faire sur un serveur Apache comme IIS. Je vais prendre ici l'exemple IIS. Le principe et l'implémentation sont relativement simples puisqu'il ne vous faudra définir qu'un header particulier. En reprenant l'exemple précédent pour qu'IE8 émule les sites avec IE7 :

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <clear />
        <add name="X-UA-Compatible" value="IE=EmulateIE7" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

Pour définir une réponse http personnalisée dans IIS, il vous suffit de créer une nouvelle réponse http avec comme nom « X-UA-Compatible » et pour valeur le mode de compatibilité voulu (ex : « IE=EmulateIE7 »).

Que se passe-t-il si je ne précise pas de balise *meta* ?

Internet Explorer 8.0 va alors rechercher le DOCTYPE de la page afin d'identifier si une information sur le mode de rendu a été définie.



Si ce n'est pas le cas, la page sera automatiquement en mode quirks (IE5). Si la balise DOCTYPE spécifie un document standard, alors le navigateur interprétera le rendu avec son moteur de rendu. A noter que quelques exceptions existent sur ce comportement comme la définition de l'**affichage de compatibilité** qui surcharge la définition du DOCTYPE [Fig.17].

Enfin une dernière méthode consiste à modifier une clé de registre pour spécifier un comportement global d'IE8 sur un poste de travail.

```
HKEY_LOCAL_MACHINE (or HKEY_CURRENT_USER)
SOFTWARE
Microsoft
Internet Explorer
Main FeatureControl
FEATURE_BROWSER_EMULATION

iexplore.exe = (DWORD)
```

Comme on a pu le voir, Internet Explorer met à notre disposition un ensemble de modes de compatibilité qui nous accompagneront dans nos migrations progressives. C'est également intéressant pour comparer les affichages et donc les interprétations CSS selon le mode choisi.

## Liens

Plus d'information sur le debug avec l'outil de développement (en Anglais) :

[http://msdn.microsoft/en-US/library/dd565625\(VS.85\).aspx](http://msdn.microsoft/en-US/library/dd565625(VS.85).aspx)

Evolutions du DOM avec IE9 :

[http://msdn.microsoft.com/en-us/library/ff986083\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff986083(v=VS.85).aspx)

■ Vincent **Bellet** - Spécialiste Solutions Applicatives  
Microsoft France - [vincent.bellet@microsoft.com](mailto:vincent.bellet@microsoft.com)

# L'information permanente

• L'actu de Programmez.com : le fil d'info quotidien

• La newsletter hebdo : la synthèse des informations indispensables.

Abonnez-vous, c'est gratuit !

[www.programmez.com](http://www.programmez.com)



# egilia<sup>®</sup>

## LEARNING

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

Faire de vos succès  
notre réussite

[www.egilia.com](http://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 .

# Java 7 : en



Que le chemin fut laborieux !

Maintes fois reporté, maintes fois modifié, Java 7 pointe enfin son nez, et officiellement. Début décembre, la Java Community Process (l'organisation qui gouverne Java et les nouvelles spécifications de l'ensemble de la plateforme) a voté en faveur des fonctions et JSR (les spécifications Java) de la 7<sup>e</sup> version majeure de Java et même de la v8 prévue en 2012 (si tout va bien).





# fin le bout du tunnel !

**D**ans Programmez!, nous vous relatons depuis plus d'un an les rumeurs, les revirements, les nouvelles fonctions finalement retirées, Java 7 qui se dédouble en Java 7 et Java 8, etc. Oracle avait entamé un début de clarification en septembre dernier lors de la JavaOne en annonçant deux versions de Java pour les 2 à 3 prochaines années : Java 7 et Java 8.

Il ne manquait plus que le vote de la gouvernance de Java, la JCP. C'est chose faite. En réalité, la JCP a voté et approuvé 4 JSR importantes :

- JSR 334 : projet Coin pour les améliorations du langage
- JSR 335 : projet Lambda pour les expressions lambda dans Java
- JSR 336 : Java SE 7
- JSR 337 : Java SE 8.

Oracle n'a pas manqué dès le 7 décembre de faire état du vote de la JCP et de préciser qu'OpenSDK proposait les dernières pré-versions de la JDK 7... Et de rappeler que l'éditeur proposait OpenSDK pour Windows, Linux, Solaris et très bientôt pour MacOS X suite à l'arrivée d'Apple dans le projet open source...

## La fondation Apache part !

En même temps, la rumeur disait donc vrai sur la menace de retrait du JCP de la Fondation Apache qui avait posé une exigence : ne pas voter pour l'acceptation de Java SE 7 et obtenir une licence pour l'outil de tests TCK.

C'est finalement la conclusion de plusieurs années de conflits, à l'époque Sun était encore une société indépendante.

En cause, le projet Harmony, l'implémentation de la fondation Apache de Java. Or, pour être une implémentation officielle, il faut obtenir une licence Java et passer les tests TCK. Sur fond de conflits internes au JCP.

Bien avant le rachat par Oracle de Sun, JCP était parfois accusé d'être peu ouvert. Et Sun tentait de proposer une version de Java payante mais avait dû reculer devant le tollé de la communauté.

Aujourd'hui, la fondation Apache dit que les droits aux "implémenteurs" ne sont pas respectés et que l'acceptation de la licence Oracle de TCK pour Java SE 7 casse l'intégrité de la licence de la JCP... Et on parle en même temps de la mainmise trop grande d'Oracle sur le JCP. Depuis, Oracle souhaite qu'Apache reconsidère sa position et revienne.

C'est l'avenir même du projet Harmony qui est en jeu car avec les revers encaissés depuis quelques mois, ce projet Java a-t-il encore un réel avenir ? Oracle veut, plus que jamais, garder la main, le récent accord avec IBM, l'autre grand contributeur de Java, en est la preuve. Reste à voir comment évoluera la plainte en justice d'Oracle contre Android (Google)...

## Source :

[https://blogs.apache.org/foundation/entry/the\\_asf\\_resigns\\_from\\_the](https://blogs.apache.org/foundation/entry/the_asf_resigns_from_the)

## Il faut bouger !

Rendez-vous en été 2011 pour la version finale de Java 7. En attendant, nous faisons le point sur les nouveautés, les fonctions, les améliorations.

■ François Tonic



# Java 7 retour sur le futur



Annoncée il y a bientôt 4 ans, la première version entièrement Open Source du JDK, plus connue sous le nom de OpenJDK, peine à voir le jour. La faute aux nombreuses batailles que se livrent les géants gravitant autour de l'univers Java, mais également aux attermoissements passés de Sun. Au milieu de cet univers tumultueux, Oracle, qui a apparemment décidé de reprendre les choses en main, a fait un certain nombre d'annonces en septembre dernier qui semblent éclaircir la situation de Java 7 ... Enfin ! Tour d'horizon et perspectives.

**L**ancées il y a bientôt 4 ans, les discussions autour de Java 7 et d'OpenJDK n'ont jamais permis à la communauté Java de se mettre d'accord sur le contenu de cette nouvelle mouture. Une modularisation du JDK via le projet Jigsaw, ou via la mise en place d'une implémentation OSGI ? Une API pour la gestion des dates basée sur Joda-Time ? (la fameuse JSR-310 ...) Le projet Coin et ses simplifications de la syntaxe de Java afin de le rendre plus moderne et plus simple, à l'image de ce qui se fait dans les nombreux langages de script de l'écosystème Java ?

## Une longue histoire

Tout ces sujets n'ont eu de cesse d'agiter la communauté depuis fin 2006, mais il faut bien le dire, ce n'est rien comparé à tout le patatouille généré par l'introduction des closures au sein du langage Java ! Plusieurs implémentations différentes ont été proposées par des leaders d'opinion de la communauté et chacune d'entre elles ont eu leurs partisans et leurs défenseurs, farouches quelquefois même. Au final, le projet Lambda a bien été mis sur pied par Sun à l'époque afin de préparer une implémentation des closures au sein de Java qui reprendrait plus ou moins ce qui ressortirait des discussions et des propositions techniques de la communauté. Force est de constater que dans le cas présent, on serait plutôt dans le « moins », tant le projet Lambda ne fait pas l'unanimité autour de lui !

C'est dans ce climat déjà tendu que Sun, en difficulté depuis un certain temps déjà, a été racheté par Oracle au milieu de l'année 2009 pour plus de 7 milliards de dollars.

Ce rachat a tout de suite déclenché les haros de la communauté Java et plus largement de la communauté Open Source. Ces dernières voyaient d'un mauvais œil ce rachat craignant fortement pour les projets Open Source portés par Sun tels que MySQL, OpenSolaris ou OpenOffice.org et donc pour le futur OpenJDK 7. Un délit de "sale gueule" en quelque sorte, puisque même si Oracle n'est pas connu pour faire de la philanthropie, il contribue au monde Open Source et ses apports récents ne sont pas négligeables. On peut ainsi citer les projets Berkeley DB ou EclipseLink (anciennement Toplink) par exemple.

Tout ceci n'annonçait rien de bon pour un Java 7, qui en s'appuyant sur le projet Open JDK, devait sortir pour la fin de l'année 2009 ! Les investigations menées par les USA et la Communauté Euro-

péenne pour valider ce rachat n'ont évidemment rien arrangé, puisque Bruxelles n'a définitivement entériné ce dernier qu'au début de l'année 2010. Dès lors, Oracle a commencé à effectuer un certain nombre d'annonces soufflant le chaud quelquefois, en confirmant que le serveur d'applications GlassFish était un projet stratégique au même titre que son produit existant, Weblogic par exemple ; mais le plus souvent en soufflant le froid en annonçant par exemple l'arrêt de son support au projet OpenSolaris en Août dernier, ou en attaquant en justice Google pour violation de brevets concernant Android et sa machine virtuelle Dalvik plus particulièrement ...

La situation au mois d'Août n'était guère encourageante pour Java 7 ... Surtout que le calendrier d'Open JDK redéfini 1 an auparavant prévoyait une sortie au cours du dernier semestre 2010, ce qui apparaissait clairement comme impossible depuis quelques mois déjà pour la plupart des observateurs de la communauté. Le mois de Septembre est arrivé et avec lui la grand-messe annuelle de la communauté Java à savoir JavaOne. Pour la première d'Oracle, elle a été accolée à Oracle OpenWorld, la conférence annuelle de la société de Larry Ellison.

Boycottée par Google et ses ingénieurs vedettes, la conférence JavaOne aura eu le mérite de voir Oracle préciser enfin ses plans pour Java 7 et définir une roadmap réaliste divisée en 2 parties.

La première partie détaille les fonctionnalités probables pour Java SE 7 ainsi qu'une date de sortie à mi-2011 alors que la seconde présentait des propositions faites par Oracle pour le contenu de Java SE 8 avec une sortie prévue en fin d'année 2012.

De ces annonces, on peut ainsi comprendre qu'Oracle compte bien continuer à sortir un Java standard labellisé Oracle, puisque très rapidement les JSR portant ces fonctionnalités devraient être soumises au JCP, selon les dires de Mark Reinhold, architecte en charge de Java chez Oracle.

En outre, on s'aperçoit également que beaucoup de ces fonctionnalités sont déjà en très bonne voie, voire finalisées et qu'il ne reste plus grand chose à faire si ce n'est corriger quelques bugs et définir les JSR correspondantes pour les intégrer dans Java SE 7.

C'est cela qui nous fait dire que le calendrier annoncé par Oracle est réaliste cette fois. Une pré-version devrait être sortie quand vous lirez ces lignes.

## Support des langages à typage dynamique

Commençons notre tour d'horizon par la JSR 292 qui doit amener le support des langages à typage dynamique au sein de Java. Actuellement, le bytecode Java supporte seulement 3 modes d'invocation de méthodes : *invokestatic*, *invokespecial*, *invokeinterface* ou *invokevirtual*. Ces derniers permettent uniquement d'appeler une méthode dont la signature est connue à la compilation. Java étant un langage fortement typé, il pouvait paraître logique à l'origine que le bytecode produit par le compilateur javac ne supporte pas l'invocation dynamique de méthodes.

Cependant, la donne a changé depuis et le bytecode Java n'est plus exclusivement généré à partir de programmes écrits en Java. Ainsi, de nombreux langages de script génèrent du bytecode Java à partir de leur syntaxe propre parmi lesquels Groovy, Scala, Clojure ou JRuby pour ne citer que les plus célèbres. Ces langages montent en puissance depuis quelques années maintenant et ils arrivent à un point où ils commencent à être bien intégrés en entreprise. Malgré cela, aucun changement majeur au sein du bytecode et de la JVM n'avait été fait pour faciliter leur intégration, même s'il est vrai que la JSR 223 permettant l'invocation de langages dynamiques avait été un premier pas allant dans le bon sens.

De ce fait, les ingénieurs en charge de ces langages avaient dû redoubler d'ingéniosité pour pouvoir produire du bytecode Java compatible en faisant un usage outrancier du type *Object* car étant typés dynamiquement, ils ne peuvent pas connaître à l'avance le type de chaque objet. En « contournant » les restrictions de Java par ces techniques, ils perdent ainsi les avantages amenés par la JVM et notamment sa capacité à optimiser le code à l'exécution, ce qui induit forcément des performances moins bonnes.

En s'intéressant de plus près au bytecode Java, on remarque qu'il est globalement non-typé et que la JVM n'impose que 2 restrictions majeures :

- Les appels de méthodes sont typés statiquement. Les paramètres et le type de retour doivent être connus à la compilation
- Les appels de méthodes doivent se faire sur des types Java ainsi que des méthodes Java

La JSR 292 arrive à ce niveau-là et pour pallier ce manque, elle introduit l'instruction de bytecode *invokedynamic* qui permet d'appeler une méthode qui n'est pas encore définie en déléguant la résolution du code appelé à du code Java. En tenant compte de cette nouvelle instruction, les langages à typage dynamique tournant sur la JVM auront des performances bien meilleures.

Outre ce changement qui ne concerne pas vraiment les développeurs, une nouvelle API permettant de travailler avec l'instruction *invokedynamic* voit le jour. Celle-ci ne fait pas partie de la JSR 292 à proprement parler, mais elle ne peut en être dissociée dans les faits puisqu'elle permet aux développeurs de tirer parti au quotidien de ses apports. Cette API est introduite via le package *java.dyn* et possède comme classes les plus importantes :

- *MethodHandle*, référence vers une méthode réelle qui permet sa manipulation
- *MethodHandles*, classe de méthodes utilitaires pour les objets de type *MethodHandle*
- *MethodType*, qui représente la signature d'une méthode
- *InvokeDynamic*, classe de type final qui sert de marqueur syntaxique pour l'appel de méthodes dynamiques. Elle possède de fait un nombre infini de méthodes statiques
- *CallSite*, lie un *invokedynamic* avec une instance de *MethodHandle*

- *Linkage*, fait le lien entre les instances de *CallSite* et les appels dynamiques de méthodes

Afin de mieux comprendre l'utilisation que l'on peut faire de cette API et de ces méthodes, voici un bout de code adapté d'exemples présentés par Oracle :

```
public class TestDynamic {
    // Méthode utilisée pour les appels dynamiques
    static void dynamicMethod(String str) {
        System.out.println("Dynamic Method called with : " + str);
    }
    public static void main(String[] args) {
        // Appel statique de dynamicMethod
        dynamicMethod(args[0]);
        // Itération sur les arguments du programme
        for (String arg : args) {
            // Génère un invokevirtual MethodHandle.invoke(String)
            void
            <void>handleDynamicMethod.invoke(arg); // Appel fortement
            typé
            Object tmp = arg;
            // Génère un invokedynamic MethodHandle.invoke(Object)Object
            InvokeDynamic.hail(tmp); // Appel typé dynamiquement
        }
    }
    // Création de la signature de la méthode
    static MethodType signatureDynamicMethod = MethodType.method
    Type(void.class, String.class);
    // Création du lien entre dynamicMethod et notre signature de méthode
    static MethodHandle handleDynamicMethod = MethodHandles.lookup
    ().findStatic(TestDynamic.class, "dynamicMethod", signature
    DynamicMethod);
    // Mise en place du bootstrap de la méthode
    static { Linkage.registerBootstrapMethod("bootstrapDynamic"); }

    // Méthode chargée d'effectuer le bootstrap d'une méthode
    dynamique
    // quand une instruction invokedynamic est trouvée dans le byte
    code private static CallSite bootstrapDynamic(Class caller, String
    name, MethodType type) {
        MethodHandle target = MethodHandles.convertArguments(handle
    DynamicMethod, type);
        CallSite site = new CallSite(caller, name, type);
        // mise en place de la méthode réelle à appeler
        site.setTarget(target);
        return site;
    }
}
```

Cette JSR et cette nouvelle API font partie d'un projet plus large, connu sous le nom de « Da Vinci Machine Project ». Ambitieux, il a pour volonté de permettre à la JVM d'être un socle d'exécution optimal pour les différents langages s'appuyant dessus en offrant des performances équivalentes au langage Java.

## Meilleure gestion des Entrées / Sorties

Déjà opérationnelle depuis près de 2 ans, la nouvelle API NIO 2 a pour but d'apporter un framework haut niveau pour la manipulation





des Entrées / Sorties en Java. Jusqu'à présent, force est de constater que ces dernières n'étaient guère avancées et qu'il était bien souvent nécessaire de bidouiller pour arriver à ses fins ! En effet, qui n'a jamais eu de problème lors de la suppression d'un fichier et est resté étonné de ne voir aucune erreur Java levée ? Qui n'est pas resté frustré devant le manque d'extensibilité de l'API actuelle ? Les manques de l'API actuelle étaient nombreux et NIO 2 a vu le jour pour répondre à ces problématiques. Poursuivant le travail entamé par le projet NIO à l'époque du JDK 1.4, NIO 2 apporte une nouvelle API de fichier, définie dans le nouveau package dédié `java.nio.file`. La notion de système de fichier représentée par la classe `FileSystem` en est au cœur, de même que la notion de path représentée par la classe éponyme `Path` [Fig.1].

L'exemple suivant illustre l'utilisation de ces dernières :

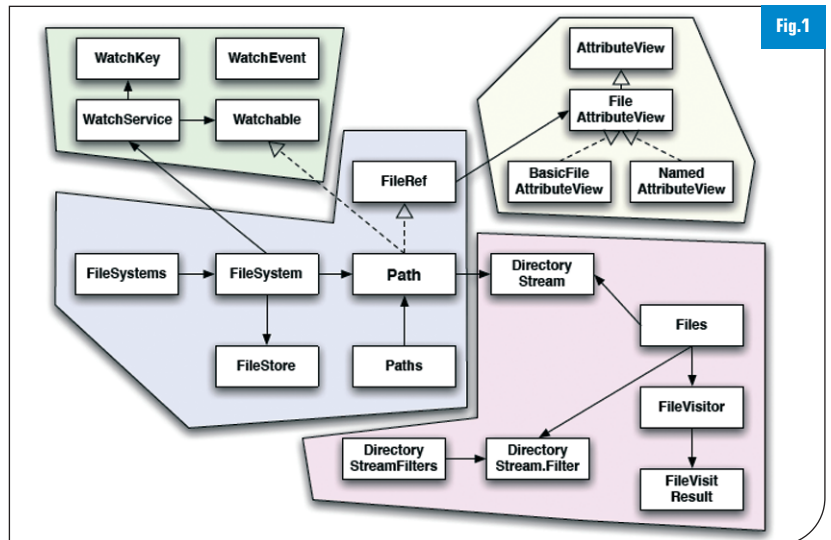


Fig.1

Diagramme de classes simplifié de l'API NIO 2

```

import java.io.* ;
import java.nio.file.* ;
import static java.nio.file.StandardOperation.* ;

// récupération du système de fichier par défaut
FileSystem fileSystem = FileSystems.getDefault() ;
Path homeDir = fileSystem.getPath("/home/sylsau") ;
// récupération en passant par la classe utilitaire Paths
Path homeDir = Paths.get("/home/sylsau") ;
// résolution de chemins
Path relativeTmp = Paths.get("tmp") ;
Path absoluteTmp = relativeTmp.resolve(homeDir) ;
// Ecriture dans un fichier
OutputStream outputStream = absoluteTmp.newOutputStream(CREATE,
APPEND) ;
outputStream.write(...) ;
    
```

NIO 2 amène également des fonctionnalités haut niveau permettant de copier ou de déplacer directement des fichiers représentés par des objets `Path` via les méthodes `copyTo` et `moveTo` de cette classe. Une amélioration notable pour les développeurs qui jusqu'ici devaient réaliser eux-mêmes leurs fonctions de copie et de déplacement. Quant à la suppression de fichiers, elle se fait désormais via la classe `Path` et sa méthode `delete` qui peut lancer une exception de type `IOException` lorsque la suppression n'a pu être réalisée correctement :

```

try {
    path.delete() ;
} catch (IOException ioe) {
    // avertissement de l'échec de la suppression
}
    
```

La création de liens symboliques est désormais possible et la gestion des propriétés de fichiers se voit dotée d'une hiérarchie de classes basée sur l'interface `FileAttributeView`.

Alors qu'avant on devait se contenter de l'accès aux propriétés de fichiers communes à l'ensemble des systèmes d'exploitation, NIO 2 offre maintenant un accès à toutes les propriétés spécifiques au système sous-jacent, aussi bien en lecture qu'en écriture. Ainsi, l'API propose notamment les implémentations suivantes :

- `BasicFileAttributeView`, qui permet d'accéder à un certain nombre de propriétés basiques, implémentées sur la plupart des systèmes de fichier
- `DosFileAttributeView`, qui offre un accès aux propriétés spécifiques aux systèmes Windows (readonly, hidden, system et archive)
- `PosixFileAttributeView`, qui donne l'accès aux informations spécifiques aux systèmes Unix (propriétaire, groupe et permissions de base du fichier)

D'autre part, le parcours des éléments d'un répertoire est également nettement amélioré et simplifié avec l'itérateur `DirectoryStream` récupérable depuis un objet `Path` et qui offre en outre la possibilité de sélectionner des éléments respectant un certain pattern.

Il est également possible de parcourir les éléments représentés par un objet `Path` et de leur appliquer un traitement via l'utilisation du design pattern `Visitor` au sein de l'API NIO 2.

Cette fonctionnalité est utilisable via la méthode statique `walkFileTree` de la classe utilitaire `Files` comme on peut le voir dans l'exemple ci-dessous :

```

// récupération de tous les fichiers java d'un répertoire
Path dir = Paths.get("/home/sylsau/files") ;
DirectoryStream<Path> javaFiles = dir.newDirectoryStream("*.java") ;

for(Path javaFile : javaFiles)
    System.out.println(javaFile.getName()) ;
// parcours récursif des éléments d'un fichier et application
d'un traitement spécifique
Files.walkFileTree(dir, new FileVisitor<Path>() {
    public FileVisitResult preVisitDirectory(Path dir, BasicFile
Attributes attrs) throws IOException {
        // TODO
        return FileVisitResult.CONTINUE;
    }
    public FileVisitResult visitFile(Path file, BasicFileAttributes
attrs) throws IOException {
        // TODO
        return FileVisitResult.CONTINUE;
    }
}
    
```



```
}
public FileVisitResult visitFileFailed(Path file, IOException
exc) throws IOException {
    // TODO
    // On arrête à la première erreur
    return FileVisitResult.TERMINATE;
}
public FileVisitResult postVisitDirectory(Path dir, IOException
exc) throws IOException {
    // TODO
    return FileVisitResult.CONTINUE;
}
});
```

Enfin, NIO 2 introduit un ensemble de classes offrant des possibilités de surveillance de fichiers ou répertoires représentés par des objets Path.

Il est de plus possible de préciser le type d'évènements pour lesquels on souhaite être alerté comme le montre l'exemple suivant :

```
import static java.nio.file.StandardWatchEventKind.*;
Path dirToWatch = Paths.get("C:/temp");
WatchService watcher = dirToWatch.getFileSystem().newWatch
Service();
dirToWatch.register(watcher, ENTRY_CREATE, ENTRY_MODIFY, ENTRY_
DELETE);

for(;;) {
    WatchKey key = watcher.take();
    for(WatchEvent<?> event : key.pollEvents()) {
        // affichage de l'événement
        System.out.println(event.kind() + " - " + event.context());
    }
    key.reset();
}
```

Comme on vient de le voir, NIO 2 apporte d'importantes possibilités au niveau de la gestion des fichiers en Java.

Cependant, ses apports sont bien plus larges puisqu'ils concernent également les sockets avec l'apparition de l'interface NetworkChannel qui vient terminer le travail entamé par NIO sur le sujet.

Les Entrées / Sorties asynchrones sont désormais supportées pour les fichiers et les sockets via l'utilisation des objets de type Future, et cerise sur le gâteau elles supportent la gestion des pools de threads :

```
// Création du channel asynchrone sur le fichier file.txt
AsynchronousFileChannel channel = AsynchronousFileChannel.open(
    Paths.get("/home/sylsau/file.txt"),
    StandardOpenOption.READ);
ByteBuffer buffer = ByteBuffer.allocate(capacity);
// Lecture asynchrone
Future<Integer> result = channel.read(buffer, 200);
// Result est terminé
if (result.isDone()) {
    // Nbre de bytes lus
    int bytesRead = result.get();
}
```

## Framework de parallélisation

Avec la montée en puissance des machines multi-cœurs, Java ne pouvait se contenter de l'API Concurrent apparue avec le JDK 5. Certes, celle-ci aura eu le mérite de rendre accessible au plus grand nombre des fonctionnalités de la programmation concurrente réservées jusque-là à des développeurs chevronnés, mais cela n'était clairement pas suffisant pour un langage de l'importance de Java ! Avec Java SE 7, Oracle va plus loin en nous amenant un framework de parallélisation à part entière dont la conception a été menée par Doug Lea, spécialiste des problématiques de concurrence dans le monde Java plus particulièrement.

Au programme de ce framework, l'introduction du mécanisme fork / join permettant d'exécuter facilement des traitements parallèles et ainsi de gagner en performance. Au cœur de l'API, on trouve la classe ForkJoinPool et un ensemble de tâches permettant la décomposition de problèmes.

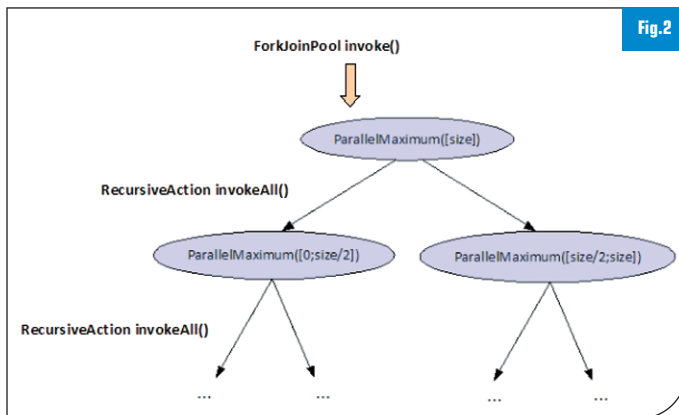
L'utilisation de cette classe n'est cependant pas automatique puisqu'il est nécessaire de bien analyser le problème rencontré pour préparer au mieux la décomposition en sous-problèmes dont la résolution sera parallélisée.

Pour illustrer l'utilisation du fork / join, nous allons étudier le programme suivant qui se charge de calculer la valeur maximum d'un tableau d'entiers de manière parallélisée :

```
public class ParallelMaximum extends RecursiveAction {
    private int[] list;
    public int result;

    public ParallelMaximum(int[] array) { list = array; }
    // méthode de lancement
    protected void compute() {
        if (list.length == 1) {
            result = list[0];
        } else {
            int midPoint = list.length / 2;
            // on divise le problème en 2 sous problèmes
            ParallelMaximum parallelMax1 = new ParallelMaximum(Arrays
.copyOfRange(list, 0, midPoint));
            ParallelMaximum parallelMax2 = new ParallelMaximum(Arrays
.copyOfRange(list, midPoint, list.length));
            // résolution en parallèle des 2 sous problèmes via fork / join
            invokeAll(parallelMax1, parallelMax2);
            // on calcule le max du résultat des 2 sous problèmes
            result = parallelMax1.result > parallelMax2.result ?
parallelMax1.result : parallelMax2.result;
        }
    }

    public static void main(String[] args) {
        // Création du problème initial avec un tableau d'entiers
        en paramètre
        ParallelMaximum parallelMax = new ParallelMaximum(new int
[] {99,1,1,45,9,250,89,-45,98,-22,106,24});
        // Création du ForkJoinPool avec le nbre de processeurs
        disponibles
        ForkJoinPool pool = new ForkJoinPool(Runtime.getRuntime()
.availableProcessors());
        // Lancement du traitement
        pool.invoke(parallelMax);
    }
}
```



Mécanisme du Fork / Join appliqué au problème ParallelMaximum

```

System.out.println("Max : " + parallelMax.result);
}
}

```

Ici, on remarquera l'héritage de la classe abstraite RecursiveAction qui impose la définition de la méthode compute au sein de laquelle le traitement à effectuer doit être placé. Au sein de cette méthode, on pourrait décider de paralléliser ou non le traitement suivant un seuil de granularité qui pourrait être le volume de données à traiter par exemple. En outre, la classe RecursiveAction nous donne accès à la méthode invokeAll qui va nous permettre d'appliquer le fork / join sur nos sous-problèmes [Fig.2] et ensuite d'en récupérer le résultat une fois que ceux-ci seront tous terminés. Le lancement du traitement principal se fait via un objet ForkJoinPool et sa méthode invoke à laquelle nous passons notre objet héritant de RecursiveAction.

Compte tenu de l'importance que prend le multi-cœur aujourd'hui et de celle qu'il aura dans le futur, on ne peut que se féliciter des apports de ce framework pour le monde Java. On peut espérer que d'autres améliorations s'y ajouteront dans le futur, notamment celles concernant la parallélisation des traitements applicables sur des collections de données. Et ceci afin de combler le retard pris par Java dans ce domaine face à ses concurrents.

## Simplification du langage

La montée en puissance des langages de script tournant sur la JVM a dévoilé au grand jour une complexité et une lourdeur certaines de la syntaxe du langage Java face à ses adversaires tels que Groovy, Scala ou Clojure. Afin de moderniser le langage Java et de simplifier sa syntaxe, tout en s'inscrivant dans une démarche de pérennité à long terme avec la fameuse compatibilité ascendante, le projet Coin a été lancé dans le but de recueillir les propositions de la communauté. Ces dernières ont été nombreuses et une première sélection a été effectuée par Oracle pour Java SE 7. Cette dernière correspond en fait aux 6 premières améliorations qui ont déjà été implémentées au sein du projet Open JDK au jour d'aujourd'hui. On retrouve donc :

- Une amélioration de la syntaxe pour les valeurs numériques avec l'apparition du préfixe « Ob » pour les valeurs binaires qu'on utilise comme suit : `int val = Ob1000000 ; // 128` . On notera également la possibilité d'utiliser le caractère underscore « \_ » de manière arbitraire entre les chiffres de valeurs numériques afin de simplifier la lecture du nombre pour le développeur : `double val =`

`1_000_000.00 // 1 million`

- Le switch avec String : `switch(str){ case "foo" : return 1; case "bar" : ... }`
- L'opérateur diamond <> pour simplifier la création d'objets génériques : `Map<String, List<String>> anagrams = new HashMap<>();`
- La réduction du nombre de warnings lors de l'invocation de méthodes varargs. En effet, ces dernières utilisent des tableaux en interne et donc implicitement on peut se retrouver dans le cas où des tableaux de types paramétrés sont créés. Ceci étant interdit en Java, on obtient alors des warnings à chacune des utilisations du paramètre de la méthode invoquée. Ce warning concerne en réalité seulement le cas où on modifierait le paramètre vararg au sein de la méthode appelée, ce qui est fortement déconseillé. Désormais, un seul warning sera généré et il concernera uniquement la déclaration de la méthode incriminée.
- Une amélioration notable de la gestion des exceptions avec la prise en charge du multi-catch d'exceptions comme suit : `try { ... } catch (final IOException | SQLException exc) { ... }` . En outre, le rethrow d'exception devient désormais possible. Ceci permettra d'intercepter toutes les exceptions au niveau d'un bloc catch afin de les logger par exemple et ensuite de lancer l'exception réellement attendue :

```

public void method() throws IOException {
    try { // code pouvant lancer une exception IOException }
    catch (final Exception exc) {
        LOGGER.info(exc);
        throw exc;
    }
}

```

- La dernière simplification concerne la gestion automatique des ressources qui contrairement à la mémoire ne sont pas gérées par le Garbage Collector de la JVM. Ainsi le pattern suivant a été utilisé de longue date pour les ressources en Java :

```

// création de la ressource
try { // utilisation de la ressource
} finally { // libération de la ressource
}

```

Le projet Coin permettra maintenant de se passer de l'étape de libération des ressources dans le bloc finally. Cette libération se fera de manière automatique à condition d'instancier les ressources à libérer automatiquement au sein d'un try (...) . Ainsi la copie d'un flux d'entrée vers un flux de sortie pourra débuter de la sorte :

```

try (InputStream input = new FileInputStream(sourceFile); OutputStream output = new FileOutputStream(destFile)) {
    // utilisation de la ressource
}

```

Ce mécanisme fonctionne grâce à l'introduction des interfaces AutoCloseable et Closeable qui présentent une méthode close(). Cette dernière est ensuite appelée par la JVM sur les classes qui les implémentent en sortie de bloc try avec gestion automatique des ressources, à condition que celles-ci soient instanciées au sein du try (...). Ainsi, cette gestion automatisée des ressources peut être





étendue à des types spécifiques par les développeurs comme cela est le cas avec l'interface `Iterable` introduite dans le JDK 5 pour la boucle `foreach`.

Ces petits "plus" syntaxiques ne sont pas à proprement parler une révolution mais plutôt une évolution positive de la syntaxe Java vers quelque chose de plus léger. Bien évidemment, beaucoup d'observateurs au sein de la communauté trouveront à redire sur ces simplifications en arguant du fait qu'elles sont trop légères et qu'il aurait fallu aller plus loin dès Java SE 7. Il faudra être patient pour ces derniers pour voir des modifications plus ambitieuses mais l'espoir reste permis puisque le projet Coin reste actif et continuera à travailler afin d'inclure de nouvelles modifications du langage pour la version 8 de Java.

## Divers

Outre les axes majeurs d'amélioration abordés dans cet article, d'autres petites améliorations vont venir se greffer au sein de Java SE 7. En premier lieu, on peut citer l'intégration du Garbage Collector G1 au sein de la JVM qui va offrir de meilleures performances au niveau de la gestion de la mémoire.

Le Look And feel Nimbus est mis à disposition au sein de l'API Swing qui se voit également ajouter le composant `JXLayer` issu du projet SwingLabs. Enfin, Java supportera désormais la norme Unicode 6.0

et la gestion automatique des ressources est appliquée aux ressources JDBC via l'héritage de l'interface `AutoCloseable` et de ce fait l'API passe en version 4.1.

## Conclusion

Le rachat de Sun par Oracle aura provoqué beaucoup de remous au sein d'un écosystème Java déjà fragilisé par l'incertitude régnant autour du projet OpenJDK et de la version 7 de Java SE. Les dernières annonces d'Oracle vont dans le sens du pragmatisme avec la définition d'une roadmap réaliste pour Java SE 7 mais également pour Java SE 8 dont les versions sont respectivement programmées pour le milieu de l'année 2011 et pour la fin de l'année 2012. En outre, le soutien officiel apporté par IBM au projet Open JDK montre la volonté d'Oracle de fédérer un acteur majeur du monde Java avec lui, ce qui est une bonne nouvelle pour l'avancement du projet. Enfin, d'un point de vue plus technique, les nouveautés apportées par Java 7 ne constitueront pas à coup sûr une révolution pour le développeur. Elles s'inscrivent plutôt dans une logique d'amélioration des performances de la plateforme Java mais également de simplification du langage comme le souhaite Oracle.

■ Sylvain Saurel – Ingénieur d'Etudes Java / JEE  
sylvain.saurel@gmail.com

# L'INFO permanente

- L'actu : le fil d'info quotidien de la rédaction
- La newsletter hebdo : abonnez-vous, comme 46 000 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](http://programmez.com)
- Les tutoriels : une solution en quelques clics !
- Le téléchargement : récupérez les nouveautés.

[www.programmez.com](http://www.programmez.com)



# Générer des web services Axis Spring + Hibernate à l'aide d'AndroMDA et de RSM (IBM)

L'approche Model Driven Architecture, définie par l'OMG fête cette année ses 8 ans. Depuis ce temps, elle définit des bonnes pratiques de réalisation semi-automatique des applications d'entreprise. Au cœur de ces bonnes pratiques, les modèles PIM (*Platform Independant Model*), PSM (*Platform Specific Model*), PDM (*Platform Dependand Model*) viennent guider l'analyse, la conception et la génération d'un squelette d'une application.

**D**ans cet article, nous présenterons une mise en œuvre de la définition d'un modèle PSM jusqu'à la génération du code, à l'aide des outils RSM (Rational Software Modeler) d'IBM et du framework AndroMDA, générateur extensible de code. En particulier, nous nous intéresserons à la génération d'un web service Axis (Spring+ hibernate) partagé par des applications « santé » qui doivent réaliser des recherches sur des patients (bénéficiaires) à propos de leur couverture sociale. Ce web service offre deux principales fonctions de recherche : obtenir un patient sur la base de son NIR (numero SS) et obtenir la liste des patients enregistrés au sein d'une base de données.

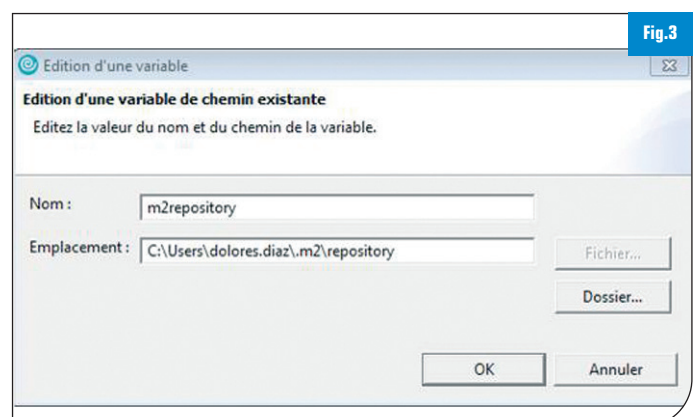
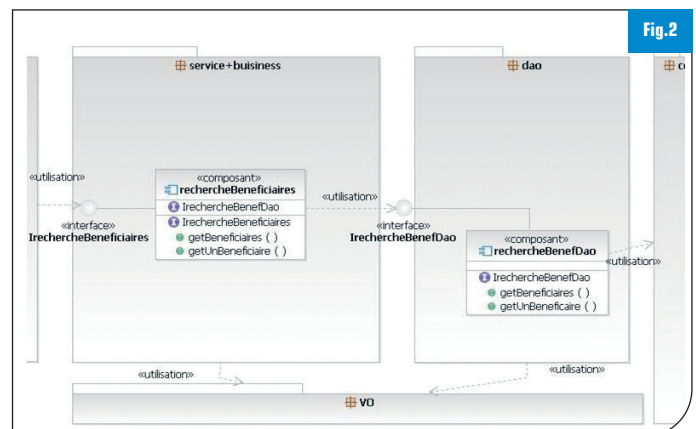
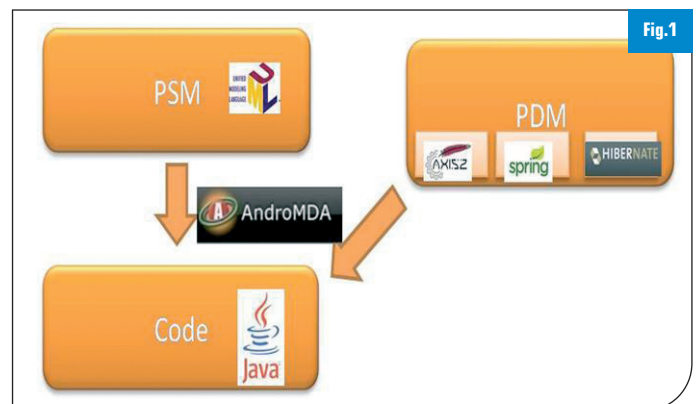
## Présentation générale de la mise en œuvre d'une définition d'un PSM vers la génération de code

La figure suivante fournit une vue d'ensemble de mise en œuvre de la définition d'un modèle PSM. Le PSM est défini à l'aide de diagrammes UML. Ces derniers doivent être enrichis par l'intermédiaire de stéréotypes ou valeurs taggées de chaque modèle PDM, en vue de générer une couche logicielle de l'application. Ainsi, 3 cartouches Axis, Spring, et Hibernate (ensemble de stéréotypes et compréhensibles par AndroMDA) ont été choisis pour générer le squelette du web service de santé [Fig.1].

Plus précisément, l'architecture logicielle cible de la recherche d'un assuré sera articulée autour d'un composant Spring et d'un composant Hibernate. L'ensemble sera exposé à l'aide d'Axis2 [Fig.2].

**Installation et paramétrage d'outils pré-requis :** Voici la liste des outils choisis et à paramétrer pour notre exemple de web service. Vérifier

- L'installation d'un JDK 1.6 et la déclaration d'une variable d'environnement JAVA\_HOME avec son chemin d'installation.
- L'installation de Maven 2.0.5 et la déclaration d'une variable d'environnement M2\_HOME avec son chemin d'installation et en option l'installation d'un repository.
- L'ajout des commandes JAVA et MAVEN dans la variable PATH System.
- L'installation du plugin Maven AndroMDA v3.2[1].
- L'installation RSM 6.0 mais surtout la déclaration de l'import des profils AndroMDA dans RSM par l'intermédiaire d'une variable d'environnement m2repository qui pointe sur le repository MAVEN [Fig.3].



[1] Pour l'installation du plugin Maven Andro V3.2, nous invitons le lecteur à se rendre sur le site AndroMDA : [http://www.andromda.org/index.php?option=com\\_content&view=article&id=105:environment-setup&catid=42:getting-started-java&Itemid=89](http://www.andromda.org/index.php?option=com_content&view=article&id=105:environment-setup&catid=42:getting-started-java&Itemid=89)

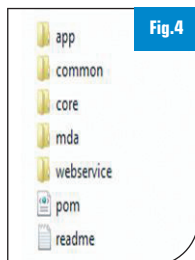
## Créer un projet AndroMDA

La création d'un projet AndroMDA passe la commande Maven suivante dans une console Shell.

```
mvn org.andromda.maven.plugins:andromdapp-maven-plugin:3.2:generate
```

AndroMDA pose une liste de questions en vue de créer un projet conforme à l'architecture logicielle attendue.

```
Please choose the type of application to generate.
j2ee Please enter the location in which your new application
will be created
C:\Users\dolores.diaz\IBM\rational\sd6.0\workspace
Please enter your first and last name. Dolores Diaz
Which kind of modeling tool will you use ? Emf-uml2
Please enter the name of your J2EE project SanteSoft_Code
Please enter an id for your J2EE project. SanteSoftCode
Please enter a version for your project 1.0
Please enter the root package name for your J2EE project
fr.valtech.sante Would you like an EAR or standalone WAR ear
ear Please enter the type of
transactional/persistence cartridge to use Spring
Please enter the database backend for the persistence layer
Oracle Will your project need workflow engine capabilities?
No Please enter the hibernate version number ? 3
Will your project have a web user interface?
No Would you like to be able to expose your services as web services?
Yes Would you like to use Axis or Xfire as
your SOAP Stack ? Axis
Would you like to use Axis 1.x or Axis 2.x 2
```



Voici le résultat de la génération du projet AndroMDA : [Fig.4].

Le résultat de cette première génération est un projet Maven qui contient des sous-projets Maven. Chaque projet Maven implémente une couche logicielle de l'application SanteSoftCode. Ainsi le projet **common** contient la génération d'objets Java qui vont se charger

de transférer des données : les Value Objects (VO.). Le projet **core** contient les ressources et le code définissant la couche service+business. Le projet **app** contient les ressources de configuration de l'application. Le projet **webservice** contient les ressources de déclaration du web service. En vue de compléter ce squelette de l'application SanteSoftCode, passons à la définition du PSM réalisé à l'aide de diagrammes UML.

## Définir et enrichir les diagrammes UML du PSM.

Le principe de définition d'un PSM repose sur le même découpage du projet Maven de l'application SanteSoftCode. Le PSM est défini et complété dans RSM et se compose d'autant de diagrammes de classes qu'il existe de couches logicielles [Fig.5].

C'est à ce moment que rentre en jeu l'utilisation des modèles PDM



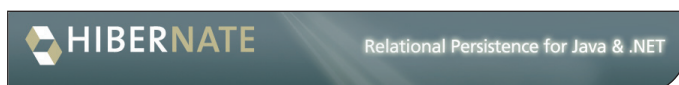
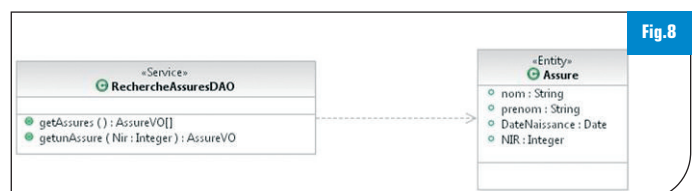
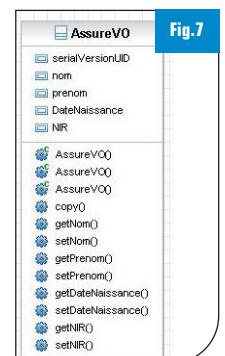
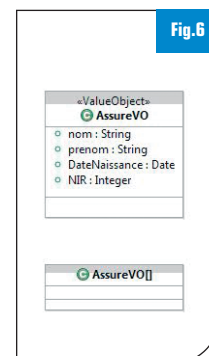
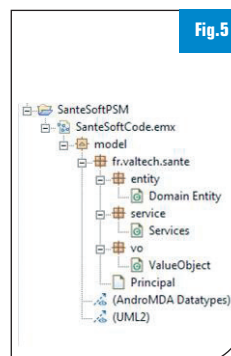
(Platform Dependand Model) par l'intermédiaire des cartouches AndroMDA. A ce jour, AndroMDA propose plus d'une dizaine de cartouches caractérisant frameworks et langages de programmation les plus répandus sur le marché : (Struts, Spring, Axis, Hibernate, EJB, EJB3,...). Chaque cartouche AndroMDA définit un ensemble de stéréotypes et de valeurs taggées spécifiant une technologie. Ces stéréotypes et valeurs taggées sont ensuite interprétés par le moteur AndroMDA en vue de compléter le projet Maven de l'application SanteSoftCode.

La définition du modèle PSM du projet SanteSoft débute par la modélisation de la couche des VO : les données manipulées dans l'application concernent les **Assures**. En conséquence, le diagramme UML **Value Objet**, contient **AssureVO** et le type de liste **AssureVO[]**. Dans cette couche, seul le stéréotype **<<ValueObject>>** indique à AndroMDA la génération de la classe Java correspondante [Fig.6]. Plus précisément, AndroMDA génère dans l'arborescence **/common/target/src/fr/valtech/sante/vo** la classe Java correspondante : [Fig.7].

Dans un second temps, la couche de persistance, représentée par le diagramme UML **Domain entity**, contient le composant hibernate permettant d'accéder à une source de données. La classe UML **RechercheAssureDAO**, stéréotypée **<<Service>>** représente un composant DAO et La classe UML **Assuré**, stéréotypée **<<Entity>>** représente les composants hibernate qui vont être mappés aux tables d'une source de données. Des valeurs taggées telles que **andromda.persistence.schema** permettent de préciser le nom du schéma cible et complètent les fichiers de configuration du data-source cible [Fig.8].

AndroMDA génère pour chaque classe UML stéréotypée « Entity » 2 classes Java : une classe abstraite **Assure.java** et sa classe concrète **AssureImpl.java** ainsi que le fichier de configuration hibernate correspondant : **Assure.hbm.xml** [Fig.9].

Enfin la couche métier, celle qui contient le composant Spring, expo-





sé en web service est définie par l'intermédiaire du dernier diagramme UML **Services**. Nous sommes restés sur la même logique des précédentes opérations de recherches d'assurés qui se retrouvent maintenant exposées et générées dans le WDSL du web service SanteSoftCode, indiqué par le stéréotype **<<WebServiceOperation>>** au niveau de chaque opération du service RechercheAssureService [Fig.10].

Pour chaque classe UML stéréotypée **<<Service>>**, AndroMDA génère un trio de classes Java. Par exemple, pour RechercheAssureBusiness, il y a une interface RechercheAssureBusiness.java, une classe abstraite RechercheAssureBusinessBase.java et une classe concrète RechercheAssureBusinessImpl.java. Cette dernière est placée dans un package à part : **core/src/main/java/fr/valtech/sante** et n'est pas écrasée lors d'une prochaine itération de génération de code. Ceci permet au développeur de venir compléter le code en intervenant notamment sur les implémentations des opérations, sans que cela soit écrasé lors d'une nouvelle génération de code par AndroMDA [Fig.11].

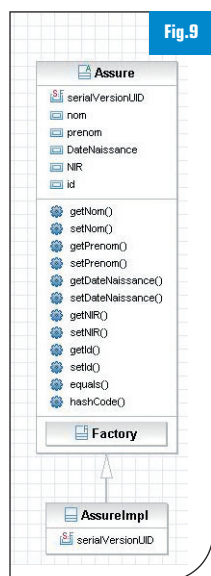


Fig.9

A l'issue de cette première modélisation, nous obtenons dans le répertoire **app\target** une archive **ear**, nommée **SanteSoftCode-1.0.ear** contenant les archives suivantes : SanteSoftCode-webservice-1.0.war, SanteSoftCode-core-1.0.jar, SanteSoftCode-common-1.0.jar. En déployant directement cette archive dans un serveur d'application, tel que par exemple JBOSS, il est possible d'accéder directement aux opérations du webservices. Par ailleurs, AndroMDA génère une page de présentation des opérations, accessible depuis l'url [http://<IP\\_Serveur>:8080/SanteSoftCode-ws/listServices.jsp](http://<IP_Serveur>:8080/SanteSoftCode-ws/listServices.jsp) [Fig.12].

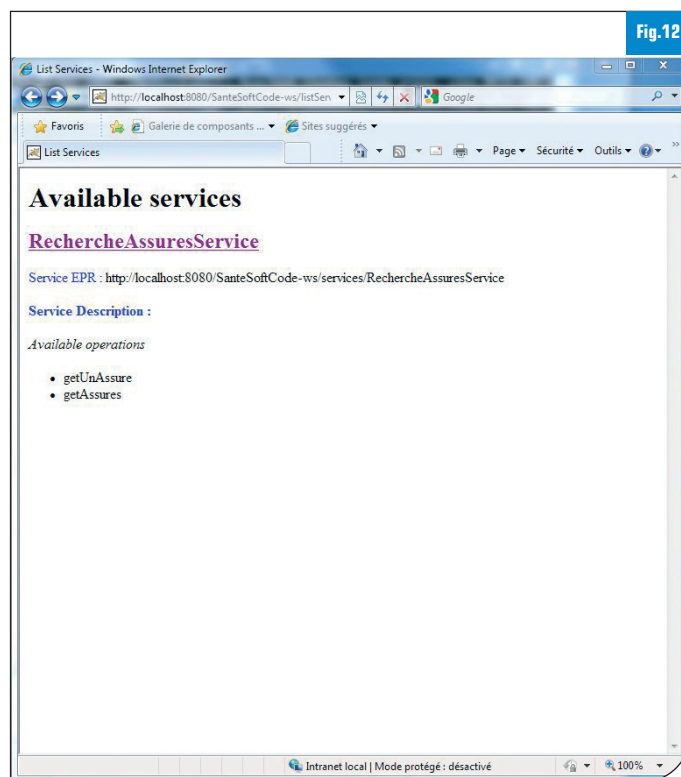


Fig.12

## Conclusion

Le framework AndroMDA reste un des rares générateurs de code, libre et extensible, qui est indépendant de l'outil de modélisation. Il n'est pas nécessaire d'investir dans un outil payant pour avoir une génération utilisable et intéressante de squelette d'applications. En effet, AndroMDA s'interface avec MagicDraw, ArgoUML et comme nous venons de le voir avec RSM (IBM), par l'intermédiaire des formats d'échange EMF(Eclipse), XML1.2 ou encore XML 2.0. Ceci étant, s'il existe une indépendance entre AndroMDA et les outils de modélisation pour la définition d'un PSM, la définition et l'utilisation des modèles PDM sont extrêmement liés à l'outil les supportant. Par exemple, l'ensemble des cartouches AndroMDA ne sont interprétables que par le moteur AndroMDA. Ceci constitue la raison principale pour laquelle les entreprises sont si réticentes à l'approche MDA. Cette limite vient du manque de standardisation de ces modèles PDM et tant qu'il n'y aura pas de profils normés OMG pour chaque technologie (Struts, Spring, WebServices, ...), les générateurs de codes continueront à proposer leur propre stéréotypes et valeurs taggées.

Rappelons que l'approche par objet a mis 20 ans pour s'imposer.

## Références :

Site MDA : <http://omg.org/mda/>, Site OMG : <http://www.omg.org/>,  
Site AndroMDA : <http://www.andromda.org/index.php>,  
Site RSM (IBM) : <http://www.ibm.com/developerworks/rational/products/rsm/>



■ Dolores Diaz  
Consultante chez Valtech Technology

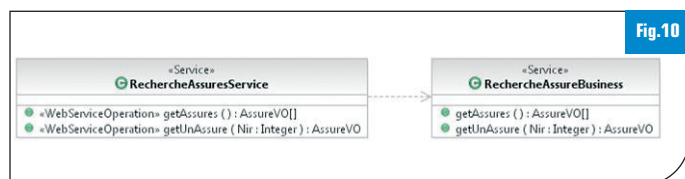


Fig.10

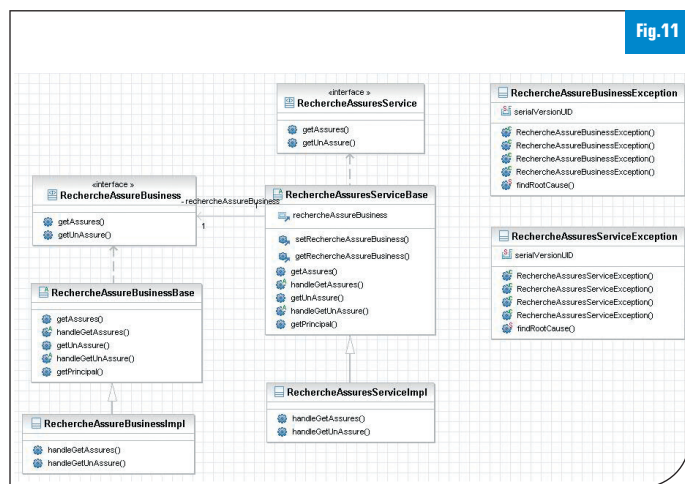


Fig.11

 **In The DataCenter**

 **In The Cloud**

 **In The Security**

 **In The SaaS**

160 exposants - 34 conférences  
pour un système d'information plus sûr et performant...



**In The Business**

8\* - 9 - 10 mars 2011 Paris - Porte de Versailles

\*à partir de 14h

Diamond Sponsors



Platinum Sponsor



Gold Sponsors



Silver Sponsor



Partenaires



email : [informations@inthe-business.com](mailto:informations@inthe-business.com)

[www.inthe-business.com](http://www.inthe-business.com)

# Epitech Innovative Projects

## Le cru 2010

Pour la 5<sup>e</sup> année, l'école informatique, Epitech, a organisé le forum des Epitech Innovative Projects (EIP), fin novembre dernier. Une trentaine de projets furent présentés. Une édition de très bon niveau et des équipes motivées !

**P**rojet de fin d'études d'envergure qui s'étend de la 4<sup>e</sup> à la 5<sup>e</sup> année, l'EIP est la résultante d'un travail en équipe, de 5 étudiants au minimum. Cette réalisation aboutie et de qualité professionnelle, concrétise les 5 ans d'enseignement à Epitech et illustre la pédagogie incitative défendue par l'école. L'EIP per-

met aux étudiants de développer leur capacité à gérer une équipe, à travailler en équipe et mener à bien un projet, tout en développant un réseau de contact auprès des entreprises.

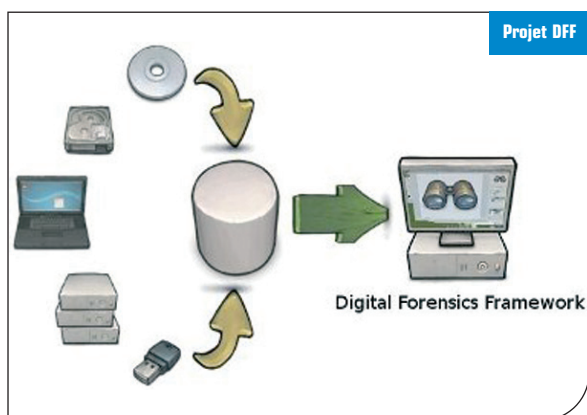
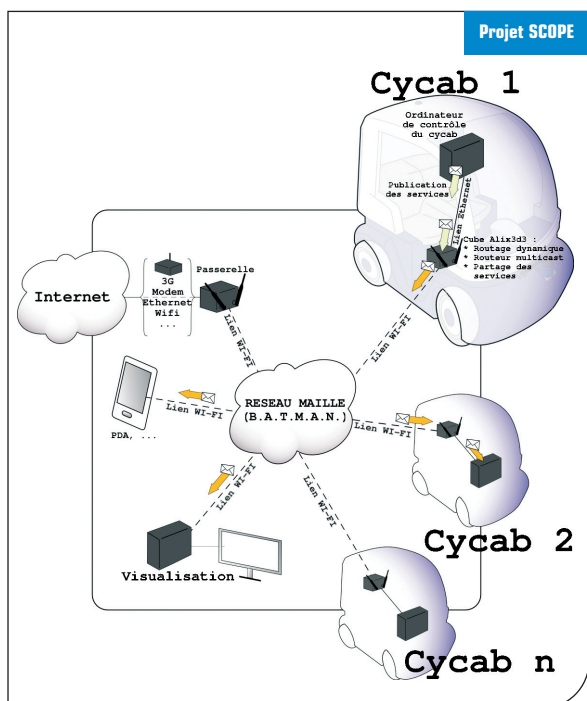
Cette année, 5 thèmes étaient retenus : améliorer le bien-être des individus, mieux diffuser le progrès technique pour tous, rendre plus efficaces les outils destinés aux entreprises, ouvrir de nouveaux horizons aux passionnés et développer un EIP existant en prolongeant ainsi la recherche sur un projet porteur. A Programmez ! nous avons été sensibles à six projets. Petit tour technologique de cette édition 2010.

### Des projets ambitieux et passionnants

Le projet **Kiwings** ambitionne de créer une technologie pour interagir avec l'ordinateur en fonction de la position du corps et d'éléments du corps dans l'espace. Ces mouvements sont capturés par une webcam connectée à l'ordinateur où est installé le logiciel de traitement. L'un des défis, selon l'équipe, a été de traiter le mouvement en temps réel et d'obtenir une qualité de 30 images par seconde, sur un netbook bien moins puissant, le traitement tombe à 15 images. Les prémices du projet ont été lancés il y a deux ans. Le but est de faciliter l'accessibilité des personnes ayant des handicaps. L'architecture est relativement simple. Le logiciel s'articule autour de trois modules : MotionManager (détermine les formes et les mouvements), MotionDriver (envoi des événements clavier et souris au système), ConfigGenerator (associe

les mouvements à des événements clavier et souris). Le tout repose sur du C++, les API OpenCV (pour capturer le flux vidéo). L'éditeur de code est Visual Studio 2008. Pour l'interface, le choix s'est porté sur GTK+. Mais si aujourd'hui le projet tourne en multithread, le but est de le passer, à l'avenir, en code parallèle pour améliorer les performances. L'une des difficultés du projet a été de traiter les mouvements et d'optimiser OpenCV. Site : <http://eip.epitech.eu/2011/epimotion>

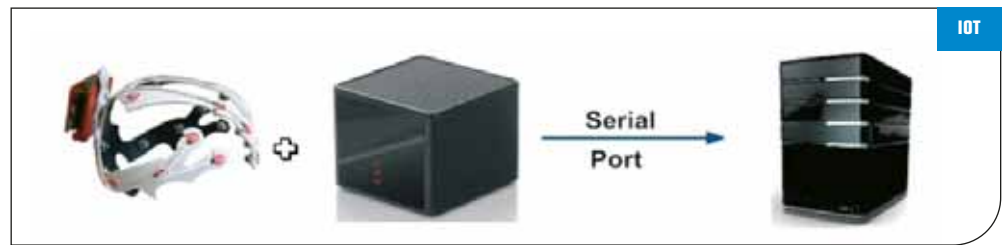
Le projet **Athena** peut se définir comme "l'ordinateur qui pense à vous". Athena ajoute à un ordinateur une qualité rare et dont il était jusqu'à présent dépourvu : l'empathie. En analysant les actions de l'utilisateur à travers le clavier et la souris, cet EIP est capable de deviner l'état émotionnel de son utilisateur. Son intelligence artificielle sait reconnaître les réactions positives, négatives ou neutres en fonction du comportement face à l'écran. Athena se destine en priorité aux éditeurs et développeurs de logiciels. Couplé à des tests en situation réelle, ce logiciel permet d'avoir un retour sur l'état émotionnel des volontaires. Ces précieuses informations permettent ainsi de modifier le fonctionnement d'une application pour répondre aux attentes des utilisateurs. Afin de calibrer les premières mesures, Athena propose, à l'aide de mini-jeux, de mettre en place des profils personnalisés pour ses testeurs. Ce projet est réalisé dans le cadre d'un partenariat avec le laboratoire ACSEL d'Epitech, qui centre ses recherches dans le domaine des intelligences artificielles. Côté technique : Visual Studio 2008, Qt (interface), C.





L'une des difficultés du projet est de définir, d'étalonner l'utilisateur pour mieux analyser son comportement. Cela passe par la création d'un profil suite à une série de jeux plaçant l'utilisateur dans différentes situations. L'équipe a travaillé avec un psychologue pour comprendre le comportement, et comment faire pour le "traduire" en informatique. Les étudiants ont dû longuement se plonger dans les ouvrages spécialisés afin de mieux appréhender l'Intelligence Artificielle. Le travail est loin d'être terminée et l'équipe actuelle espère trouver de nouveaux étudiants pour faire vivre le projet...

Le projet **DFF (Digital Forensic Framework)** est peut être CELUI qui nous a le plus intéressé. On parle ici de technique forensic, donc d'informatique légale. C'est-à-dire que l'on n'intervient pas avant les attaques et intrusions mais après pour analyser les actions, les données, retrouver les données effacées, comprendre l'attaque, etc. Cette informatique est malheureusement encore trop peu connue en France. Digital Forensic Framework (DFF) est différent, car à l'inverse de ses homologues, il est libre, gratuit, multiplateforme, et propose tout le nécessaire de l'informatique légale en un seul logiciel. Pensé pour des professionnels de l'investigation, DFF "autopsie", maintient des données provenant de différents supports informatiques (disques durs, téléphones portables, clés USB, etc.). Il peut ainsi constituer des preuves légales (comme des échanges de SMS ou des tentatives d'intrusion) pour appuyer des cas judiciaires. Ses possibilités sont regroupées sous une bibliothèque de fonctions ou framework. Ce projet est déjà ancien et a fait l'objet de la création d'une société, ArxSys. Le projet est en double licence. Le travail réalisé par l'équipe de cette année a été important : refonte de l'architecture, ouverture du support à NTFS, prise en compte des systèmes de fichiers mobiles, amélioration du module réseau. L'interface a été largement modifiée. Le projet s'appuie sur C++, Python et Qt. Le tout est codé sur



Visual Studio 2008 et 2010 ainsi que sur emacs. L'objectif à court terme : continuer le projet et le faire connaître !

Des partenariats existent aussi avec l'INRIA. C'est le cas avec le projet **SCOPE**, que nous avons déjà pu observer il y a deux ans. Le but du projet est de créer une voiture robotisée, sans pilote. Scope existe depuis 5 ans et travaille activement avec le laboratoire IMARA. Il ne s'agit pas ici de construire une voiture sans pilote mais d'estimer la faisabilité du projet, créer les briques techniques nécessaires. Cette année, l'équipe Epitech s'est beaucoup investie dans la partie communication qui a été entièrement réécrite pour la rendre plus performante, plus rapide dans les traitements. La communication entre les voitures se fait en réseau wifi via un réseau maillé. Le travail a été important car il a fallu passer en Ipv6, revoir la topologie réseau, créer une nouvelle couche de découverte de service. Et si aujourd'hui cette partie est hébergée sur un Windows XP, demain, ce sera un Windows 7 embarqué. Dans quelques mois, une expérimentation à plusieurs voitures sera réalisée sur 6 mois, sur un parcours défini à La Rochelle. Occasion pour les ingénieurs de tester les technologies, la réactivité et la logique du système.

Autre projet pas si futuriste que cela : **IOT (I only think)**. Destiné principalement aux personnes à mobilité réduite, il permet de transformer à l'aide d'un casque les impulsions électriques du cerveau en instructions compréhensibles par un logiciel. Il suffit de regarder un menu pour le voir s'ouvrir, ou de se concentrer sur une application pour la lancer. Grâce au partenariat avec l'Institut national de

recherche en informatique et automatique (INRIA), IOT n'a pas besoin d'une méthode intrusive pour fonctionner : le casque OpenEEG se pose et s'adapte sur la tête pour permettre de contrôler l'ordinateur par la pensée. Développé pour le plus grand nombre à partir de la technologie OpenViBE, cet EIP est multiplateforme (Windows, Linux, Mac OS) et libre. Son utilisation est totalement personnalisable pour faciliter sa « prise en main » et s'adapter aux méthodes de travail de l'utilisateur. Site : <http://eip.epitech.eu/2011/iot>

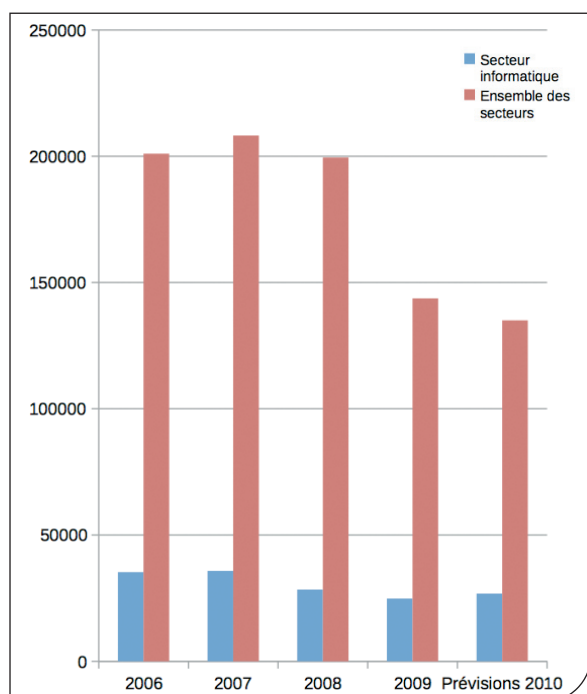
Rendez-vous à l'automne 2011 pour la prochaine édition.

■ François Tonic



# L'emploi est timidement reparti !

Le marché retrouve un dynamisme dans l'Informatique : à l'occasion de la sortie de la première édition de leur baromètre semestriel commun, l'APEC et Syntec numérique mettent en évidence les chiffres clés de l'emploi cadre informatique.



## Près d'un cadre sur 10 en France

Le secteur informatique demeure un des tous premiers secteurs employeurs. Il compte près de 246 000 cadres en France, soit 9% des effectifs de cadres du privé. Le secteur se caractérise par sa jeunesse : un cadre sur deux a moins de 40 ans, et par la prédominance du sexe masculin (73%).

Par ailleurs, 44% des cadres informaticiens sont basés en Ile de France, alors qu'ils ne sont que 38% tous secteurs confondus.

## 2e semestre 2010 : la reprise s'est confirmée

Ce premier baromètre montre un redressement en 2010 de la situation de l'emploi dans l'informatique. 90% des entreprises du secteur ont

recruté au moins un cadre au cours du troisième trimestre 2010, alors qu'elles n'étaient que 80% l'année précédente. En 2010, les estimations s'élèvent à 26 000 recrutements de cadres dans le secteur, en progression par rapport à 2009. Pour 58% des entreprises du secteur, le volume de recrutement est plus élevé que l'an passé.

54% des entreprises de plus de cent salariés ont recruté au moins un cadre au cours du 3e trimestre 2010.

L'année 2009 s'était soldée, pour le secteur, par une destruction de près de 3 670 postes cadres, soit un taux de perte d'emploi de 1,3%. Le secteur s'est toutefois maintenu pendant la période de récession parmi les premiers créateurs d'emplois cadres (recrutements bruts).

La conjoncture de l'emploi cadre du secteur est donc meilleure que la moyenne tous secteurs d'activité confondus. Cependant, cette reprise est assez timide. Le nombre de recrutement est en hausse en 2010 par rapport à 2009, mais reste inférieur à 2008, et encore plus aux années précédentes.

## INSERTION PROFESSIONNELLE DES JEUNES DIPLÔMÉS

	Situation professionnelle en 2009 des diplômés de 2008		Situation professionnelle en 2010 des diplômés de 2009	
	Filière informatique	Ensemble	Filière informatique	Ensemble
Taux d'emploi	81	68	70	64
Taux d'insertion	86	75	73	70

	Situation professionnelle en 2009 des diplômés de 2008		Situation professionnelle en 2010 des diplômés de 2009	
	Filière informatique	Ensemble	Filière informatique	Ensemble
Secteur Public	13	35	13	37
Secteur Privé	87	65	87	63
Etudes, R&D	11	21	11	16
<b>Informatique</b>	<b>78</b>	<b>7</b>	<b>80</b>	<b>6</b>
Autres fonctions	11	72	9	78
Cadre ou fonctionnaire A	91	64	86	63
AM ou fonctionnaire B	3	9	10	11
Employé ou fonctionnaire C	6	27	4	26
CDI ou titularisé	88	61	82	45
CDD ou contractuel	11	34	16	41
Intérim	0	3	1	4
Autres	1	2	1	10
Industrie construction	12	19	16	22
Commerce	2	6	1	6
Services	86	71	83	72
dont services informatiques	51	6	45	5

Source Apec : « Les jeunes diplômés de 2009 : situation professionnelle en 2010 »

## 37 candidatures par offre d'emploi

Effet du ralentissement conjoncturel, les offres d'emploi en informatique attirent davantage de candidatures que par le passé : 37 en moyenne sur les offres diffusées par les entreprises au cours du 1er trimestre 2010 contre 23 en moyenne un an auparavant. Elles restent, néanmoins, moins attractives que l'ensemble des offres tous secteurs confondus qui suscitent, en moyenne, 57 candidatures.

Ainsi deux postes sur trois restent difficiles à pourvoir. Les recruteurs déclarent, plus souvent que la moyen-

ne, qu'il leur a été difficile de trouver des candidatures adéquates : 64% pour les activités informatiques contre 58% tous secteurs confondus.

### 82% des jeunes diplômés décrochent un CDI

La proportion de jeunes diplômés de la filière informatique (bac +4 ou plus) sortis en 2009 qui sont en activité en 2010, reste supérieure à la moyenne des sortants toutes disciplines confondues : 70% contre 64%. Mais l'écart s'est réduit, la promotion de 2009 ayant été pénalisée par la crise avec une perte, par rapport à la promotion de 2008, de 11 points sur le taux d'activité et de 13 points sur le taux d'insertion (part des jeunes diplômés ayant connu une insertion).

### L'AVIS DES RECRUTEURS

Diriez-vous que vous avez trouvé des candidats adaptés à ce poste...

	Secteur informatique	Ensemble des secteurs
Très facilement	4%	5%
Plutôt facilement	32%	36%
Plutôt difficilement	50%	44%
Très difficilement	14%	14%
Total	100%	99%

NSP = 1%

Source : Enquête Apec de suivi des offres

Les jeunes diplômés de la filière informatique s'orientent surtout dans le secteur privé (87% contre 63% pour l'ensemble). Ils débutent dans la fonction informatique (80%) et dans une moindre mesure en Recherche & Développement (11%). Entre les sortants de 2008 et ceux de 2009, ces proportions sont restées assez stables. Ils obtiennent un meilleur sta-

tut que la moyenne des jeunes diplômés : 86% sont cadres ou fonctionnaires A (contre 63%) et 82% ont un CDI ou sont titularisés (contre 45%). S'ils restent mieux lotis que les autres débutants, leur statut s'est néanmoins un peu dégradé par rapport à la promotion de 2008 : 91% étaient cadres (- 5 points) et 88% en CDI (-6 points).

## Une forme originale de recrutement : formations gratuites pour les candidats

Objet Direct, spécialisée sur les architectures objet et web, invite gratuitement à ses formations internes les candidats à un emploi. Candidats et collaborateurs suivront ensemble ces formations sur mesure sur les dernières technologies du monde .Net, Java et Internet.

**A**vec cette opération, Objet Direct, filiale de Viseo, entend se positionner comme un recruteur actif de nouveaux talents, en suscitant la motivation par l'immersion. Les formations seront en effet animées par des consultants-formateurs d'Objet Direct et intégreront également des salariés de la société dans le cadre de leur plan de formation annuel.

« En 2010, nous avons lancé un plan de recrutement ambitieux de 100 postes au niveau du groupe Viseo, dont 35% pour Objet Direct », indique Franck Priore, Directeur Général d'Objet Direct. Mais le contexte économique complique le recrutement car il noie les candidats dans les sollicitations. « On constate qu'une partie des candidats ne sait pas ce qu'apporte réellement une société spécialisée technologiquement comme la nôtre par rapport à une SSII généraliste. Il nous faut donc sortir des moyens de recrutement traditionnels, et faire connaître notre politique de gestion des Ressources Humaines basée sur la



proximité et sur un plan de formation aux nouvelles technologies, continu et ambitieux.

Il nous est alors apparu tout naturel d'ouvrir nos formations aux candidats qui, non seulement assimileront une matière pédagogique de premier ordre, mais auront également l'opportunité d'échanger avec les consultants Objet Direct qui participent à

ces sessions ; ainsi ils pourront découvrir l'ambiance, le niveau d'expertise et la réalité des valeurs de l'entreprise, et ce, grandeur réelle, et non plus seulement dans le cadre des entretiens de sélection», poursuit Franck Priore.

Inscription en ligne sur

<http://formez-vous.objetdirect.com>

En savoir plus : [www.objetdirect.com](http://www.objetdirect.com)





# Gardez le fil sur internet

Que serait le web et les sites web sans les fils et les flux d'informations et de données RSS, Restful, ou encore provenant de Twitter ? Une des forces du web est d'agréger différents fils dans une page web. Mais au-delà du simple site web, les fils s'installent un peu partout : applications riches, portails Sharepoint, Facebook, etc. Plus que jamais, il faut savoir rester connecté à ses feeds préférés.

Dans ce dossier, nous allons étudier plusieurs cas de figures : Sharepoint, Twitter, et comment mettre en place une structure RESTful avec JSON.

Il est temps de tricoter son web !

■ François Tonic

# Ecrire un client pour l'API Twitter

Le service de microblogging Twitter expose une API que nous mettons en œuvre afin d'être capable de suivre les 'tweets' depuis une application de bureau, plutôt qu'à travers un navigateur.

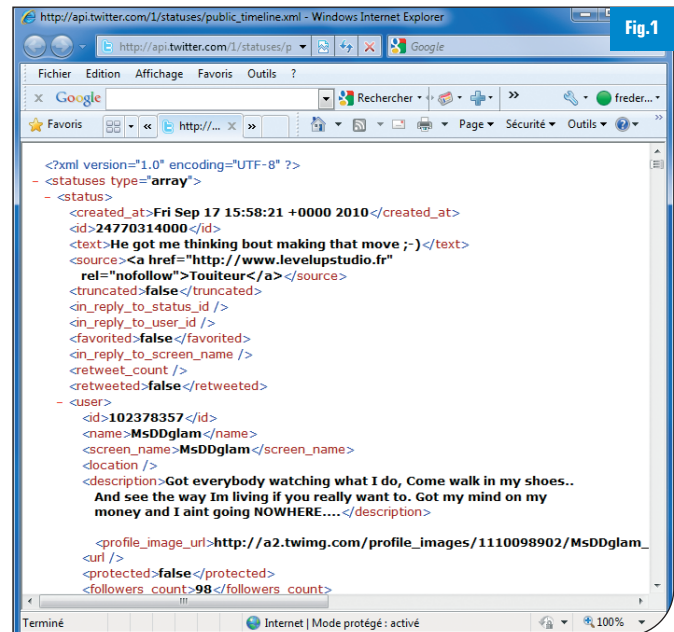
Les réseaux sociaux sont plus que jamais à la mode et modifient le paysage d'Internet. Parmi ceux qui font fureur, nous avons Twitter, un service de micro-blogging. Ce service permet de publier, dans un flux, des messages très concis de 140 caractères au maximum. Chaque utilisateur détenteur d'un compte est détenteur de son flux de messages. Il peut y joindre, en s'y abonnant les messages des flux d'autres utilisateurs, et les autres utilisateurs peuvent faire de même. Au départ, Twitter est conçu pour que les flux de messages soient visionnés dans un navigateur. Mais il n'est pas interdit de préférer une autre solution, par exemple afficher les flux de messages dans une application de bureau. Twitter met pour cela une API à disposition. Nous allons étudier le maniement de cette API et donner les informations nécessaires afin que chacun puisse construire l'application qui lui convient. Pour les besoins de cet article nous avons créé deux comptes twitter, et abonné le second au flux du premier. Faites quelque chose de semblable pour vos expérimentations. Ces comptes sont maintenant désactivés, ce qui signifie que les tokens montrés dans cet article ne sont plus valables.

## Une API en REST

L'API exposée par Twitter est une API en REST ce qui la rend accessible à n'importe quel langage, du moment que l'on peut ouvrir une connexion avec un socket. C'est l'aspect aisé du travail avec Twitter. On ne peut pas en dire autant du mécanisme d'authentification imposé depuis août 2010: OAuth. Ce mécanisme complexe est un sujet à part entière. Nous allons simplement nous débrouiller avec lui aujourd'hui.

## Qu'est-ce que REST ?

REST, ou REpresentational State Transfert n'est pas un langage mais une architecture. En fait c'est ni plus ni moins que l'architecture du Web. Sur le Web une ressource est identifiée par une URI ou identifiant unique. On y accède par une URL, qui permet de localiser la ressource, et le protocole de communication. Ainsi dans :<http://www.programmez.com/index.php> la ressource est index.php, un fichier qui contient du code PHP. La ressource est localisée dans le domaine [www.programmez.com](http://www.programmez.com) et le protocole est http. Ce principe s'applique à l'API de Twitter. On accède à la documentation en cliquant sur le minuscule mot API en bas des pages de Twitter. Ceci vous amènera aux pages dédiées aux développeurs. Cliquez alors sur documentation dans le menu. Vous arrivez à une page dont la colonne de droite énumère les ressources disponibles. Il y a des ressources pour tous les aspects du travail avec Twitter : comptes utilisateurs, abonnements, etc. Au centre se situent les ressources 'timeline', ce qui dans le jargon de Twitter signifie les flux de messages. Le flux de messages de la page d'accueil est public. Cette ressource est pointée par `statuses/public_timeline`, un status dans le jargon de twitter dési-



gnant simplement un "tweet", c'est-à-dire un message. Pour accéder à cette ressource, nous devons connaître le domaine qui est `api.twitter.com`. Enfin nous devons savoir que nous devons ajouter le numéro de version de l'API puis spécifier dans quel format nous désirons recevoir les données parmi les possibilités que sont xml, json, rss, et atom. Choisissons par exemple XML. L'URL finale est [http://api.twitter.com/1/statuses/public\\_timeline.xml](http://api.twitter.com/1/statuses/public_timeline.xml). Il suffit de saisir cette URL dans un navigateur, qui recevra aussitôt les données comme illustré [Fig.1].

## Le protocole http

Lorsque nous saisissons une URL dans un navigateur, celui-ci émet une requête HTTP vers le serveur qui détient la ressource, puis il reçoit la réponse de ce dernier. Ce protocole a été décrit plus en détail dans l'article consacré à Google Analytics dans Programmez! 131. Voici le code Java qui constitue une requête GET conforme à ce protocole, l'envoi à Twitter via un socket puis lit la réponse. Nous obtenons ainsi les flux publics de Twitter au format XML, comme nous l'avons obtenu avec le navigateur.

```
System.out.println(requete);
try {
    InetAddress twitter =
        InetAddress.getByName("api.twitter.com");
    SocketFactory socketFactory =
        (SocketFactory) SocketFactory.getDefault();
    Socket s = socketFactory.createSocket(twitter, 80);
    BufferedReader in = new BufferedReader(
```





```

    new InputStreamReader(s.getInputStream());
    DataOutputStream out =
        new DataOutputStream(s.getOutputStream());
    out.writeBytes(requete);
    String responseLine;
    while ((responseLine = in.readLine()) != null) {
        System.out.println(responseLine);
    }
    in.close();
    out.close();
    s.close();
} catch (UnknownHostException uex) { uex.printStackTrace(); }
} catch (IOException e) { System.out.println("Erreur" + e); }
} catch (Exception ex) { ex.printStackTrace(); }
}
}

```

Il est possible d'affiner une requête en y ajoutant des paramètres qui sont généralement optionnels. Ici, l'exemple propose en commentaire une requête alternative pour réduire les volumes de données relatives à l'utilisateur avec le paramètre `trim_user`:

```
GET /1/statuses/public_timeline.xml?trim_user=true HTTP/1.1\r\n
```

## Travailler avec OAuth

OAuth est un protocole qui, lorsqu'il se déroule bien, fournit à celui qui veut s'identifier deux jetons ou 'token'. Ces tokens sont des chaînes de caractères complexes, humainement illisibles. Programmer une authentification OAuth est parfaitement ennuyeux. Mais avec Twitter il est possible de contourner la difficulté, car la validité des jetons n'est pas limitée dans le temps. Une fois ceux-ci obtenus au moyen d'un utilitaire tiers, nous pourrions les utiliser dans notre code. Cependant, la première chose à faire est d'enregistrer une application Twitter. Une application Twitter n'a d'application que le nom. Il s'agit d'une entité totalement opaque qui n'est finalement qu'un point d'accès pour s'authentifier avec le protocole OAuth. Pour enregistrer cette application, rendez-vous sur <http://dev.twitter.com>, puis cliquez sur 'Register an App'. Renseignez le formulaire. Choisir 'Client' comme type d'application et choisir un accès en lecture/écriture (Read & Write). Comme nous créons une application de bureau, nous ne devons pas donner d'URL de rappel (Callback URL). Une fois l'opération terminée, consultez les détails de votre application. Vous trouverez 2 tokens sous les noms de 'Consumer Key' et Consumer Secret' respectivement. [Fig.2] Nous les utiliserons bientôt.

## Twurl

Twurl est un client pour le service Twitter à la manière de Curl pour les API REST en général. Twitter propose ce service en ligne. Pour y accéder, depuis la page récapitulative de vos applications, cliquez sur le lien Twurl console. En utilisant ce service, souvent fort capricieux, vous pourrez expérimenter avec l'API. Un intérêt majeur est qu'il montre les requêtes http générées, ce qui vous sera éventuellement très utile, si vous voulez écrire vous-même des requêtes avec authentification. Partisans du moindre effort, nous optons pour une autre solution, nous allons utiliser le twurl qui vient avec le langage Ruby. Nul besoin de connaître ce langage,

du moment qu'il est installé sur votre machine. Il suffit alors d'installer twurl : **gem install twurl**

Maintenant, attaquons la procédure pour obtenir nos tokens permanents. Nous commençons par utiliser les 2 tokens récoltés précédemment (le copier/coller est notre ami -) :

```
twurl authorize --consumer-key=3J216t8ou8mJTD7jX4pw --consumer-secret=B8HqE30Gy52bK2yxwycYsXdxXLOGX5BTxKvJn7zRUU
```

A ce moment twurl émet une sortie absconse et attend que vous saissiez quelque chose :

```
Go to https://api.twitter.com/oauth/authorize?oauth_callback=
oob&oauth_consumer_key=3J216t8ou8mJTD7jX4pw&oauth_nonce=PNM7jIgXxG0zNrrSWDkVuF
cYvtF4jf9xtV0cZpCY&oauth_signature=oAHj3nHPBFbu43GisZICMad
rQQ%253D&oauth_signature_method=HMAC-HA1&oauth_timestamp=12
84757958&oauth_token=zKicX1tygPCATUyPn6ekYfKTvyc6fe2ySDqwOc
0C0&oauth_version=1.0 and paste in the supplied PIN
```

Faites ce qui est demandé: copiez/collez cette URL gigantesque dans un navigateur. Vous verrez alors apparaître miraculeusement une page affichant un code PIN, une fois que vous aurez cliqué sur un bouton 'Autoriser'. [Fig.3] Saisissez alors le code PIN et validez. Vous devriez alors voir le message 'Authorization successful'. L'affaire est dans le sac: les tokens d'identification permanents sont déposés par twurl dans un fichier nommé `.twurlrc` et situé dans votre répertoire utilisateur. Nous pouvons alors utiliser les librairies de notre choix parmi celles proposées sur le site de Twitter. Mais vous pouvez aussi facilement vous faire la main sur l'API avec twurl. Par exemple :

```
twurl /1/statuses/user_timeline.xml
```

## Avec Python

Nous choisissons `auto-python-twitter2` qui est constitué d'un seul fichier: `oauthtwitter.py`. Il y a un petit bug dedans: modifiez la ligne :

```
url = req.get_normalized_http_url()
```

en :

```
url = req.normalized_url
```

En outre, vous devez avoir les modules `simplejson` et `oauth2` installés, ce qui est une formalité. Utilisez par exemple le script `easy_install` de votre distribution Python :

```
easy_install simplejson
easy_install oauth2
```

Voici maintenant un script qui affiche le flux de vos tweets conjoints à ceux auxquels vous êtes abonnés, puis qui poste deux tweets. Bien remarquer l'initialisation qui utilise les 4 tokens: le consumer-key, le consumer secret et les deux tokens permanents.

```
import oauthtwitter
twitter = oauthtwitter.OAuthApi("3J216t8ou8mJTD7jX4pw",
    "B8HqE30Gy52bK2yxwycYsXdxXLOGX5BTxKvJn7zRUU",
    "190776305-Xcwp3vOMwpa1RjQgZLMLNjXyexReZNBqkuawb3b",
    "oNLFF2kvzBt3bnvyk3NjnF3oLEyG4iK0Un24UjPG5c")
data = twitter.ApiCall("statuses/friends_timeline", "GET")
print data
```



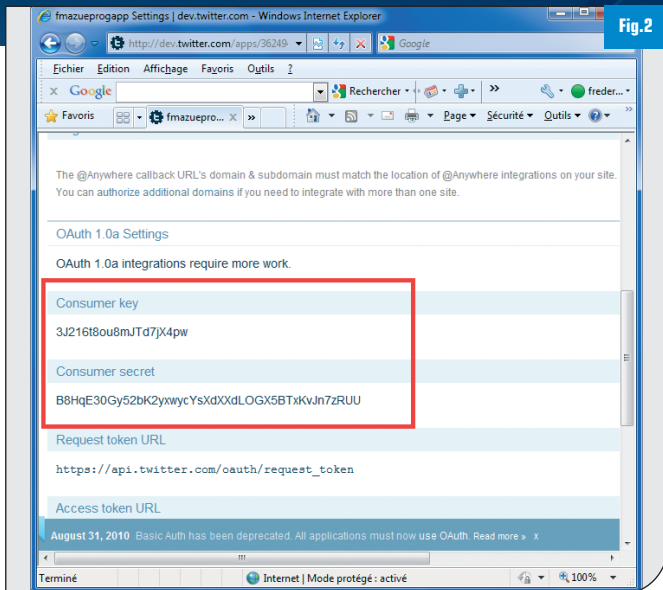


Fig.2



Fig.3

```
print [item['text'] for item in data]
print [(item['user']['screen_name'], item['user']['id']) for
item in data]
print
data = twitter.ApiCall("statuses/friends_timeline", "GET",
{'trim_user' : 'true'})
print data
print [item['text'] for item in data]
# avec trim_user = true le screen_name disparaît
print [item['user']['id'] for item in data]
# tweet !! :- )
twitter.UpdateStatus('Python rules :- )')
twitter.ApiCall("statuses/update", "POST", {'status' :
'Python rules with params'})
```

Ce module Python super malin nous donne le choix : ou préciser ressources, paramètres et méthode (GET ou POST) ou utiliser des méthodes de plus haut niveau, comme ici UpdateStatus pour poster un tweet. Mais ce module a un intérêt encore plus grand : il est facile d'y lire dans le code comment il procède pour constituer la partie relative à OAuth dans les requêtes http.

### Avec Java

Là encore il existe plusieurs bibliothèques. Je vous recommande Twitter4j car elle est très aboutie et elle nous permet d'utiliser les tokens permanents sans souci :

```
package com.programmez.twitter;
import twitter4j.ResponseList;
import twitter4j.Status;
```

```
import twitter4j.Twitter;
import twitter4j.TwitterFactory;
import twitter4j.http.AccessToken;

public class DemoTwitter4J {
    public static void main(String args[]) throws Exception {
        TwitterFactory factory = new TwitterFactory();
        Twitter twitter = factory.getInstance();
        twitter.setOAuthConsumer("3J216t8ou8mJTd7jX4pw",
            "B8HqE30Gy52bK2yxwycYsXdxXLOGX5BTxKvJn7zRUU");
        AccessToken accessToken =
            new AccessToken("190776305-Xcwp3vOMwpa1RjQgZLMLLNjXyex
            ReZNBqkuawb3b",
            "oNLFF2kvzBt3bnvyk3NjnF3oLEyG4iK0Un24UjPG5c");
        twitter.setOAuthAccessToken(accessToken);
        ResponseList<Status> statuses = twitter.getHomeTimeline();
        for (Status status : statuses) {
            System.out.println(status.getUser().getName()
                + ":" + status.getText());
        }
        Status status = twitter.updateStatus("Une tasse de café ?");
        System.exit(0);
    }
}
```

Cet exemple affiche les tweets (texte et auteur) qui apparaissent sur notre page Twitter, puis poste à son tour un petit message. Bon tweeting ! :-)

■ Frédéric Mazué - [fmaue@programmez.com](mailto:fmaue@programmez.com)

## Nouveau : économisez jusqu'à 50%

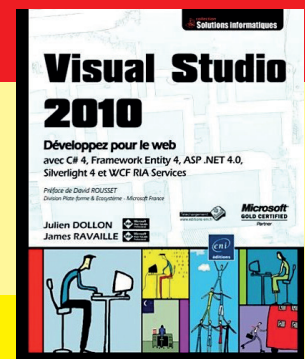
### Abonnement 2 ans au magazine + 1 livre numérique ENI

• **79€ 2 ans au magazine** au lieu de 130,90 (valeur de 22 numéros) *Tarif France métropolitaine*  
+ un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€ 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI**

**Livres à Choisir :** • Visual Studio 2010 • PHP5.3 • Bing Maps • MySQL 5, Administration et optimisation  
• Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring.

**Coupon d'abonnement p. 77 et sur [www.programmez.com/abonnement.php](http://www.programmez.com/abonnement.php)**





# REST, Restlet : garder le contact

Tout le monde connaît les concepts de base du Web à savoir : les URL, les documents hypertextes, les navigateurs et les serveurs correspondants. Actuellement le Web évolue avec l'adoption de plus en plus en plus importante d'AJAX et la venue de nouvelles techniques telles que HTML 5 ainsi que de nouveaux modes d'accès (smartphones, tablettes, objets communicants, etc.) avec toutes les implications qui s'en suivent en termes de complexité de développement.

**D**ans ce contexte, REST apparaît comme une solution innovante et simplificatrice capable de tirer pleinement parti des possibilités et choix faits initialement par les concepteurs du Web et de son protocole HTTP. Elle décrit des principes d'architecture pour les applications distribuées sur Internet et se positionne également comme une alternative au style RPC, notamment les services web basés sur SOAP.

Cette architecture est d'autant plus d'actualité avec l'usage du cloud computing, concept visant à abstraire l'infrastructure matérielle, afin d'offrir une plate-forme web de développement aux capacités de traitement et de stockage extensibles et consommables à la demande.

Nous nous baserons pour cet article sur Restlet Framework afin d'illustrer les concepts constitutifs de l'architecture REST et de décrire sa mise en œuvre. Cet outil propose un socle léger et très flexible afin d'exposer et de consommer des applications RESTful et par là même de tirer parti de toute la puissance du Web.

## Qu'est-ce que REST?

REST provient d'une thèse écrite par un des fondateurs du Web, Roy T. Fielding, et notamment d'un chapitre en particulier dénommé « REpresentational State Transfer (REST) ». Y sont décrits les concepts et les choix des fondateurs du Web et sont discutées les architectures applicatives distribuées. Sont abordés notamment les concepts de composants, connecteurs, ressources et représentations.

REST est de plus en plus utilisé pour la réalisation d'architectures orientées services utilisant des services web destinés à la communication entre applications. Il est important de remarquer que l'architecture n'impose pas de technologies ou langages aussi bien au niveau du client que du serveur, permettant ainsi une interopérabilité entre systèmes hétérogènes.

Les applications suivant ces principes REST sont appelées RESTful. Ces dernières tirent parti au maximum des principes de fonctionnement du web en utilisant le protocole HTTP. Dans la suite de l'article, nous nous concentrerons sur les notions de ressource et représentation.

## Notion de ressource

Le concept de ressource se positionne au centre de l'architecture REST. Les ressources correspondent en effet aux briques permettant de construire le Web. La notion de ressource est relativement abstraite puisqu'elle décrit n'importe quel élément digne

d'intérêt qu'une application pourrait exposer de manière distante afin qu'elle soit utilisée par d'autres applications.

Il peut s'agir d'un service bancaire pour la consultation d'un solde de compte ou d'un service de consultation de bulletin météo pour une ville. Une importante caractéristique des ressources reste néanmoins leur capacité à être liées les unes avec les autres afin de définir un ensemble distribué qui peut être navigué et découvert dynamiquement.

Chaque ressource est identifiée par un nom unique dénommé URI (Uniform Ressource Identifier) qui permet de l'identifier et de l'accéder via Internet. Une ressource expose son état sous forme de représentations. Ces représentations permettent aussi bien d'avoir accès à son état que de le modifier. Comme nous le verrons par la suite, les contenus peuvent être échangés en utilisant différents formats. La figure suivante illustre ces deux aspects :



Décrivons maintenant comment utiliser et interagir avec une ressource REST en HTTP.

## Comment gérer les états des ressources ?

Différentes méthodes sont mises à disposition afin d'interagir avec la ressource. Le protocole HTTP intègre ce type de mécanismes d'interrogation et modification de ressources par l'intermédiaire des méthodes GET, PUT, POST et DELETE.

Tout d'abord la méthode GET permet de récupérer la représentation de l'état courant d'une ressource. La méthode PUT permet quant à elle de créer ou mettre à jour des ressources. La méthode DELETE permet de supprimer la ressource. La méthode POST offre, elle, la possibilité de réaliser la création d'une ressource et de lui associer une URI.

Deux autres méthodes viennent compléter cette liste: HEAD renvoie seulement les meta données de la représentation et OPTIONS donne accès aux capacités de communication de la ressource, notamment les méthodes autorisées.

Deux autres termes sont également associés à REST. Il s'agit tout d'abord de la notion de sûreté d'une méthode décrivant qu'un état

reste inchangé après appel. GET et HEAD sont clairement sûres tandis que PUT et DELETE ne le sont pas. Dans le cas de POST, cela dépend de ce que réalise l'opération.

Le second terme correspond à la notion d'idempotence. Cette dernière signifie que le résultat sera identique suite à l'exécution répétée de la même requête. C'est le cas de GET, HEAD, PUT et DELETE. Par contre la méthode POST ne remplit pas nécessairement cette condition. C'est notamment pourquoi les navigateurs demandent une confirmation de l'utilisateur lors de la ré-exécution d'une requête POST.

Après avoir abordé les principes généraux de l'architecture REST, décrivons maintenant sa manière de négocier le type de contenu attendu pour la représentation.

## Négociation de contenu

HTTP intègre nativement un mécanisme de négociation de contenu basé sur des en-têtes afin de définir le format de données, la langue, l'encodage ou le jeu de caractères attendu pour la réponse ou tout du moins ceux supportés ainsi que des niveaux de préférence.

Ce mécanisme se fonde notamment sur l'en-tête HTTP *Accept* dans lequel peuvent être précisés le ou les formats de données supportés par le client, comme illustré ci-dessous.

```
GET /contact/1 HTTP/1.1
Accept: text/xml
Host: localhost:8182
User-Agent: Restlet-Framework/2.0.3
Content-Length: 0
```

Si la partie serveur de l'application RESTful supporte correctement la négociation de contenu, elle est alors en mesure de déterminer quel format de contenu doit être retourné en fonction de ceux qu'elle supporte. Nous avons maintenant planté le décor de l'architecture REST. Rentrions maintenant dans sa mise en œuvre avec Restlet Framework.

## Mise en œuvre d'une ressource avec Restlet

Comme nous l'avons évoqué, tous les concepts de l'architecture REST sont basés sur les concepts du Web et sont donc indépendants des langages et technologies choisis pour l'implémentation des applications RESTful.

Afin de mettre en œuvre les concepts décrits précédemment, nous avons choisi d'utiliser Restlet Framework. Ce dernier fournit une API Java simple d'utilisation et flexible afin de développer des applications RESTful et de les accéder. Cet outil est open-source et disponible à l'adresse <http://www.restlet.org/>. Modulaire, il met à disposition un ensemble d'extensions facilitant l'utilisation de technologies telles que Jackson pour JSON ou Spring dans l'implémentation de telles applications.

Ce framework couvre les différents concepts et briques décrites dans l'architecture REST et le protocole http, et fournit un cadre d'utilisation et une API très flexibles. Regardons tout d'abord la manière de définir une ressource avec ce framework.

Restlet met à disposition la classe *ServerResource* qu'il convient d'étendre afin d'implémenter ses propres traitements sur l'appel de méthodes HTTP.

L'outil offre des méthodes de traitement associées par l'intermédiaire d'annotations. Par exemple, la méthode *getContact* est

annotée avec *@Get* afin qu'elle soit accessible via une méthode GET. Ces méthodes suivent des conventions au niveau de leur signature afin de permettre la gestion des représentations.

L'exemple ci-dessous décrit la mise en œuvre d'une ressource de gestion de contacts et plus particulièrement les méthodes de récupération d'un contact et de modification de son contenu. Nous nous basons sur une représentation gérant le format JSON par l'intermédiaire de la bibliothèque Jackson.

La méthode *getContact* renvoie les données d'un contact pour un identifiant spécifié dans l'adresse d'accès au format JSON tandis que la méthode *storeContact* reçoit le nouveau contenu du contact et le met à jour. Pour l'interaction avec la source de données, un service dédié dénommé *ContactService* est utilisé.

```
public class SimpleContactServerResource
    extends ServerResource {
    private ContactService contactService
        = new ContactServiceImpl();

    @Get
    public Representation getContact(Variant variant) {
        Map<String, Object> attributes
            = getRequest().getAttributes();
        String contactId = (String) attributes.get("id");
        Contact contact = contactService.getContact(contactId);
        return new JacksonRepresentation<Contact>(contact);
    }

    @Put
    public void storeContact(
        Representation representation, Variant variant) {
        Contact contact = (new JacksonRepresentation<Contact>(
            representation, Contact.class)).getObject();
        contactService.storeContact(contact);
    }
}
```

Il est à noter que Restlet est suffisamment flexible pour masquer l'utilisation des classes *Representation* et *Variant* en paramètres et retour des méthodes. Il intègre en effet un mécanisme de conversion automatique bean / représentation. Nous avons fait le choix ici de ne pas le décrire et de montrer explicitement l'utilisation des concepts REST avec l'outil.

Une fois la ressource implémentée, il convient de spécifier à quelle adresse cette dernière pourra être accédée. Cet aspect se réalise au sein de la classe implémentant l'application Restlet et dans la méthode *createInboundRoot*. Cette dernière définit notamment les mécanismes de routage et de filtrage de la requête qui offrent la possibilité d'insérer des traitements (sécurité, filtre ...).

```
public class ContactApplication extends Application {
    public Restlet createInboundRoot() {
        Router router = new Router(getContext());
        router.attach("http://localhost:8182/contact/{id}",
            SimpleContactServerResource.class);
        return router;
    }

    public static void main(String[] args) {
        try {
            Server server = new Server(Protocol.HTTP, 8182);
            server.setNext(new ContactApplication());
            server.start();
            Thread.sleep(6000);
        } catch (Exception ex) {}
    }
}
```

Le même mécanisme est supporté pour les requêtes clientes avec la méthode *createOutboundRoot*. Dans notre exemple, il n'est pas nécessaire de redéfinir cette méthode.

Maintenant que nous avons vu comment implémenter une application RESTful avec Restlet, décrivons maintenant comment interagir avec elle à distance.





## Appel d'une ressource avec Restlet

La classe centrale de Restlet pour interagir de manière distante avec une ressource REST est *ClientResource*. Remarquez que la ressource n'est pas forcément implémentée avec Restlet ni même avec Java.

La classe utilise une adresse (URI) pour localiser la ressource puis offre des méthodes pour l'appeler avec la méthode HTTP choisie. De ce côté également les mécanismes de sérialisation / désérialisation des données se basent sur le concept de représentation. Cette dernière correspond à une représentation objet des données échangées dans le corps de la requête ou de la réponse. Le code suivant permet de récupérer simplement un contact à partir de la ressource REST implémentée précédemment.

```
public class ContactApplicationClientPut extends Application {
    public static void main(String[] args) {
        try {
            ClientResource clientResource = new ClientResource(
                "http://localhost:8082/contact/1");

            Contact contact = new Contact();
            contact.setId("1");
            contact.setNom("Nom");
            contact.setPrenom("Prénom");
            JacksonRepresentation<Contact> representation
                = new JacksonRepresentation<Contact>(contact);
            clientResource.put(representation);
        } catch (Exception ex) {}
    }
}
```

Le format utilisé ici est JSON et le contenu de la réponse est le suivant :

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Wed, 08 Dec 2010 07:31:18 GMT
Accept-Ranges: bytes
Server: Restlet-Framework/2.0.3
Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept
Content-Type: application/json

{"id":"1","nom":"Nom","prenom":"Prénom"}
```

La modification d'un contact se réalise de la manière suivante depuis le client toujours en se basant sur le format JSON pour échanger les données.

```
public class ContactApplicationClientGet extends Application {
    public static void main(String[] args) {
        try {
            ClientResource clientResource = new ClientResource(
                "http://localhost:8082/contact/1");
            Representation representation = clientResource.get();
            Contact contact = (new JacksonRepresentation<Contact>(
                representation, Contact.class)).getObject();
            System.out.println(contact.getId()
                + " - " + contact.getNom() + " " + contact.getPrenom());
        } catch (Exception ex) {}
    }
}
```

Nous n'avons montré ici qu'une infime partie des possibilités offertes par Restlet Framework mais nous voyons bien la puissance de l'outil à mettre à disposition les mécanismes de REST. Il est à noter que l'outil permet également de masquer complètement l'utilisation des représentations et d'ajouter des traitements dans la chaîne d'exécution des requêtes aussi bien côté client que serveur.

## Relations entre REST et HTTP

Comme vous avez pu le voir, REST et HTTP apparaissent très proches. Ils sont cependant de nature différente. En effet REST définit un type d'architecture tandis que HTTP correspond à un

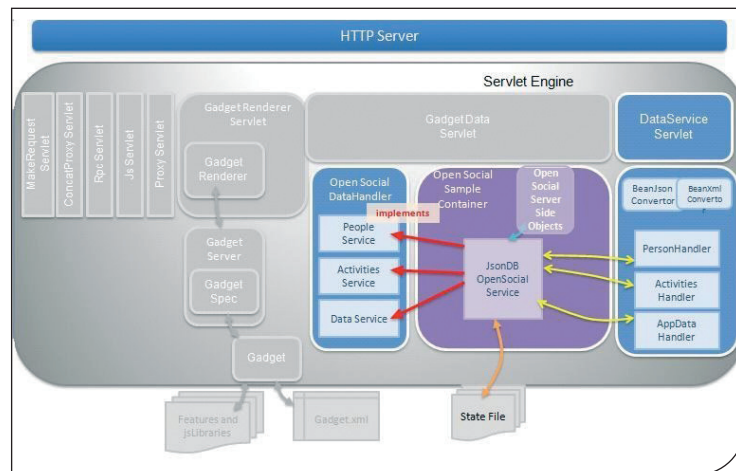


Schéma d'implémentation Rest (Java Shindig)

protocole concret de communication. De ce fait, bien que ce protocole facilite la mise en œuvre d'applications RESTful, il n'est pas nécessaire de l'utiliser pour faire du REST. De même l'utilisation de HTTP n'implique pas qu'une application soit RESTful puisqu'elle ne suit pas forcément tous les concepts décrits précédemment.

## En résumé

Dans cet article, nous avons tout d'abord décrit les concepts de l'architecture REST, architecture visant à tirer pleinement parti des concepts du Web afin de mettre en œuvre des applications distribuées et une approche orientée service sur Internet.

Le concept central consiste en la ressource intégrant différents traitements et correspondant à la brique pour construire des applications RESTful. Cette ressource intègre un état qui lui est propre et peut être interrogée via des méthodes afin d'avoir accès à cet état et le modifier. REST se pose en alternative au style architectural RPC et à la plupart des cas d'utilisation de SOAP avec ce paradigme fondé sur les ressources.

HTTP supporte également un mécanisme permettant de préciser le contenu attendu lors des réponses suite à des appels de ressources. Cette fonctionnalité porte le nom de négociation de contenu.

Bien que REST soit indépendant de la plateforme, du langage et des outils utilisés, nous avons choisi de décrire concrètement la mise en œuvre de ses principes en se basant sur le protocole HTTP et Restlet Framework. Ce dernier fournit un cadre flexible au-dessus des différents concepts de REST. Il permet très facilement d'implémenter des applications RESTful et de les appeler.

■ Jérôme Louvel ([jerome.louvel@noelios.com](mailto:jerome.louvel@noelios.com))

Leader Technique – Restlet Framework

Co-auteur de l'ouvrage *Restlet in action* (Manning) et contributeur à l'ouvrage *RESTful Web Services* (O'Reilly)

■ Thierry Templier ([templth@yahoo.fr](mailto:templth@yahoo.fr))

Architecte Java, Java EE et Web

Co-auteur des ouvrages *Spring par la pratique* et *JavaScript pour le Web 2.0* (Eyrolles).

■ Thierry Boileau ([thierry.boileau@noelios.com](mailto:thierry.boileau@noelios.com))

Développeur principal – Restlet Framework

Co-auteur de l'ouvrage *Restlet in action* (Manning)

# Architecture Restful SharePoint 2010

Une architecture Rest (REpresentational State Transfer) permet de communiquer avec des serveurs en utilisant des URI. Ainsi on peut demander l'exécution de procédure côté serveur avec juste une adresse URI. L'architecture Rest se répand de plus en plus vite à cause de sa simplicité de mise en place et sa simplicité de compréhension. Ainsi les nouveaux Frameworks comme par exemple MVC pour asp.net, utilisent l'architecture Rest.

Un exemple d'url Rest en architecture MVC est l'url <http://localhost/Controller/View>, qui indique au Framework MVC le controller et la view à utiliser pour exécuter la page. L'architecture RestFul permet de faire en plus d'une navigation en lecture, des requêtes en Lecture/Ecriture, c'est ce qu'on appelle du CRUD (Create, Read, Update, Delete). L'envoi des informations se fait en utilisant les verbes http dans l'ordre Put, Get, Post et Delete.

## ListData.svc

ListData.svc est la base de l'architecture rest de SharePoint. Ce service WCF page se trouve dans le répertoire virtuel \_vti\_bin comme tous les services WCF de SharePoint. L'appel à ce service peut donner l'erreur « Could not load type System.Data.Services.Providers.IDataServiceUpdateProvider, cela vient du fait que le framework .net 3.5 doit être mis à jour avec Ado.net Data Services update for .net 3.5 SP1 pour que ListData.svc fonctionne.

Une fois la mise à jour faite, l'appel à `listdata.svc` donne : [Fig.1](#).

Comme vous pouvez voir c'est un fichier XML qui donne l'ensemble des listes et des librairies de documents du site contextuel de SharePoint (en effet cet ensemble de listes est contextuel a ce qui précédait `/_vti_bin/listdata.svc`). Pour lister l'ensemble des items d'une liste, il suffit d'indiquer le nom de la liste à la suite de `listdata.svc` (`/_vti_bin/listdata.svc/Documents`) par exemple : [\[Fig.2\]](#).

L'architecture restfull de SharePoint est très poussée car elle permet de filtrer, ordonner, sélectionner les items de notre liste, on peut véritablement parler de requête restful.

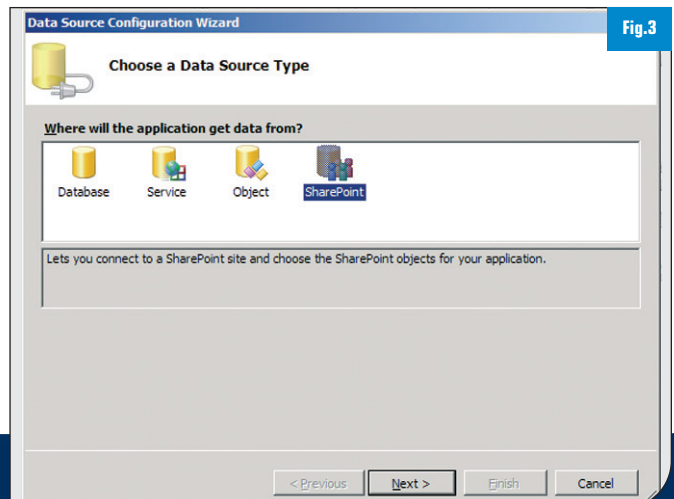
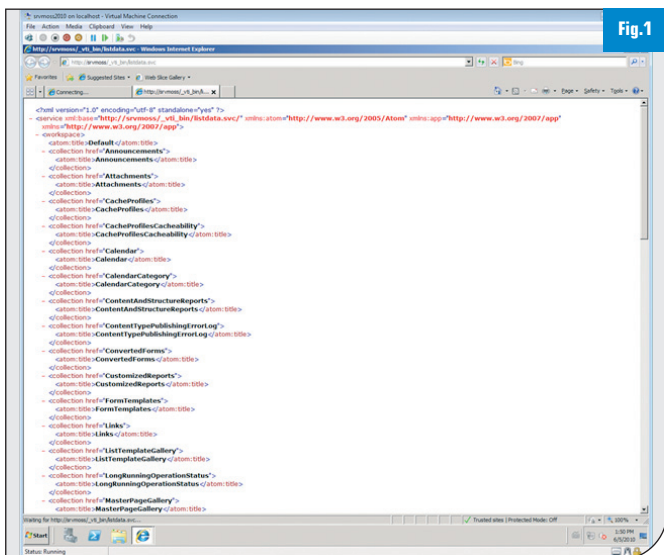
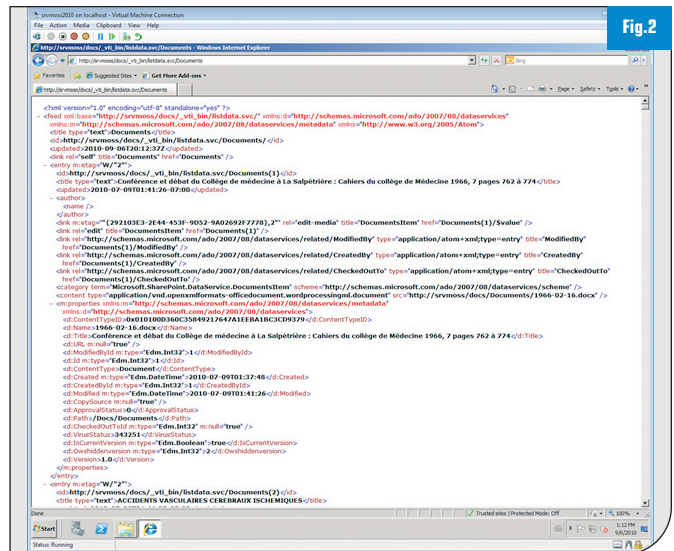
Voici quelques exemples et leur signification

Requête	Retourne
listdata.svc/Documents(15)	le 15e item de la liste
listdata.svc/Documents?\$top=10	les 10 premiers éléments
listdata.svc/Documents?\$top=10&\$skip=15	Les éléments 15 à 25
listdata.svc/Documents?\$orderby=Title	Tous les éléments Trier les éléments par « Title »
listdata.svc/Documents? \$select=Title,Modified	Seulement les propriétés « Title » et « Modified » de tous les éléments
listdata.svc/Documents?\$filter=Title eq 'Document1'	L'élément dont le Titre est égal à Document1
listdata.svc/Documents/\$count	Le nombre d'éléments de la liste

## SharePoint DataSource

La SharePoint DataSource se base sur l'architecture restFul de SharePoint. Prenons un exemple si je fais une application Windows et que je crée une nouvelle DataSource : [\[Fig.3\]](#).

On doit mettre le lien vers ListData.svc : [Fig.4].



Lorsqu'on exécute l'application, avec l'outil Fiddler on remarque les appels à l'architecture restFul de SharePoint par la DataSource SharePoint [Fig.5].

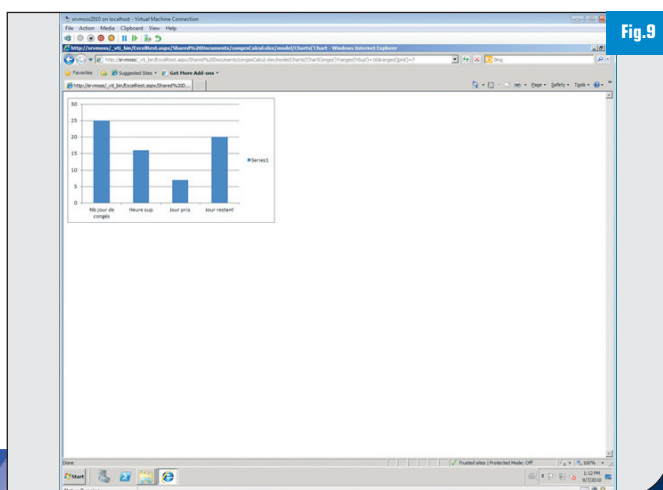
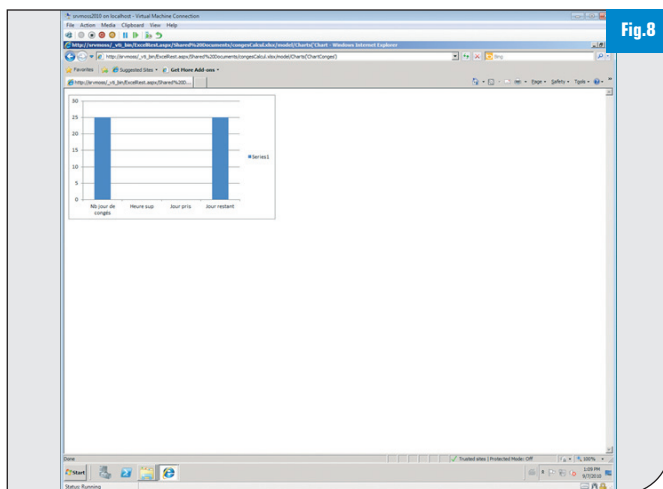
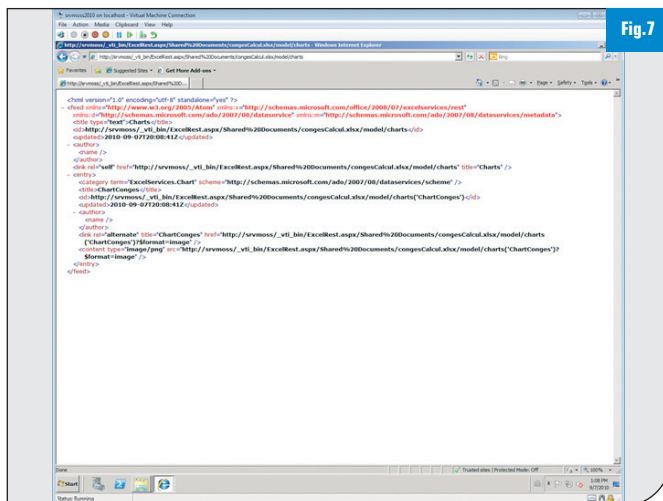
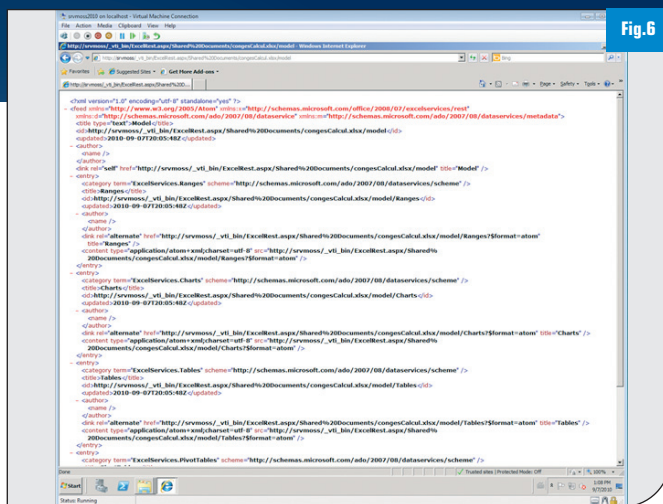
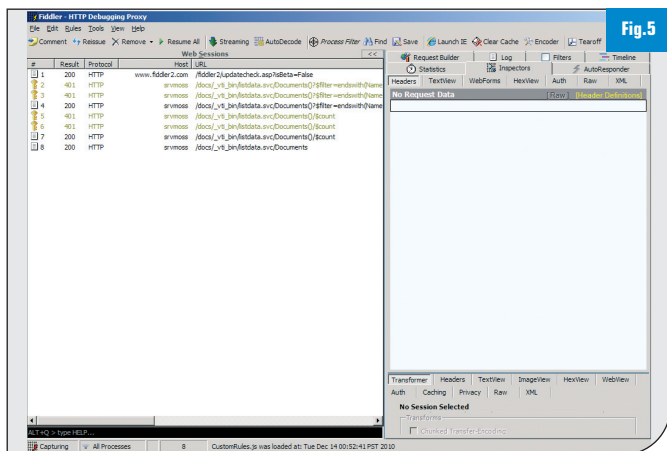
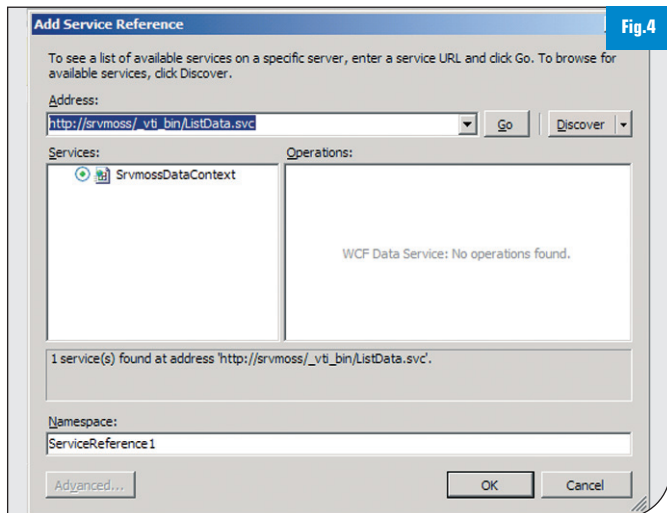
## ExcelRest.aspx

Dans la même configuration de ListData.svc, on a ExcelRest.aspx qui permet de récupérer des informations sur des classeurs Excel publiés dans SharePoint et de les manipuler pour en exécuter les calculs. Prenons un exemple : j'ai publié une feuille Excel dans SharePoint et je fais appel à :

[http://srvmoss/\\_vti\\_bin/ExcelRest.aspx/Shared%20Documents/CongesCalcul.xslx/model](http://srvmoss/_vti_bin/ExcelRest.aspx/Shared%20Documents/CongesCalcul.xslx/model) ce lien est composé de l'url vers ExcelRest.aspx, suivi du lien *relative* vers le classeur Excel et enfin de /model pour récupérer les informations sur notre classeur : [Fig.6]. Si on ajoute Charts à la fin, on a les informations sur les graphiques incluses dans le classeur : [Fig.7]. Comme on peut le voir, on a un Chart nommé ChartConges que l'on peut donc appeler en ajoutant ('ChartConges') à la fin de l'url : [Fig.8]. Ce graphique est basé sur des paramètres qu'on peut aussi mettre à jour juste avec l'url (ici je vais mettre à jour Ranges nommé HSup et jpris avec l'url : [Fig.9].

[http://srvmoss/\\_vti\\_bin/ExcelRest.aspx/Shared%20Documents/CongesCalcul.xslx/model/Charts\('ChartConges'\)?ranges\('HSup'\)=16&ranges\('jpris'\)=7](http://srvmoss/_vti_bin/ExcelRest.aspx/Shared%20Documents/CongesCalcul.xslx/model/Charts('ChartConges')?ranges('HSup')=16&ranges('jpris')=7)  
L'architecture Rest d'Excel permet aussi de récupérer les tables ou les pivots tables et donc de pouvoir utiliser Silverlight pour récupérer des graphes par exemple.

■ Xavier Vanneste - <http://blog.xvanneste.com>





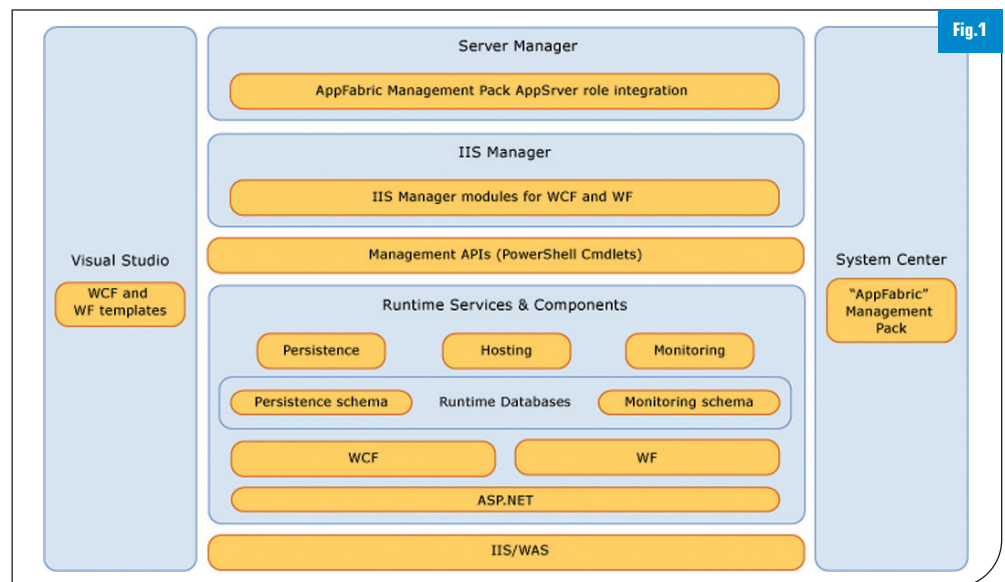
# AppFabric : l'accélérateur de développement d'applications multi-tiers

AppFabric est la solution middleware de Microsoft. Deux variantes existent aujourd'hui sur le marché : Windows Server AppFabric, dédié au développement classique et Windows Azure AppFabric, dédié au développement dans le Cloud. Même si les deux produits restent encore séparés, Microsoft est en train de réunir le meilleur des deux environnements de développement dans une nouvelle génération d'infrastructure d'application.

L'adoption du Cloud Computing par Microsoft s'est traduite en 2007 par la vision « S+S » (Software + Services), visant à tirer parti à la fois des logiciels classiques et des services logiciels disponibles sur Internet. En 2009, Microsoft a amélioré cette vision en adoptant la stratégie « trois écrans et un nuage », faisant référence aux trois écrans PC, TV, Mobile et au nuage Internet. L'objectif est de proposer une expérience utilisateur similaire sur différents types d'écrans tout en les rendant communicants grâce à Internet. Les applications monopostes, autonomes, déconnectées tendent donc à disparaître au profit d'applications massivement multi-utilisateurs, communicantes, distribuées, connectées - partiellement ou totalement - à des services divers. En effet, il est aujourd'hui banal de trouver des sites web de commerce où les fonctions de recherche, de géolocalisation des magasins, de paiement multi-support,... s'appuient sur des services existants, mis à disposition sur Internet. Les nouveaux défis à relever pour réaliser de tels types d'applications vont se concentrer autour de :

- la connectivité entre applications et services,
- la performance des applications et la réactivité perçue par l'utilisateur final,
- la sécurité mais aussi la simplicité d'accès aux applications, en adoptant si possible l'authentification unique (SSO: Single Sign-On).

Techniquement, il est tout à fait possible d'exploiter le framework .NET et la plateforme Windows pour résoudre ces problématiques, mais il reste encore souvent à écrire trop



d'infrastructure logicielle, ce qui retarde inévitablement le développement des fonctions métiers, seules visibles aux yeux de l'utilisateur.

Microsoft apporte une solution à travers deux nouveaux produits : Windows Server AppFabric et Windows Azure AppFabric, destinés à combler certaines fonctions d'infrastructure inexistantes aujourd'hui chez Microsoft, respectivement en entreprise avec Windows Server et le cloud avec Windows Azure.

## Windows Server AppFabric

Windows Server AppFabric est une extension de Windows Server 2008 SP2 et Windows Server 2008 R2 disponible depuis juin 2010, qui fournit un environnement d'hébergement très avancé ainsi qu'un cache distribué qui peut être utilisé par les applications Web et les services middle-tiers. L'environnement d'hé-

bergement délivré par AppFabric ajoute des fonctionnalités de gestion pour Internet Information Services (IIS), Windows Process Activation Service (WAS) et pour le Framework .NET 4. Le déploiement, la configuration et la gestion des services basés sur Windows Communication Foundation (WCF) et Windows Workflow Foundation (WF) sont ainsi facilités. Grâce aux fonctionnalités de cache d'objets distribués en mémoire, les performances des applications .NET et plus spécialement des applications ASP.NET peuvent être largement améliorées. La figure montre tous les composants d'AppFabric ainsi que les composantes du Framework .NET 4 et IIS qui sont utilisées par AppFabric [Fig.1].

## Hosting Services (précédemment Dublin)

La mise en place d'un environnement d'hébergement pour une application

qui comprend des services basés sur WCF et WF peut être complexe et fastidieuse. Bien qu'IIS et WAS fournissent un environnement d'hébergement robuste pour de telles applications, leur utilisation peut nécessiter un effort de développement considérable. Une fois la solution déployée et configurée, la gestion des services n'est pas aisée.

L'utilisation de l'environnement d'hébergement délivré par AppFabric facilite largement le déploiement, la configuration et le monitoring, car ils étendent WAS en ajoutant une configuration par défaut. Pour chaque nœud de serveur, AppFabric simplifie la gestion en appliquant des para-

mètres par défaut (exemples: sql-WorkflowInstanceStore (persistance), etwTracking (monitoring), service-Throttling (performance), service-Credentials/serviceCertificate (security), etc...). Le fichier « Web.config » racine contient des paramètres qui peuvent être utilisés par toute application hébergée.

Pour les services WCF et WF déjà existants et hébergés dans WAS, les services d'hébergement d'AppFabric permettent de les lister et de les gérer. Si, par contre, ces services sont hébergés dans un service Windows ou une application autonome, il faudra les migrer vers l'environnement WAS pour pouvoir tirer pleinement avantage des extensions fournies par AppFabric.

Le service d'hébergement d'AppFabric met à disposition un magasin de persistance basé sur SQL Server et un fournisseur de persistance SQL. Tout workflow hébergé dans WAS peut exploiter ces fonctionnalités, simplifiant ainsi la configuration de la persistance. Il est bien entendu possible de configurer un magasin de persistance personnalisé avec un autre fournisseur.

Les outils d'administration permettent de gérer les instances des workflows en cours en les contrôlant (verrouillage/nouvelle tentative, démarrage automatique, minuteurs durables et file d'attente de commandes) et en les supervisant. Ils permettent également de paramétrer les adresses des services, de choisir l'activation automatique ou de configurer les paramètres de sécurité et de performance [Fig.2]. Des extensions du Gestionnaire des services Internet pour WCF et WF sont disponibles dans la Console de gestion IIS standard pour configurer les tâches les plus courantes [Fig.3].

AppFabric offre également un tableau de bord dans la Console de gestion IIS qui contient les informations recueillies lors de la surveillance des applications et enregistrées dans la base de données de surveillance [Fig.4]. Cette base de données SQL Server peut, bien sûr, être requêtée manuellement. Les fonctionnalités fournies sont donc les suivantes :

- Déploiement et gestion améliorés de services WCF et WF hébergés dans WAS.
- Configuration facilitée de la persistance pour les workflows.
- Windows PowerShell cmdlets.
- Monitoring personnalisable des services hébergés.
- Démarrage automatique d'applications pour minimiser la latence d'accès aux services.

## In-Memory Distributed Cache (précédemment Velocity)

Dans des applications ayant une large volumétrie de données, la récupération de ces données est souvent un facteur de faibles performances. Afin d'améliorer cette situation, l'utilisation du cache est une méthode connue et répandue car la lecture de données en mémoire est évidemment bien plus rapide que la lecture de données depuis la base (bien que la base de données gère aussi son propre cache).

Actuellement chaque application doit implémenter son propre cache avec une configuration propre. Les impacts sur l'application d'une mauvaise implémentation du cache sont importants. Afin de standardiser la gestion d'un cache, AppFabric fournit les briques d'une implémentation complète et centralisée pour l'ensemble des applications. Le cache d'AppFabric peut être utilisé en tant que cache distribué pouvant être configuré pour s'exécuter sur plusieurs serveurs, augmentant ainsi les limites de la mémoire, améliorant encore les performances et assurant un service continu en cas d'incidents sur l'un des serveurs (haute disponibilité). Concernant la gestion du cache, des outils d'administration sont mis à disposition afin d'offrir des fonctionnalités de gestion centralisée [Fig.5].

Tout objet sérialisable peut être mis en cache, y compris les DataSets, les DataTables, les données binaires ou du XML. Il est aussi possible, dans le cadre d'ASP.NET, de stocker les sessions dans le cache. L'API .NET est très efficace, et l'implémentation est donc très facile à mettre en œuvre. AppFabric Cache, solution gratuite et complètement intégrée dans Win-

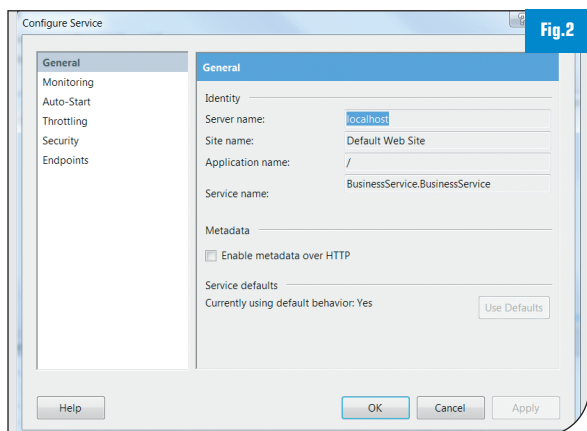


Fig.2

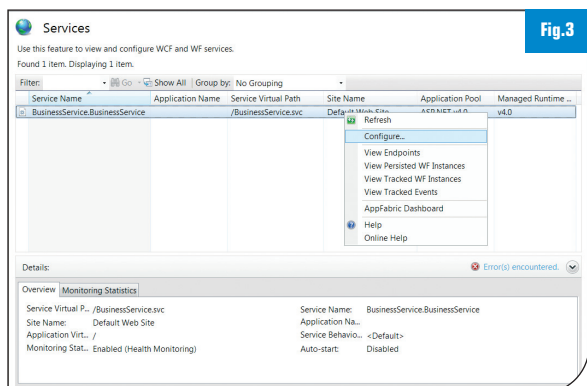


Fig.3

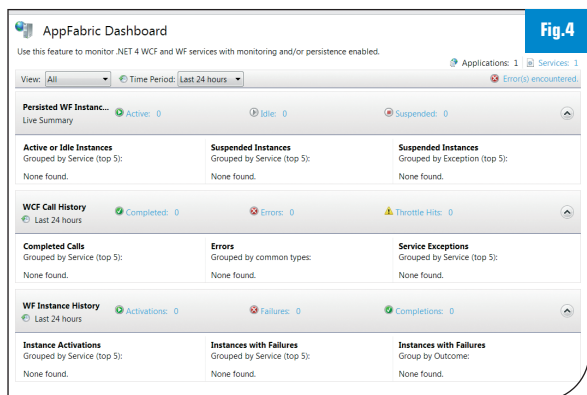


Fig.4

dows, trouve sa place parmi les autres solutions du marché telles que Oracle Coherence, NCache, ScaleOut ou Memcached (open source).

## Windows Azure AppFabric

La plateforme Azure contient plusieurs produits :

- Windows Azure : la plateforme d'hébergement d'applications et services web,
- SQL Azure : la plateforme d'hébergement de données relationnelles,
- (Windows Azure) AppFabric : la plateforme d'intégration entre services en entreprise et sur Internet.

Windows Azure AppFabric couvre deux sujets techniques souvent très critiques : la connectivité entre services et le contrôle d'accès aux applications.

## Service Bus

Connecter des applications et/ou des services dans le réseau d'une entreprise est chose courante et relativement aisée. Par contre, lorsque les applications ou les services sont hébergés sur Internet ou sur des réseaux différents, deux principaux problèmes apparaissent : le franchissement des firewalls et des NAT, rendant ainsi la communication très difficile à mettre en œuvre. Pour éviter cet obstacle il existe plusieurs solutions : déplacer les applications ou les services dans la DMZ, configurer de manière spécifique les reverse proxies, développer des couches de communication réseau ad-hoc... Dans tous ces cas, les manipulations sont lourdes, difficiles et coûteuses. Le Bus de Service est une infrastructure de communication intermédiaire, permettant de gérer tous ces problèmes à votre place et évitant ainsi

d'intervenir sur la topologie des réseaux [Fig.6].

Voici les principales fonctionnalités apportées par le Bus de Service :

- Franchissement des firewalls et NAT.
- Communication bidirectionnelle depuis tout équipement connecté à Internet.
- Mise en relation directe lorsque c'est possible.
- Mise en buffer des messages pour les applications faiblement connectées.
- Mise en tunnel de protocoles.
- Support des patterns de communication : multicast, multiplex, publish / subscribe.
- Publication et découverte des services grâce à l'annuaire du Bus.

Le tout avec l'assurance d'une solution supportant une haute-disponibilité, une montée en charge automatique, les protocoles classiques d'Internet (HTTP, REST, XML, ...) et de manière sécurisée, grâce à l'Access Control (voir ci-dessous).

Pour mettre en œuvre le Bus de Service, il est nécessaire de posséder un compte Azure (à créer auprès de Microsoft) et d'utiliser des bindings WCF particuliers pour joindre ou exposer un service au travers du Bus. (voir tableau ci-contre).

Disponible depuis le mois d'avril 2010, le Bus de Service est facturé (tarifs indicatifs) :

- 2,83 € par connexion, ou bien de 7,06 € à 705,65 € le lot de 5 à 500 connexions,
- 0,071 € in / 0,106 € out par Giga-octet de données transférées (0,213 € in / 0,319 € out en Asie)

## Access Control

Lorsqu'il s'agit d'interconnecter des

systèmes, applications ou services sur Internet, il est impératif de maîtriser les utilisateurs accrédités et leurs autorisations. Or c'est une tâche délicate en raison de la diversité technologique des applications et du morcellement fréquent des identités et des autorisations dans des annuaires disparates.

La fédération d'identité est un principe conçu pour faciliter cette gestion. Il consiste à séparer les parties impliquées lors de l'accès à une application : l'authentification, la fourniture des autorisations et l'application elle-même [Fig.7].

Les avantages sont multiples :

- l'authentification peut être déléguée à un tiers externe à l'entreprise, très pratique lorsqu'il s'agit d'authentifier des utilisateurs externes (partenaires, fournisseurs, ...) sans s'embarrasser de la gestion de leurs identités,
- la mise en œuvre du SSO (Single Sign-On) est facilitée si le même fournisseur d'authentification est réutilisé pour tout un ensemble d'applications,
- l'administration s'en trouve également simplifiée puisqu'elle peut s'appuyer sur un référentiel d'identités commun,
- enfin, les autorisations sont exprimées via un concept de Claims

Binding WCF	Binding Service Bus Relay
BasicHttpBinding	BasicHttpRelayBinding
WebHttpBinding	WebHttpRelayBinding
WSHttpBinding	WSHttpRelayBinding
WS2007HttpBinding	WS2007HttpRelayBinding
WSHttpContextBinding	WSHttpRelayContextBinding
WS2007HttpFederationBinding	WS2007HttpRelayFederationBinding
NetTcpBinding	NetTcpRelayBinding
NetTcpContextBinding	netTcpRelayContextBinding
n/a (proche de NetMsmqBinding)	NetOnewayRelayBinding
n/a (proche de NetTcpPeerBinding)	NetEventRelayBinding

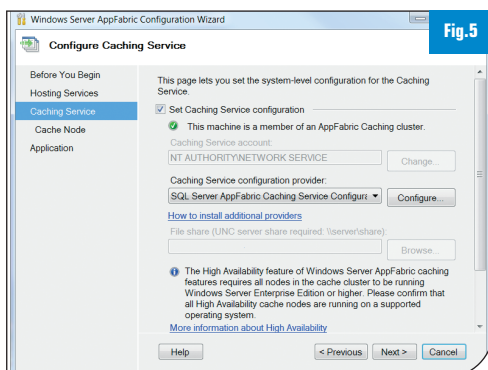


Fig.5

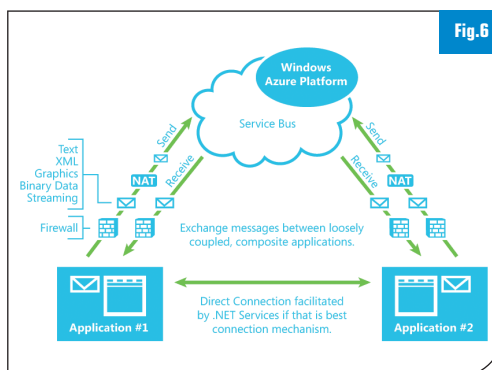


Fig.6

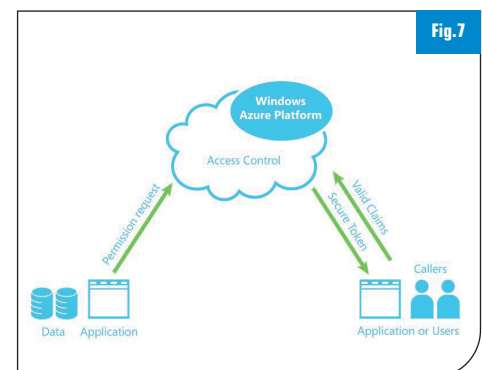


Fig.7



(revendications), assez généraliste pour représenter tout type de permission.

Access Control est un élément de la fédération d'identité, présent dans Windows Azure, dont l'objectif est de fournir des revendications pour une application donnée en fonction d'un ensemble de règles. A ce titre, il est qualifié de STS : Security Token Service. Ses fonctionnalités sont :

- Déléguer l'authentification à un gestionnaire d'identités externe
  - typiquement en se basant sur un annuaire connu : Active Directory, LiveID, OpenID (Google, Yahoo), Facebook...
  - en utilisant des protocoles standards : OAuth WRAP, WS-Trust, WS-Federation
  - et des formats standards : SAML 1.1/2.0, SWT
- Transformer les revendications obtenues suite à l'authentification en d'autres revendications cette fois-ci utiles à l'application, à savoir les autorisations compréhensibles par l'application.

Pour mettre en œuvre Access Control, il est bien sûr nécessaire de posséder un compte Azure et de configurer les transformations des revendications au moyen d'un outil fourni avec le SDK : Access Control Service Management Browser.

L'Access Control est facturé 1,41 € toutes les 100 000 transactions (tarif approximatif, sans compter les données transférées au même tarif que le Service Bus).

## Comparaison

Bien que portant le même nom, les deux AppFabric sont des produits aux fonctionnalités bien différentes. C'est également le cas en termes de packaging : Windows Server AppFabric est inclus dans la licence Windows Server et doit être installé sur la machine, alors que Windows Azure AppFabric est disponible dans les DataCenters de Microsoft et est facturé à la consommation (volumes échangés, nombre de transactions, ...). Le choix d'un nom identique ne peut évidemment pas être justifié sur la base de ces éléments. Pour comprendre ce choix, il faut revenir à l'objectif général

de AppFabric : il s'agit de faciliter la gestion et le développement d'applications communicantes en apportant des solutions d'infrastructures logicielles, jusqu'alors développées de manière spécifique ou basées sur des produits d'éditeurs tiers. Or, les infrastructures logicielles actuelles sont différentes en entreprise et sur Internet, car les attentes et les contraintes ne sont pas les mêmes. Sur Internet par exemple, les défis sont la montée en charge ou l'interopérabilité multiplateforme, tandis qu'en entreprise ils sont concentrés autour des performances brutes ou le suivi des processus métiers vitaux. Les deux produits AppFabric apportent donc des compléments différents, pour répondre au mieux aux problématiques de l'entreprise et de l'Internet. Il faut donc retenir que les deux AppFabric ont une même finalité, mais avec des moyens différents parce que les contextes sont différents.

Deux autres raisons, bien plus pragmatiques, expliquent que les fonctionnalités des deux produits sont aujourd'hui différentes :

- Il faut composer de fait avec l'histoire logiciel de Microsoft : certaines fonctionnalités de Windows Azure AppFabric ne sont pas présentes dans Windows Server AppFabric parce qu'elles sont déjà couvertes par d'autres produits en entreprise. C'est le cas pour :
  - Le Service Bus d'Azure dont certaines fonctionnalités sont couvertes par Biztalk Server en entreprise,
  - De même pour la fédération d'identité, dont Active Directory Federation Services représente le maillon de transformation de revendications en entreprise.

En termes de composantes d'infrastructure logicielle disponible, il est donc important de prendre en compte AppFabric mais aussi d'autres produits Microsoft fonctionnellement proches.

- La plateforme Azure est très récente et les systèmes Windows Server et Windows Azure n'ont pas encore atteint la symétrie promise. La plateforme Server ayant une étape

d'avance, il en est de même pour son AppFabric, l'objectif de symétrie annoncé par Microsoft se concrétisera donc progressivement par un enrichissement de Windows Azure AppFabric.

Lors de la PDC (Professional Developers Conference) le 28 octobre dernier, Microsoft a notamment annoncé de nouvelles fonctionnalités pour Windows Azure AppFabric qui seront disponibles dès la fin 2010 ou courant 2011 selon les cas. Parmi elles, il est prévu d'avoir un cache distribué (comme dans Windows Server AppFabric), mais aussi une extension pour Biztalk Server 2010 (Connect) facilitant l'utilisation des services exposés sur le Windows Azure Services Bus.

## Conclusion

Cet article a présenté ce que contiennent les deux produits : Windows Server AppFabric et Windows Azure AppFabric. Très différents en apparence, ces produits offrent tous deux des facilités pour le développement des couches middle-tiers en apportant une infrastructure logicielle adaptée respectivement au monde de l'entreprise et de l'Internet.

Les nouveautés annoncées à la PDC 2010 confirment une meilleure symétrie des deux produits, mais surtout une meilleure fluidité entre eux. Le Cloud Computing n'ayant pas pour vocation à remplacer toutes les applications actuellement en entreprise, on s'oriente vraisemblablement vers le modèle hybride où les applicatifs des deux environnements devront communiquer davantage et plus facilement. Et c'est là tout l'intérêt d'AppFabric !

■ Jason De Oliveira  
Practice Manager &  
Solutions Architect, .Net  
Rangers by Sogeti  
Blog :  
<http://jasondeoliveira.com>

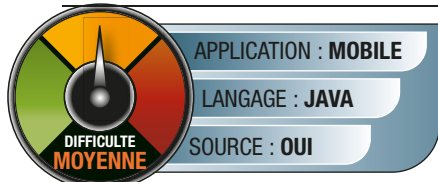


■ Jean-Luc Boucho  
Solutions Architect & Leader  
Windows Azure à Winwise  
Blog :  
<http://blog.jeanlucboucho.com>

# La persistance sous Android

Cet article est le troisième d'une série consacrée à Android pour couvrir les plus grands thèmes de la programmation Android. Le premier article concernait la mise en place de votre environnement de développement. La dernière fois, nous avons abordé différents aspects de la conception d'interfaces graphiques

Aujourd'hui c'est la persistance qui est au programme. Le chargement, la sauvegarde et la recherche d'informations sont des parties évidemment moins remarquées dans une application, mais ces fonctionnalités sont au cœur de nombreux cas d'utilisation.



## LES DIFFÉRENTS TYPES DE PERSISTANCE

Le concept de persistance est assez vague si l'on ne

précise pas un peu son contour. Dans cet article nous allons aborder ces différents types de persistance :

- les préférences,
- les fichiers,
- les bases de données.

La documentation aborde des notions réseau en même temps. Il est vrai que les différents mécanismes de connexion et de transfert de données permettent d'enregistrer et de consulter des données. Mais le sujet est assez vaste, il fera certainement l'objet d'un article dédié prochainement ! Parmi ces différentes catégories qui ne sont pas tout à fait comparables puisqu'elles n'ont pas la même utilité, nous développerons plus la partie base de données qui est à la fois la plus puissante et la moins triviale.

## GÉNÉRALITÉS SUR LA PERSISTANCE ANDROID

Avec Android, toutes les données de l'application sont privées et en suivant des bonnes pratiques de programmation, elles sont accessibles uniquement par l'application qui les gère. Néanmoins, une application Android peut exposer ses données grâce à des mécanismes proposés par le kit de développement : les fournisseurs de contenu (Content providers). Ces fournisseurs de contenu sont donc des composants (proposés par l'API pour certains ou redéveloppés si nécessaire) proposant une syntaxe standard d'échange en lecture et/ou écriture. Nous n'aborderons pas le sujet des fournisseurs de contenu dans l'article. Nous restons dans un contexte de création ou de modification des données à l'usage unique de l'application.

## LES PRÉFÉRENCES

Le mécanisme des préférences est généralement utilisé pour enregistrer les réglages de l'application (préférences utilisateur) via un système de paire clé/valeur. Pour charger ou sauvegarder des préférences, on utilise la méthode `getSharedPreferences()` du contexte. Pour les enregistrer, on a également besoin d'un objet `Editor` :

```
SharedPreferences preferences =
getSharedPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor editor = preferences.edit();
editor.putString("nom_preference", "valeur");
```

```
// Indispensable pour valider les modifications
editor.commit();
```

On remarque de suite une sorte de portée véhiculée par la constante `MODE_PRIVATE`. On peut changer cette portée pour permettre un partage vers d'autres applications, mais nous l'avons déjà dit, la bonne pratique pour le partage est l'utilisation des fournisseurs de contenu. Côté lecture, voici les quelques lignes nécessaires :

```
SharedPreferences preferences =
getSharedPreferences(Context.MODE_PRIVATE);
String preference = preferences.getString("nom_preference");
```

## LES FICHIERS

En plus des possibilités offertes par le package `java.io`, Android propose deux méthodes pour manipuler des fichiers depuis un contexte local : `openFileInput()` et `openFileOutput()`. Ces deux méthodes permettent de manipuler des flux en lecture et écriture :

```
// Ouverture d'un fichier en écriture
FileOutputStream FOS = openFileOutput("fichier.ext", MODE_PRIVATE);

// Ouverture en lecture
FileInputStream FIS = openFileInput("fichier.ext");
```

## LES BASES DE DONNÉES

Android, comme beaucoup de plateformes mobiles modernes, utilise SQLite comme moteur de base de données relationnelles. C'est un programme du domaine public, compact et rapide.

SQLite est compatible avec des standards SQL pour créer ou requêter les différentes tables. Nous allons d'ailleurs rapidement revoir les deux ou trois instructions SQL indispensables à notre application. Ensuite nous allons pratiquer pour réaliser une gestion de contacts succincte.

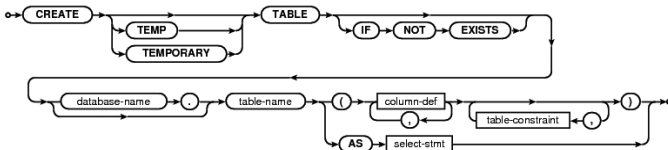
## Quelques rappels sur SQL

Pour réaliser des opérations sur la structure de la base, manipuler des données ou même régler les accès aux bases, on utilise le langage SQL. Ce langage est divisé en trois grandes parties :

- DCL (Data Control Language) : gestion des règles d'accès aux données,
- DDL (Data Definition Language) : gestion de la structure de la base (tables, vues, ...),

- DML (Data Manipulation Language) : interrogation de la base de données et altération de ces données.

Nous allons revoir quelques notions concernant les deux derniers volets de ce langage afin de créer la structure de la base de données, d'y insérer des données et enfin de consulter le contenu de la base. Du côté de la création de la base de données elle-même, nous n'utilisons pas (en Android) de commande SQL mais du code Java. Nous passons donc à l'étape suivante concernant la structure de la base. La commande la plus importante est CREATE TABLE. Elle permet la création d'une table en spécifiant les différentes colonnes (les différents champs) d'une ligne (d'un tuple). La syntaxe complète est :

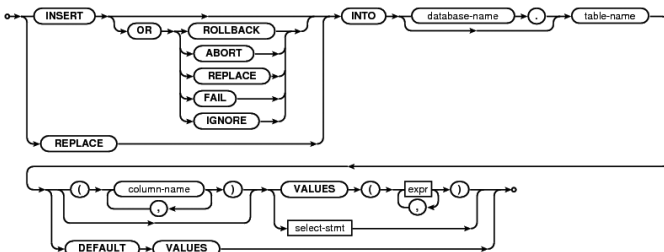


Syntaxe de la commande CREATE TABLE extraite de la documentation.

Sa syntaxe simplifiée est :

```
CREATE TABLE nom_table (nom_colonne1 type_colonne1, nom_colonne2
type_colonne2, ...);
```

Ensuite, nous avons besoin d'insérer des données dans les différentes colonnes de la table. Pour cela vous connaissez certainement la commande INSERT. La syntaxe complète est :

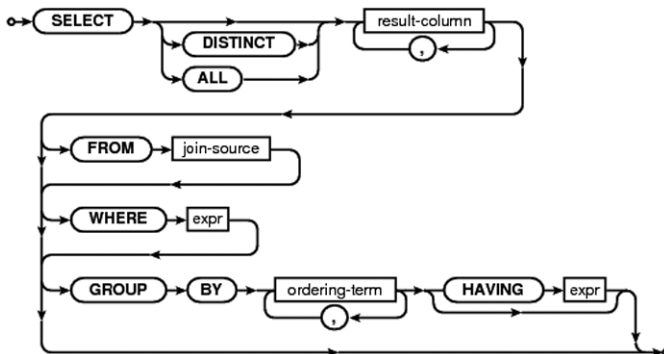


Syntaxe de la commande INSERT extraite de la documentation.

Sa syntaxe simplifiée :

```
INSERT INTO nom_table (nom_colonne1, nom_colonne2, ...) VALUES
(valeur1, valeur2, ...);
```

Il ne reste plus qu'à sélectionner des éléments présents dans une table de la base et c'est la commande SELECT qui va nous le permettre. Une partie de la syntaxe de cette commande est :



Syntaxe partielle de la commande SELECT extraite de la documentation.

Sa syntaxe simplifiée pour nos besoins :

```
SELECT nom_colonne1, nom_colonne2, ... FROM nom_table;
```

## Créer une liste de contacts

Nous allons créer une base simple qui contiendra des contacts avec des champs nom et prénom renseignés.

Pour faire bien les choses, nous devons créer *SQLiteOpenHelper* qui permet de créer la base si elle n'existe pas ou de la mettre à jour en fonction de la version présente et de la version attendue. Ce mécanisme de mise à jour est assez puissant.

```
package com.programmez.android;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase.CursorFactory;

public class DBHelper extends SQLiteOpenHelper {
    private static final String REQUETE_CREATION =
        "CREATE TABLE IF NOT EXISTS contacts (nom VARCHAR, prenom
        VARCHAR);";
    private static final String
        REQUETE_INSERTION = "INSERT INTO contacts (nom, prenom)
        VALUES ('Guignard', 'Damien');";

    public DBHelper(Context context, String name, CursorFactory factory,
        int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(REQUETE_CREATION);
        db.execSQL(REQUETE_INSERTION);
    }

    @Override
    public void onUpgrade(SQLiteDatabase bd, int ancienneVersion,
        int nouvelleVersion) {
        // Prévoir le cas de l'upgrade non pertinent pour notre exemple !
    }
}
```

Nous ne nous occupons ici que de la création de la table avec l'insertion d'un tuple pour les tests. Ensuite, toujours dans l'idée de bien architecturer notre application, nous allons créer un adaptateur qui contiendra toutes les méthodes nécessaires à la manipulation de la base : méthodes pour ouvrir, fermer ou requêter la base. Ici nous n'implémentons que quelques méthodes :

```
package com.programmez.android;

import java.util.ArrayList;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

public class BDAdapter {
```



```

private static final String DATABASE_NAME = "Contacts";

private DBHelper helper;
private SQLiteDatabase database;

public BDAdapter(Context context) {
    helper = new DBHelper(context, DATABASE_NAME, null, 1);
}

public void open() {
    database = helper.getWritableDatabase();
}

public void close() {
    database.close();
}

public void insert(String nom, String prenom) {
    database.execSQL("INSERT INTO contacts (nom, prenom), VALUES ('" + nom + "', '" + prenom + "')");
}

public ArrayList<String> getPrenoms() {
    ArrayList<String> prenom = new ArrayList<String>();
    Cursor cursor = database.query(DATABASE_NAME,
        new String[] {"prenom"}, null, null, null, null, null);
    if (cursor.moveToFirst()) {
        do {
            String prenom = cursor.getString(0);
            prenom.add(prenom);
        } while (cursor.moveToNext());
    }

    return prenom;
}
}

```

Il est possible de générer un peu plus ce code, mais pour l'essentiel, tout est bien découpé en méthodes qui se servent de la référence à la base obtenue par notre helper. La méthode `open()` ouvre la base en écriture. C'est l'occasion de tout de suite rappeler qu'il ne faut pas oublier de fermer proprement cette connexion grâce à la méthode `close()`. Côté requête, la méthode `getPrenoms()` récupère les prénoms présents dans la base grâce à un curseur. Pour finaliser cette application, il ne reste plus qu'à réaliser la chaîne complète des méthodes implémentées et à présenter le résultat dans une liste :

```

package com.programmez.android;

import java.util.ArrayList;

import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;

public class ListeContacts extends ListActivity {

```

```

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    BDAdapter adapter = new BDAdapter(this);
    adapter.open();
    ArrayList<String> prenom = adapter.getPrenoms();
    setListAdapter(new ArrayAdapter<String>(
        this, android.R.layout.simple_list_item_1, prenom));
    adapter.close();
}
}

```



Voici le résultat de l'application étudiée :  
▼ *Rendu de l'application.*

## Utiliser adb pour examiner les bases sqlite

Dans vos développements futurs, pensez que le kit de développement Android nous offre la possibilité de nous connecter directement à cette base depuis l'ordinateur de développement, en ligne de commande :

- lancez l'émulateur;
- utilisez la commande `adb shell` dans une console;
- nous allons nous rendre là où notre base est stockée en allant dans le répertoire `data/data` puis dans le répertoire correspondant à notre package (`com.programmez.android`). La commande complète est : `cd /data/data/com.programmez.android`;
- ce répertoire contient un répertoire `databases` qui lui-même contient notre base;
- entrez la commande `sqlite3 contacts.db`. Vous venez d'ouvrir la base avec SQLite;
- vous pouvez saisir des requêtes à exécuter;
- `.exit` vous permet de sortir de l'invite de commande SQLite;

## CONCLUSION

Grâce aux différentes notions vues, vous êtes maintenant à même d'appréhender la manipulation de données à l'intérieur d'une application Android.

Vous pouvez réaliser des applications qui gèrent du contenu, synchronisent de l'information.

Les idées d'applications associées vont de l'éternelle "todo list" à une application type visiteurs médicaux (saisie de commandes ou consultation offline d'informations puis synchronisation) en passant par des relevés de prix à la volée. A vous de jouer !

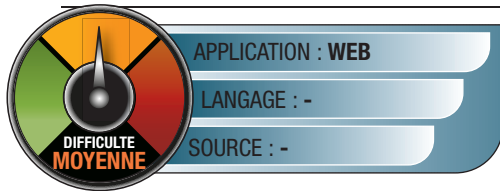
## Références

- Android par Google : <http://developer.android.com>
- SQLite Query Language : <http://www.sqlite.org>
- Pour aller plus loin avec les fournisseurs de contenu : <http://developer.android.com/guide/topics/providers/content-providers.html>

■ Damien Guignard

# Embarquer une station web

L'accessibilité au web est utile pour accéder à un site internet ou un projet en cours de développement. Cependant, lors de la présentation d'une réalisation, il peut être utile de pouvoir montrer les évolutions de votre projet ou même de pouvoir effectuer une démonstration d'un ancien projet.



Une station embarquée, permet d'avoir à portée de main, un certain nombre d'outils applicatifs que vous utilisez régulièrement pour le web, comme un serveur, un éditeur de texte, un logiciel de retouche de photos, etc.

Certains de ces logiciels sont proposés dans les systèmes d'exploitation des ordinateurs mais ne correspondent pas toujours à vos besoins et aussi ne sont pas toujours actifs. Une question se pose : lorsque vous accédez à un ordinateur qui ne vous appartient pas, vous n'êtes pas certain de trouver l'ensemble des outils que vous avez l'habitude d'utiliser et surtout aurez-vous l'autorisation d'installer quelque chose sur cette machine puisqu'elle ne vous appartient pas.

L'article d'aujourd'hui va vous montrer une approche qui vous sera utile si vous êtes amenés à vous déplacer, ou à maintenir des développements à distance en dehors de votre bureau, ou bien encore réaliser des projets sur différents ordinateurs. Une station embarquée correspond principalement à posséder un environnement AMP (Apache, MySQL, PHP).

Chaque programme est maintenu et développé séparément mais lorsque vous les associez, ils vont vous permettre d'obtenir un environnement web.

## Environnement embarqué

Un environnement de développement Web se compose avant tout d'un environnement

AMP (Apache, MySQL, PHP) qui doit pouvoir s'installer sur un support de faible capacité comme :

- Un disque dur externe
- Une clef USB
- Tous supports de stockage (carte SD par exemple)
- Un dossier différent

Il existe différentes distributions suivant votre système d'exploitation (Linux, Mac, Windows, Solaris). Cependant, certaines distributions ne sont pas spécifiques à un seul OS, mais peuvent exister sur l'ensemble des OS.

Bien entendu, si vous ne connaissez pas le système d'exploitation que vous utiliserez, vous devez posséder au moins une version des outils disponibles par système d'exploitation. Ces distributions ont pour but de pouvoir s'utiliser sans être obligé de faire à chaque utilisation, une nouvelle installation et un paramétrage.

Ces environnements pré-configurés offrent aussi un environnement web sous la forme AMP (Apache/MySQL/PHP). Il existe différents kits, les principaux étant : Mov'AMP, WampMSS, EasyPHP, MampServer2Go, UwAmp.

## Mov'AMP

Il s'agit d'une des toutes premières distributions proposant un environnement AMP puisque les premières versions fonctionnaient avec PHP 4. La version actuelle ne fonctionne que sous Windows (98, Me, 2000, XP) et aucune installation n'est nécessaire.

Vous pouvez utiliser cette distribution en décompressant simplement l'archive. Elle se compose de Apache, PHP 5.2.3 et MySQL 4.x. <http://www.movamp.net>

## WampMSS

Inspirée de WampServer, cette distribution est une version modifiée de celui-ci. Cette distribution fonctionne sous Windows et propose plusieurs versions comprenant des versions différentes de PHP, MySQL, PhpMyAdmin, SQLite. <http://uni-d.net>

## EasyPHP

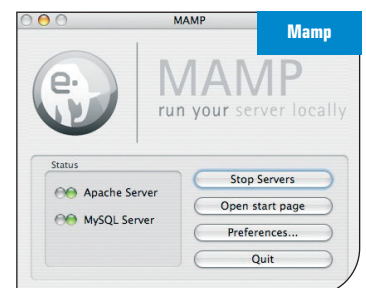
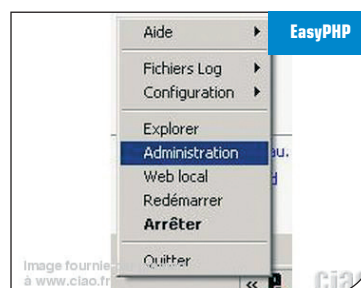
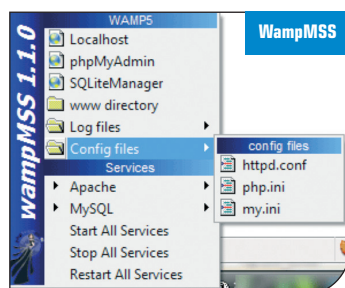
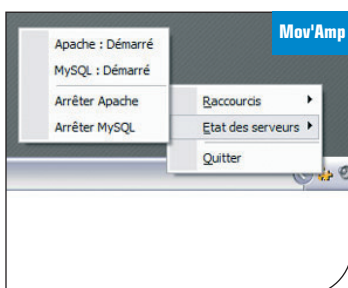
EasyPHP a été la première distribution connue en France proposant un environnement AMP pour Windows, qui était disponible seulement pour un environnement bureautique. Depuis la version 2, le kit s'installe sur des supports amovibles et dérivés en plusieurs versions. Une version principale avec PHP 5.2.x, MySQL 5, SQLite et PECL. L'autre version propose PHP 5.3.x, MySQL 5. <http://www.easyphp.org>

## Mamp

Mamp est la distribution pour le système d'exploitation MAC OS. Elle se compose d'un environnement Apache, PHP, MySQL, SQLite. Il existe une version standard et une version professionnelle. La première version est suffisante pour s'installer sur les différents supports car elle est moins gourmande au niveau de l'espace. <http://mamp.info>

## Server2Go

Server2Go est un serveur web, proposé sous trois formes pour Windows, Mac et Linux. Cette distribution est proposée avec différents composants pré-définis. A la différence des autres distributions, il existe 4 différentes versions de base pour Windows avec le langage PHP et PERL, les bases de données MySQL et SQLite. Par ailleurs, elle est aussi dérivée avec 2 versions Apache différentes.



Les autres versions proposent moins de choix mais sont assez complètes pour faire tourner celui-ci auprès des autres systèmes d'exploitation. <http://www.server2go-web.de/>

## UwAmp

Il s'agit d'une distribution française très jeune. Elle propose un serveur Apache, PHP, MySQL, SQLite. Aucune installation n'est nécessaire car vous devez juste décompresser l'archive pour faire fonctionner cet environnement.

Ce qui la distingue des autres distributions, c'est la communication d'informations supplémentaires comme le Monitoring CPU et l'utilisation de Multi-version de PHP.

Ainsi, vous pouvez varier les différentes versions de PHP pour effectuer des tests de compatibilité suivant la configuration des serveurs sans obligation de changer de dossier ou de support.

## Environnement bureautique

L'environnement bureautique est un peu différent de la partie précédente, car même si on peut utiliser ces distributions sur des supports amovibles, il n'est pas facile de les faire fonctionner sur un autre ordinateur sans être obligé de refaire une installation ou un paramétrage. Mais elles existent, et il est important de les connaître.

## Wampserver

Actuellement, WampServer est la distribution la plus connue depuis de nombreuses années, qui existe depuis la sortie de PHP 5. WampServer propose un logiciel modulaire avec la possibilité d'installer des add-ons et des versions différentes pour Apache, PHP et MySQL.

L'ensemble des possibilités offertes permet de changer de configuration du serveur en un minimum de clics de souris. Par ailleurs, toutes les options de PHP, sont intégrées mais non activées, grâce à cela vous pouvez paramétrer votre propre environnement sous Windows à votre convenance et surtout suivant la configuration nécessaire de vos différents projets.

## Xampp

Cette distribution existe pour Windows, Mac, Linux et propose Apache, MySQL et PHP en version de base. Cependant Xampp est plus qu'une station web car il propose un environnement FTP, un serveur Mail, Perl... Il existe différentes versions : une simple et une autre avec l'ensemble des options définies par défaut.

<http://www.apachefriends.org/fr> ou <http://portableapps.com/apps/development/xampp>

## Performance et sécurité

Quels que soient la distribution et l'environnement que vous utilisez, la performance est associée tant au niveau de la configuration que des supports utilisés. Les performances risquent d'être limitées car même si votre environnement est optimisé, vous pouvez vous rendre compte de certaines limites, liées au niveau du matériel comme les connectiques USB ou certains types de supports de stockage.

Or, les environnements embarqués sont destinés avant tout à vous permettre de proposer une démonstration de l'avancement de vos travaux ou de permettre des stations de développements complémentaires. Comme ceci, vous n'êtes pas freiné dans votre projet. Du côté de la sécurité, les risques sont très limités car vous ne partagez pas l'accès à votre serveur. Même si cela est possible par l'intermédiaire de l'ip de l'ordinateur, celui-ci n'a pas vocation d'être déployé en production.

## Compilation

Ces distributions sont déjà compilées et par conséquent aucune manipulation supplémentaire n'est nécessaire par rapport au système choisi. L'énorme avantage vient du fait que vous devez juste récupérer celle-ci et l'exécuter.

## Différents packages

Certaines distributions sont proposées et déjà configurées sous la forme de « Packs » pour être utilisées avec certaines applications web comme les CMS. Actuellement, il

existe 2 distributions déjà configurées comme Mov'amp et EasyPHP.

## Mov'AMP

Le CMS Joomla propose une version de Mov'AMP, légèrement modifiée, car vous pouvez utiliser ce CMS dans cette distribution sans aucune configuration spécifique. Grâce à cette opération, vous pouvez continuer à faire évoluer votre projet web sur un ordinateur Windows sans que celui-ci vous appartienne.

## EasyPHP

EasyPHP est aussi un environnement embarqué sur support amovible. Il propose différentes versions avec différents modules pré-installés. Vous allez pouvoir associer ce kit avec les CMS WordPress ou Spip, mais aussi avec du e-commerce comme Prestashop.

## Utilisation

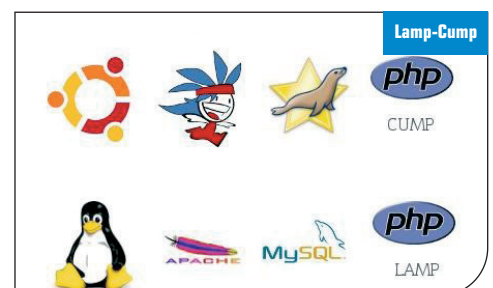
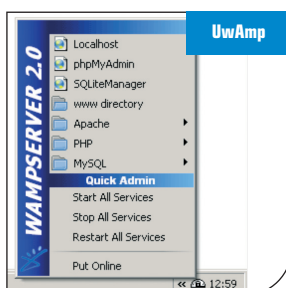
Lorsque vous décidez d'utiliser un environnement embarqué, le but principal doit vous permettre de retrouver votre environnement favori sur un support amovible. Lorsque celui-ci est lancé, vous pouvez installer toutes sortes d'applications CMS, e-Commerce, CRM, ERP... Vous n'aurez aucun problème pour effectuer ce genre d'opération, car les fichiers de configuration serveur (Apache), ou langage (PHP), sont déjà paramétrés, ce qui est un point important à prendre en considération.

## Conclusion

Avec Apache comme serveur HTTP, un environnement embarqué est très facile à utiliser et à déployer sur différents supports. Cependant, si vous ne souhaitez pas utiliser ce type de serveur, vous pouvez utiliser IIS ou Cherokee, mais avec l'obligation d'effectuer la configuration vous-même.

### ■ Christophe Villeneuve

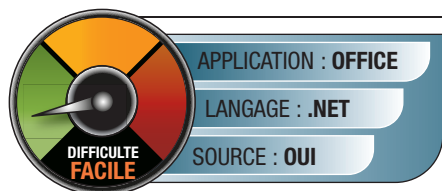
Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...





# Office 2010 : le nouveau VSTO

Pour cet article je me suis intéressé aux Visual Studio Tools for Office. VS 2010 apporte son lot de nouveautés et supporte la nouvelle suite Office 2010. Concrètement, les VSTO permettent de créer des applications OBA (Office Business Applications) qui sont des programmes .NET s'intégrant à la suite bureautique de Microsoft.



De la création d'un modèle de document spécifique à la customisation du ruban, tout y est ! Le remplaçant de VBA est-il simple d'utilisation ? C'est la question à laquelle j'essaierai de répondre. Je m'attarderai également sur l'aspect génération/lecture de document avec le SDK OpenXML. Un petit article qui vous fera faire le tour de ce que nous propose .NET pour Office 2010.

La première question que l'on peut se poser est : pourquoi une application OBA ? Créer un logiciel intégré c'est profiter de la puissance d'Office mais surtout ne pas brusquer l'utilisateur. Quand on sait que l'utilisateur met dix secondes à se faire un avis sur un logiciel, utiliser office c'est s'assurer d'une adoption rapide. Visual Studio 2010 supporte la création de projet pour Office 2010 mais aussi Office 2007. Un projet OBA se représente par une assembly .NET (un Add-in) que l'on "greffe" à la suite Office. Il existe deux types d'add-ins : ceux qui se chargent pour un certain type de document et ceux qui se chargent pour toute l'application.

- Les add-ins de document: L'assembly se chargera dès qu'un document bien spécifique sera ouvert.
- Les add-ins d'application: Dès que l'application s'ouvre ou si l'utilisateur en fait la demande, l'assembly se charge

Par défaut, la création d'un nouveau projet Visual Studio nous propose ces templates : [Fig.1].

Dans l'image, les projets de types Add-in correspondent aux Add-in d'applications et les \*\*\* Template/Workbook aux applications de documents. Sans installations complémentaires on se retrouve dans un environnement que nous connaissons avec les mêmes possibilités qu'avec un autre projet .NET (Debug...). Le code .NET que nous écrivons peut accéder au modèle objet du document (un peu comme pour SharePoint, sauf que le modèle objet héritant de COM il n'y a pas d'héritage). Ce modèle objet est accessible grâce à une assembly PIA (Primary Interop Assembly) qui est un wrapper entre .NET et les objets COM. Les VSTO utilisent donc les mêmes objets que VBA grâce à de l'interopérabilité (attention aux performances!).

Le chargement d'un add-in se fait de deux façons :

• "At Startup": Le plug-in démarre au lancement de l'application office

- "On Demand": C'est l'utilisateur qui peut charger le plug-in suivant ses besoins

Le Runtime VSTO possède un VSTO Loader qui vérifie avec quelle version de framework nous travaillons. Dans le cas du framework 3.5, il faut que l'utilisateur installe les objets PIA dont je vous ai parlé plus haut. S'il utilise le framework 4.0, nous verrons dans le chapitre "déploiement" qu'il existe un moyen d'embarquer les PIA [Fig.2]. Nous profiterons des nouveautés du langage C# 4 pour éviter d'avoir des appels à des méthodes interminables grâce aux paramètres optionnels du langage.

## NOUVEAUTÉS PROPRES À VS 2010

Les VSTO existent depuis 2005 et ont subi une grosse évolution lors du passage à la version 2008. Les principales nouveautés propres à la version 2010 sont les suivantes :

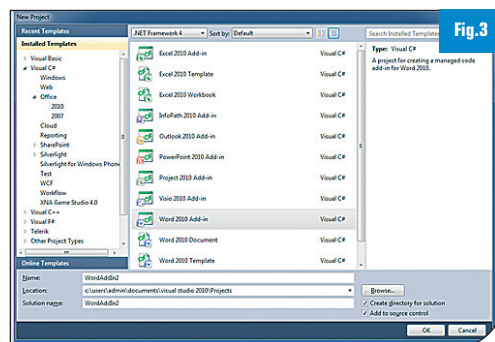
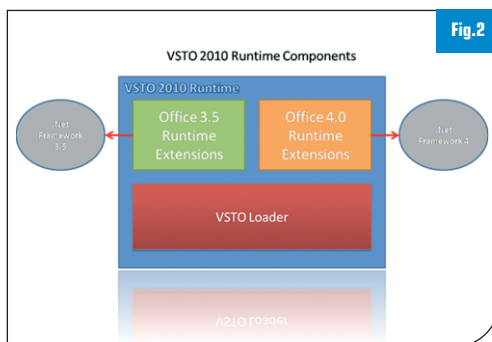
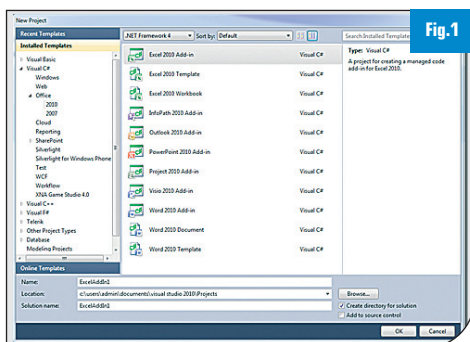
- Support d'Office 2010
- Support de la personnalisation du ruban pour Project, Outlook, InfoPath et Visio
- Support du développement avec le Framework 4.0
- Support des installations MSI
- Déploiement de plusieurs solutions en un seul package
- Designer visuel pour l'édition du ruban
- ...

## LE MODÈLE OBJET PIA

Les PIA sont les wrappers entre les objets COM et .NET. Par exemple quand on crée un projet pour Office, des références sont automatiquement ajoutées au projet appelé "\*\*\*.Interop.\*.\*". Il existe d'autres assemblies type PIA, toutes sont stockées dans ce dossier :

C:\Program Files (x86)\Microsoft Visual Studio 10.0\Visual Studio Tools for Office\PIA

Les PIA doivent être installés dans le GAC de la machine qui exécute un programme VSTO. Ils sont automatiquement déployés lors d'une installation d'office. Quelques exemples de PIA :



- Microsoft.Office.Interop.Access.dll
- Microsoft.Office.Interop.Excel.dll
- Microsoft.Office.Interop.InfoPath.dll
- Microsoft.Office.Interop.Outlook.dll
- Microsoft.Office.Interop.PowerPoint.dll
- Microsoft.Office.Interop.Publisher.dll
- Microsoft.Office.Interop.SharePointDesigner.dll
- Microsoft.Office.Interop.Visio.dll
- Microsoft.Office.Interop.Word.dll

A noter que l'on peut référencer les PIA dans une application .NET basic (pas VSTO) pour travailler sur les documents office.

## QUEL LOGICIEL DE LA SUITE OFFICE 2010 POUVONS-NOUS PERSONNALISER ?

La liste des PIA vue précédemment doit déjà vous avoir donné une petite idée des logiciels personnalisables. C'est globalement toute la suite Office que l'on peut modifier. Seuls Excel et Word supportent les plug ins s'appliquant à un document. Dans la suite de cet article, nous allons concevoir :

- Des plug in de "complément" qui peuvent être attribués à toute la suite Office.
  - Personnalisation du ruban
  - Personnalisation du menu contextuel
  - Personnalisation du backstage
  - Personnalisation du custom action pane
  - ...
- Des plug in plus spécifiques
  - Plug in de document pour Word
  - Plug in de document pour Excel
  - Plug in pour Outlook

Il faut que côté client, l'utilisateur possède le Framework que nous ciblons. Voici les seuls Frameworks supportés par les VSTO :

- Framework .NET 3.5
- Framework .NET 4.0
- Framework .NET 3.5/4.0 Client Profile

Le principal intérêt de cibler le framework 4.0 est de pouvoir déployer nos applications sans PIA. En effet, par défaut, si nous créons un projet avec le .NET Framework 4.0, les propriétés des références de types PIA sont "Embed Interop Types" à vraies. Cela signifie qu'à la compilation, les PIA sont embarqués dans l'assembly générée. Pratique si vous souhaitez avoir seulement une assembly pour des clients hétérogènes (Office 2007 et 2010).

## PERSONNALISATION DU RUBAN

Le ruban office est personnalisable à partir d'une assembly VSTO. Pour cela, direction Visual Studio pour la création d'un projet de "complément" (ou Addin) Office : [Fig.3]. A ce stade, notre projet contient un fichier ThisAddIn.cs avec deux méthodes :

- ThisAddIn\_Startup: S'exécutant lors du chargement de l'add-in
  - ThisAddIn\_Shutdown: S'exécutant lors de la fermeture de l'add-in
- Il existe deux méthodes pour personnaliser le ruban : grâce à un designer pour les cas simples ou avec un fichier XML pour les cas compliqués. Nous ajoutons un élément à notre projet de type Ribbon : [Fig.4]. Et un fichier Ribbon1.cs est créé permettant la visualisation de notre ruban : [Fig.5]. La boîte à outils de Visual Studio nous propose des contrôles de base : [Fig.6]. Que nous disposons sur le ribbon afin de s'abonner à des événements comme le "Click" avec du code C#. Nous pouvons à tout moment demander d'exporter le ruban au format XML lors de situations complexes : [Fig.7]. Suite à cette action, un fichier Ribbon.xml est généré et nous pouvons modifier à la main les éléments de celui-ci :

```

1. <customUI onLoad="Ribbon_Load">
2.   <ribbon>
3.     <tabs>
4.       <tab idMso="TabAddIns">
5.         <group id="group1" label="group1">
6.           <button id="button1" label="button1"
7.             showImage="false" />
8.           <dialogBoxLauncher>
9.             <button id="group1_btn" />
10.          </dialogBoxLauncher>
11.        </group>
12.      </tab>
13.    </tabs>
14.  </ribbon>
15. </customUI>

```

Lors de l'exécution, Word s'ouvre avec le plug in activé. Le plug-in est installé d'Office (Notez le jeu de mots 😊) et il y a bien un appel à la méthode ThisAddIn\_Startup. Depuis Office 2010, l'utilisateur peut choisir les plug-ins à activer ou non, de ce fait nous pouvons à la main désactiver le plug-in (ce qui provoquera l'appel à la méthode Shutdown) [Fig.8].

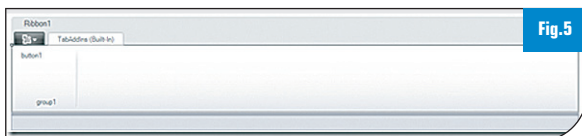


Fig.5

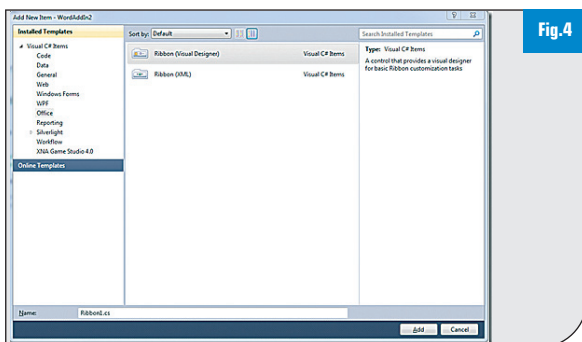


Fig.4

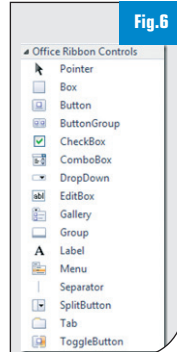


Fig.6

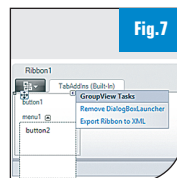


Fig.7

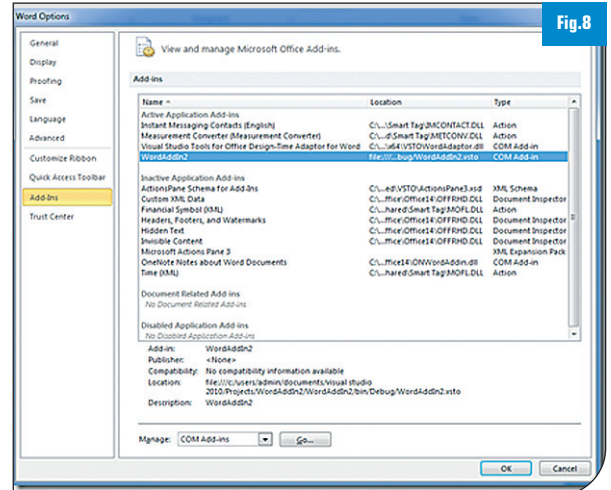


Fig.8

## PERSONNALISATION DU BACKSTAGE

Le BackStage est la page récapitulative du document. On utilise généralement le ribbon pour ajouter des éléments qui vont traiter sur le contenu du document. A l'inverse, nous utilisons le BackStage pour réaliser des actions plus générales comme l'ajout de métadonnées, la publication sur un SharePoint... Là encore nous n'avons pas de designer, nous ajoutons un fichier XML à notre projet.

Nous vérifions que l'espace de nom XML utilise bien la nouvelle DTD de Microsoft : [Fig.9]

```
xmlns=»http://schemas.microsoft.com/office/2009/07/customui«
```

Un BackStage se représente ainsi : [Fig.10].

Avant de commencer la configuration du fichier XML, il faut enregistrer notre nouvel élément de configuration dans notre projet VSTO. Pour cela, direction la classe ThisAddin.cs pour redéfinir la méthode CreateRibbonExtensibilityObject et renvoyer une instance de notre classe Ribbon1.cs :

```
1. public partial class ThisAddIn
2. {
3.     private void ThisAddIn_Startup(object sender, System.EventArgs e)
4.     {
5.
6.     }
7.     protected override Office.IRibbonExtensibility CreateRibbonExtensibilityObject()
8.     {
9.         return new Ribbon1();
10.    }
11.
12.    private void ThisAddIn_Shutdown(object sender, System.EventArgs e)
13.    {
14.    }
15. }
```

Et nous ajoutons une balise <backstage> pour définir un espace personnalisé avec un bouton et une textbox.

```
1. <?xml version=»1.0« encoding=»UTF-8«?>
2. <customUI xmlns=»http://schemas.microsoft.com/office/2009/07/customui« onLoad=»Ribbon_Load«>
3.     <backstage>
4.         <tab id=»tabNew« label=»Mon Tab« insertAfterMso=»TabInfo«>
5.             <firstColumn>
6.                 <group id=»grpNew« label=»Mon groupe« helperText=»
```

```
Mon groupe personnalisé«>
7.             <primaryItem>
8.                 <button id=»btnNew« label=»Mon bouton« getImage=»GetImage« />
9.             </primaryItem>
10.            <topItems>
11.                <editBox id=»txtNew« label=»StringTest« sizeString=»WWW« />
12.            </topItems>
13.        </group>
14.    </firstColumn>
15. </tab>
16. </backstage>
17. </customUI>
```

A noter que le bouton retrouve son image avec l'appel de la méthode GetImage que nous définissons dans Ribbon1.cs :

```
1. public System.Drawing.Bitmap GetImage(Office.IRibbonControl control)
2. {
3.     if (control.Id == «btnNew»)
4.         return Properties.Resources.Background;
5.     else
6.         return null;
7. }
```

[Fig.11] Nous pouvons personnaliser, entre autres, le style du groupe, par exemple :

```
1. <group id=»grpNew« label=»Mon groupe«
2.     helperText=»Mon groupe personnalisé«
3.     getStyle=»GetWorkStatusStyle«>
```

```
1. public BackstageGroupStyle GetWorkStatusStyle(Office.IRibbonControl control)
2. {
3.     return BackstageGroupStyle.BackstageGroupStyleError;
4. }
```

[Fig.12] Pour finir, nous captions le clic sur le bouton pour afficher dans une messagebox le contenu de la textbox et exporter le fichier au format PDF.

```
1. <button id=»btnNew« label=»Mon bouton«
2.     getImage=»GetImage« onAction=»OnClick« />
3.
```

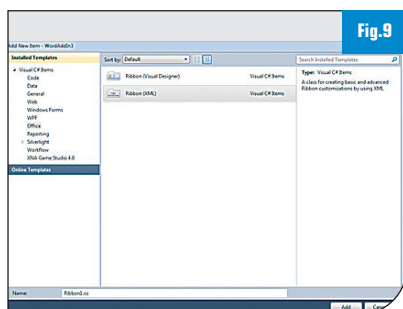


Fig.9

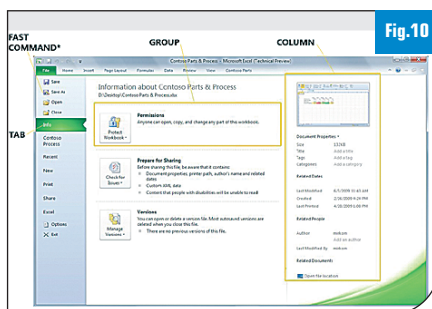


Fig.10

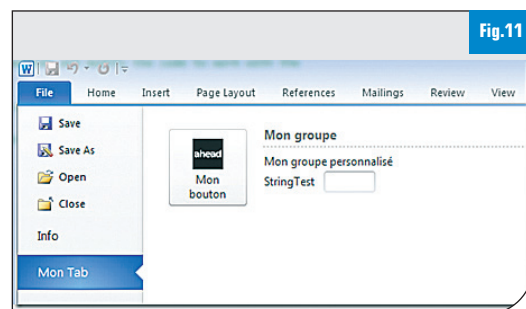


Fig.11



```
Bar», Office.MsoBarPosition.msoBarPopup);
6.
7.    var coucouButton = (CommandBarButton)commandBar.Controls
.Add();
8.
9.    coucouButton.Style = MsoButtonStyle.msoButtonCaption;
10.   coucouButton.Caption = «Mettre en gras»;
11.   coucouButton.FaceId = 113;
12.
13.   coucouButton.Click += ButtonClick;
14. }
```

Le but du menu contextuel étant de mettre en gras, la bonne pratique veut que nous attribuions au bouton que nous développons le bon ID. Les ID des contrôles Word sont téléchargeables à cette adresse (113 pour la fonction gras) :

<http://www.microsoft.com/downloads/details.aspx?FamilyID=3f2fe784-610e-4bf1-8143-41e481993ac6&displaylang=en> [Fig. 14]

C'est dans la méthode `WindowBeforeRightClick` que nous affichons ce fameux menu contextuel :

## PERSONNALISATION DU MENU CONTEXTUEL

Toujours dans notre add-in Word, nous pouvons contrôler l'application Word, et entre autres, redéfinir un menu contextuel.

Nous utilisons la méthode Startup pour réaliser deux choses :

- Capturer l'événement `WindowBeforeClick` qui affichera notre menu personnalisé
- Initialiser notre menu personnalisé

```
1. private void ThisAddIn_Startup(object sender, System.EventArgs e)
2. {
3.     this.Application.WindowBeforeRightClick += Application_WindowBeforeRightClick;
4.     ConfigureMenu();
5. }
```

ConfigureMenu est une méthode qui initialise le nouveau menu contextuel:

```
1. private CommandBar commandBar;  
2. private void ConfigureMenu()  
3. {  
4.  
5.     commandBar = Application.CommandBars.Add(«MySuperCommand
```

```

1. private Word.Selection selection;
2. void Application_WindowBeforeRightClick(Word.Selection Sel,
ref bool Cancel)
3. {
4.     commandBar.ShowPopup();
5.     selection = Sel;
6.     Cancel = true;
7. }
8.
9. private void ButtonClick(CommandBarButton ctrl, ref bool
canceldefault)
10. {
11.     selection.BoldRun();
12. }

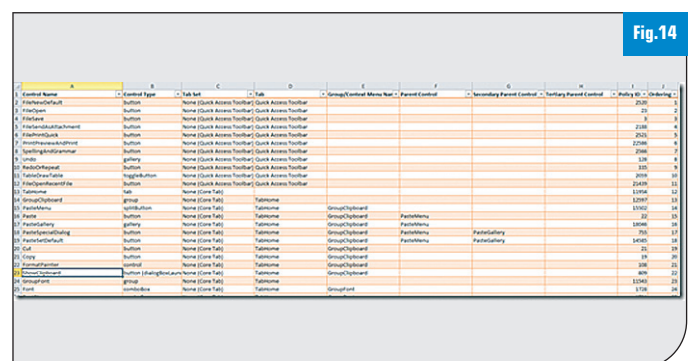
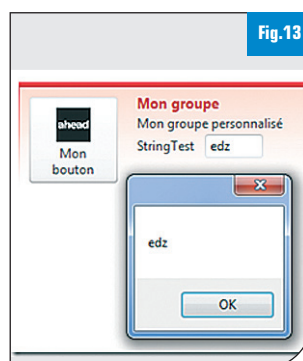
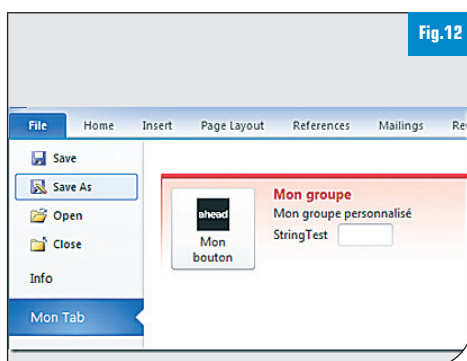
```

Il est dommage de ne pas avoir de designer en 2010 pour faire ce genre de choses 😞



L'autre solution pour implémenter un menu contextuel est d'utiliser le fameux fichier XML que nous utilisons depuis le début pour le ribbon ou le BackStage. Exemple :

1. <customUI
2. xmlns=»http://schemas.microsoft.com/office/2009/07/customui«



```

3. onLoad=>Ribbon_Load>
4. <contextMenus>
5. <contextMenu idMso=>ContextMenuText>
6. <button id=>MyButton label=>Hello World/>
7. <menuSeparator id=>MySeparator />
8. <toggleButton id=>MyToggle label=>My Toggle Button />
9. </contextMenu>
10. </contextMenus>
11. </customUI>

```

Résultat : [Fig.15]

## LES CUSTOMS ACTIONS PANE

Les Actions Pane sont les volets que l'on voit sur le côté comme la recherche par exemple. Ils supportent l'hébergement de contrôle Winforms (et de ce fait WPF grâce au contrôle ElementHost), c'est d'ailleurs ce que nous allons faire. Nous créons un simple UserControl WPF dans le projet VSTO : [Fig.16].

Puis nous retournons dans le fichier ThisAddin.cs et nous modifions la méthode Startup pour intégrer notre UserControl :

```

1. private void ThisAddIn_Startup(object sender, System.EventArgs e)
2. {
3. //Le contrôle Winforms qui host le WPF : (
4. UserControl usercontrol = new UserControl();
5. usercontrol.Controls.Add(new ElementHost(){Child = new
UserControl1(), Dock = DockStyle.Fill});
6. //Le volet
7. Microsoft.Office.Tools.CustomTaskPane taskPane = Custom
TaskPanes.Add(usercontrol, «Volet personnalisé»);
8. taskPane.Visible = true;
9. taskPane.DockPosition = MsoCTPDockPosition.msoCTPDockPosition
Right;
10. taskPane.Width = 300;
11. }

```

Note : il faut rajouter une référence vers WindowsFormsIntegrat-ion pour utiliser ElementHost [Fig.17].

## PERSONNALISATION SPÉCIFIQUE POUR WORD

Faire un add-in, c'est bien. Faire un add-in utile, c'est mieux. Pour cela, nous avons à disposition un modèle objet COM permettant de faire des traitements sur le contenu du fichier. Voici quelques éléments en vrac que nous pouvons utiliser à partir, par exemple, d'un bouton dans un ribbon :

- Création d'un nouveau document Word

```
1. Globals.ThisAddIn.Application.Documents.Add();
```

Ouvrir un document Word en lecture seule

```
1. Globals.ThisAddIn.Application.Documents.Open(@"C:\toto.doc",
ReadOnly:true);
```

Ou Close pour la fermeture

- Enregistrer un document Word

```
1. Globals.ThisAddIn.Application.ActiveDocument.Save();
```

Insérer du texte dans un document, le mettre en gras et le sélectionner

```

1. Range rng = Globals.ThisAddIn.Application.ActiveDocument.
Range();
2. rng.Text = «My Text»;
3. rng.Bold = 1;
4. rng.Select();

```

## MODÈLE ET DOCUMENT WORD

Nous avons vu les add-ins d'applications qui s'appliquent à un logiciel de la suite Office. Il existe des add-ins de documents ou de modèles qui sont propres à un fichier spécifique. Pour utiliser l'add-in, l'utilisateur doit créer un fichier à partir du modèle Word ou en ouvrant le document concerné. Prenons l'exemple de la création d'un modèle : [Fig.18]. La création de ce projet nous ouvre un éditeur Visual Studio représentant Word : [Fig.19].

Ce modèle peut contenir des contrôles Winforms traditionnels (ou WPF avec HostElement) ou des contrôles propres à Word. Comme dans une Windows Forms, nous pouvons nous abonner aux événements des contrôles. A l'exécution, Word s'ouvre avec un document généré à partir de ce modèle.

## PERSONNALISATION SPÉCIFIQUE POUR EXCEL

Comme pour Word, Excel peut être personnalisé par des add-ins d'applications et de documents (classeur ou modèle). Voici quelques opérations spécifiques pour Excel :

- Créer un classeur

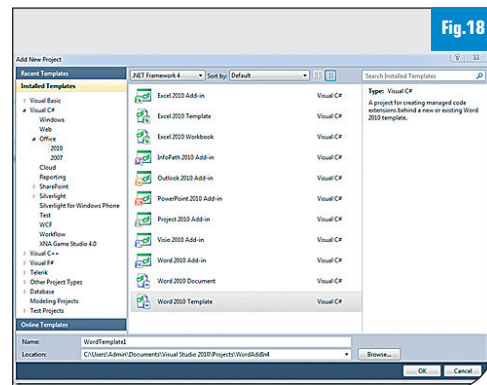
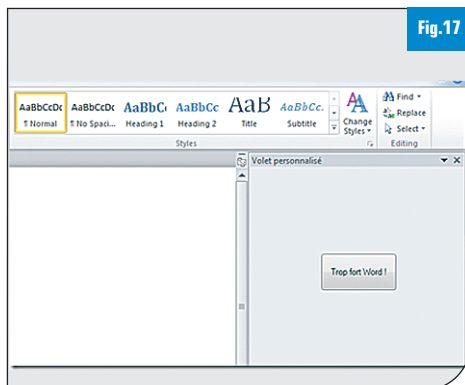
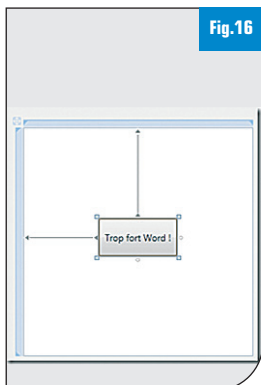
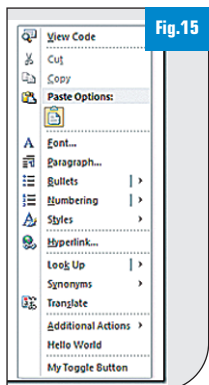
```
1. Globals.ThisAddIn.Application.Workbooks.Add();
```

Exécuter un calcul par Excel

```

1. var active = Globals.ThisAddIn.Application.ActiveSheet;
2. NamedRange range = active.Controls.AddNamedRange(Globals.

```



```
ThisAddIn.Application.get_Range("A1");
3. range.Calculate();
```

## PERSONNALISATION SPÉCIFIQUE POUR OUTLOOK

Outlook ne déroge pas à la règle et peut être modifié grâce aux VSTO. Une spécificité d'Outlook est la création de régions (élément graphique venant se greffer à l'interface de Outlook) [Fig.20].

Lors de la création de la région, des choix s'imposent :

- Sa position
- Le cadre où l'afficher (lecture, composition)
- Le type de fenêtre (Mail, Appointment...) [Fig.21]

## DÉPLOIEMENT

Le déploiement VSTO nécessite une préparation préalable du poste utilisateur. Celui-ci doit avoir :

- Le .NET Framework 3.5 ou 4.0 (Client ou complet)
- Visual Studio 2010 Tools pour Office Runtime
- Les PIA si le framework n'est pas le 4.0

Le déploiement peut se faire avec un Windows Installer ou Click Once.

## COMPARAISON VBA / VSTO

La comparaison VBA/VSTO est inévitable. Le VBA profite du fait qu'il suffit de partager un document sans se soucier de l'installation utilisateur en face. Mais VBA souffre de la réputation des développeurs qui l'utilisent (manque de rigueur...) et trouve vite ses limites en termes de sécurité ou déploiement. Les VSTO utilisent le modèle objet office mais aussi .NET. A noter que l'on peut appeler du code VBA à partir de VSTO et inversement.

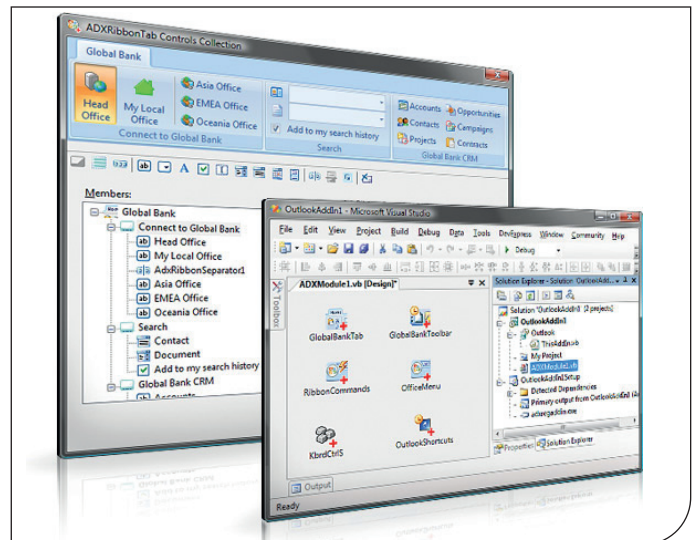
## LE SDK OPEN XML

Puisque le sujet de cet article est l'utilisation d'Office en .NET, j'en profite pour présenter rapidement le SDK OpenXML. Ce SDK va nous permettre de lire/générer/manipuler des fichiers Office sans passer par les PIA. La version actuelle est la 2.X et elle est téléchargeable ici : <http://www.microsoft.com/downloads/details.aspx?FamilyID=c6e744e5-36e9-45f5-8d8c-331df206e0d0&displaylang=en#filelist>

Deux fichiers sont à disposition :

- OpenXMLSDKv2.msi : représente le SDK de développement
- OpenXMLSDKTool.msi : représente des outils tels que OpenXML Class Explorer ou OpenXmlDiff

Ce SDK ne nécessite pas Office installé sur sa machine. Les per-



sonnes ayant travaillé avec ce SDK en production expliquent sur leurs blogs que c'est parfois compliqué de modifier des contenus riches. Des alternatives existent comme l'API d'Aspose. N'hésitez pas à faire un petit tour des API disponibles.

Un document OpenXML est un fichier ZIP qui regroupe des fichiers aux formats XML (content, metadata...), des contenus de type medias... Pour commencer le développement avec OpenXML il faut référencer DocumentFormat.OpenXml.dll. Un outil très utile vient s'ajouter à Visual Studio 2010. Il s'appelle Open XML Package Editor Power Tool. Il permet, entre autres, d'ouvrir/éditer/créer des fichiers OpenXML avec l'EDI Visual Studio.

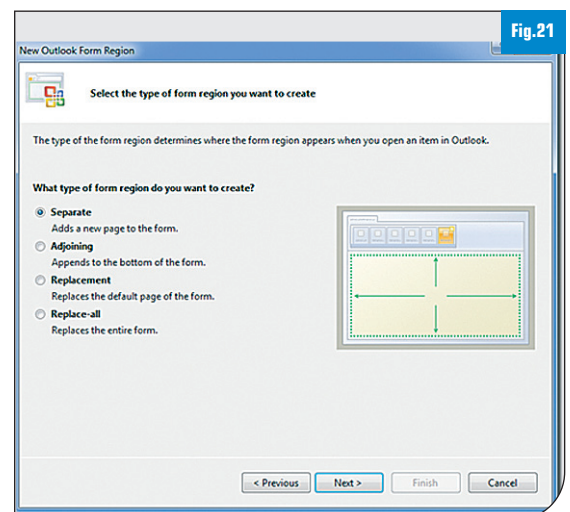
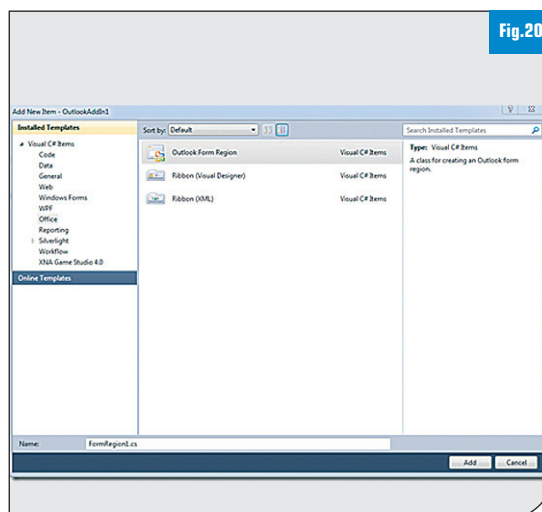
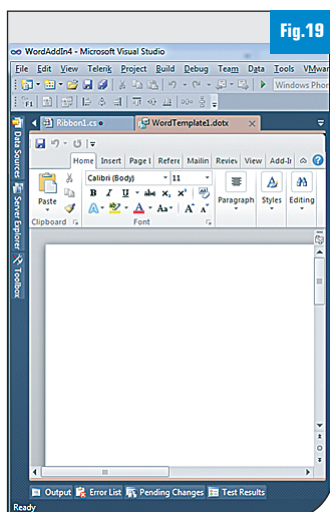
## CONCLUSION

VSTO nous permet de rester dans l'environnement .NET lors de nos développements et apporte des fonctionnalités en plus, notamment en termes de déploiement. Cependant, entre les designers manquants et l'interopérabilité sur les objets COM on a l'impression d'être resté sur un système un peu archaïque. A quand une suite Office réalisée en .NET ?



### Julien Dollon

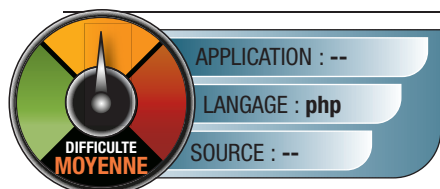
*Most Valuable Professional, Consultant Trainer I'FORM/Exakis, Dotnet-France Community Leader, SUPINFO Full Professor .NET Technologies I AHEAD Scrum Master*





# Interopérabilité en PHP

La communication entre les langages a toujours existé. Cependant au fil des années, des évolutions et des différents rapprochements entre sociétés, l'interopérabilité des langages est de plus en plus flagrante, surtout à travers le Web. Maintenant vous pouvez communiquer plus facilement avec les langages dynamiques sous licences payantes et les solutions open source.



Il n'existe pas de définition figée sur ce thème. Mais vous pouvez considérer que cela correspond à l'aptitude de mettre en relation deux ou plusieurs systèmes à

fonctionner ensemble, en utilisant les standards communs.

Cette technologie a aussi un but fort intéressant, appelée « web services » ou « services métiers ». Les web services sont des applications développées avec des langages statiques (Basic, C, C++, Assembleur...) et peuvent communiquer facilement avec les langages dynamiques. Le langage PHP est considéré comme un langage souple et évolutif. Par ailleurs, il est modulable et permet de réaliser de nombreuses tâches. Ce langage a aussi trouvé sa place pour répondre aux attentes des différents acteurs du serveur et aux nouvelles fonctionnalités. Enfin, l'objectif doit permettre de communiquer le plus simplement en supprimant la complexité et la diversité des technologies.

## PRÉSENTATION

La communication entre les langages peut se faire de deux façons :

- Les extensions COM – DCOM
  - L'utilisation de différentes méthodes XML – RPC ou REST ou SOAP
- Par conséquent, le langage PHP peut avoir un rôle de « Maître » ou aussi d'« Esclave », c'est-à-dire recevoir ou envoyer des ordres. Ces ordres peuvent s'exécuter du côté client ou du côté serveur.

### Côté client

Le côté client correspond à la partie FRONT d'un site web, c'est-à-dire la partie visible. Le langage PHP permet d'interagir facilement entre les différents langages. Les langages principaux sont HTML, JS, Ajax, XML / JSON, SWF (Flash). La majorité d'entre eux utilisent le PHP pour communiquer facilement avec les bases de données et par conséquent le PHP est utilisé en mode esclave car ce n'est pas lui qui dirige le traitement des informations. Avec l'utilisation du XML, vous pouvez communiquer avec des logiciels bureautiques ou applicatifs. La communication peut s'effectuer de la façon suivante :

```
<?php
$doc = new DOMDocument();
$doc->load('book.xml');
echo $doc->saveXML();
?>
```

Le source montre qu'il faut déclarer un objet, le charger et ensuite le sauvegarder. Vous pouvez effectuer différents traitements lorsque votre objet est défini.

### Côté serveur

Le côté serveur est différent de ce qui a été vu car il n'est pas visible pour l'internaute.

Cependant il est très utile à connaître car vous allez pouvoir communiquer avec d'autres langages, se situant sur d'autres serveurs et environnements.

L'interopérabilité entre les langages va permettre d'avoir une ouverture et une compatibilité entre les systèmes informatiques.

Ce secteur est en constante évolution mais oblige les développeurs à connaître certaines bases des autres langages. L'article va présenter quelques relations disponibles

## PHP & JIEE

JIEE fut l'un des premiers langages dynamiques orienté Web, à s'ouvrir vers les autres. Ce langage basé sur la technologie Java a évolué au niveau applicatif.

Vous pouvez utiliser une classe Java dans un environnement PHP. Cette classe est alors manipulable en toute transparence dans l'environnement PHP.

Pour effectuer une connexion entre JIEE et PHP, le code peut se présenter de la façon suivante :

```
<?php
$chaine = new java('java.lang.StringBuffer', 'Bienvenue ');
$chaine->append('à ');
$chaine->append('Java');
$chaine->insert(0, '<p>');
$chaine->append('</p>');
$longueur = $chaine->length();
echo $chaine->toString() . ' ('.$longueur.')';
?>
```

Pour utiliser ce script, il est nécessaire d'activer la librairie Java et de la paramétrer

## PHP & .NET

Les langages .NET (C# et VB.Net) sont développés et maintenus par Microsoft. Depuis quelques années, une certaine ouverture vers les langages ouverts est apparue. Pour communiquer entre ces 2 langages, vous devez déclarer la librairie pour son bon fonctionnement :

```
<?php
$stack = new DOTNET("mscorlib", "System.Collections.Stack");
$stack->Push(".Net");
$stack->Push("Bonjour ");
echo $stack->Pop() . $stack->Pop();
?>
```

## PHP & PYTHON

L'interopérabilité peut aussi s'appliquer autour de 2 langues dynamiques Open Source. La réalisation est possible par l'intermédiaire d'une passerelle, intitulée WSGI. Elle permet d'exécuter les processus PHP à l'intérieur du langage Python. L'opération est réalisable par l'intermédiaire d'une passerelle WSGI Python et WSGI routage middleware. Le code peut se présenter de la façon suivante :

```
from wphp import PHPApp

my_php_app = PHPApp('/path/to/php-files',
                    php_options={'magic_quote_gpc': 'Off'})
```

## Les mixtes

Une nouvelle forme d'interopérabilité a été mise en avant en 2010, il s'agit d'un mélange entre la partie client (visible) et la partie serveur (cachée) il s'agit de Hip Hop réalisé par Facebook.

La technologie Hip Hop est basée en partie sur une extension XHP, qui est un regroupement de XML et PHP. Celle-ci permet d'augmenter la syntaxe de PHP et d'écrire des fragments HTML dans un même script. Du côté serveur, le langage PHP, développé en C++ a été compilé en G++ pour obtenir une version comprimée et allégée du langage dynamique PHP. Cette nouvelle façon de programmer se présente de la façon suivante :

```
<?php

if ($_POST['name']) {

    echo <span>Hello, {$_POST['name']}</span>;

} else {

    echo

    <form method="post">

        What is your name?<br />

        <input type="text" name="name" />

        <input type="submit" />

    </form>;

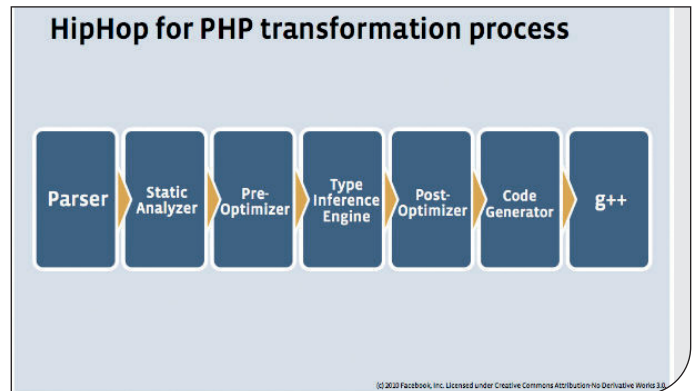
}

?>
```

Cette nouvelle façon de développer induit certains nettoyages au niveau de la syntaxe, ce qui permet d'avoir un code épuré au maximum. Cette approche a en plus permis de gagner du temps processeur et donc économisé au niveau du nombre de serveurs.

## LES INFRASTRUCTURES

Les relations étroites qui existent entre les langages et les serveurs distants s'effectuent facilement. Alors le contenu affiché peut proposer des services supplémentaires en toute transparence pour l'internaute. Mais l'interopérabilité ne se limite pas seulement au



Architecture de Facebook HipHop.

langage, vous pouvez effectuer la même chose avec les bases de données, tout en gardant votre langage favori comme noyau principal. Ainsi, par l'intermédiaire d'interfaces, vous pouvez proposer des web services ou des API SOAP.

L'ensemble de ces relations a permis de faire évoluer le langage PHP, avec de nouvelles fonctionnalités, de nouveaux environnements et bien sûr également au niveau de la plateforme.

## Plateforme Windows

Windows Azure est une plateforme ouverte pour le système d'exploitation Windows. Il propose des solutions pour les développeurs, permettant de supporter les langages dynamiques tel que PHP, des outils de développement (Eclipse ou Visual Studio) pour créer des applications qui s'exécuteront sur Windows Azure.

Cette plateforme va s'occuper de gérer les protocoles standards internet, tout en laissant la portabilité des données vers d'autres utilisations. De nombreux plug-ins ont été mis à disposition des différents logiciels open source comme Eclipse pour permettre aux développeurs de réaliser les applications facilement. Cependant des SDK par langage sont proposés pour vous aider.

## Plateforme Linux

Les plateformes Linux voient apparaître régulièrement de nouvelles versions pour permettre la gestion d'interopérabilité.

## OPENSTACK

OpenStack est un des projets les plus anciens pour le web et lancé par la société Rackspace. Son objectif est une plateforme de cloud computing open source sous licence Apache 2.0. Par ailleurs, il se présente comme une pile de logiciels permettant de déployer une architecture complète de service d'hébergement et composé de :

- une technologie de stockage
- une gestion des instances de serveurs virtuels gérés par modules

## CONCLUSION

Dans l'interopérabilité, les relations sont à tous les niveaux et à toutes les étapes. Certaines sont plus utilisées que d'autres, mais de nombreuses sociétés commencent à proposer des offres ou différentes solutions open source ou payantes. Les possibilités offertes sont importantes et donneront lieu à d'autres articles sur ce sujet.

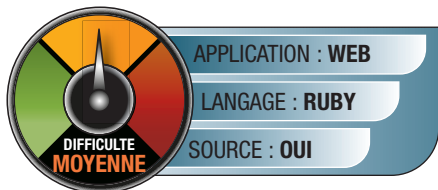
### ■ Christophe Villeneuve

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

# La boîte à outils (presque) parfaite, Rails 3

2<sup>e</sup> partie

Nous poursuivons notre découverte de Ruby on Rails 3.0. Les “gems” sont des paquets de code Ruby qui peuvent être installés sur les machines disposant de Ruby en utilisant le gestionnaire de paquets RubyGems. 



RubyGems est un gestionnaire de Package Similaire à "apt-get" ou "port", mais qui est codé en pur Ruby, et sert à installer/supprimer/mettre à jour des modules Ruby. Les sources du code présenté ici sont disponibles sur : <http://github.com/gdurelle/myblog>

## BUNDLER

Bundler est un gem qui permet de gérer les dépendances de votre application. A la racine de chaque projet Rails, on peut trouver un fichier Gemfile, qui en est le manifeste. Celui-ci contient la liste de tous les gems nécessaires au bon fonctionnement de l'application. Cela permet notamment, lorsque l'on travaille en équipe, de s'assurer que tous ont tous les gems nécessaires, et surtout la même version de ces derniers ! Cela permet également d'installer facilement la totalité des gems d'un projet dans un environnement vierge. Pour installer la totalité des gems, il suffit de taper : **bundle** (qui est un raccourci pour «bundle install»)

## RAKE

Rake est également un gem. Il permet d'exécuter de nombreuses commandes. Essayer **rake -T** pour voir la liste des commandes à votre disposition. Vous pouvez même créer vos propres rake tasks en Ruby. Pour cela créez simplement un fichier monfichier.rake dans le repertoire lib/tasks. Pour l'exécuter il suffira alors de faire **rake monfichier:matache**

ex : créez le fichier lib/tasks/myblog.rake  
Ecrivez y le code suivant :

```
1 namespace :myblog do
2   desc "Show myblog's stats"
3   task :stats => :environment do
4     nb_articles = Article.count
5     nb_authors = Author.count
6     avg = nb_articles/nb_authors
7     puts "Articles : " + nb_articles.to_s
8     puts "Authors : " + nb_authors.to_s
9     puts "AVG articles/authors : " + avg.to_s
10  end
11 end
```

Et lancez la commande : **rake my\_blog:stats**

## LE JAVASCRIPT ET RAILS 3

L'un des apports majeurs de Rails3 est la refonte des helpers Javascript. Pourquoi ? Car ils génèrent à présent du code non-intrusif. Cela signifie un code plus "propre" car vous séparez bien votre code javascript de votre code html. Tout le javascript est dans un fichier "rails.js" à part. De plus, le découplage étant l'un des mots d'ordre de Rails 3, vous pourrez utiliser n'importe quelle bibliothèque Javascript à la place de celle par défaut. La bibliothèque par défaut utilisée

pour ces mécanismes est Prototype, que vous pouvez facilement remplacer par JQuery ou Mootools.

### Pourquoi changer ?

Simplement parce que si vous utilisez/préférez un autre framework que celui imposé par défaut, il est plus simple de ne pas avoir à gérer la compatibilité entre les 2 !

En effet la plupart des frameworks JavaScript utilisent les mêmes formalismes pour manipuler le DOM et de nombreux noms de méthode peuvent être les mêmes. Ceci pouvant provoquer des comportements totalement imprévisibles.

### Comment changer ?

Rendez vous dans le dossier public/javascript/. Vous devriez voir plusieurs fichiers Javascript qui sont inclus automatiquement via le layout application.html.erb :

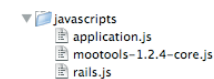
```
<%= javascript_include_tag :defaults %>
```

Le fichier qui vous intéresse est le fichier rails.js. C'est ce fichier qui contient les helpers javascript. Téléchargez les helpers rails.js de votre framework JavaScript préféré : [http://mootools.net/forge/p/rails\\_3\\_driver](http://mootools.net/forge/p/rails_3_driver) ou <http://github.com/rails/jquery-uis>

Remplacez le fichier rails.js par ce dernier. Pensez également à ajouter le fichier de la bibliothèque du framework choisi. Il ne reste plus qu'à changer l'inclusion des fichiers javascript dans le layout :

<%= javascript\_include\_tag :defaults %> devient donc :

```
<%= javascript_include_tag "mootools-1.2.4-core.js", "rails.js" %>
```



Et voilà ! Vous pouvez effacer les fichiers controls.js, dragdrop.js, effects.js et prototype.js (ce sont des bibliothèques ScriptAcuLoUs, un plugin Prototype).

### Et si on l'utilisait pour voir ?

En fait vous utilisez déjà JavaScript ! Dans views/articles/index.html.erb vous pouvez voir la méthode delete :

```
<td><a href="#" data-confirm="Are you sure?" data-method="delete" rel="nofollow">Destroy</a></td>
```

Si vous regardez la source html du site, vous voyez alors que cette ligne génère le code suivant :

```
<td><a href="/articles/1" data-confirm="Are you sure?" data-method="delete" rel="nofollow">Destroy</a></td>
```

Ici, pas de JavaScript inclus sauvagement, les attributs "data-method" et "data-confirm" sont récupérés par le code présent dans rails.js qui s'occupe de tout. Pour Ajax, il suffit d'ajouter l'option **":remote => true**" aux éléments que vous désirez faire exécuter des



requêtes Ajax (formulaires, liens, boutons, ...).

En pratique : Dans `views/articles/_form.html.erb` :

```
1 <%= form_for(@article, :remote => true) do |f| %>
```

Par ce simple ajout, nous avons rendu notre formulaire "Ajax" ! Ainsi c'est une requête Javascript (XHRHttpRequest) qui va effectuer l'envoi du formulaire. Vous remarquerez que la première conséquence est que nous ne sommes plus redirigé après l'envoi du formulaire, ce qui peut sembler déroutant, mais qui est parfaitement normal, puisque nous n'avons pas encore géré le callback de notre envoi de formulaire asynchrone. Pour cela, nous allons surcharger les callbacks existants dans `rails.js`, afin d'appliquer le traitement que nous voulons ! Le fichier `application.js` semble tout indiqué pour cela. Pensez à ajouter `application.js` à la liste des fichiers javascript à charger dans votre layout !

```
4 window.addEventListener('DOMContentLoaded', function() {
5   if($('new_article')){
6     $('new_article').ajaxComplete(function(){
7       window.location = "/articles"
8     });
9   }
10 });
```

Ce script va récupérer l'id du formulaire et ajouter un callback sur l'événement indiquant la réussite de la requête.

Ici nous nous contentons d'une redirection, mais vous pouvez facilement imaginer effacer des données, lancer un effet graphique, etc.

## ACTIONMAILER

Rails 3 a également permis une normalisation de l'API d'envoi de mails. Celle-ci possède à présent une catégorie bien séparée, dans le repertoire `app/mailers`.

Devenue très complète, elle permet d'envoyer et personnaliser des emails complexes (pièces jointes, multi-part), mais également d'en recevoir. Commençons par définir où et comment il sera possible d'envoyer des emails. Vous vous souvenez de la vue "about" du contrôleur "Welcome" ? Cela semble parfait pour y coller un formulaire de contact qui enverra un email ! Cette zone est accessible à l'utilisateur par l'url "myblog.dev/welcome/about". Les fichiers correspondants sont "app/controllers/welcome\_controller.rb" et "app/view/welcome/about.html.erb".

Rajoutons le lien dans le header :

```
<%= link_to "About", welcome_about_path %>
```

Créons le mailer :

```
rails g mailer ContactMailer
```

Dans le repertoire `app/mailers` viens d'apparaître `contact_mailer.rb`. Une ligne de code est déjà présente, qui permet d'indiquer une série de valeurs par défaut :

```
default :from => "from@example.com"
```

La plupart du temps c'est vous et non l'utilisateur qui envoyez un email (newsletter, inscription, etc), c'est donc la valeur du `:from` qui est indiquée par défaut. Rajoutons-y la méthode suivante :

```
4 def contact_email(email, title, content)
5   @email = email
6   @content = content
7   mail(:from => email, :to => "contact@myblog.dev", :subject => title)
8 end
```

Cette méthode va permettre d'envoyer un email à l'adresse indiquée (et non pas celle par défaut), avec pour expéditeur, sujet et contenu les valeurs qui seront indiquées dans le formulaire de contact.

Dans "views/" est apparu un répertoire "action\_mailer/".

Créez y un fichier qui porte le nom de la méthode de votre "Mailer" : "contact\_email.html.erb", et insérez-y le contenu de l'email :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
5   </head>
6   <body>
7     <h1><%= @email %> says :</h1>
8   <p>
9     <%= @content %>
10  </p>
11 </body>
12 </html>
```

C'est une bonne idée de créer son équivalent "plain text", car il arrive que certaines personnes désactivent l'affichage html des emails.

Créez simplement un autre fichier nommé "contact\_email.text.erb", et insérez-y du contenu pur texte :

```
1 <%= @email %> says :
2
3 <%= @content %>
```

Maintenant il faut appeler cette méthode pour envoyer l'email.

```
11 def contact_form
12   message = params[:contact_form]
13   email = message[:email]
14   subject = message[:subject]
15   content = message[:content]
16   ContactMailer.contact_email(email, subject, content).deliver
17
18   redirect_to welcome_about_path
19 end
```

Allez dans "app/controllers/welcome\_controller", et créez une méthode "contact\_form": Cette méthode recevra les paramètres du formulaire et enverra l'email via l'appel :

```
ContactMailer.contact_email(email, title, content).deliver
```

N'oubliez pas de créer la route pour y accéder !

```
post "welcome/contact_form"
```

Rajoutons le formulaire à "app/views/welcome/about.html.erb"

```
1 <h1>Contact us :</h1>
2
3 <%= form_tag(:action => "contact_form") do %>
4   <p>
5     <%= label(:contact_form, :email, "Your Email : ") %><%= email_field :contact_form, :email %>
6   </p>
7   <p>
8     <%= label(:contact_form, :subject, "Subject : ") %><%= text_field :contact_form, :subject %>
9   </p>
10  <p>
11    <%= label(:contact_form, :content, "Message : ") %><br/>
12    <%= text_area :contact_form, :content %>
13  </p>
14  <p>
15    <%= submit_tag "Send", :disable_with => "Sending..." %>
16  </p>
17 <% end %>
```

Vous avez un formulaire de contact entièrement fonctionnel ! Afin de vérifier son fonctionnement en développement, ouvrez le fichier de log "/log/development.log" (l'application Console sous MacOSX est parfaite pour cela). Voici un extrait épuré des parties "inutiles" :

```
Started POST "/welcome/contact_form" for 127.0.0.1 at Tue Oct 26 12:19:57 +0200 2010
Processing by WelcomeController#contact_form as HTML
Parameters: {"commit"=>"Send", "contact_form"=>{"subject"=>"NonTitre", "content"=>"Lorem Ipsum of course !", "email"=>"unmail@mail.com"}}
Rendered contact_mailer/contact_email.html.erb (0.4ms)
Rendered contact_mailer/contact_email.text.erb (0.2ms)

Sent mail to contact@myblog.dev (18ms)
Date: Tue, 26 Oct 2010 12:19:59 +0200
From: unmail@mail.com
To: contact@myblog.dev
Subject: NonTitre

<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
  </head>
  <body>
    <h1>unmail@mail.com says :</h1>
    <p>
      Lorem Ipsum of course !
    </p>
  </body>
</html>
```

On y voit non seulement que l'email est multi-part, mais on voit également s'afficher le contenu de l'email ! Très pratique pour détecter rapidement le moindre problème.

## I18n

L'internationalisation en Rails est relativement aisée, puisqu'elle consiste à utiliser un fichier par langue. Par défaut le fichier `en.yml`

est déjà présent, vous pouvez rajouter un `fr.yml`, `es.yml`, etc. Le principe est simple : une clef «globale», une valeur pour chaque clef dans chaque fichier.

```
config/locales/en.yml :
```

```
en:
  hello: "Hello world"
```

```
config/locales/fr.yml :
```

```
fr:
  hello: "Bonjour le monde"
```

L'utilisation est simple :

```
<h1><%= I18n.t(:hello) %></h1>
```

le 't' est un raccourci pour "translate"

Ecrivez dans le contrôleur `Welcome` :

```
24 def set_locale
25
26   if I18n.locale == :en
27     I18n.locale = :fr
28   else
29     I18n.locale = :en
30   end
31
32   redirect_to root_path
33 end
```

La méthode `set_locale` servira à changer la valeur de la "locale". La "locale" est la langue actuellement utilisée. Le "redirect\_to" précise de rediriger vers la page principale (`welcome/index`), une fois l'opération terminée. Le cas échéant, il aurait fallu une vue associée à la méthode.

On décrit la route dans le fichier `config/routes.rb` :

```
get "welcome/set_locale"
```

On précise que ce sera un GET qui sera effectué vers la méthode `set_locale` et pour finir le code à insérer dans `welcome/index.html.erb` :

```
1 <h1>
2   <%= I18n.t(:hello) %>
3   <span id="change_locale">
4     <span id="actual_locale">
5       &nbsp;&nbsp;&nbsp;<%= I18n.locale.to_s %>
6     </span>
7     <%= link_to (I18n.locale == :en ? :fr : :en), welcome_set_locale_path, :id => "wanted_locale" %>
8   </span>
9 </h1>
```

Ce code est assez simple, il affiche simplement la "locale" courante, et un lien proposant le changement de locale. Vous pouvez continuer en traduisant tous les morceaux statiques de votre application.

## DÉPLOIEMENT

### Le repertoire public

Ce repertoire contient tout ce qui sera directement accessible sur le serveur. On y trouve donc logiquement l'`index.html` que vous devriez avoir supprimé au début, ainsi que tous les autres fichiers auxquels vous voulez avoir accès publiquement, tels que les images, fichiers javascript et feuilles de style.

### servers, passenger, heroku, ...

Il existe divers moyen d'héberger votre application Rails, que ce soit sur un serveur dédié, ou sur le "Cloud".

#### Amazon AWS

La PAAS (Platform As A Service) AWS est composée de nombreux services distribués, dont entre autres :

- EC2 (Elastic Compute Cloud) : puissance
- S3 (Simple Storage Service) : stockage datas
- RDS (Relational Satabase Service) : MySQL
- SQS (Simple Queue Service) : process

Tous ces services étant accessibles via REST, n'importe quelle application Rails peut en profiter. Cependant la complexité d'utilisation d'AWS fait qu'il est plutôt réservé aux développeurs expérimentés.

#### Heroku (Racheté par Salesforce)

Ce PAAS permet de déployer facilement une application Rails sur un nuage de serveur. En fait, il utilise les services d'Amazon (EC2, RDS, et S3), mais vous apporte la simplicité d'utilisation. Il vous suffit d'installer le Gem Heroku et de déployer votre projet via celui-ci.

`sudo gem install heroku`. Si vous utilisez Git : `git push heroku master` dans votre projet, et voilà !

#### Phusion Passenger

C'est un module pour Apache & Nginx pour la mise en production de vos applications.

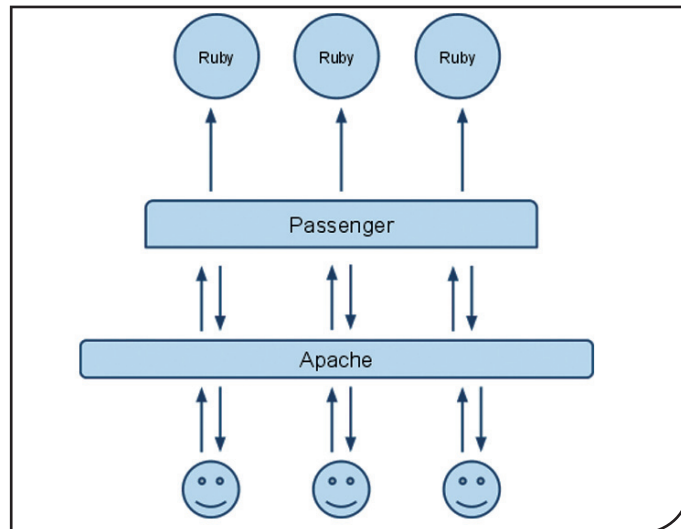


Schéma représentant simplement ce que Phusion Passenger fait pour vous

Constitué par un gem, Phusion Passenger va placer Apache (ou nginx) comme serveur principal, chargé de distribuer les requêtes, et celui-ci communiquera les requêtes qui le nécessitent au serveur Mongrel... Pour installer Passenger, rien de plus simple :

```
gem install passenger
```

puis

```
sudo passenger-install-apache2-module
ou sudo passenger-install-nginx-module selon vos préférences.
```

### Capistrano

Capistrano est un gem qui permet d'effectuer le déploiement de vos application très facilement. Il automatise les tâche de déploiement, permet les "rollback", et tout cela sur plusieurs machines. Idéal si, comme souvent, vous avez divers environnements (development, staging, production)

```
gem install capistrano
capify
```

Il suffit ensuite de compléter le fichier `config/deploy.rb` avec vos informations, et `cap deploy` fera le reste. Capistrano est un gem, pensez donc à le rajouter à votre Gemfile ! Il suffit normalement de décommenter la ligne `# gem 'capistrano'` pré-générée par Rails.

## CONCLUSION

A présent que vous savez comment créer votre projet Rails, et comment le déployer, nous verrons pour la fin comment améliorer votre application à l'aide des plugin et gem les plus connus, et nous verrons également comment utiliser le caching... Toutes les sources sont disponibles sur : <http://github.com/gdurelle/myblog>

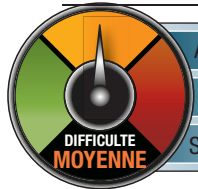


■ Gregory Durelle  
■ Romain Durritçague

# Découverte de Samsung Bada

2<sup>e</sup> partie

Le court projet que nous vous proposons de réaliser ici va vous permettre d'utiliser la caméra du Samsung Wave (ou alors la webcam de votre PC, si vous travaillez dans le simulateur) et de comparer deux photos afin de vérifier si les visages présents sur les images correspondent.



APPLICATION : MOBILE

LANGAGE : C++

SOURCE : OUI

Pour cela nous allons utiliser principalement deux classes du namespace `Osp::Uix`, `FaceDetector` et `FaceRecognizer` et la classe `Camera` du namespace `Osp::Media`.

Notre projet sera composé de deux formulaires : `MainForm` et `CameraForm`. Le premier formulaire servira à comparer les deux photos prises avec l'appareil photo et le deuxième offrira à l'utilisateur la possibilité de capturer une photo. Nous rajouterons aussi la détection de visage sur l'aperçu de la caméra.

## FACERECOGNIZERPROJECT :

### Les prérequis

Il vous faut tout d'abord créer le projet. Choisissez comme type : bada Form Based Application.

Nom du formulaire: `MainForm`

Avant de continuer plus loin, nous allons modifier le fichier `Manifest.xml` afin de rajouter les privilèges suivants :

- IMAGE
- CAMERA

Le premier privilège IMAGE nous permettra d'utiliser les classes relatives à la gestion des bitmaps et différentes ressources graphiques au sein de notre application tandis que le second nous permettra d'accéder au périphérique caméra depuis le code.

### Gérer la caméra

Pour cela nous allons créer une classe `CameraManager` qui héritera de la classe `Osp::Media::Camera`.

Dans le SDK Bada il est nécessaire de différencier deux modes, un mode capture et un mode preview. Le mode capture correspond au résultat que l'on souhaite, alors que le second mode nous permet d'afficher l'aperçu en temps réel de la caméra.

Dans notre classe `CameraManager`, nous ajouterons deux méthodes publiques et quatre attributs en privé :

```
public:
    result StartCamera();
    result StopCamera();
private:
    Osp::Graphics::Dimension* __pCurrentResolutionCapture;
    Osp::Graphics::Dimension* __pCurrentResolutionPreview;
    Osp::Graphics::PixelFormat __currentPixelFormatCapture;
    Osp::Graphics::PixelFormat __currentPixelFormatPreview;
```

Les deux premiers attributs nous permettront de stocker la résolution en cours du mode capture et la résolution en cours du mode preview. Les attributs suivants nous permettent de stocker le type de sortie en fonction du mode utilisé. Nous allons maintenant implé-

menter notre méthode `StartCamera()`. Cette méthode a pour but de démarrer le matériel, mais aussi de définir la résolution ainsi que le format d'image géré par la caméra. Pour cela nous utiliserons les méthodes suivantes afin de configurer la caméra :

- `GetSupportedPreviewResolutionListN()` ; récupère la liste des résolutions disponibles
- `GetSupportedCaptureResolutionListN()` ; récupère la plus grande résolution pour la capture
- `GetSupportedPreviewFormatListN()` ; récupère la liste des formats disponibles pour la preview
- `GetSupportedCaptureFormatListN()` ; récupère la liste des formats disponibles pour la capture

Après l'avoir configuré, il suffit de la démarrer :

```
StartPreview(null, true);
```

Le premier paramètre de `StartPreview()` nous permet de récupérer le flux vidéo et de l'afficher directement (par le biais du buffer `Osp::Graphics::BufferInfo`) cependant, nous utiliserons l'interface `ICameraEventListener` qui nous permettra de récupérer le contenu grâce à l'évènement `OnCameraPreviewed()`. Nous nous servirons aussi de cette méthode pour détecter un visage sur l'aperçu de la caméra. La méthode `StopCamera()` arrête la caméra. Il est très important de le faire, sous peine de voir la caméra inutilisable dans toutes les applications (natives comprises)! Et ce, jusqu'au redémarrage du téléphone.

### Création du formulaire principal

Rajoutez la déclaration de méthode suivante dans le fichier `MainForm.h` :

```
virtual result OnDraw(void);
```

Cette méthode nous permettra de dessiner directement les deux photos dans le canvas du formulaire.

Nous allons définir deux variables statiques au début du fichier `MainForm.m` correspondant aux chemins des fichiers dans lesquels seront stockées les photos :

```
static String photo1Path = L"/Home/FRD/test1.jpg";
static String photo2Path = L"/Home/FRD/test2.jpg";
```

Complétez ensuite le code avec l'implémentation de la méthode `OnDraw` : voir code complet sur notre site.

On effectue une première vérification sur l'existence du fichier de la photo1 (`IsFileExist`). Si le fichier est disponible dans la mémoire du téléphone, alors on récupère l'image (`bitmapDecoder->DecodeN(photo1Path, BITMAP_PIXEL_FORMAT_ARGB8888)`) et on la stocke dans l'objet `photo1`. Il ne nous reste plus qu'à dessiner cette image dans la vue et à adapter sa taille à l'aide du canvas (`canvas->DrawBitmap(Rectangle(x1Step, 200, maxWidth1, newHeight1), *photo1)`).

Vous pouvez préparer la méthode `OnActionPerformed` de la classe



MainForm afin qu'elle utilise le formulaire CameraForm. Lors de la création de l'objet CameraForm on initialisera le mainform avec le formulaire courant et on définira le nom de l'image dans lequel il faudra sauvegarder la capture (pictureName).

```
if (actionId == ID_BUTTON_PHOTO1
    || actionId == ID_BUTTON_PHOTO2){
    //Création de la form qui gèrera la caméra
    CameraForm *cameraForm = new CameraForm();
    cameraForm->Initialize();
    cameraForm->setMainForm(this);
    String picName;
    picName = (actionId == ID_BUTTON_PHOTO1) ? «test1.jpg» :
«test2.jpg»;
    cameraForm->setPictureName(picName);

    Frame *frame = Osp::App::Application::GetInstance()->Get
AppFrame()->GetFrame();
    result r = frame->AddControl(*cameraForm);
    if (r != E_SUCCESS){
        r = frame->SetCurrentForm( *this );
        if( r == E_SUCCESS ) {
            this->Draw();
            this->Show();
        }
    }
    else{
        frame->SetCurrentForm(*cameraForm);
        cameraForm->Draw();
        cameraForm->Show();
    }
}
```

La première grosse partie du projet est déjà terminée, il ne nous reste plus qu'à enchaîner sur la gestion de la caméra dans le formulaire CaméraForm.

## Utiliser la caméra

Le SDK bada nous permet d'interagir avec l'appareil photo du Samsung Wave grâce à la classe `Osp::Media::Camera`. Il est possible d'utiliser aussi bien la caméra frontale que la caméra arrière, bien évidemment le capteur offrant la meilleure résolution est celui qui se trouve à l'arrière. La classe Camera offre aussi la possibilité de contrôler différents paramètres pour les prises de vue : Contraste, ISO, balance des blancs, zoom, flash, focus et effets. Les images prises grâce à la caméra seront sauvegardées en JPEG. Pour afficher l'aperçu en temps réel de la caméra il est nécessaire d'utiliser un contrôle du type `Osp::Ui::Controls::OverlayPanel`. Nous allons passer dès à présent à la création du formulaire qui utilisera le CameraManager, rajoutez donc un formulaire CameraForm en utilisant le wizard dans UIBuilder.

Insérez les inclusions suivantes :

```
#include «CameraForm.h»
#include «CameraManager.h»
#include <FGraphics.h>
#include <FApp.h>
#include <FIO.h>
```

Et faites dériver la classe CameraForm de l'interface `ICameraEventListener`, afin que l'on puisse s'enregistrer et recevoir les événements de la caméra :

```
public Osp::Media::ICameraEventListener
```

Rajoutez les définitions des méthodes de l'interface `ICameraEventListener` :

```
//ICameraEventListener
virtual void OnCameraAutoFocused (bool completeCondition);
virtual void OnCameraCaptured (Osp::Base::ByteBuffer &captured
Data, result r);
virtual void OnCameraErrorOccurred (CameraErrorReason r);
virtual void OnCameraPreviewed (Osp::Base::ByteBuffer &previewed
Data, result r);
```

Ajoutez les variables suivantes :

```
protected:
static const int ID_SOFT_KEY0 = 100;
static const int ID_SOFT_KEY1 = 101;
private:
    MainForm *mainForm;
    Osp::Base::String __pictureName;
    Osp::Ui::Controls::OverlayPanel* __pOverlayPanel;
    Osp::Graphics::Canvas* __pOverlayPanelCanvas;
```

Le contrôle `OverlayPanel` nous permettra d'afficher l'aperçu de la caméra et le canvas `__pOverlayPanelCanvas` nous servira à dessiner les rectangles récupérés grâce à la détection de visage, directement sur l'aperçu. Ajoutez un mutateur sur `mainForm` :

```
void setMainForm(MainForm *aForm);
```

Cette méthode permettra de garder une référence vers le formulaire principal, afin de pouvoir revenir dessus, lorsque l'on appuiera sur le bouton back. et un autre mutateur sur `__pictureName` :

```
void setPictureName(Osp::Base::String pName);
```

Cette méthode nous permettra de connaître le nom de l'image dans laquelle la capture de la caméra devra être sauvegardée.

Dans le fichier d'implémentation `CameraForm.cpp`, modifiez la méthode `Initialize()` : voir code complet sur notre site. Modifiez ensuite la méthode `OnActionPerformed` de la manière suivante :

```
void
CameraForm::OnActionPerformed(const Control& source, int actionId)
{
    switch(actionId) {
        case ID_SOFT_KEY0: {
            // on prend la photo
            result r = __pCamera->Capture();
            if (IsFailed(r)){

            }
        }
        break;
        case ID_SOFT_KEY1: {
            // on retourne sur le précédent formulaire
            this->backToPreviousForm();
        }
    }
}
```

```
break;
}
}
```

La capture de la photo sera faite lors de l'appel `__pCamera->Capture()`, qui appellera sur le listener `ICameraEventListener` la méthode `OnCameraCaptured`.

Nous allons dès à présent implémenter les méthodes :

```
CameraForm::OnCameraCaptured (Osp::Base::ByteBuffer
&capturedData, result r)
CameraForm::OnCameraPreviewed (Osp::Base::ByteBuffer
&previewedData, result r)
```

- La première méthode nous permet de sauvegarder la photo lors de la capture. L'image en elle-même (son contenu) est la référence `capturedData`, grâce à la méthode `osp::Io::File::Write`, nous enregistrons l'image dans la mémoire interne.
- La seconde méthode nous permet de récupérer chaque frame de l'aperçu. Dans notre cas, le but principal de cette méthode, sera d'afficher un rectangle autour du visage de la personne visée. Nous allons donc modifier l'image source afin justement d'y ajouter le rectangle désiré. Pour cela, dans un premier temps nous allons enregistrer chaque frame reçu de l'appareil photo dans un overlay panel : `overlayPanel.SetInputBuffer(Dimension( 800, 480), previewedData, Dimension(previewWidth,previewHeight), bufferPixelFormat);` Puis nous allons utiliser la classe `osp::Uix::FaceDetector` qui va nous permettre, comme son nom l'indique, de détecter les visages d'une image ou d'un flux. Créons un nouvel attribut à notre classe `__pFaceDetector`. Cet attribut évitera au téléphone de créer à chaque frame reçu un nouvel objet. On gagne ainsi en performance et de plus l'objet `__pFaceDetector` se servira des précédents frames pour améliorer la détection des visages.

Dans la fonction `Initialize()` de notre classe on instancie et on configure notre objet `FaceDetector`.

```
__pFaceDetector = new FaceDetector();
__pFaceDetector->Construct();
__pFaceDetector->SetProperty(FACEDETECTOR_MAXNUMBER_IMAGE, 1);
__pFaceDetector->SetProperty(FACEDETECTOR_SCALE_IMAGE, 1);
long minFaces, maxFaces, step, initial;
__pFaceDetector->GetRange(FACEDETECTOR_MAXNUMBER_VIDEO,minFaces,
maxFaces, step, initial);
__pFaceDetector->SetProperty(FACEDETECTOR_MAXNUMBER_VIDEO,max
Faces);
__pFaceDetector->SetProperty(FACEDETECTOR_SCALE_VIDEO, 1);
```

Note : Il est possible de détecter plusieurs visages en même temps.

A chaque frame reçu par l'appareil photo (dans la méthode `OnCameraPreviewed`), rajoutez les lignes suivantes :

```
// on recherche les visages
IList *infos = __pFaceDetector->DetectFacesFromVideoStreamN
(previewedData, __pCamera->GetPreviewResolution(), __pCamera->
GetPreviewFormat());
// si il detecte au moins un visage
if (infos->GetCount() > 0){
// on créé et on affiche le rectangle dans le canvas de
l'overlayPanel
Rectangle* pRect = (Rectangle *)infos->GetAt(0);
__pOverlayPanelCanvas->SetBackgroundColor(Color(0,0,0,0));
```

```
__pOverlayPanelCanvas->Clear();
__pOverlayPanelCanvas->DrawRectangle(*pRect);
__pOverlayPanelCanvas->Show();
}
```

Ici on utilise l'objet `__pFaceDetector` afin de détecter un visage directement à partir du flux vidéo renvoyé par la caméra (`previewedData`) et si le face detector retourne des résultats (`infos->GetCount()->0`) alors on affichera le rectangle correspondant au visage directement sur l'aperçu de la caméra (`__pOverlayPanelCanvas->DrawRectangle(*pRect)`). Maintenant que les images sont capturées et sauvegardées, il ne nous reste plus qu'à passer à la reconnaissance de visage qui sera effectuée dans le formulaire `MainForm`.

## Retour sur le premier formulaire

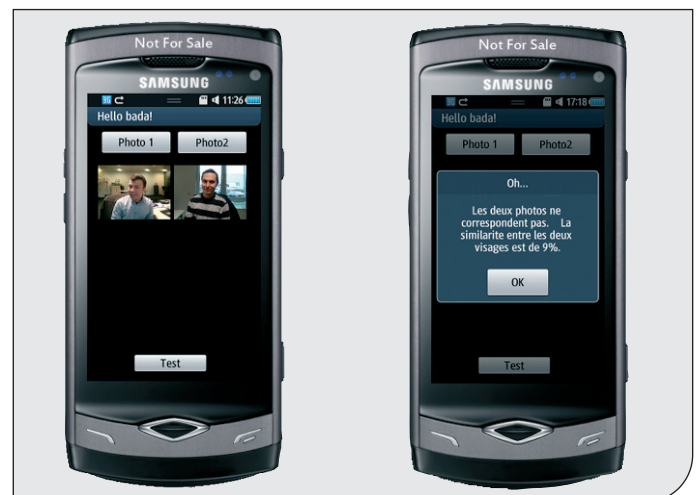
Une fois la photo réalisée, nous retournons sur l'écran principal. Sur cet écran nous proposons à l'utilisateur de comparer les photos prises. Nous ajoutons donc un bouton qui analysera les photos. Pour les analyser nous utiliserons la classe `osp::Uix::FaceRecognizer` qui permet comme la classe `FaceDetector` de détecter des visages sur une photo ou vidéo, mais elle est capable aussi d'évaluer deux photos en fonction d'un pourcentage (voir code complet sur notre site). Le projet est maintenant terminé, vous pouvez passer à la compilation et lancer l'application.

Une fois les deux photos capturées, vous pourrez sur le formulaire principal appuyer sur le bouton test qui se chargera de comparer les deux images. Un message sera ensuite affiché dans le cas où les deux visages trouvés sur la photo correspondent à la même personne ou alors comme c'est le cas dans l'exemple ici, un pourcentage de ressemblance entre les deux visages sera indiqué. Les résultats sont plutôt convaincants si l'on place bien la tête du sujet de face, évidemment. Comme vous avez pu le constater, la mise en œuvre de la détection et de la reconnaissance de visages est assez aisée grâce aux API très simples fournies dans le SDK Bada.

Pragma Consult a réalisé dans le cadre du concours mondial de développement Samsung Bada, l'application Doogies qui a remporté le premier prix de la catégorie Mates and Dates. Plus d'infos sur <http://developer.bada.com/challenge/main.do?menu=CM01010000&regionID=L000000000>.

■ Alexis Détré, élève ingénieur, SUPINFO Metz

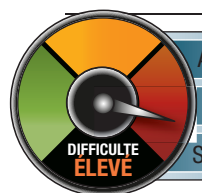
■ Yvan Moté, Mobile Technology Consultant chez Pragma Consult, Luxembourg



# Communiquer avec les périphériques USB sous Windows



L'atout majeur des périphériques USB est qu'on les insère dans le système et qu'on les utilise sans se poser de question. Mais un geek s'en posera une tôt ou tard : « comment communiquer avec le périphérique par programmation » ? Nous tentons d'y répondre.



APPLICATION : USB

LANGAGE : C++

SOURCE : OUI

La facilité d'utilisation des périphériques USB est un must pour l'utilisateur lambda. Mais pour le programmeur c'est autre chose.

Comment savoir si le périphérique est présent ou s'il est brutalement retiré pendant que notre application tente de l'utiliser ? Comment communiquer avec lui ? Imaginons un exemple concret. Nous voudrions écrire une application qui conserve des données secrètes, cryptées. Il n'est possible d'accéder aux données décryptées que si l'utilisateur insère une clé USB très particulière sur laquelle une clé de cryptage est cachée quelque part, non dans un fichier, mais dans un secteur. Cet article donne les éléments pour construire une telle application sous Windows. Et bien sûr, le contenu de cet article est directement transposable pour travailler avec n'importe quel périphérique USB dont on dispose de la documentation. On suppose que le lecteur est déjà très familiarisé avec la programmation Windows. Nous avons écrit nos exemples, dont les sources complètes sont disponibles sur notre site, avec Visual Studio 2008. N'importe quel compilateur C++ conviendra. Les exemples ont été testés sous Windows 7. Cet article fait allusion au DDK (Driver Development Kit), mais il n'est pas nécessaire d'installer celui-ci.

## 1 WINDOWS ET LES PÉRIPHÉRIQUES

Penchons-nous d'abord sur la façon dont Windows gère les périphériques. Le sujet est complexe et même obscur tant que l'on ne plonge pas dans le Windows Driver Development Kit (DDK), ce qui n'est pas envisageable dans le cadre de cet article. Disons simplement qu'un périphérique est un matériel qui donne des informations sur son identité, sur son constructeur, qui utilise des bibliothèques constituant son pilote, etc. Au final un périphérique devient une abstraction constituée par des données qui seront donc rangées dans une base de données. Le gestionnaire de périphériques que nous connaissons tous donne une représentation de cette base de données [Fig.1]. Celle-ci est en grande partie la base de registre du système, mais pas uniquement pour des raisons historiques. Dans une base de données les données sont fréquemment rangées en face d'index. Nous trouvons ici un équivalent aux index avec les GUID. Les GUID (Global Unique Identifier) sont des valeurs 128 bits dont l'unicité est garantie. Un GUID recense une classe de périphériques. Il y a par exemple un GUID pour les disques. Il existe bien sûr un GUID pour les périphériques USB. Ceci implique qu'un disque USB est rangé deux fois, une fois dans chaque classe. Deux fois au moins... Il existe une classe de périphériques amovible. Un disque USB étant

amovible a priori, il est rangé dans 3 classes. Au moins... Car un disque a sans doute aussi un volume, et il existe une classe de volume... Arrêtons-nous là :) Nous en savons assez pour ne pas être surpris lors de nos essais.

## 2 ÊTRE NOTIFIÉ DU (DÉ)BRANCHEMENT D'UN PÉRIPHÉRIQUE USB

Le système d'exploitation Windows prévoit de notifier les applications lors d'une modification de l'arborescence des périphériques USB. Concrètement, quand un périphérique est ajouté ou retiré. Cette notification se fait par envoi d'un message WM\_DEVICECHANGE à la procédure de fenêtre. L'application doit donc posséder une fenêtre qui peut, si besoin, être invisible. Le message WM\_DEVICECHANGE apporte d'abord un certain nombre d'informations avec son premier paramètre (wParam). Voici les valeurs les plus intéressantes :

wParam de WM_DEVICECHANGE	Signification
DBT_DEVNODES_CHANGED	L'arborescence des périphériques a été modifiée. Ce wParam n'existe pas sous Windows NT 4.0 et systèmes inférieurs.
DBT_DEVICEARRIVAL	Un nouveau périphérique a été installé (ou connecté dans le cas d'un périphérique USB)
DBT_DEVICEREMOVECOMPLETE	Un périphérique a été désinstallé (ou déconnecté dans le cas d'un périphérique USB)

Le premier élément de ce tableau est très important puisqu'il est présent dans le message WM\_DEVICECHANGE de toute modification. Malheureusement il n'est pas très explicite et il n'y a pas d'information complémentaire dans le lParam associé. Prenons un cas concret: vous allumez votre webcam. A ce moment le système diffuse le message WM\_DEVICECHANGE avec DBT\_DEVNODES\_CHANGED en guise de wParam. Mais vous n'avez aucune information vous permettant de savoir que c'est une webcam qui a été activée. La documentation MSDN le précise bien: «une application qui maintient une liste des périphériques doit rafraîchir cette liste à tout moment.» Nous verrons comment procéder un peu plus loin. Les deux autres valeurs sont plus intéressantes car le lParam associé permet de déterminer la lettre de volume concerné, par exemple lors de l'enfichage d'une clé USB. Voici maintenant un extrait de l'exemple DemoDeviceChange (disponible complet sur notre site) qui illustre ceci :

```
void PrintDevice(HWND hOutputWnd, LPARAM lParam)
{
    WCHAR drive[] = L"x:";
    DWORD mask;
    WCHAR i;
```



```
StringCchPrintf(buffer, LENBUF, TEXT("\n"));
OutputMessage(hOutputWnd, 0, (LPARAM)buffer);

PDEV_BROADCAST_HDR dbh = (PDEV_BROADCAST_HDR)lParam;
PDEV_BROADCAST_DEVICEINTERFACE dbd =
    (PDEV_BROADCAST_DEVICEINTERFACE)lParam;
PDEV_BROADCAST_OEM dbo = (PDEV_BROADCAST_OEM)lParam;
PDEV_BROADCAST_VOLUME dbv = (PDEV_BROADCAST_VOLUME)lParam;

switch(dbh->dbch_devicetype)
{
case DBT_DEVTYP_DEVICEINTERFACE:
    StringCchPrintf(buffer, LENBUF,
        TEXT("DBT_DEVTYP_DEVICEINTERFACE\n"));
    OutputMessage(hOutputWnd, 0, (LPARAM)buffer);

    StringCchPrintf(buffer, LENBUF,
        TEXT("Name: %s\n"), dbd->dbcc_name);
    OutputMessage(hOutputWnd, 0, (LPARAM)buffer);
    break;
//etc
default:
    break;
}

case WM_DEVICECHANGE:
{
    switch (wParam)
    {
    case DBT_DEVICEARRIVAL:
        PrintDevice(hOutputWnd, lParam);
    }
}
```

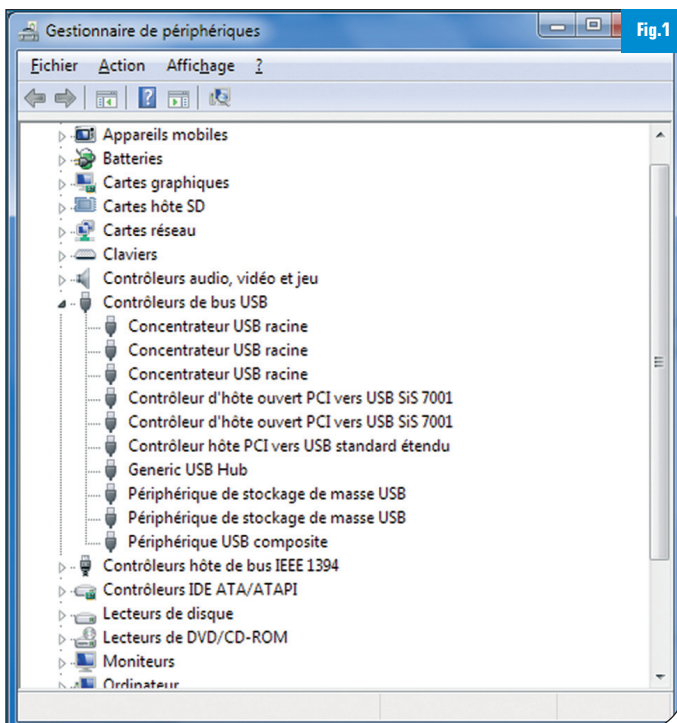
```
StringCchPrintf(buffer, LENBUF,
    TEXT("DBT_DEVICEARRIVAL\n"));
    break;
//etc
default:
    StringCchPrintf(buffer, LENBUF, 7
        TEXT("cas WM_DEVICECHANGE non traité\n"));
    break;
}
OutputMessage(hOutputWnd, wParam, (LPARAM)buffer);
}
break;
```

Quelques remarques à propos de ce code. D'abord le nom de volume n'est pas fourni directement par le système (pourquoi ?...) mais sous la forme d'un masque. La routine PrintDevice montre comment exploiter ce masque. La fonction OutputMessage, est tirée de MSDN et n'est donc pas reproduite ici. Il est possible d'affiner un petit peu le comportement d'une application exploitant WM\_DEVICECHANGE au moyen de l'API RegisterDeviceNotification, mais cette API est surtout valable dans le cas d'un service. Enfin ce code ne fonctionne pas correctement, même dans le cas de volumes! En effet comme on peut le voir sur l'illustration [Fig.2], le retrait d'un concentrateur sur lequel sont enfichées deux clés ne provoque qu'une notification de retrait de volume mais provoque bien deux notifications d'ajout lorsqu'on remet le concentrateur en place. Au final on retiendra que WM\_DEVICECHANGE n'est pas vraiment exploitable en dehors du seul fait de notifier «qu'il se passe quelque chose». A nous de faire tout le travail ensuite, ce que nous abordons maintenant.

### 3 ENUMÉRER LES PÉRIPHÉRIQUES DU SYSTÈME

Sous Windows, tous les périphériques, qu'ils soient installés de façon classique ou à chaud (cas de l'USB), relèvent de l'API d'installation, la SetupApi qui devra donc être ajoutée à votre projet, comme illustré pour Visual Studio [Fig.3]. Le fichier SetupApi.lib fait partie du SDK Windows. Cette librairie propose des fonctions pour s'enquérir de tous les périphériques connus par un système. Au centre de ces fonctions, nous avons SetupDiGetClassDevs. Cette fonction permet de lister tous les périphériques connus, ou tous les périphériques présents, et de les filtrer éventuellement par classes, lorsqu'on lui fournit en argument un de ces GUID de classes dont nous avons parlé plus haut. En retour, la fonction fournit un ensemble de données que l'on peut parcourir au moyen de la fonction SetupDiEnumDeviceInfo pour extraire, à chaque appel, une structure DeviceInfoData qui décrit un périphérique. SetupDiEnumDeviceInfo reçoit en argument un index basé zéro que le programmeur doit se charger d'incrémenter. La fonction retourne FALSE lorsque l'ensemble a été totalement parcouru. Nous donnons ci-dessous un exemple, DemoEnumDevices, qui interroge le système pour toutes les classes de périphériques présents. Ensuite nous utilisons SetupDiGetDeviceRegistryProperty pour afficher certaines des caractéristiques des périphériques:

```
#include <windows.h>
#include <setupapi.h>
#include <iostream>
```



Le gestionnaire de périphériques donne une représentation de la base de données des périphériques du système.

```
#include <fstream>

using namespace std;

wofstream logfile("logfile.txt", ios::ate|ios::out);
//define wcout logfile

// Se débarrasser d'éventuels caractères à problème
// pour wcout
void FixWChar(WCHAR* p) {
    while(*p) {
        if(*p == 8217) *p = 39;
        p++;
    }
}

void PrintDeviceRegistry(HDEVINFO di, PSP_DEVINFO_DATA dd)
{
    const DWORD lenbuf = 512;
    WCHAR strbuf[lenbuf];
    // Obtenir la classe du device
    SetupDiGetDeviceRegistryProperty(di, dd, SPDRP_CLASS, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"Classe: " << strbuf << endl;
    // Obtenir le GUID classe du device
    SetupDiGetDeviceRegistryProperty(di, dd, SPDRP_CLASSGUID, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"GUID de classe: " << strbuf << endl;
    // Obtenir le FriendlyName du device
    SetupDiGetDeviceRegistryProperty(di, dd, SPDRP_FRIENDLYNAME, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"FriendlyName: " << strbuf << endl;
    // Obtenir le LOCATION_INFORMATION du device
    SetupDiGetDeviceRegistryProperty(di, dd,
        SPDRP_LOCATION_INFORMATION, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"Location information: " << strbuf << endl;
    // Obtenir le LOCATION_PATH du device
    SetupDiGetDeviceRegistryProperty(di, dd, SPDRP_LOCATION_PATHS, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"Location path: " << strbuf << endl;
    // Obtenir le PHYSICAL_DEVICE_OBJECT_NAME du device
    SetupDiGetDeviceRegistryProperty(di, dd,
        SPDRP_PHYSICAL_DEVICE_OBJECT_NAME, 0,
        (PBYTE)strbuf, lenbuf, 0);
    //FixWChar(strbuf);
    wcout << L"Physical device object name: " << strbuf << endl;
    SetupDiGetDeviceRegistryProperty(di, dd, SPDRP_DEVICEDESC, 0,
        (PBYTE)strbuf, lenbuf, 0);
    FixWChar(strbuf);
    wcout << L"Desc: " << strbuf << endl;
}

int main(int argc, char** argv) {
```

```
HDEVINFO hDevInfo;
SP_DEVINFO_DATA DeviceInfoData;
// Obtenir un handle sur la liste de
// tous les périphériques présents
hDevInfo = SetupDiGetClassDevs(NULL,
    0, 0, DIGCF_PRESENT | DIGCF_ALLCLASSES);

if (hDevInfo == INVALID_HANDLE_VALUE)
    return 1;

// Parcourir la liste de tous les périphériques
DWORD index = 0;
DeviceInfoData.cbSize = sizeof(SP_DEVINFO_DATA);
while(SetupDiEnumDeviceInfo(hDevInfo, index, &DeviceInfoData)) {
    index++;
    wcout << L"Device " << index << L" : " << endl;
    PrintDeviceRegistry(hDevInfo, &DeviceInfoData);
    wcout << endl;
}
SetupDiDestroyDeviceInfoList(hDevInfo);
return 0;
}
```

Ce code appelle quelques commentaires. Un détail d'abord: la fonction `FixWChar` sert uniquement à corriger une apostrophe (') en Unicode de derrière les fagots dans un périphérique «Contrôleur d'hôte» sur la machine de votre serveur et qui fait perdre les pédales au flux C++. Votre serveur a eu la surprise de constater, en témoigne la capture [Fig.4] que son ordinateur portable dispose de 354 périphériques! Et cela sera très probablement pareil pour votre machine. Que se passe-t-il ? Pour y voir clair, retirez le commentaire de cette ligne de code, afin que le programme produise sa sortie dans un fichier (logfile.txt).

```
#define wcout logfile
```

En examinant le contenu de fichier on comprend que chaque périphérique apparaît plusieurs fois. Ainsi une clé va apparaître comme périphérique USB, comme lecteur de disque, et comme volume. Par exemple sur ma machine, une clé USB donne ceci :

```
Device 20 :
Classe: DiskDrive
GUID de classe: {4d36e967-e325-11ce-bfc1-08002be10318}
FriendlyName: CBM Flash Disk USB Device
Location information: CBM Flash Disk USB Device
Location path: CBM Flash Disk USB Device
Physical device object name: \Device\00000154
Desc: Lecteur de disque
```

Ou encore en tant que volume amovible (WPD), ici F :

```
Device 136 :
Classe: WPD
GUID de classe: {eec5ad98-8080-425f-922a-dabf3de3f69a}
FriendlyName: F:\
Location information: F:\
Location path: UMBROOT(0)#UMB(WpdBusEnumRoot)#UMB(STORAGE#
VOLUME#_??_USBSTOR#DISK&VEN_CBM&PROD_FLASH_DISK&REV_2.00#6&73
8D485&0#)
```

```
Physical device object name: \Device\00000156
Desc: Flash Disk
```

Trois questions se posent maintenant : Comment affiner le filtrage des périphériques, comment mettre en correspondance un volume et un périphérique et comment ensuite utiliser ces informations pour communiquer avec le périphérique ? L'exemple suivant répondra à la première question. Pour ce qui est de mettre en correspondance les périphériques, on analysera les Location Path et Friendly Name. Ainsi on retrouve bien les informations CBM Flash Disk dans le Location path du volume F : et nous voyons qu'il s'agit bien d'un périphérique USB (USBSTOR). C'est rassurant, mais insuffisant pour deux raisons. D'abord un nom de volume ne permettra pas forcément de réaliser notre but, car lors des accès bas niveau que nous voulons faire, il est possible que premier secteur du disque et le premier secteur de la partition soient deux choses distinctes. Et enfin, ces informations pour passionnantes qu'elles soient ne permettent pas de communiquer avec les périphériques. Sous Windows on communique avec un périphérique en commençant par l'ouvrir comme un fichier au moyen de l'API CreateFile et en fournissant à celle-ci un chemin vers le dit périphérique. Malheureusement, aucun des Location Path ainsi obtenus ne fonctionne.

## 4 AFFINER LES RÉSULTATS ET OBTENIR DES INTERFACES

L'exemple suivant, DemoEnumDisk, filtre les disques présents sur le système en fournissant des GUID à la fonction SetupDiGetClassDevs. Le programme procède en deux passes : d'abord tous les disques, puis tous les disques amovibles. A la première passe trois disques sont détectés : le disque dur de la machine et deux clés USB enfichées. A la deuxième passe, seules les deux clés sont détectées. Voici un extrait de ce code :

```
GUID diskGUID = GUID_DEVCLASS_DISKDRIVE;
GUID wpdGUID = GUID_DEVCLASS_WPD;

// Obtenir un handle sur la liste de
// tous les périphériques de types Disk présents
hDevInfo = SetupDiGetClassDevs(&diskGUID,
    0, 0, DIGCF_PRESENT);
// etc...
```

```
// Obtenir un handle sur la liste de
// tous les périphériques de types Disk amovible présents
hDevInfo = SetupDiGetClassDevs(&wpdGUID,
    0, 0, DIGCF_PRESENT);
// etc...
```

Où se procurer les GUID passés à la fonction ? Ceux-ci se situent dans le fichier en-tête devguid.h du SDK Windows. Pour l'instant nos résultats sont filtrés, mais il ne sont toujours pas exploitables pour communiquer avec les périphériques. A partir de la documentation MSDN, vraiment pas claire sur ce point, on doit deviner qu'il faut filtrer les «périphériques qui supportent une interface de périphériques». Ceci s'obtient en passant le drapeau DIGCF\_INTERFACEDevice à la fonction SetupDiGetClassDevs qui comme précédemment va retourner un ensemble de données mais cette fois il s'agira d'interfaces. Cet ensemble d'interfaces doit être énuméré (ou parcouru) à l'aide d'une autre fonction, SetupDiEnumDeviceInterfaces, dont l'utilisation est similaire à SetupDiEnumDeviceInfo, avec une nuance de taille, cette fonction permet elle aussi de filtrer le type d'interfaces énumérées. Ensuite pour chaque interface obtenue lors de cette énumération, il est possible d'invoquer la fonction SetupDiGetDeviceInterfaceDetail, pour obtenir, comme le nom le suggère, des informations supplémentaires, qui se réduisent à un DevicePath qui cette fois sera utilisable par l'API CreateFile. En outre SetupDiGetDeviceInterfaceDetail a ceci de très sympathique que son appel permet également de remplir une structure DeviceInfoData (voir premier programme d'énumération), ce qui offre la possibilité de mettre en correspondance un périphérique proprement dit et son interface d'accès. Nous allons donc réécrire notre programme d'énumération de périphériques/interfaces, et nous allons filtrer ces interfaces en deux passes : d'abord pour énumérer les interfaces disques, et ensuite pour énumérer les interfaces USB, espérant ainsi n'obtenir que les disques USB. Cet exemple, DemointerfaceDisk.cpp est disponible sur notre site.

Le GUID pour les interfaces USB provient du Driver Development Kit. CE GUID est nécessaire et suffisant, il n'y a pas besoin d'installer le DDK pour cet exemple. Exemple qui nous réserve encore une surprise... Sur ma machine d'essai, un ordinateur portable, j'ai eu la surprise de voir que le disque dur a une interface USB... qui ne fonctionne pas. Le même programme sur un PC de bureau donne des

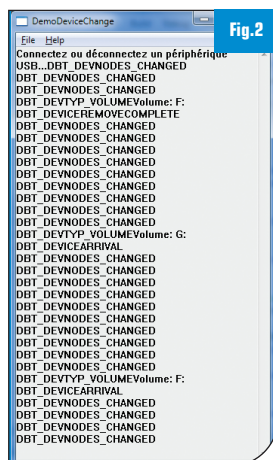


Fig.2

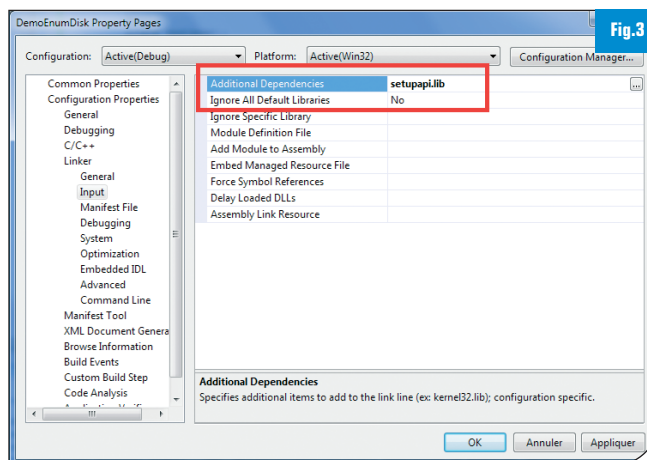


Fig.3



Fig.4

Notre application à l'écoute du message WM\_DEVICECHANGE

Ajout de la SetupApi à un projet Visual Studio.

Surprise : 354 périphériques recensés sur une seule machine !



résultats plus conformes aux attentes. Mais tout ceci démontre que seulement rechercher des interfaces USB est source de dysfonctionnement. Pour que tout fonctionne bien, il faut rechercher ces interfaces et vérifier les résultats en examinant en parallèle la nature du périphérique concerné. Comme le montre la capture [Fig.4], nous obtenons des chemins de périphérique plutôt compliqués. Par exemple sur ma machine pour une clé USB :

```
\\?\usbstor#disk&ven cdm&prod flash_disk&rev 2.00#6&738d485&0
#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Ou
\\?\usb#vid 13fe&pid 3600#079b1a0081187746#{a5dcbf10-6530-11
d2-901f-00c04fb951ed}
```

dont seul celui préfixé avec usbstor et décrivant spécifiquement un disque fonctionnera correctement avec l'exemple suivant.

## 5 ENFIN, COMMUNIQUER AVEC LE PÉRIPHÉRIQUE

Nous arrivons enfin à notre dernier exemple, qui sans doute nécessitera les droits de l'administrateur pour fonctionner. Nos clés USB étant des périphériques de type disque, nous allons en lire le premier secteur [Fig.5]. Le système pointe sur celui-ci par défaut à l'ouverture du périphérique avec la fonction CreateFile. Lire un secteur quelconque est possible théoriquement, mais peut nécessiter auparavant d'obtenir un lock sur la partition contenant le dit secteur, voire sur toutes les partitions du disque. On pointera ensuite sur le bon secteur avec la fonction SetFilePointer. Encore faut-il, pour exploiter celle-ci, connaître la géométrie du disque. Géométrie que notre exemple obtient par communication directe de bas niveau avec la fonction DeviceIoControl [Fig.6].

```
#include <windows.h>
#include <winioctl.h>
#include <iostream>
using namespace std;

// ADAPTEZ à votre MACHINE!!
//WCHAR chemin[] = L"\\\\.\\usbstor#disk&ven cdm&prod flash_disk
&rev 2.00#6&738d485&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}";
//WCHAR chemin[] = L"\\\\.\\F:";

int main(int argc, char* argv[]) {
    HANDLE device = CreateFile(chemin, GENERIC_READ, FILE_SHARE_READ,
        NULL, OPEN_EXISTING, 0, NULL);
    if(device == INVALID_HANDLE_VALUE) {
        wcout << L"Impossible d'ouvrir le device" << endl;
        return 1;
    }
    wcout << "Device: " << chemin << endl << endl;

    DWORD nbreturned;
    STORAGE_DEVICE_NUMBER sdn;
    ZeroMemory(&sdn, sizeof(STORAGE_DEVICE_NUMBER));
    DeviceIoControl(device, IOCTL_STORAGE_GET_DEVICE_NUMBER,
        0, sizeof(STORAGE_DEVICE_NUMBER),
        &sdn, sizeof(STORAGE_DEVICE_NUMBER),
        &nbreturned, 0);
```

```
wcout << L"Device number: " << sdn.DeviceNumber << endl;
wcout << L"Device type: " << sdn.DeviceType << endl;
wcout << L"Partition number: "
    << sdn.PartitionNumber << endl << endl;

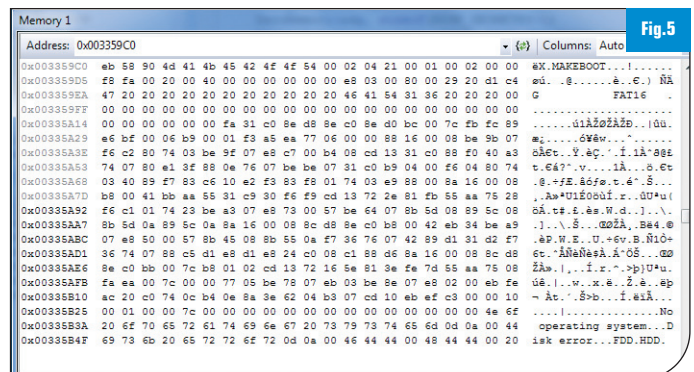
DISK_GEOMETRY dg;
ZeroMemory(&dg, sizeof(DISK_GEOMETRY));
DeviceIoControl(device, IOCTL_DISK_GET_DRIVE_GEOMETRY,
    0, 0,
    &dg, sizeof(DISK_GEOMETRY),
    &nbreturned, 0);

wcout << L"Géométrie du disque:" << endl;
wcout << L"Octets par secteur: " << dg.BytesPerSector << endl;
wcout << L"Nb de cylindres: " << dg.Cylinders.QuadPart << endl;
wcout << L"Type de media: " << dg.MediaType << endl;
wcout << L"Nb de secteurs pas piste: " << dg.SectorsPerTrack << endl;
wcout << L"Nb de pistes par cylindres: "
    << dg.TracksPerCylinder << endl;

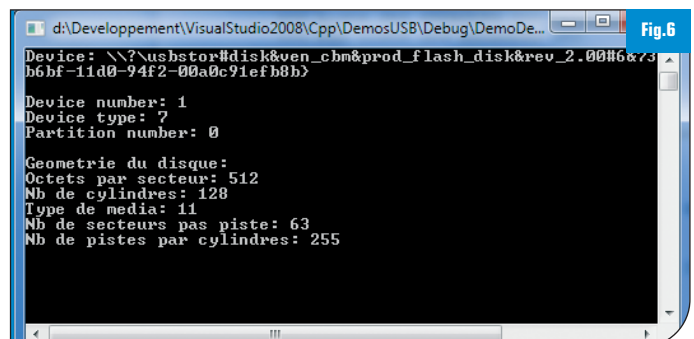
char* buffer =(char *)LocalAlloc(LPTR, dg.BytesPerSector);
ReadFile(device, buffer, dg.BytesPerSector, &nbreturned, 0);
LocalFree(buffer);
CloseHandle(device);

wcout << endl << L"Appuyez sur [RETURN]" << endl;
cin.get();
return 0;
}
```

■ Frédéric Mazué - [fmazue@programmez.com](mailto:fmazue@programmez.com)



Le premier secteur d'une clé USB lu et affiché sous Visual Studio.



La géométrie d'une clé USB obtenue avec DeviceIoControl.

# Economisez jusqu'à 50%



Programmez ! est le magazine du développement 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 ! [www.programmez.com](http://www.programmez.com)

**1** -25%

## Abonnement 1 an

**49€** au lieu de 65,45 € tarif au numéro - Tarif France métropolitaine

**2** +0,8€ par mois

## Abonnement Intégral : + archives

1 an au magazine + archives sur Internet et PDF  
**59€** Tarif France métropolitaine

**3** jusqu'à -50%

## Abonnement 2 ans + 1 livre numérique ENI

• **79€** au lieu de 130,90 (valeur de 22 numéros) Tarif France métropolitaine + un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€** 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI



**OUI, je m'abonne**

Vous pouvez vous abonner en ligne et trouver tous les tarifs [www.programmez.com](http://www.programmez.com)

☐ Abonnement 1 an au magazine : 49 € (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine

☐ Abonnement Intégral : 1 an au magazine + archives : 59 € Tarif France métropolitaine

☐ Abonnement 2 ans au magazine + livre numérique ENI : 79 € Tarif France métropolitaine

☐ Abonnement 2 ans au magazine + livre numérique ENI + archives : 89 € Tarif France métropolitaine

Livres à Choisir : ☐ Visual Studio 2010 ☐ PHP5.3 ☐ Bing Maps ☐ MySQL 5, Administration et optimisation

☐ Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring. Détails sur [www.programmez.com/abonnement.php](http://www.programmez.com/abonnement.php)

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

Prénom : \_\_\_\_\_ Nom : \_\_\_\_\_

Adresse : \_\_\_\_\_

Code postal : \_\_\_\_\_ Ville : \_\_\_\_\_

Tél : \_\_\_\_\_ (Attention, e-mail indispensable)

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](mailto:abonnements.programmez@groupe-gli.com)

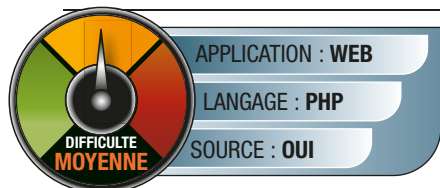
Offre limitée, valable jusqu'au 31 janvier 2011

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.

Le magazine du développement  
**PROgrammez !**

# Symfony 2.0 : PHP comme vous ne l'avez jamais vu !

Symfony est un framework PHP, entièrement développé en suivant le modèle MVC. Son code est de grande qualité, et sa documentation très complète, ce qui le rend facile à apprendre et à mettre en oeuvre.



Il peut être utilisé pour tout type de projet, et plusieurs de ses raffinements (système de template évolué, générateur de backend, couche de mapping objet-rela-

tionnel) permettent d'avoir rapidement un site fonctionnel. C'est actuellement l'un des frameworks les plus aboutis du monde PHP. Sa force vient de l'utilisation des concepts et pratiques qui ont fait leurs preuves dans d'autres langages ou frameworks (Java ou Ruby on Rails pour ne citer qu'eux). Depuis la version 1.0, le framework n'avait que peu évolué en termes de fonctionnalités, les ajouts et corrections effectués étant mineurs. Symfony2 marque un tournant, en modifiant en profondeur une grande partie de ses concepts fondamentaux, et en introduisant une nouvelle architecture plus simple et plus claire.

## PRÉ-REQUIS

L'ensemble du code présenté va utiliser le projet « Sandbox » qui permet de se lancer immédiatement dans des expérimentations autour de Symfony. Le projet « Sandbox » est directement téléchargeable sous forme d'archive sur le site de Symfony 2 (<http://symfony-reloaded.org/code>). Il est nécessaire de disposer d'au moins un PHP en version 5.3.2 pour pouvoir utiliser Symfony 2.

## ARCHITECTURE GÉNÉRALE D'UN PROJET

Voici l'architecture physique du projet «Symfony Sandbox», :

```
hello/
  cache/
  config/
  logs/
src/
  Application/
    HelloBundle/
      Controller/
      Resources/
  vendor/
    symfony/
web/
```

Les fichiers généraux de l'application (configuration, cache, logs, etc.) sont stockés dans le répertoire portant le nom de l'application (ici hello), et l'application elle-même est stockée dans le répertoire src/Application, sous la forme d'un bundle (nous y reviendrons dans la seconde partie de ce dossier). Les bibliothèques utilisées dans ce projet sont placées dans le répertoire src/vendor, ici seul le core de

Symfony est inclus, mais si l'inclusion d'autres bibliothèques est nécessaire, elles devront être placées à cet endroit. Le dernier répertoire racine, web, est le répertoire où sont placés les front controllers appelés lors d'une requête (cf. partie concernant les différents environnements). Ce sera donc ce répertoire qui devra être défini comme répertoire racine lors de la mise en production du projet, pour plus de sécurité. Notre projet « Réseaux Sociaux » va s'appuyer sur ce projet «Symfony Sandbox», ce qui va nous permettre de développer rapidement les fonctionnalités attendues, sans avoir à effectuer la première phase de configuration.

## BUNDLES

Le système des "bundles" est une des modifications les plus importantes introduite dans Symfony2. Ce système remplace celui des plugins de la version précédente, mais il va aussi beaucoup plus loin. Maintenant tout dans un projet est bundle, et doit être implémenté sous la forme de bundle :

- le core de Symfony2,
- le code de tiers-partie,
- et même l'application elle-même.

De cette manière, le partage de fonctionnalités entre les applications est grandement facilité, de même que la compréhension d'un projet existant.

## Qu'est-ce qu'un bundle ?

Un bundle est un répertoire contenant un ensemble de fichiers qui peuvent être organisés de manière très flexible, mais un ensemble de bonnes pratiques ont été définies pour optimiser cette organisation. Ces bonnes pratiques sont en fait une implémentation du modèle MVC (Modèle-Vue-Contrôleur), et tirent parti des namespaces PHP. De la même manière qu'un plugin ou qu'un module dans la version précédente de Symfony, un bundle contient un ensemble de fonctionnalités logiquement reliées entre elles (en l'occurrence notre bundle TwitterBundle contient, comme son nom l'indique, un ensemble de fonctionnalités permettant l'accès au réseau social Twitter).

## Création du bundle TwitterBundle

La première étape est la création de l'arborescence du bundle. Le modèle MVC est implémenté de la manière suivante au sein d'un bundle :

- les modèles sont indépendants et peuvent être placés dans un sous-répertoire Model,
- les contrôleurs doivent être placés dans le répertoire Controller,
- la ou les vues sont stockées dans le répertoire Resources/views.

D'autres répertoires composent un bundle, et stockent un certain nombre de ressources utiles : les scénarios de test (dans le répertoire Tests/), les ressources web (scripts Javascript, images, etc. dans



Resources/web/), ou encore la configuration (Resources/config/). Le fichier principal TwitterBundle.php est le suivant :

```
namespace Bundle\TwitterBundle;

use Symfony\Framework\Bundle\Bundle;

class TwitterBundle extends Bundle
{
}
```

Un bundle doit être nommé de manière précise en utilisant un namespace, composé de plusieurs arguments :

- le namespace principal (Bundle pour un plugin, Application pour un bundle d'application),
- le possesseur du bundle (facultatif),
- la catégorie du bundle (si besoin pour organiser un grand nombre de bundles),
- enfin le nom du bundle lui-même.

Ce qui pourrait donner par exemple comme namespace pour notre bundle d'accès à Twitter Bundle/Valtech/Social/TwitterBundle, avec comme nom ValtechSocialTwitterBundle (on voit bien que le nom, bien que long, doit être tout à fait clair et doit décrire correctement le bundle). Comme notre application est simple, nous avons choisi d'utiliser le namespace Bundle\TwitterBundle.

## REQUEST HANDLER

Le request handler est le système permettant l'association d'une requête de l'utilisateur (requête HTTP) à la meilleure réponse correspondante.

```
// src/Bundle/TwitterBundle/Resources/config/routing.yml
twitter:
    pattern: /twitter/:userId
    defaults: { _controller: TwitterBundle:Twitter:index }
```

Cette règle est associée à la méthode indexAction de la classe TwitterController ci-dessous :

```
// src/Bundle/TwitterBundle/Controller/TwitterController.php
namespace Bundle\TwitterBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Bundle\TwitterBundle\Model\Timeline; // model inclusion
using namespace

class TwitterController extends Controller {

    /**
     * Display timeline of user defined by userId
     * @param String $userId twitter user screen name or id
     */
    public function indexAction($userId) {
        $tl = new Timeline($userId);
        return $this->render('TwitterBundle:Twitter:index',
            array('status'=>$tl->getTextStatus(), 'user'=>$tl->getUserId()));
    }

    // ...
}
```

Son utilisation est facile à appréhender : chaque route associe une règle à une action d'un contrôleur, ce système pouvant être utilisé de deux manières différentes.

## Projets structurés

La première méthode est la méthode classique, déjà implémentée dans les versions précédentes de Symfony.

Les routes sont déclarées dans les fichiers de routing de l'application ou de chaque bundle. Cette méthode permet de structurer facilement les projets en regroupant les ensembles de routes selon la logique du projet, et d'en améliorer la maintenabilité.

```
// hello/config/routing.yml
homepage:
    pattern: /
    defaults: { _controller: FrameworkBundle:Default:index }

twitter:
    resource: TwitterBundle/Resources/config/routing.yml
```

La route par défaut de l'application (homepage) est déclarée directement dans le fichier de configuration de l'application, alors que les routes concernant notre bundle Twitter sont incluses grâce au fichier de configuration contenu dans le bundle.

## Micro framework

La seconde correspond à une méthode utilisée dans le monde des micro-frameworks, où tout un projet est contenu dans un seul fichier (comme par exemple le framework Sinatra utilisant le langage Ruby). Dans ce cas un fichier unique est créé, liant directement une route à une réponse : il est ainsi plus facile d'avoir une vision globale d'un projet. Cette méthode n'est actuellement pas exploitable directement dans Symfony2, mais le projet Silex, développé par Fabien Potencier, est un mini-framework basé sur Symfony2 exploitant pleinement cette méthode de développement.

## ENVIRONNEMENTS

Symfony propose un système d'environnement simple à comprendre et à mettre en oeuvre. Chaque phase d'un projet s'effectue dans un environnement clos, donnant accès à un certain nombre d'outils destinés à faciliter le développement. Trois types d'environnements sont proposés par défaut, avec chacun leurs spécificités et leurs outils, tout en ayant la possibilité de configurer des environnements plus spécifiques si nécessaire.

## Développement

Le premier, et le principal, est l'environnement de développement. Il permet aux développeurs d'avoir accès à un grand nombre d'informations facilitant le débogage et l'optimisation d'un projet. Il est accessible en préfixant toutes les URL du projet par index\_dev.php. Les informations sont fournies par la «debug bar», présente en bas de chaque page du projet. Celle-ci donne accès à un certain nombre de données (nombre de requêtes vers la base de données, temps de génération, poids de la page, etc.), mais la grande nouveauté est le Profiler, qui permet d'accéder plus facilement à l'ensemble des détails utiles :

- le détail de la requête effectuée pour accéder à la page,
- les exceptions ayant pu apparaître,

- les différents événements ayant conduit au rendu de la page,
- enfin les logs concernant la page en cours.

Une autre force de cet environnement est la page affichée en cas d'erreur, qui donne une vue compréhensible de celle-ci, de la stack trace (pile d'appel) ainsi que des logs.

## Tests

Le second environnement, l'environnement de test, permet de réaliser automatiquement des tests unitaires et fonctionnels à l'aide de PHPUnit (un framework de test open source dédié au langage PHP), ce qui est généralement une des tâches les plus fastidieuses à réaliser «à la main». Leur mise en place est simple, l'arborescence d'un bundle (en quelque sorte l'unité logique de développement, nous y reviendrons un peu plus tard) est répliquée dans un sous-répertoire Tests, où les tests sont rédigés. Les tests fonctionnels ont un système de rédaction spécifique : ils se basent sur un navigateur virtuel, qui va effectuer les mêmes opérations qu'un internaute réel. Leur rédaction passe donc par deux étapes : la requête (requête http, simulation de la soumission d'un formulaire ou du clic sur un lien) et le test de la réponse (où le document est analysé pour vérifier la validité des données qu'il contient). Ci-dessous un exemple de test réalisé pour l'exemple d'application Symfony Sandbox.

```
namespace Application\HelloBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class HelloControllerTest extends WebTestCase {

    public function testIndex() {
        $client = $this->createClient();
        $crawler = $client->request('GET', '/hello/Fabien');
        $this->assertTrue($crawler->filter('html:contains
("Hello Fabien")')->count() > 0);
    }
}
```

L'environnement de test est automatiquement sélectionné lors du lancement de tests avec PHPUnit.

## Production

Dernier environnement, il est celui qui va permettre de mettre le projet en production, avec tout ce que cela comporte : activation du cache, masquage des erreurs, données définitives. Il est utilisé par défaut, lors de l'appel d'une page sans utiliser d'URL spécifique.

## EVENT DISPATCHER

Pour rendre notre application plus intelligente et lui permettre de se connecter à plusieurs réseaux sociaux, nous allons introduire l'« Event Dispatcher ». Le concept utilisé par Symfony 2 est celui des notifications du framework Cocoa (<http://developer.apple.com/documentation/Cocoa/Conceptual/Notifications/index.html>). Un « event dispatcher » est un objet sur lequel vont venir se connecter des émetteurs et des récepteurs d'événements. Nous allons utiliser ce dispatcher pour venir connecter des agents dédiés à différents réseaux sociaux qui recevront un événement du type « timeline » ou « profile » pour leur demander d'effectuer un traitement particulier. L'émetteur de l'événement n'aura pas à se soucier de savoir quels sont les agents connectés. Il est important de noter que l'Event Dispatcher n'est pas

un singleton, de ce fait il est nécessaire d'injecter la même instance à l'ensemble des acteurs de l'événement (émetteur et récepteurs).

Plusieurs méthodes existent sur l'objet Event Dispatcher :

- *connect* : permet de se connecter à un dispatcher
- *disconnect* : permet de se déconnecter d'un dispatcher
- *notify* : permet d'émettre un événement (cet événement est typé par un nom)
- *notifyUntil* : permet d'émettre un événement jusqu'à ce qu'un accusé de traitement soit reçu
- *filter* : permet d'émettre un événement en lui associant un objet. Cet objet peut ainsi être traité et enrichi par les différents récepteurs

Pour ne pas mélanger les codes, nous allons déployer un nouveau Bundle nommé SocialNetworksBundle. Cette nouvelle application est disponible dans le Bundle SocialNetworkBundle.

La création de l'EventDispatcher et son injection dans les agents Twitter et Plurk se fait lors de la création du contrôleur :

```
class SocialNetworksController extends Controller {
    private $eventDispatcher = null;

    public function __construct() {
        $this->eventDispatcher=new EventDispatcher;
        new Twitter($this->eventDispatcher);
        new Plurk($this->eventDispatcher);
    }
}
```

Les agents une fois créés vont s'abonner à un message du type « profile » et « timeline » sur l'EventDispatcher :

```
function __construct(EventDispatcher $eventDispatcher) {
    // Abonnement à l'événement « timeline »
    $eventDispatcher->connect('timeline', function(Event $event,
$value) {
        $document = new \DOMDocument();
        try {
            $document->load(sprintf("http://twitter.com/
statuses/user_timeline/%s.xml",$value["userId"]));
        } catch (\ErrorException $e) {
            return $value;
        }
        // Traitement de la timeline Twitter
        $data = array();
        $xpath = new \DOMXPath($document);
        foreach ($xpath->query(Parser::cssToXPath('status
> text')) as $node) {
            $data[] = $node->nodeValue;
        }

        $value["timelines"]["Twitter"]=$data;
        return $value;
    });

    // Abonnement à l'événement « profile »
    $eventDispatcher->connect('profile', function(Event $event,
$value) {
        $document = new \DOMDocument();
        try {
            $document->load(sprintf("http://twitter.com/
statuses/user_timeline/%s.xml",$value["userId"]));
        } catch (\ErrorException $e) {
```

```

        return $value;
    }
    // Traitement du profile Twitter
    $data = array();
    $xpath = new \DOMXPath($document);
    // get required values
    $data['screen_name'] = $xpath->query(Parser::cssToXpath('status > user > screen_name'))->item(0)->nodeValue;
    $data['name'] = $xpath->query(Parser::cssToXpath('status > user > name'))->item(0)->nodeValue;
    $data['description'] = $xpath->query(Parser::cssToXpath('status > user > description'))->item(0)->nodeValue;
    $data['url'] = $xpath->query(Parser::cssToXpath('status > user > url'))->item(0)->nodeValue;
    $data['friends_count'] = $xpath->query(Parser::cssToXpath('status > user > friends_count'))->item(0)->nodeValue;
    $data['followers_count'] = $xpath->query(Parser::cssToXpath('status > user > followers_count'))->item(0)->nodeValue;

    $value["informations"]["Twitter"]=$data;
    return $value;
});
}

```

Listener/Twitter.php

```

function __construct(EventDispatcher $eventDispatcher) {
    $eventDispatcher->connect('timeline', function(Event $event, $value) {
        try {
            $handle = fopen(sprintf("http://www.plurk.com/API/Profile/getPublicProfile?api_key=mCfV9V6XGBpPebFYB1EK77uJVD9L84sg&user_id=%s", $value["userId"]), "r");
            $contents = stream_get_contents($handle);
            fclose($handle);
            $plurks=json_decode($contents);
            } catch (\ErrorException $e) {
                return $value;
            }
            $posts = array();
            foreach($plurks->plurks as $i=>$plurk) {
                array_push($posts, "<strong>".$plurk->qualifier."</strong>".$plurk->content);
            }
            $value["timelines"]["Plurk"]=$posts;
            return $value;
        });
        $eventDispatcher->connect('profile', function(Event $event, $value) {
            try {
                $handle = fopen(sprintf("http://www.plurk.com/API/Profile/getPublicProfile?api_key=mCfV9V6XGBpPebFYB1EK77uJVD9L84sg&user_id=%s", $value["userId"]), "r");
                $contents = stream_get_contents($handle);
                fclose($handle);
                $plurks=json_decode($contents);
                } catch (\ErrorException $e) {
                    return $value;
                }
            }
        }
    }

```

```

$userInfos = array();
foreach($plurks->user_info as $i=>$info) {
    $userInfos[$i] = $info;
}
$value["informations"]["Plurk"]=$userInfos;
return $value;
});
}

```

Listener/Plurk.php

Les fonctions appelées lors du routage des pages (par exemple timeline) vont déclencher les événements adéquats sur l'EventDispatcher :

```

public function timelineAction($userId) {
    $tl = new Timeline($userId, $this->eventDispatcher);
    return $this->render('SocialNetworksBundle:SocialNetworks:timeline', array('timelines'=>$tl->getTimeline(), 'user'=>$tl->getUserId()));
}

```

Controller/SocialNetworksController.php

```

function __construct($userId = "", EventDispatcher $eventDispatcher) {
    if ($userId != "") {
        $this->setUserId($userId);
    }
    $value = array();
    $value["timelines"]=array();
    $value["userId"]=$this->getUserId();
    $event = $eventDispatcher->filter(new Event(null, 'timeline'), $value);
    $this->setTimeline($event->getReturnValue());
}

```

Model/Timeline.php

La méthode utilisée pour l'envoi du message étant la méthode « filter », l'ensemble des agents écoutant ce message vont traiter, l'un après l'autre, cet ordre en interrogeant à leur manière le réseau social concerné, et vont ajouter les informations reçues et décodées à l'objet « \$value » transmis en paramètre de la méthode filter. Ce tour d'horizon de l'application SocialNetworkBundle a permis de mettre en évidence l'utilisation de l'EventDispatcher de Symfony.

## CONCLUSION

Bien que Symfony 2 soit un framework toujours en phase de développement, et dont la version finale n'est pas prévue avant le premier trimestre 2011, cette nouvelle version marque une évolution très importante et semble prometteuse. La flexibilité qu'elle apporte est un vrai plus par rapport à l'architecture figée des précédentes versions. De plus, un effort a été fait pour rendre compréhensibles les opérations auparavant « magiques » (l'appel d'un template pour le rendu par exemple), ce qui permet une compréhension facilitée pour les nouveaux utilisateurs.

- Aurélien **Olivier**, développeur Web, Valtech
- Patrick **Antivackis**, directeur technique, Valtech



## OUTILLAGE

## LaTeX l'essentiel

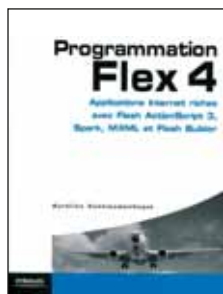


Difficulté : \*\*\*  
 Editeur : Pearson  
 Auteur : collectif  
 Prix : 27 €

Un des piliers de l'informatique : LaTeX. Il s'agit d'un système permettant de rédiger, de mettre en forme des articles scientifiques, des mémoires... Outil particulièrement puissant, il nécessite tout de même une solide expérience pour le maîtriser, le mettre en œuvre. Ce livre présente l'essentiel de ce qu'il faut savoir pour composer des documents avec LaTeX. Le lecteur est guidé pas à pas dans son apprentissage grâce à un cours progressif, illustré d'exemples pratiques. L'ouvrage aborde l'installation, la réalisation de formules, documents spécifiques comme les classes Beamer, la gestion des index, etc. Des exercices permettent de s'exercer à la fin de chaque chapitre. Un ouvrage qui fera rapidement référence pour les utilisateurs et la communauté LaTeX !

## WEB

## Programmation Flex 4



Difficulté : \*\*\*  
 Editeur : Eyrolles  
 Auteur : Aurélien Vannieuwenhuyze  
 Prix : 42 €

Flex 4 méritait bien 600 pages ! L'ouvrage explique aux développeurs web, qu'ils soient ou non rompus à Flash et ActionScript, comment utiliser le framework Flex pour concevoir et créer des applications web dites riches (RIA), à l'instar des applications Ajax ou Silverlight. Tout en rappelant les langages sur lesquels s'adosse cette technologie (ActionScript, MXML), l'ouvrage passe en revue l'intégralité des techniques de développement Flex : maîtrise de l'environnement de travail, création d'interfaces interactives et évoluées avec les vues, transitions et thèmes, gestion des données et échanges avec le serveur via RPC, mais aussi gestion et création des composants, débogage et optimisation, tests unitaires avec Flex Unit. Il les met ensuite en situation avec deux études de

cas détaillant la création d'un catalogue interactif (où la gestion des produits met en œuvre Adobe Air 2), puis l'incorporation de l'API de géolocalisation de Google. Aurélien publie régulièrement depuis plusieurs années des articles dans Programmez !.

## ALM

## TFS 2010

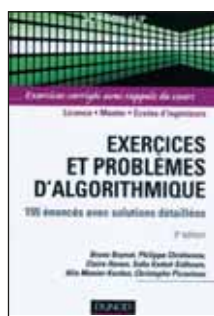


Difficulté : \*\*\*  
 Editeur : éditions Eni  
 Auteur : Florent Santin  
 Prix : 29 €

Team Foundation Server 2010 est la plateforme ALM de Microsoft et la version 2010 introduit de très nombreuses nouveautés et améliorations. Dans les premiers chapitres, vous découvrirez le mode opératoire pour installer le serveur en fonction des diverses architectures envisageables. Le chapitre suivant s'attarde sur les procédures et l'outillage requis pour l'administration tels que la sauvegarde, la reprise en cas d'incident, la création de projets ou bien la gestion des droits utilisateurs. La notion de projet d'équipe est ensuite détaillée afin de présenter ce point d'entrée au référentiel de projets commun à tous les utilisateurs... Il s'agit de l'ouvrage, en français, le plus complet et le plus actuel sur TFS 2010 !

## SYSTÈME

## Exercices et problèmes d'algorithmique



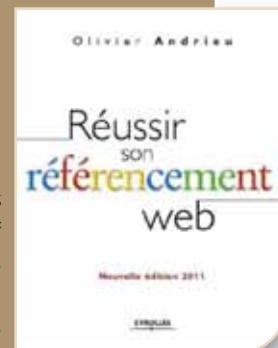
Difficulté : \*\*\*  
 Editeur : Dunod  
 Auteur : collectif  
 Prix : 39 €

Cet ouvrage s'adresse aux étudiants en licence et master de mathématiques et d'informatique, ou en cursus universitaire équivalent de bac+3 à bac+5, ainsi qu'aux élèves en écoles d'ingénieurs. On y trouvera, pour chaque notion, des exercices visant la compréhension du cours. Pour chaque question, une correction entièrement rédigée, rigoureuse et complète est fournie. 155 exercices permettent d'apprendre, de s'entraîner, avec une progression dans la difficulté !

## LIVRE DU MOIS

## Réussir son référencement web édition 2011

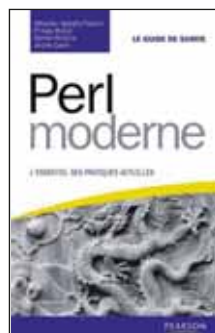
Difficulté : \*\*\*  
 Editeur : Eyrolles  
 Auteur : Olivier Andrieu  
 Prix : 29,90 €



Le référencement est plus que jamais le nerf de la guerre des sites web. Et la question à résoudre paraît très simple : comment être bien référencé sur les moteurs de recherche ? Comment optimiser sa présence ? Comment l'entretenir ? L'auteur nous livre ici un précieux sésame vers un univers de plus en plus complexe. Le référencement est un art en perpétuelle évolution car les algorithmes des moteurs comme Google changent plusieurs fois par semaine ! Et de nouveaux concepts apparaissent : modelage de pagerank, indexation rapide PubSub Hubbub, temps de chargement d'une page, monétisation, pages sur le web mobile, etc. Avec de nombreux exemples, des conseils très précis, cet ouvrage sera votre bible du web de l'année 2011... jusqu'à la prochaine édition...

## LANGAGE

## Perl moderne



Difficulté : \*\*\*  
 Editeur : Pearson  
 Auteur : collectif  
 Prix : 22 €

Perl est un des langages historiques de l'informatique et demeure très utilisé. Ce Guide de survie est l'outil indispensable pour programmer en Perl aujourd'hui. Il présente les dernières évolutions de Perl 5 avec ses versions 5.10 et 5.12, fortement empreintes de la version 6 en cours de finalisation. Plus de 350 séquences de code pour répondre aux situations les plus courantes et exploiter efficacement les fonctions et les bibliothèques d'un langage qui s'est radicalement modernisé.

# 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.*



Cas clients

Actu triée par secteur

Avis d'Experts



Actus / Evénements / Newsletter / Vidéos

[www.solutions-logiciels.com](http://www.solutions-logiciels.com)

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, 22 rue René Boulanger, 75472 PARIS - ou par fax : 01 55 56 70 20

**1 an : 30€ au lieu de 36€, prix au numéro** (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 36€ - Algérie, Maroc, Tunisie : 36€ - Canada : 48€ - Dom : 45€ - Tom : 60€  
6 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : ..... Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM ..... Prénom

N° ..... rue

Complément

Code postal : ..... Ville

Adresse mail

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

# AVEC LE CLOUD, LE MEILLEUR DE VOTRE BUSINESS EST DEVANT VOUS.

Vous avez le pouvoir de voir très grand. Dès le départ.

Vous maîtrisez vos coûts, quels que soient les à-coups.

Vous ajustez vos ressources à vos besoins, en temps réel.

Vous êtes proche de vos données, même quand vous en êtes loin.

Vous n'opposez plus grandes idées et petits budgets.

Vous économisez l'énergie. Et l'espace.

Vous transformez plus vite vos projets en réalité business.

Votre architecture IT a votre imagination pour seule limite.

Avec la puissance du Cloud, vous disposez de la plus large gamme de solutions, hautement disponibles, ouvertes, agiles et simples à utiliser.

Vous aussi, tirez parti de toute la puissance du Cloud. Maintenant.

**Microsoft**



**Cloud Power**

**LA PLUS LARGE GAMME DE SOLUTIONS DANS LE NUAGE. SUR TERRE.**

Microsoft® Office 365 • Windows Azure™ • Windows Server® Hyper-V® • Microsoft Dynamics™ CRM Online.  
En savoir plus : [www.microsoft.fr/cloud](http://www.microsoft.fr/cloud)